# OMIS 6000

## Week 3:

- Range of optimality and feasibility
- Shadow prices and reduced costs
- Primal and dual linear programs
- Complementary slackness

**Adam Diamant, Schulich School of Business, Winter 2025**

# Sensitivity Analysis

- It is an analysis that is used to determine how the **optimal solution** is affected by changes in:
  - The objective function coefficients (**OFC**s).
  - The right-hand side (RHS) values of the constraints.

- **Sensitivity analysis** is important to the manager who must operate in environments with imprecise estimates for the parameters of a problem.

- It allows a manager to ask **"what-if"** questions and determines how sensitive the optimal solution is to data errors and parameter misspecifications.

# What should you be looking for?

Changes to the formulation could affect:

1) Which **corner point** is the optimal one.
2) The size and shape of the **feasible region**.
3) The **objective function value** at the optimal solution (even if it is the same corner point).

Each type of change has a different effect on the model. It also gives some insight as to the ***robustness*** of the optimal solution.

# Flair Furniture Example

# Flair Furniture Example

- **Two products:** Chairs and Tables
- **Decision:** How many of each to make this month?
- **Objective:** Maximize profit

| | **Tables** (per table) | **Chairs** (per chair) | |
|---|---|---|---|
| **Profit Contribution** | $7 | $5 | **Hours Available** |
| **Carpentry** | 3 hrs | 4 hrs | 2400 |
| **Painting** | 2 hrs | 1 hr | 1000 |

**Restrictions:** Make no more than 450 chairs and at least 100 tables

# Flair Furniture Example

**Maximize** $\boldsymbol{Z} = 7T + 5C$ (profit)

**Subject to the constraints:**

$$3T + 4C \leq 2400 \quad \text{(carpentry hrs)}$$

$$2T + 1C \leq 1000 \quad \text{(painting hrs)}$$

$$C \leq 450 \quad \text{(max \# chairs)}$$

$$\mathrm{T} \geq 100 \quad \text{(min \# tables)}$$

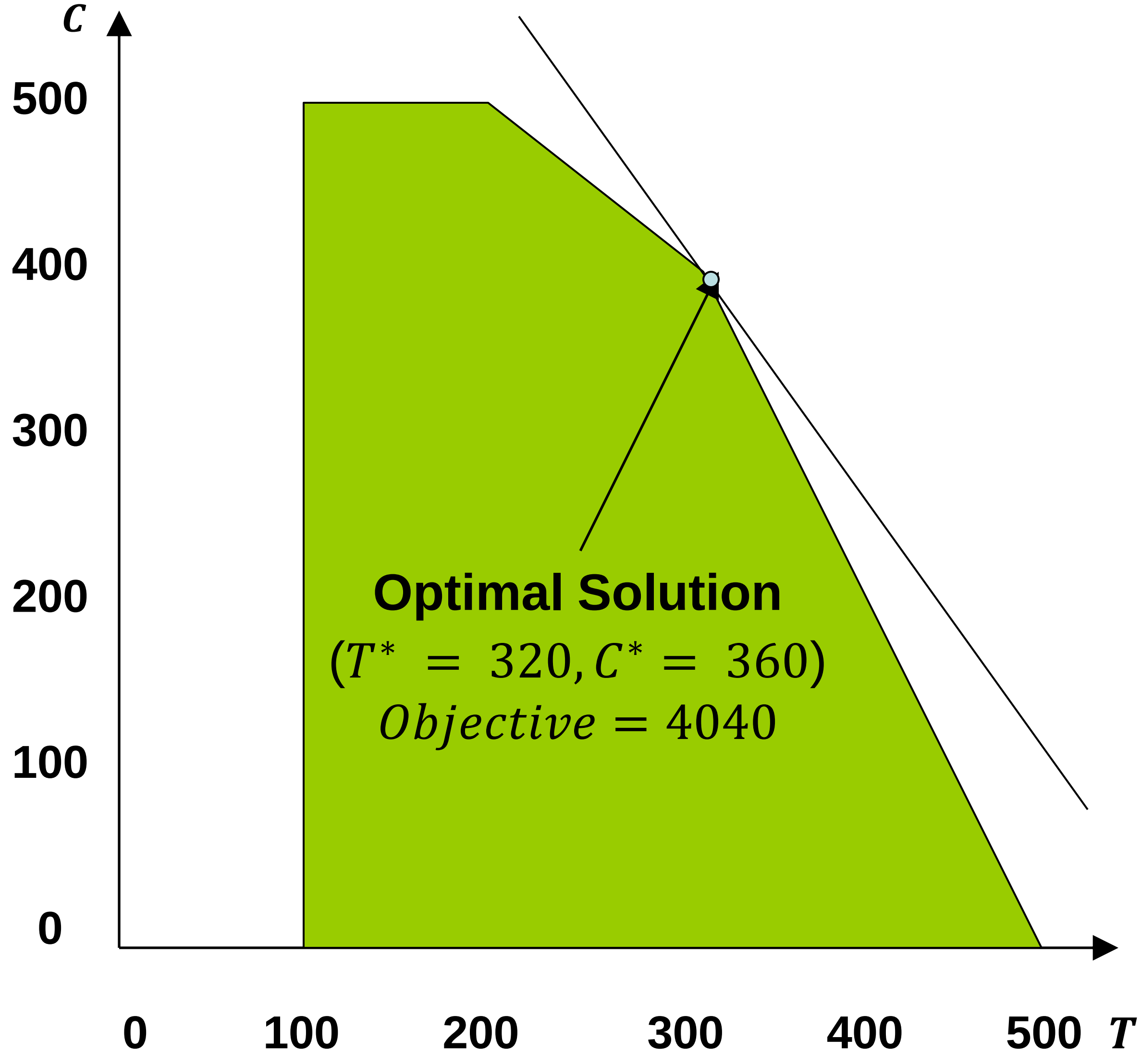
$$T, C \geq 0 \quad \text{(non-negativity)}$$

$$\boldsymbol{T = Table}$$
$$\boldsymbol{C = Chair}$$

# Flair Furniture Example

**Objective Function**

$7T + 5C = Profit$



# Flair Furniture Example: Linear Programming Solutions

Target Cell (Max)

| Cell | Name | Original Value | Final Value |
|---|---|---|---|
| $D$13 | Profit Total | 4040 | 4040 |

objective.getValue()

or

model.ObjVal

Adjustable Cells

| Cell | Name | Original Value | Final Value |
|---|---|---|---|
| $B$10 | Tables | 320 | 320 |
| $C$10 | Chairs | 360 | 360 |

name_of_variable.x

Constraints

| Cell | Name | Cell Value | Formula | Status | Slack |
|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | $D$16<=$F$16 | Binding | 0 |
| $D$17 | Painting Used | 1000 | $D$17<=$F$17 | Binding | 0 |
| $D$18 | Minimum Tables Used | 320 | $D$18>=$F$18 | Not Binding | 220 |
| $D$19 | Maximum Chairs Used | 360 | $D$19<=$F$19 | Not Binding | 90 |

name_of_constraint.slack

# Impact of Possible Changes

1. **What happens if you change the value of an objective function coefficient (OFC)?**
   – The slope of the objective function line will be different which may change the optimal solution.

2. **What happens if you change the right-hand-side (RHS) value of a constraint?**
   – This distorts the size and shape of the feasible region and may alter the optimal solution.

# Objective Function Coefficients

What would happen if the **value** of one of the objective function coefficients (**OFC**s) changed? Would we still have made the same optimal decision? If so, how much can this **OFC** differ without changing the current optimal solution?

This analysis is referred to as the:

**Range of Optimality**

# Flair Furniture Example

What if the profit contribution for tables changed from <u>$**7**</u> to <u>$**8**</u> per table?

$$\text{Maximize } \overset{8}{\cancel{7}}\, T + 5\, C \quad \text{(profit)}$$

- Clearly **profit goes up**, but would we want to make more tables and less chairs?
  - That is, does the **optimal solution** change?

# Flair Furniture Example

**Original:**

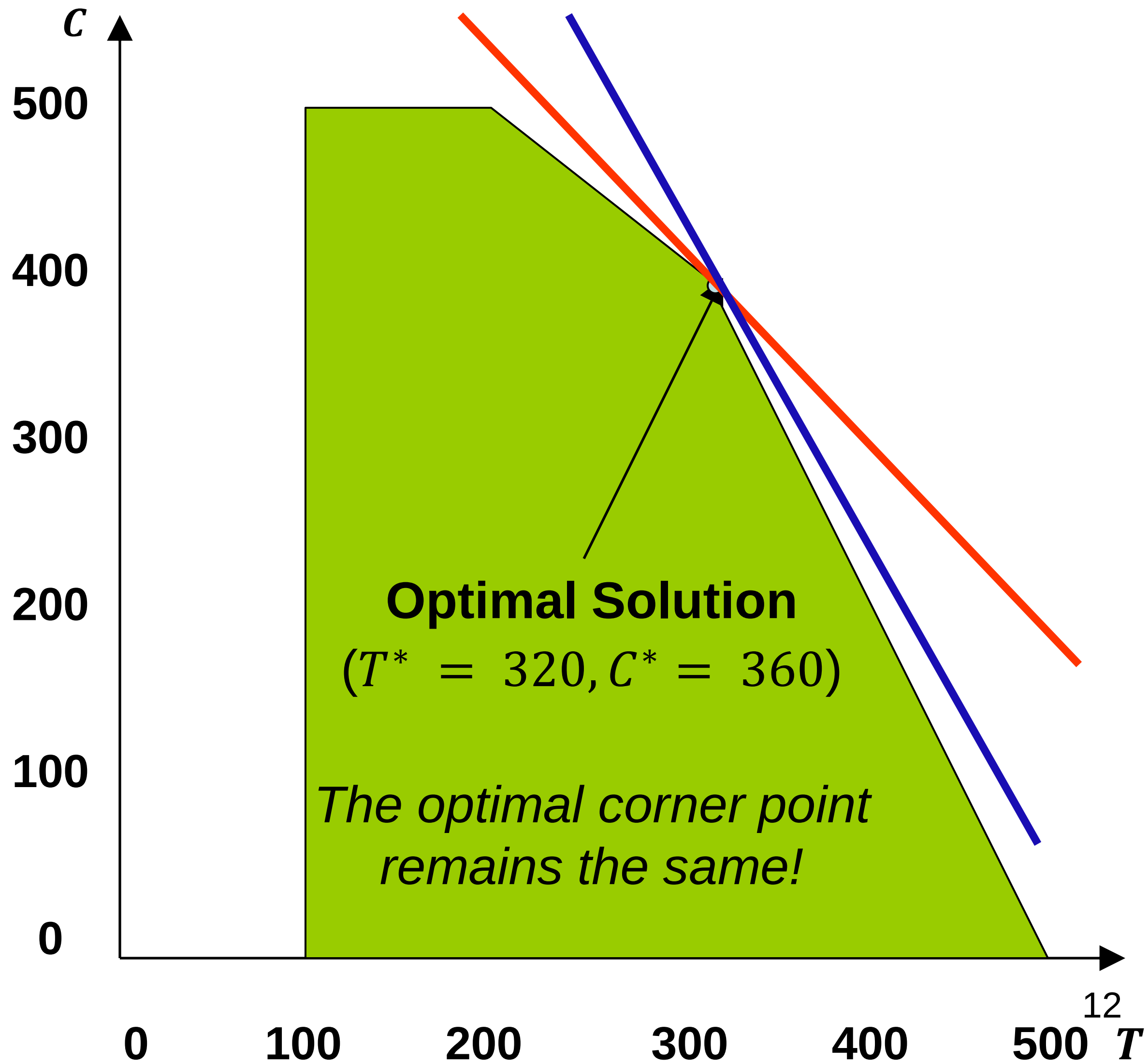
**Objective Function**

$7T + 5C = 4040$

**Revised:**

**Objective Function**

$8T + 5C = 4360$



# Range of Optimality

- The **range of optimality** of an **OFC** is found by determining an interval for the coefficient in which the **original optimal solution remains the same** while keeping all other problem data constant.
  - The shape of the **feasible region** **does not** change.

- If the **OFC** changes beyond its **range of optimality**, a new corner point becomes optimal. The **range of optimality**, then, is an upper and lower limit where the current **optimal solution** does not change.

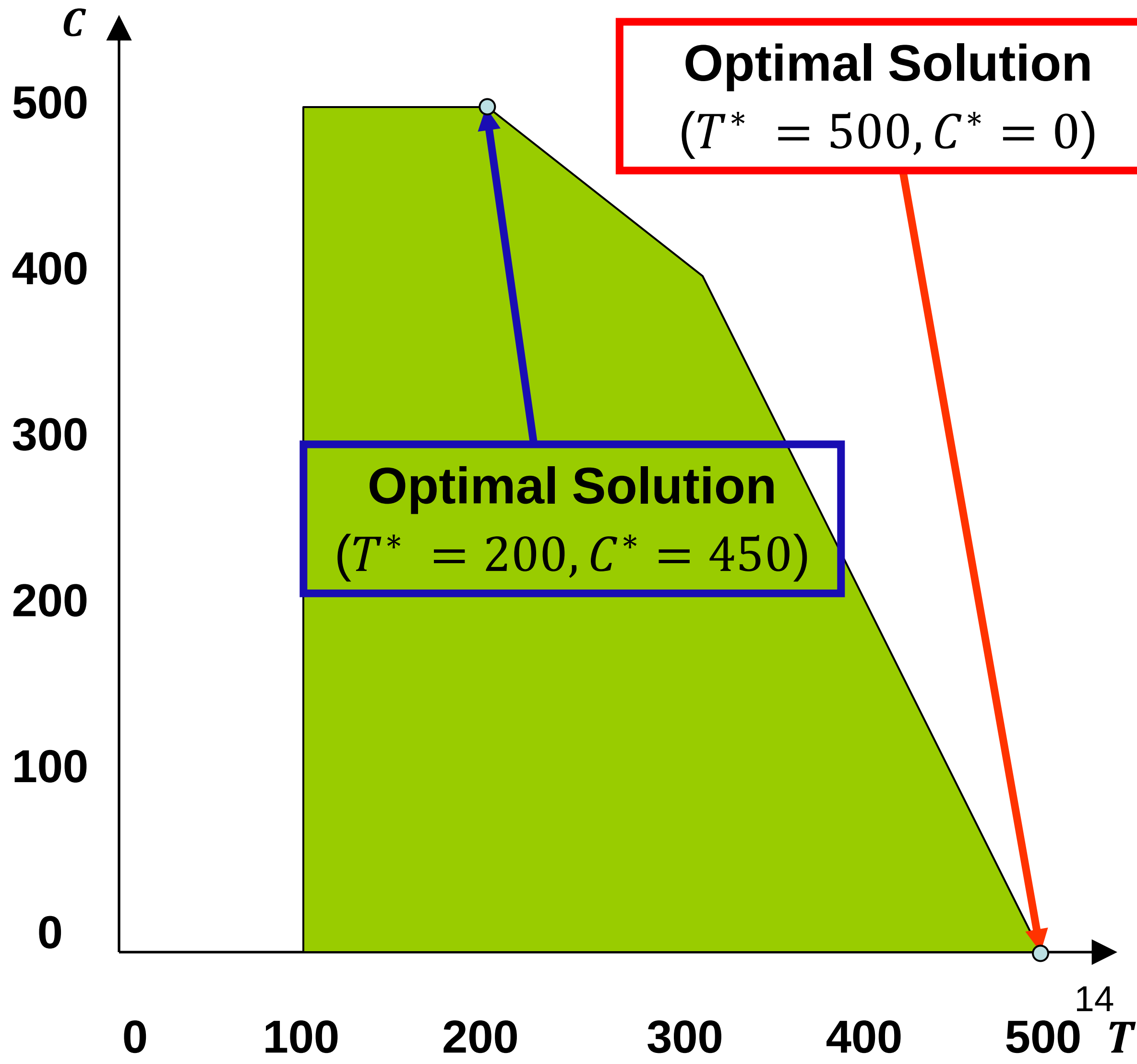
# Flair Furniture Example

**Objective Function**

$$11T + 5C = 5500$$

**Objective Function**

$$3T + 5C = 2850$$



# Flair Furniture Example: Range of Optimality

**Upper Range of Optimality:** **OFC + Allowable Increase**

**Lower Range of Optimality:** **OFC - Allowable Decrease**

**Range of Optimality**

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Flair Furniture Example: Range of Optimality

**Upper Range of Optimality ($T$):** $7 + 3 = 10$

**Lower Range of Optimality ($T$):** $7 - 3.25 = 3.75$

**Range of Optimality**

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Flair Furniture Example: Range of Optimality

**Upper Range of Optimality ($C$):** $5 + 4.333 = 9.333$

**Lower Range of Optimality ($C$):** $5 - 1.50 = 3.50$

**Range of Optimality**

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Flair Furniture Example: Range of Optimality

**Upper Range of Optimality in gurobipy:**

```
name_of_variable.SAObjUp
```

**Range of Optimality**

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Flair Furniture Example: Range of Optimality

**Lower Range of Optimality in [gurobipy](gurobipy):**

```
name_of_variable.SAObjLow
```

**Range of Optimality**

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Changes to Objective Function Coefficients (OFCs)

Provided the change in an **OFC** is **within** (non-inclusive of boundary) the **range of optimality** and no other problem parameters are modified, the objective function value changes by

$$\mathbf{\Delta Objective = \Delta OFC \times Optimal\ Solution}$$

**Flair Furniture Example:**

If the **OFC** corresponding to $C$ increases by \$2, it is still within the **range of optimality** and thus, the objective function value will increase by $\$2 \times 360 = \$720$.

# Flair Furniture Example: Sensitivity Analysis

**Reduced Cost:** The incremental benefit associated with including an extra unit of that variable in the optimal solution.

```
name_of_variable.RC
```

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# What happens with variables in the optimal solution?

- The **reduced cost** is ***<u>zero</u>*** for a decision variable that is currently in the *optimal solution* (e.g., $\text{chairs.RC} = 0$).
  - Suppose the reduced cost $> \ 0$ for a maximization problem.
    - A one-unit increase would lead to a **<u>better</u>** solution.
  - Suppose the reduced cost $< \ 0$ for a minimization problem.
    - A one-unit increase would lead to a **<u>better</u>** solution.
- In both cases, there is a *contradiction* because the solution we have, *by definition*, is the **optimal one**!

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

# What happens with variables <u>not</u> in the optimal solution?

- The **reduced cost** is ***<u>non-zero</u>*** for decision variables *<u>not in</u>* the *optimal solution* (e.g., $\text{desks.RC} < 0$).
  - It represents how much ***<u>less valuable</u>*** a decision variable is to the best objective function value.
  - **Maximization Problems:** Decision variables that equal zero will have **negative** reduced costs. For every unit increase of the decision variable, the objective function would **decrease** by the reduced cost amount.
  - **Minimization Problems:** Decision variables that equal zero will have **positive** reduced costs. For every unit increase of the decision variable, the objective function would **increase** by the reduced cost amount.

# What happens with variables <u>not</u> in the optimal solution?

- The **reduced cost** is ***<u>non-zero</u>*** for decision variables *<u>not in</u>* the *optimal solution* (e.g., $\text{desks.RC} < 0$).
  - It represents how much ***<u>less valuable</u>*** a decision variable is to the best objective function value.
  - **Maximization Problems:** Non-zero reduced costs also represent how much the objective coefficient of that decision variable would have to **increase** before it would become part of the optimal solution.

$$\boldsymbol{OFC > 60 + |-20| = 80}$$

Adjustable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$11 | Desks | 0 | -20 | 60 | 20 | 1E+30 |

# What happens with variables <u>not</u> in the optimal solution?

- The **reduced cost** is ***<u>non-zero</u>*** for decision variables *<u>not in</u>* the *optimal solution* (e.g., $\text{desks.RC} > 0$).
  - It represents how much ***<u>less valuable</u>*** a decision variable is to the best objective function value.
  - **Minimization Problems:** Non-zero reduced costs also represent how much the objective coefficient of that decision variable would have to **decrease** before it would become part of the optimal solution.

$$\boldsymbol{OFC < 60 - |20| = 40}$$

Adjustable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$11 | Desks | 0 | 20 | 60 | 1E+30 | 20 |

# Impact of Possible Changes

1. **What happens if you change the value of an objective function coefficient (OFC)?**
   – The slope of the objective function line will be different which may change the optimal solution.

2. **What happens if you change the right-hand-side (RHS) value of a constraint?**
   – This distorts the size and shape of the feasible region and may alter the optimal solution.

# Impact of Possible Changes

- To perform a sensitivity analysis, we must ensure that the RHS value of every constraint contains only <u>**numbers**</u> (no decision variables).
  - Assume that all terms with decision variables have been moved to the LHS of the constraint.
  - The practical interpretation is that the number on the RHS represents the *amount of a resource*.
- When solving an LP, you do not have to ensure this holds. This is only necessary when performing a **sensitivity analysis**.

# Flair Furniture Example: Sensitivity Analysis

**Optimal LHS value of the constraint**

`model.getRow(constraint).getValue()`

**Current RHS Value**

`constraint.RHS`

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Flair Furniture Example: Sensitivity Analysis

**<u>Question:</u>** What happens if the right-hand-side (RHS) value of one of the constraints changed? Will we still have made the same decision? What happens to the objective?

**Current RHS Value**

`constraint.RHS`

Variable Cells

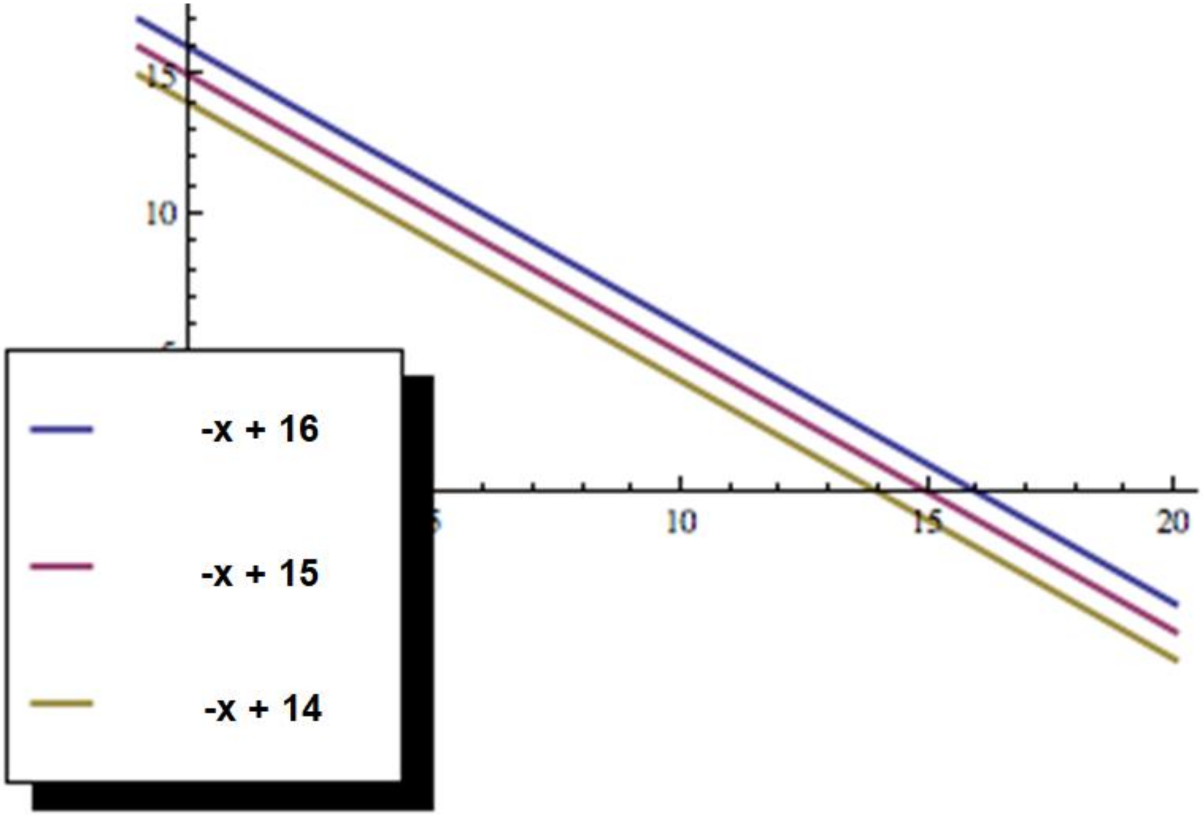
| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# RHS Coefficient Changes

- When a RHS value of a constraint changes, the new constraint moves parallel to its old self.



# RHS Coefficient Changes

- When a RHS value of a constraint changes, the new constraint moves parallel to its old self.

- What happens to the optimal solution and the objective function value depends on whether the line represents a **binding** or **nonbinding** constraint.

- **Common terms:**
  - **Shadow Price** (also known as the **Dual Value**).
  - The **Range of Feasibility** of the resource limits.

# RHS Coefficient Changes

**<u>Shadow Price</u>:** The amount that the objective function value will **change** per unit **increase** in the right-hand side (RHS) value of a ***<u>single</u>*** constraint.

`constraint.pi`

- All other data is assumed to remain the same. That is, other parameters are fixed or constant.
- The insights only hold when the change in the RHS is within the **range of feasibility**, i.e., the maximum allowable increase (`constraint.SARHSUp`) and decrease (`constraint.SARHSLow`) in the RHS value until a new solution becomes optimal.

# Flair Furniture Example: Sensitivity Analysis

**Shadow Prices**

constraint.pi

**Range of Feasibility**

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Flair Furniture Example: Sensitivity Analysis

**Upper Range of Feasibility: RHS + Allowable Increase**

**Lower Range of Feasibility: RHS - Allowable Decrease**

**RHS**
`constraint.RHS`

**Range of Feasibility**

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Flair Furniture Example: Sensitivity Analysis

**Upper Range of Feasibility (Carpentry):** $2400 + 225 = 2625$

**Lower Range of Feasibility (Carpentry):** $2400 - 900 = 1500$

**RHS**
constraint.RHS

**Range of Feasibility**

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Flair Furniture Example: Sensitivity Analysis

**Upper Range of Feasibility (Painting):** $1000 + 600 = 1600$

**Lower Range of Feasibility (Painting):** $1000 - 150 = 850$

| RHS `constraint.RHS` | Range of Feasibility |
|---|---|

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Flair Furniture Example: Sensitivity Analysis

**Upper Range of Feasibility in gurobipy:** `constraint.SARHSUp`

**Lower Range of Feasibility in gurobipy:** `constraint.SARHSLow`

**RHS** `constraint.RHS`

**Range of Feasibility**

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

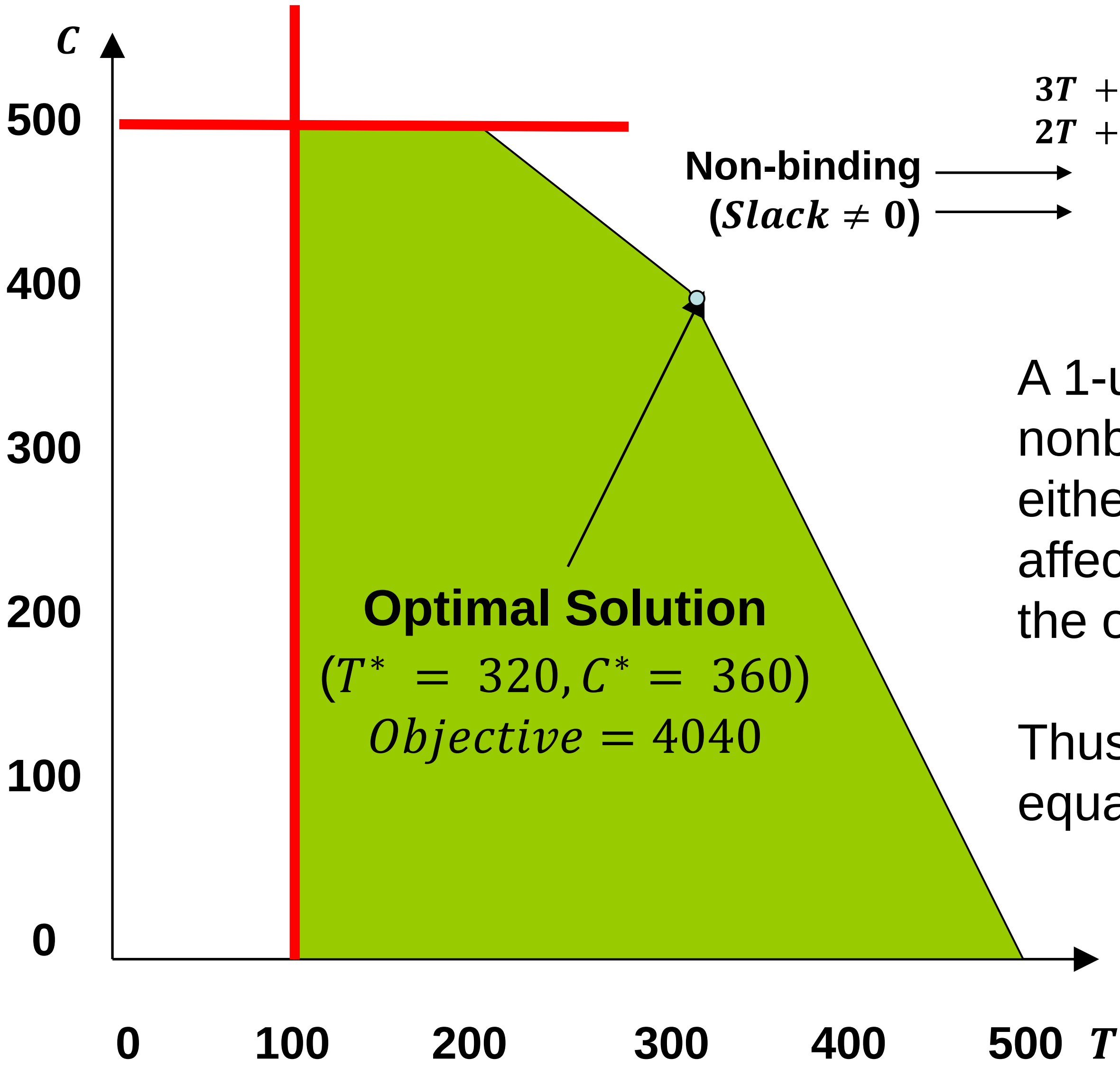
| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# RHS Coefficient Changes

- Shadow prices for *<u>nonbinding constraints</u>* are **zero**.
  - **Interpretation:** A non-binding constraint indicates that we have not used up all of a resource. Thus, changing its capacity ***<u>will not</u>*** affect the objective function.

- Shadow prices for *<u>binding constraints</u>* are **non-zero**.
  - **Interpretation:** A binding constraint indicates that we have used up all of a resource. Thus, changing the capacity ***<u>will</u>*** affect the objective function.

# Flair Furniture Example: Nonbinding Constraints




$3T + 4C \leq 2400$ (carpentry hrs)
$2T + 1C \leq 1000$ (painting hrs)
$C \leq 450$ (max # chairs)
$T \geq 100$ (min # tables)

**Non-binding ($Slack \neq 0$)**

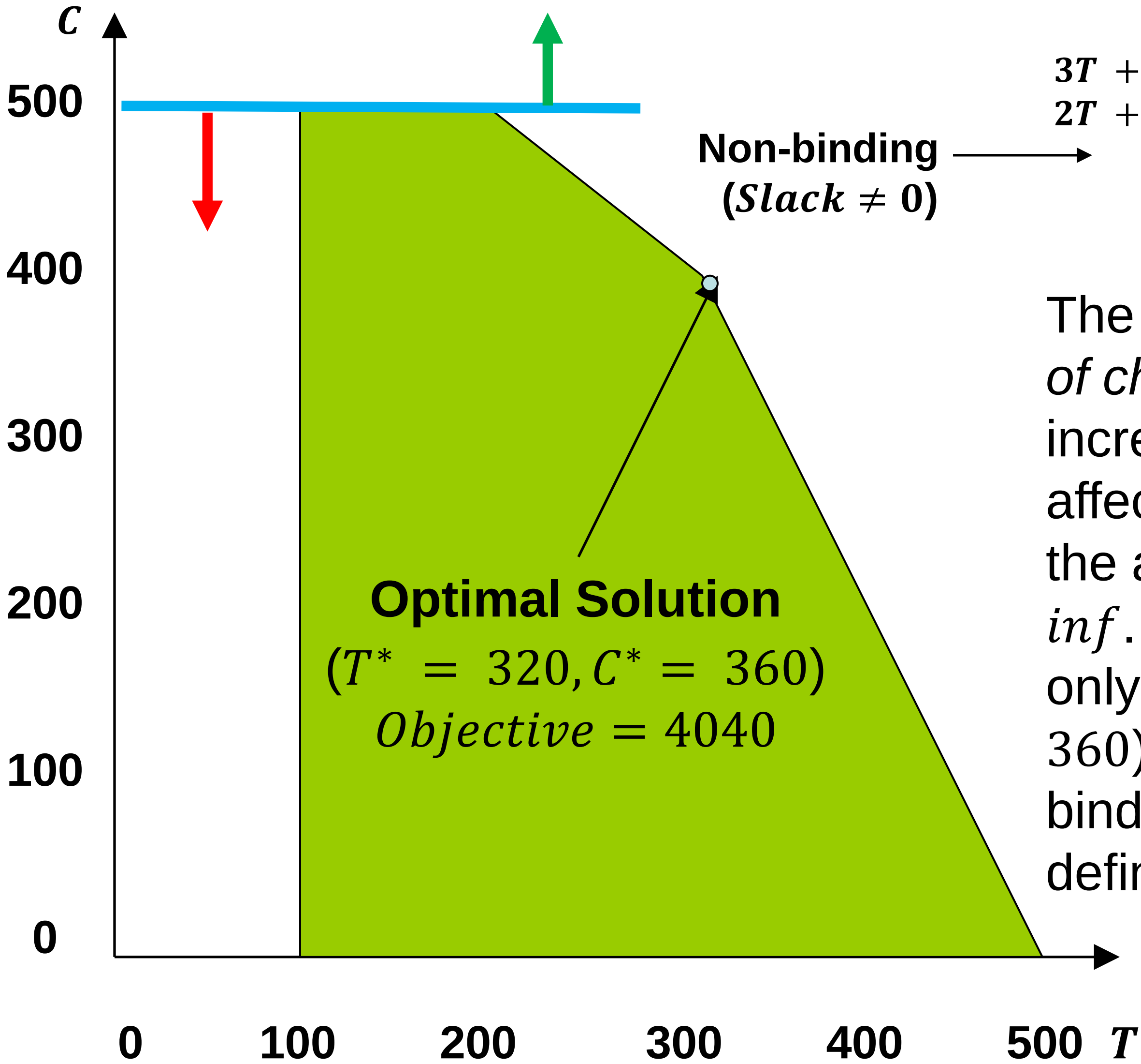
A 1-unit change in the RHS of nonbinding constraints in either direction ***does not*** affect the optimal solution nor the objective function value.

Thus, the shadow prices equal 0 for these constraints.

# Flair Furniture Example: Nonbinding Constraints

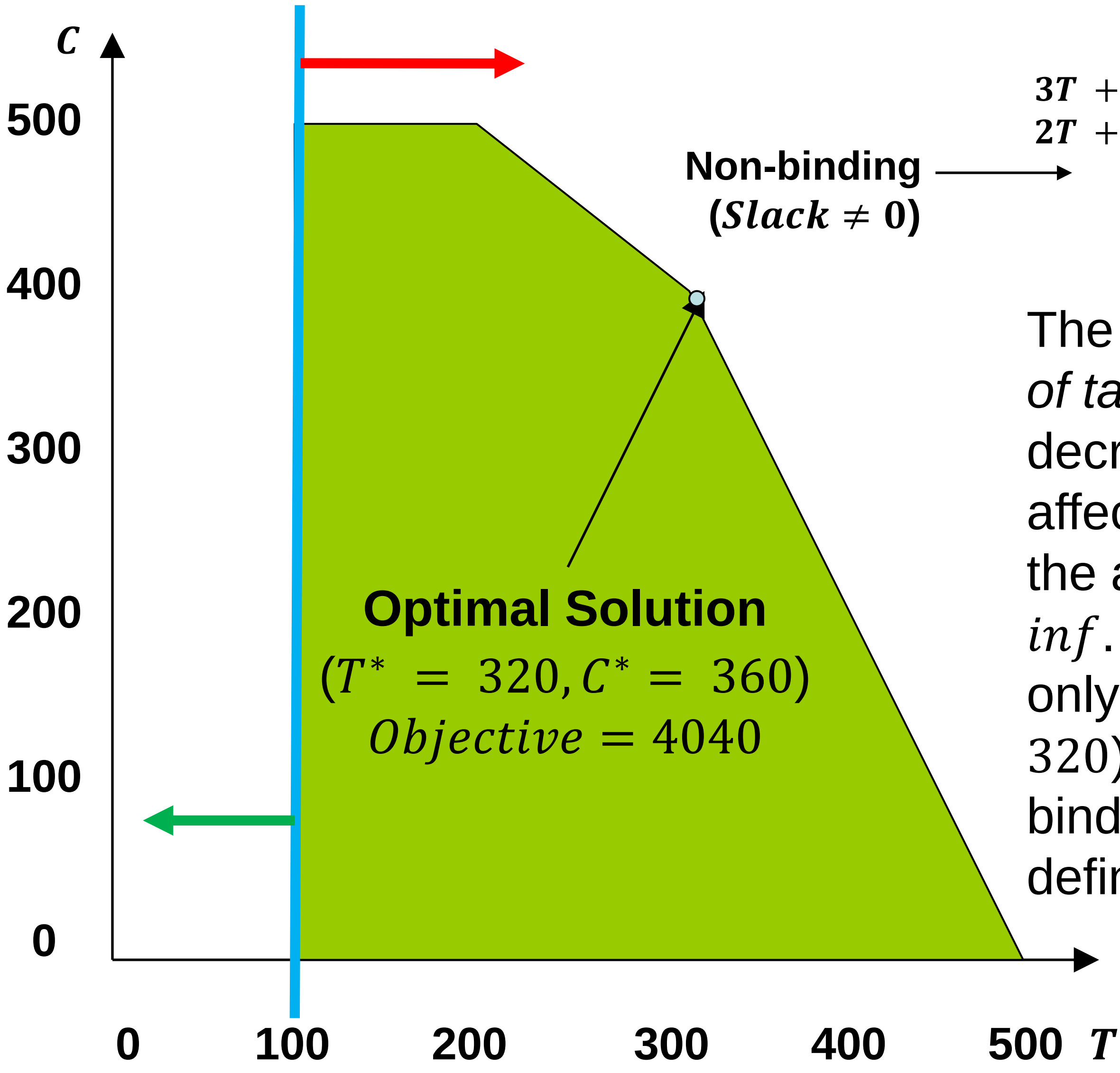
$$\begin{aligned} 3T + 4C &\leq 2400 && \text{(carpentry hrs)} \\ 2T + 1C &\leq 1000 && \text{(painting hrs)} \\ C &\leq 450 && \text{(max \# chairs)} \\ T &\geq 100 && \text{(min \# tables)} \end{aligned}$$

Non-binding ($Slack \neq 0$) → $C \leq 450$




The RHS of the *max number of chairs constraint* can be increased infinitely without affecting feasibility. Hence, the allowable increase is $inf$. However, the RHS can only be decreased by $90$ (to $360$) until it becomes a binding constraint and thus, defines an optimal solution.

# Flair Furniture Example: Nonbinding Constraints




Non-binding ($Slack \neq 0$) → $C \leq 450$

$3T + 4C \leq 2400$ (carpentry hrs)

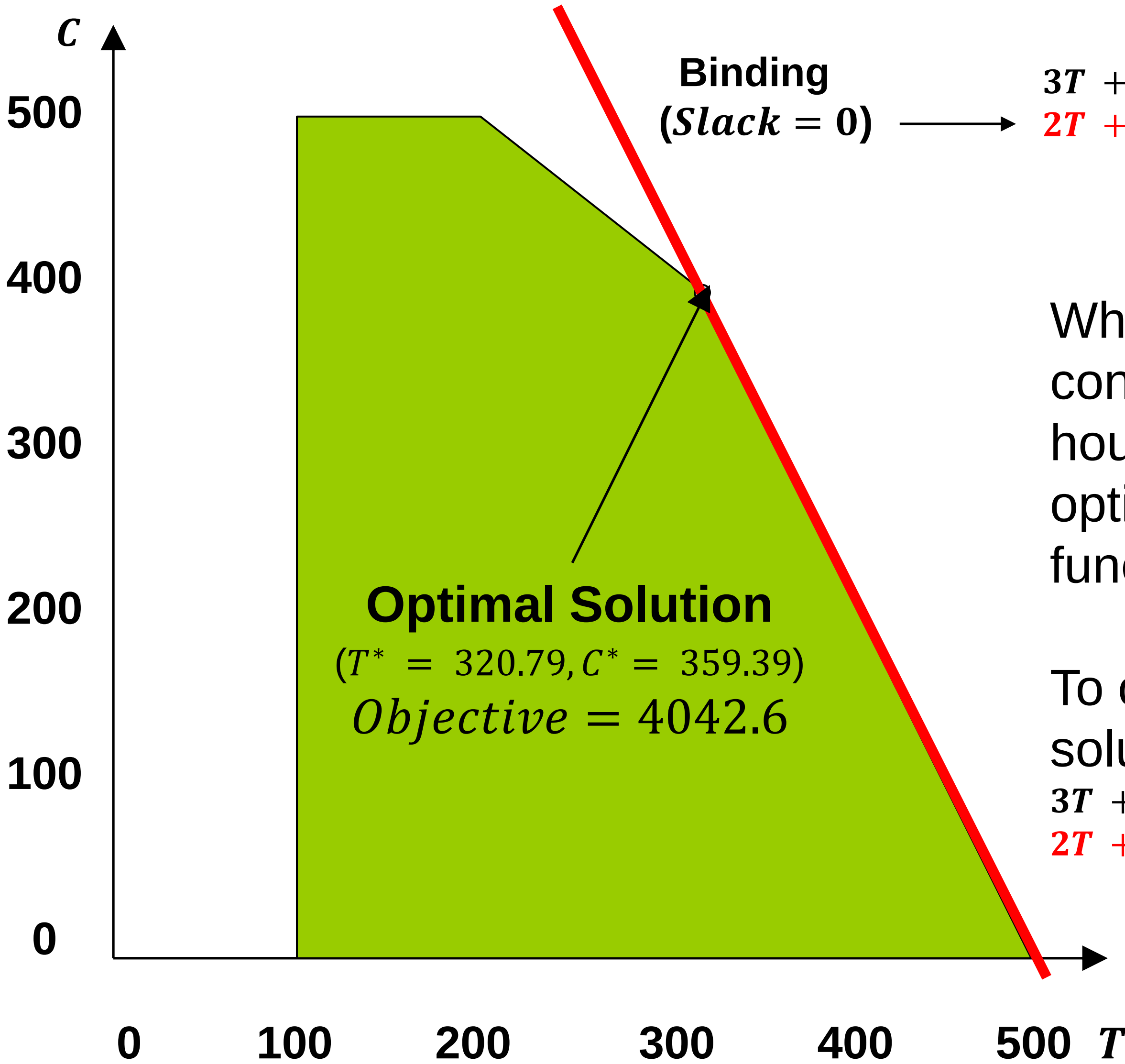
$2T + 1C \leq 1000$ (painting hrs)

$C \leq 450$ (max # chairs)

$T \geq 100$ (min # tables)

The RHS of the *max number of tables constraint* can be decreased infinitely without affecting feasibility. Hence, the allowable decrease is $inf$. However, the RHS can only be increased by $220$ (to $320$) until it becomes a binding constraint and thus, defines an optimal solution.

# Flair Furniture Example: Binding Constraints




$3T + 4C \leq 2400$ (carpentry hrs)
$2T + 1C \leq 1001$ (painting hrs)
$C \leq 450$ (max # chairs)
$T \geq 100$ (min # tables)

When the RHS of a binding constraint (e.g., painting hours) is changed, the optimal solution and objective function value change too!

To obtain the new optimal solution, solve the system:

$3T + 4C = 2400$ (carpentry hrs)
$2T + 1C = 1001$ (painting hrs)

# Flair Furniture Example: Binding Constraints

- To obtain the **original** optimal solution, solve the system:

$$\mathbf{3T + 4C = 2400} \quad \textbf{(carpentry hrs)}$$
$$\mathbf{2T + 1C = 1000} \quad \textbf{(painting hrs)}$$

- The optimal solution is:
  - $T^* = 320, C^* = 360$ and the objective is $4040$.

- To obtain the **new** optimal solution, solve the system:

$$\mathbf{3T + 4C = 2400} \quad \textbf{(carpentry hrs)}$$
$$\mathbf{2T + 1C = 1001} \quad \textbf{(painting hrs)}$$

- The optimal solution is:
  - $T^* = 320.79, C^* = 359.39$ and the objective is $4042.6$.

The difference in objective function values is $\mathbf{4042.6 - 4040 = 2.6}$ is the **shadow price** for that constraint.

# Flair Furniture Example: Sensitivity Analysis

**Shadow Prices**

**Range of Feasibility**

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Changes to RHS Values

If the change in the RHS is **within** the **range of feasibility**, we do not have to re-solve the LP to know what happens to the objective function.

$$\Delta\textbf{Objective} = \Delta\textbf{RHS} \times \textbf{Shadow Price}$$

To find the new ***optimal solution***, one must re-solve the LP or solve the system of equations the define the current corner point, regardless of whether the change in the RHS is within the **range of feasibility** or outside of this range.

# Flair Furniture Example: Sensitivity Analysis

- If we can get an extra hour of Carpentry, the profit increases by $0.6. If we lose an hour of Carpentry, the profit decreases by $0.6.
- If we can get an extra hour of Painting, the profit increases by $2.6. If we lose an hour of Painting, the profit decreases by $2.6.

**Profit is more sensitive to changes in the *Painting* constraint than in the *Carpentry* constraint.**

Variable Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$10 | Tables | 320 | 0 | 7 | 3 | 3.25 |
| $C$10 | Chairs | 360 | 0 | 5 | 4.333333333 | 1.5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$16 | Carpentry Used | 2400 | 0.6 | 2400 | 225 | 900 |
| $D$17 | Painting Used | 1000 | 2.6 | 1000 | 600 | 150 |
| $D$18 | Minimum Tables Used | 320 | 0 | 100 | 220 | 1E+30 |
| $D$19 | Maximum Chairs Used | 360 | 0 | 450 | 1E+30 | 90 |

# Why are Shadow Prices Useful?

| MAXIMIZATION PROBLEM | Constraints | | |
|---|---|---|---|
| **Constraints** | **Allowable Increase** | **Allowable Decrease** | **Shadow Price** |
| Labor ($/hour) | 80 | 60 | 20 |
| Raw Material ($/unit) | Infinity | 30 | 0 |
| Inventory Policy ($/unit) | 4 | 10 | 140 |

- After some market/financial research, you find three options:
  - Hire more employees, which would cost $10 / hour of labor.
  - Buy more raw material for $5 / unit.
  - Increase the inventory capacity, which costs $120 / extra unit.

- Assume you can choose only one option. Which would you choose to maximize the profit associated with production?

# Why are Shadow Prices Useful?

| MAXIMIZATION PROBLEM | Constraints | | |
|---|---|---|---|
| **Constraints** | **Allowable Increase** | **Allowable Decrease** | **Shadow Price** |
| Labor ($/hour) | 80 | 60 | 20 |
| Raw Material ($/unit) | Infinity | 30 | 0 |
| Inventory Policy ($/unit) | 4 | 10 | 140 |

- The marginal profit for hiring one more employee is (shadow price – cost)
  = (20-10) = $10
- The marginal profit for buying an extra unit of raw material is (shadow price – cost)
  = 0 – 5 = -$5
- The marginal profit for increasing the inventory capacity is (shadow price – cost)
  = 140 – 120 = $20

**Thus, considering the marginal profits, the best solution is to increase the inventory capacity.**

# Crop Allocation

Margaret Schlass is an American farmer with 500 acres of land who specializes in growing certified organic crops. She has a three-plant rotation: **oats, maize, and soybean**. Each winter, Margaret decides how much land to devote to each crop.

# Crop Allocation

At least 200 tons of oats and 260 tons of maize are needed for cattle feed. These amounts can also be bought from other farms. She can also sell what she grows. Over last decade, mean selling prices of oats and maize have been $220 and $260 per ton. Purchase prices are 20% more due to transportation and shipping costs. The selling price of Soybean is $55 per ton. However, the US department of agriculture has imposed a quota of 7000 tons. Soybean sold in excess of this quota are priced at $26 per ton.

| | Purchase Price (Per Ton) | Mean Selling Price Below Quota (Per Ton) | Mean Selling Price Above Quota (Per Ton) |
|---|---|---|---|
| **Oats** | $264 | $220 | - |
| **Maize** | $312 | $260 | - |
| **Soybean** | - | $55 | $26 |

# Crop Allocation

Over the last 10 years, Margaret has kept logs for the mean yield per acre. She expects to get 4.25 tons per acre for oats, 3.0 tons per acre for maize, and 20 tons per acre for soybean.

**How much land should Margaret devote to each crop to maximize her expected profits while also ensuring that she has enough food to feed her cattle?**

| | Purchase Price (Per Ton) | Mean Selling Price Below Quota (Per Ton) | Mean Selling Price Above Quota (Per Ton) |
|---|---|---|---|
| **Oats** | \$264 | \$220 | - |
| **Maize** | \$312 | \$260 | - |
| **Soybean** | - | \$55 | \$26 |

# Crop Allocation

## Define the objective

***Maximize Margaret's profit***

## Define the decision variables

$x_i$ = *acres of land devoted to crop* $i$ *where* $i = \{1,2,3\} = \{oat, maize, soybean\}$

$y_i$ = *tons of crop* $i$ *purchased where* $i = \{1,2\} = \{oat, maize\}$

$w_i$ = *tons of crop* $i$ *sold where* $i = \{1,2,3,4\} = \{oat, maize, soybean\ high, soybean\ low\}$

# Crop Allocation

## Write the mathematical objective function

**Maximize** $Z =$

| | Purchase Price (Per Ton) | Mean Selling Price Below Quota (Per Ton) | Mean Selling Price Above Quota (Per Ton) |
|---|---|---|---|
| **Oats** | \$264 | \$220 | - |
| **Maize** | \$312 | \$260 | - |
| **Soybean** | - | \$55 | \$26 |

# Crop Allocation

**Write the mathematical objective function**

**Maximize** $Z =$

$= 220w_1 + 260w_2 + 55w_3 + 26w_4$ **(selling prices)**

| | Purchase Price (Per Ton) | Mean Selling Price Below Quota (Per Ton) | Mean Selling Price Above Quota (Per Ton) |
|---|---|---|---|
| **Oats** | $264 | $220 | - |
| **Maize** | $312 | $260 | - |
| **Soybean** | - | $55 | $26 |

# Crop Allocation

## Write the mathematical objective function

**Maximize** $Z =$

$$= 220w_1 + 260w_2 + 55w_3 + 26w_4$$ **(selling prices)**

$$- 264y_1 - 312y_2$$ **(purchase costs)**

| | Purchase Price (Per Ton) | Mean Selling Price Below Quota (Per Ton) | Mean Selling Price Above Quota (Per Ton) |
|---|---|---|---|
| **Oats** | $264 | $220 | - |
| **Maize** | $312 | $260 | - |
| **Soybean** | - | $55 | $26 |

# Crop Allocation

## Formulating the constraints

There are four types of constraints:

1. Land capacity constraint
2. Cattle feed constraints
3. Quota constraints
4. Non-negativity constraints

# Crop Allocation

## Formulating the land capacity constraint

*Margaret Schlass is an American farmer with 500 acres of land who specializes in growing certified organic crops.*

# Crop Allocation

## Formulating the land capacity constraint

*Margaret Schlass is an American farmer with 500 acres of land who specializes in growing certified organic crops.*

$$x_1 + x_2 + x_3 \leq 500$$ **(acreage)**

# Crop Allocation

## Formulating the cattle feed constraint

*At least 200 tons of oats and 260 tons of maize are needed for cattle feed. These amounts can also be bought from other farms. Over the last 10 years, Margaret has also kept track of the mean yield per acre. She expects to get 4.25 tons per acre for oats and 3.00 tons per acre for maize.*

# Crop Allocation

## Formulating the cattle feed constraint

*At least 200 tons of oats and 260 tons of maize are needed for cattle feed. These amounts can also be bought from other farms. Over the last 10 years, Margaret has also kept track of the mean yield per acre. She expects to get 4.25 tons per acre for oats and 3.00 tons per acre for maize.*

$$4.25x_1 + y_1 - w_1 \geq 200 \quad \textbf{(oats)}$$

# Crop Allocation

## Formulating the cattle feed constraint

*At least 200 tons of oats and 260 tons of maize are needed for cattle feed. These amounts can also be bought from other farms. Over the last 10 years, Margaret has also kept track of the mean yield per acre. She expects to get 4.25 tons per acre for oats and 3.00 tons per acre for maize.*

$$4.25x_1 + y_1 - w_1 \geq 200 \quad \textbf{(oats)}$$

$$3.00x_2 + y_2 - w_2 \geq 260 \quad \textbf{(maize)}$$

# Crop Allocation

## Formulating the quota constraints

*She expects to get 20 tons per acre for soybean and there is no restriction on how much she can produce. However, the US department of agriculture has imposed a quota of 7000 tons for the high selling price of soybeans.*

# Crop Allocation

## Formulating the quota constraints

*She expects to get 20 tons per acre for soybean and there is no restriction on how much she can produce. However, the US department of agriculture has imposed a quota of 7000 tons for the high selling price of soybeans.*

$$w_3 + w_4 = 20x_3$$ **(soybean production)**

# Crop Allocation

## Formulating the quota constraints

*She expects to get 20 tons per acre for soybean and there is no restriction on how much she can produce. However, the US department of agriculture has imposed a quota of 7000 tons for the high selling price of soybeans.*

$w_3 + w_4 = 20x_3$ **(soybean production)**

$w_3 \leq 7000$ **(high selling price quota)**

# Crop Allocation

**Maximize** $Z = 220w_1 + 260w_2 + 55w_3 + 26w_4 - 264y_1 - 312y_2$

**Subject to:**

| | |
|---|---|
| $x_1 + x_2 + x_3 \leq 500$ | (Acreage constraint) |
| $4.25x_1 + y_1 - w_1 \geq 200$ | (Oats constraint) |
| $3.00x_2 + y_2 - w_2 \geq 260$ | (Maize constraint) |
| $w_3 \leq 7000$ | (High selling price quota) |
| $w_3 + w_4 = 20x_3$ | (Soybean production) |
| $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, y_1 \geq 0,$ | (Non-negativity constraints) |
| $y_2 \geq 0, w_1 \geq, w_2 \geq 0, w_3 \geq 0, w_4 \geq 0$ | (Non-negativity constraints) |

# Crop Allocation: Python Solution

- This is an example of aggregate planning.
- We are making an allocation decision now to maximize potential profit in the future.
- How can we use the sensitivity analysis information to comment on the robustness of the crop allocation this year?

**What managerial intuition do you get from the Python solution?**

# Reduced Cost vs. Shadow Prices

Reduced costs and shadow prices are related:

**Reduced cost of a variable** = **OFC of that variable** - **Sum (over all constraints) of the shadow prices multiplied by the constraint coefficients of that variable**

**Intuition:** Increasing a unit of a variable **improves** the objective (OFC) but incurs a **cost** (resource consumption).

# Reduced Cost vs. Shadow Prices

| Variable | Current Coefficient | Reduced Costs |
|---|---|---|
| x | $200 | 0.00 |

| Constraints | Shadow Prices |
|---|---|
| Labor | 20 |
| Raw Material | 0 |
| Inventory Policy | 140 |

Maximize $Z = 200x + 300y$
subject to

$3x + 8y \leq 240$ (labor)

$6x + 3.5y \leq 210$ (raw material)

$1x + y \leq 40$ (inventory)

$x, y \geq 0$

If you increase $x$ by one unit, how much can you ***gain***?

- Answer: $200, because you get profit from one additional unit in the objective function.

# Reduced Cost vs. Shadow Prices

| Variable | Current Coefficient | Reduced Costs |
|---|---|---|
| x | $200 | 0.00 |

| Constraints | Shadow Prices |
|---|---|
| Labor | 20 |
| Raw Material | 0 |
| Inventory Policy | 140 |

Maximize $Z = 200x + 300y$
subject to

$3x + 8y \leq 240$ (labor)
$6x + 3.5y \leq 210$ (raw material)
$1x + y \leq 40$ (inventory)
$x, y \geq 0$

If you increase $\boldsymbol{x}$ by one unit, how much do you ***lose*** due to the **labor constraint**?

- Answer: You require **3 more hours of labor** (constraint coefficient). Each hour is worth $\$\mathbf{20}$ (shadow price). Thus, $\mathbf{3 \times 20 = \$60.}$

# Reduced Cost vs. Shadow Prices

| Variable | Current Coefficient | Reduced Costs |
|---|---|---|
| x | $200 | 0.00 |

| Constraints | Shadow Prices |
|---|---|
| Labor | 20 |
| Raw Material | 0 |
| Inventory Policy | 140 |

Maximize $Z = 200x + 300y$
subject to

$$3x + 8y \leq 240 \quad \text{(labor)}$$
$$6x + 3.5y \leq 210 \quad \text{(raw material)}$$
$$1x + y \leq 40 \quad \text{(inventory)}$$
$$x, y \geq 0$$

If you increase $\boldsymbol{x}$ by one unit, how much do you **lose** due to the **raw material constraint**?

- Answer: You require **6 more units of raw material** (constraint coefficient). Each unit is worth $\$\mathbf{0}$ (shadow price). Thus, $\mathbf{6} \times \mathbf{0} = \$\mathbf{0}$.

# Reduced Cost vs. Shadow Prices

| Variable | Current Coefficient | Reduced Costs |
|---|---|---|
| x | $200 | 0.00 |

| Constraints | Shadow Prices |
|---|---|
| Labor | 20 |
| Raw Material | 0 |
| Inventory Policy | 140 |

Maximize $Z = 200x + 300y$
subject to

$3x + 8y \leq 240$ (labor)
$6x + 3.5y \leq 210$ (raw material)
$1x + y \leq 40$ (inventory)
$x, y \geq 0$

If you increase $\boldsymbol{x}$ by one unit, how much do you **lose** due to the **inventory constraint**?

- Answer: You require **1 more inventory unit** (constraint coefficient). Each unit is worth $\$\mathbf{140}$ (shadow price). Thus, $\mathbf{1} \times \mathbf{140} = \$\mathbf{140}$**.**

# Reduced Cost vs. Shadow Prices

| Variable | Current Coefficient | Reduced Costs |
|---|---|---|
| x | $200 | 0.00 |

| Constraints | Shadow Prices |
|---|---|
| Labor | 20 |
| Raw Material | 0 |
| Inventory Policy | 140 |

Maximize $Z = 200x + 300y$
subject to

$$3x + 8y \leq 240 \quad \text{(labor)}$$
$$6x + 3.5y \leq 210 \quad \text{(raw material)}$$
$$1x + y \leq 40 \quad \text{(inventory)}$$
$$x, y \geq 0$$

The reduced cost of a variable is the net marginal value of an extra unit of that variable:

**Gain per unit – Cost per unit**

**Reduced cost of** $x = \underbrace{\$200}_{\text{Gain per unit}} - \underbrace{3 \times 20 - 6 \times 0 - 140 \times 1}_{\text{Cost per unit}} = 0.00$

# Linear Programming Duality

# Linear Programming Duality

Consider the following linear program where we wish to find the optimal values for $x_i$ where $i = 1, \ldots n$.

$$\max \;\; z = \sum_{i=1}^{n} c_i \, x_i$$

$$\sum_{i=1}^{n} a_{ij} x_i \leq b_j \quad for \; all \; j = 1, \ldots, m$$

$$x_i \geq 0 \quad for \; all \; i = 1, \ldots, n$$

# Linear Programming Duality

Using vector and matrix notation, we get:

- $\boldsymbol{c} \in \mathbb{R}^{\boldsymbol{n}}$ is a vector of cost coefficients
- $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{m} \times \boldsymbol{n}}$ is matrix of constraint parameters
- $\boldsymbol{b} \in \mathbb{R}^{\boldsymbol{m}}$ is a vector of RHS values
- $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$ is a vector of decisions variables

$$\max_{\boldsymbol{x} \geq \boldsymbol{0}} \; z = \boldsymbol{c}^{\boldsymbol{T}} \boldsymbol{x} \;\; s.t. \; \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \quad (P)$$

# Linear Programming Duality

Consider the following linear program where we wish to find the optimal values for $y_j$ where $i = 1, \ldots m$.

$$\min \; z = \sum_{j=1}^{m} b_j \, y_j$$

$$\sum_{j=1}^{m} a_{ij} y_j \geq c_i \quad for\, all\, i = 1, \ldots, n$$

$$y_j \geq 0 \quad for\, all\, i = 1, \ldots, m$$

# Linear Programming Duality

Using vector and matrix notation, we get:

- $\boldsymbol{c} \in \mathbb{R}^{\boldsymbol{n}}$ is a vector of cost coefficients
- $\boldsymbol{A}^{\boldsymbol{T}} \in \mathbb{R}^{\boldsymbol{n} \times \boldsymbol{m}}$ is matrix of constraint parameters
- $\boldsymbol{b} \in \mathbb{R}^{\boldsymbol{m}}$ is a vector of RHS values
- $\boldsymbol{y} \in \mathbb{R}^{\boldsymbol{m}}$ is a vector of decisions variables

$$\min_{\boldsymbol{y} \geq \boldsymbol{0}} \; w = \boldsymbol{b}^{\boldsymbol{T}} \boldsymbol{y} \;\; s.t. \; \boldsymbol{A}^{\boldsymbol{T}} \boldsymbol{y} \geq \boldsymbol{c} \qquad (D)$$

# Weak Duality

**Theorem**: If there exists a feasible solution $x$ to $(P)$ and a feasible solution $y$ to $(D)$:

$$z \leq w$$

**Proof:** In-class

We call $(P)$ the **primal problem** and $(D)$ the **dual problem**. *Note*: the dual of the dual problem is the original primal problem again.

# Strong Duality

**Theorem**: If there exists finite optimal solutions $x^*$ to $(P)$ and $y^*$ to $(D)$:

$$z^* = w^*$$

**Proof:** In-class

We call $(P)$ the **primal problem** and $(D)$ the **dual problem**. *Note*: the dual of the dual problem is the original primal problem again.

# Linear Programming Duality

To create $(D)$ from $(P)$:

1. Switch the optimization from $\max$ to $\min$.
2. Introduce as many dual variables (i.e., $y$) in $(D)$ as there are constraints in $(P)$.
3. Define as many constraints in $(D)$ as there are variables (i.e., $x$) in $(P)$.
4. Switch the roles of the vector of coefficients ($c$) in the objective function and the vector of right-hand sides ($b$) in the inequalities.
5. Switch the inequalities in the constraints.

# Linear Programming Duality

To create $(D)$ from $(P)$:

$$
\begin{array}{ccc}
\textbf{Primal} & \longleftrightarrow & \textbf{Dual} \\
\text{Max} & \longleftrightarrow & \text{Min} \\
\sum_j a_{ij}x_j \leq b_i & \longleftrightarrow & y_i \geq 0 \\
\sum_j a_{ij}x_j \geq b_i & \longleftrightarrow & y_i \leq 0 \\
\sum_j a_{ij}x_j = b_i & \longleftrightarrow & y_i \gtrless 0 \\
x_j \geq 0 & \longleftrightarrow & \sum_i a_{ij}y_i \geq c_j \\
x_j \leq 0 & \longleftrightarrow & \sum_i a_{ij}y_i \leq c_j \\
x_j \gtrless 0 & \longleftrightarrow & \sum_i a_{ij}y_i = c_j.
\end{array}
$$

Note that if the primal problem is a minimization problem, we move right to left (the dual of the dual problem is the primal).

# Sensitivity Analysis

- In an optimal solution of the primal problem, the **shadow prices** for the constraints represent the optimal solution of the **decision variables** in the dual optimization problem.
  - If a constraint in $(P)$ is *binding* in the optimal solution (its slack is zero), the corresponding dual variable is **non-zero** and vice versa**.**
  - If a constraint in $(P)$ is *non-binding* in the optimal solution (its slack is non-zero), the corresponding dual variable is **zero** and vice versa.

# Complementary Slackness

**Theorem**: If there exists a feasible solution $x$ to $(P)$ and a feasible solution $y$ to $(D)$, then we know both $x$ and $y$ are optimal if

$$\boldsymbol{x}^T(\boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y}) = \boldsymbol{y}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$$

This condition is necessary and sufficient.

**Proof:** In-class

# Reduced Cost vs. Shadow Prices

Reduced costs and shadow prices are related:

**Reduced cost of a variable** = **OFC of that variable** - **Sum (over all constraints) of the shadow prices multiplied by the constraint coefficients of that variable**

**Intuition:** This follows from complementary slackness!

$$\boldsymbol{x}^{T}(\boldsymbol{c}-\boldsymbol{A}^{T}\boldsymbol{y})=\boldsymbol{y}^{T}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})$$

# What is the dual?

Consider the following (primal) problem:

$$\max \ \ z = 5x_1 + 4x_2$$

$$x_1 \le 4$$
$$x_1 + 2x_2 \le 13$$
$$5x_1 + 3x_2 \le 31$$
$$x_1 \ge 0, x_2 \ge 0$$

Notice that any feasible solution, such as $(3,4)$, is a lower bound to the optimal solution.

# What is the dual?

The corresponding dual problem is:

$$\min \quad z =$$

# What is the dual?

The corresponding dual problem is:

$$\min \quad z =$$

$$y_1 \geq 0, y_2 \geq 0, y_3 \geq 0$$

# What is the dual?

The corresponding dual problem is:

$$\min \quad z = 4y_1 + 13y_2 + 31y_3$$

$$y_1 \geq 0, y_2 \geq 0, y_3 \geq 0$$

# What is the dual?

The corresponding dual problem is:

$$\min \quad z = 4y_1 + 13y_2 + 31y_3$$

$$y_1 + y_2 + 5y_3 \geq 5$$

$$y_1 \geq 0, y_2 \geq 0, y_3 \geq 0$$

# What is the dual?

The corresponding dual problem is:

$$\min \quad z = 4y_1 + 13y_2 + 31y_3$$

$$y_1 + y_2 + 5y_3 \geq 5$$
$$2y_2 + 3y_3 \geq 4$$
$$y_1 \geq 0, y_2 \geq 0, y_3 \geq 0$$

Notice that any feasible solution, such as $(1,1,1)$, is an upper bound to the optimal solution.

# What is the dual?

Primal problem:

$$\max \;\; z = 5x_1 + 4x_2$$

$$x_1 \leq 4$$
$$x_1 + 2x_2 \leq 13$$
$$5x_1 + 3x_2 \leq 31$$
$$x_1 \geq 0, x_2 \geq 0$$

Dual problem:

$$\min \;\; z = 4y_1 + 13y_2 + 31y_3$$

$$y_1 + y_2 + 5y_3 \geq 5$$
$$2y_2 + 3y_3 \geq 4$$
$$y_1 \geq 0, y_2 \geq 0, y_3 \geq 0$$

***Solving both problems to optimality gives you equivalent solutions!***

# Dual Example: Python Solution

- There is a correspondence between the **shadow prices** in the primal problem and the **optimal solution** in the dual problem.
- By understanding the relationship between **slack**, **shadow prices**, **reduced costs**, and **optimal solutions**, we can better visualize the geometry of our model.

**What managerial intuition do you get from the Python solution?**

# Tower Research® Example Revisited: Python Solution

# Tower Research® Example Revisited: Python Solution

**Maximize (Primal)** $\boldsymbol{Z} = w_4$

**Subject to:**

$w_1 = 4000 + 1000 + B_1 - 1200 - I_1$ (Balance constraint #1)

$w_2 = w_1 + B_2 + 4400 - 1.03B_1 - 4800 - I_2$ (Balance constraint #2)

$w_3 = w_2 + B_3 + 5800 + 1.02I_1 - 4212 - 1.03B_2$ (Balance constraint #3)

$w_4 = w_3 + 3000 + 1.02I_2 - 1000 - 1.03B_3$ (Balance constraint #4)

$B_t \leq 3000\ for\ t = 1, \dots, 3$ (Borrowing constraints)

$I_t \geq\ 0\ for\ t = 1, .., 2$ (Non-negativity constraints)

$B_t \geq\ 0\ for\ t = 1, .., 3$ (Non-negativity constraints)

$w_t \geq\ 0\ for\ t = 1, .., 4$ (Non-negativity constraints)

# Tower Research® Example Revisited: Python Solution

**Maximize (Primal)** $\boldsymbol{Z} = w_4 + 0\sum_{t=1}^{2} I_t + 0\sum_{t=1}^{3} B_t + 0\sum_{t=1}^{3} w_t$

**Subject to:**

$w_1 = 4000 + 1000 + B_1 - 1200 - I_1$ (Balance constraint #1)

$w_2 = w_1 + B_2 + 4400 - 1.03B_1 - 4800 - I_2$ (Balance constraint #2)

$w_3 = w_2 + B_3 + 5800 + 1.02I_1 - 4212 - 1.03B_2$ (Balance constraint #3)

$w_4 = w_3 + 3000 + 1.02I_2 - 1000 - 1.03B_3$ (Balance constraint #4)

$B_t \leq 3000\ for\ t = 1, \ldots, 3$ (Borrowing constraints)

$I_t \geq\ 0\ for\ t = 1, . . , 2$ (Non-negativity constraints)

$B_t \geq\ 0\ for\ t = 1, . . , 3$ (Non-negativity constraints)

$w_t \geq\ 0\ for\ t = 1, . . , 4$ (Non-negativity constraints)

# Tower Research® Example Revisited: Python Solution

**Maximize (Primal)** $\boldsymbol{Z} = w_4 + 0\sum_{t=1}^{2} I_t + 0\sum_{t=1}^{3} B_t + 0\sum_{t=1}^{3} w_t$

**Subject to:**

$w_1 = 4000 + 1000 + B_1 - 1200 - I_1$ (Balance constraint #1)

$w_2 = w_1 + B_2 + 4400 - 1.03B_1 - 4800 - I_2$ (Balance constraint #2)

$w_3 = w_2 + B_3 + 5800 + 1.02I_1 - 4212 - 1.03B_2$ (Balance constraint #3)

$w_4 = w_3 + 3000 + 1.02I_2 - 1000 - 1.03B_3$ (Balance constraint #4)

$B_t \leq 3000 \; for \; t = 1, \ldots, 3$ (Borrowing constraints)

---------------------------------------------------------------------------------

$y_t \in \mathbb{R} \;\; for \; t = 1, .., 4$ (Wealth shadow prices)

$z_t \geq 0 \; for \; t = 1, .., 3$ (Borrowing shadow prices)

# Tower Research® Example Revisited: Python Solution

**Maximize (Primal)** $\boldsymbol{Z} = w_4 + 0\sum_{t=1}^{2} I_t + 0\sum_{t=1}^{3} B_t + 0\sum_{t=1}^{3} w_t$

**Subject to:**

$w_1 = 4000 + 1000 + B_1 - 1200 - I_1$ (Balance constraint #1)

$w_2 = w_1 + B_2 + 4400 - 1.03B_1 - 4800 - I_2$ (Balance constraint #2)

$w_3 = w_2 + B_3 + 5800 + 1.02I_1 - 4212 - 1.03B_2$ (Balance constraint #3)

$w_4 = w_3 + 3000 + 1.02I_2 - 1000 - 1.03B_3$ (Balance constraint #4)

$B_t \leq 3000 \; for \; t = 1, \ldots, 3$ (Borrowing constraints)

----------------------------------------------------------------------------------------

$y_4 \geq 1$ (Final period constraint)

$y_t \in \mathbb{R} \;\; for \; t = 1, \ldots, 4$ (Wealth shadow prices)

$z_t \geq 0 \; for \; t = 1, \ldots, 3$ (Borrowing shadow prices)

# Tower Research® Example Revisited: Python Solution

**Minimize (Dual)** $\boldsymbol{Z} = 3800y_1 - 400y_2 + 1588y_3 + 2000y_4$
$+3000\sum_{t=1}^{3} z_t$

**Subject to:**

$w_1 = 4000 + 1000 + B_1 - 1200 - I_1$ (Balance constraint #1)

$w_2 = w_1 + B_2 + 4400 - 1.03B_1 - 4800 - I_2$ (Balance constraint #2)

$w_3 = w_2 + B_3 + 5800 + 1.02I_1 - 4212 - 1.03B_2$ (Balance constraint #3)

$w_4 = w_3 + 3000 + 1.02I_2 - 1000 - 1.03B_3$ (Balance constraint #4)

$B_t \leq 3000 \; for \; t = 1, \ldots, 3$ (Borrowing constraints)

--------------------------------------------------------------------------------

$y_4 \geq 1$ (Final period constraint)

$y_t \in \mathbb{R} \;\; for \; t = 1, .., 4$ (Wealth shadow prices)

$z_t \geq \; 0 \; for \; t = 1, .., 3$ (Borrowing shadow prices)

# Tower Research® Example Revisited: Python Solution

**Minimize (Dual)** $\boldsymbol{Z} = 3800y_1 - 400y_2 + 1588y_3 + 2000y_4 + 3000\sum_{t=1}^{3} z_t$

**Subject to:**

| | |
|---|---|
| $y_1 - y_2 \geq 0$ | (Wealth constraint #1) |
| $y_2 - y_3 \geq 0$ | (Wealth constraint #2) |
| $y_3 - y_4 \geq 0$ | (Wealth constraint #3) |
| $-y_1 + 1.03y_2 + z_1 \geq 0$ | (Borrowing constraint #1) |
| $-y_2 + 1.03y_3 + z_2 \geq 0$ | (Borrowing constraint #2) |
| $-y_3 + 1.03y_4 + z_3 \geq 0$ | (Borrowing constraint #3) |
| $y_1 - 1.02y_3 \geq 0$ | (Inventory constraint #1) |
| $y_2 - 1.02y_4 \geq 0$ | (Inventory constraint #2) |
| $y_4 \geq 1$ | (Final period constraint) |
| $y_t \in \mathbb{R}\ for\ t = 1,..,4$ | (Wealth shadow prices) |
| $z_t \geq\ 0\ for\ t = 1,..,4$ | (Borrowing shadow prices) |

# Tower Research® Example Revisited: Python Solution

- What insight does the dual provide?
- Which formulation, the primal or the dual model, do you think is easier to solve?.
- How do you check whether you have formulated the correct dual problem?

**What managerial intuition do you get from the Python solution?**

# Tower Research® Example Revisited: Python Solution

- What insight does the dual provide?
- Which formulation, the primal or the dual model, do you think is easier to solve?.
- How do you check whether you have formulated the correct dual problem?
  - It is correct when the primal and dual models have **the same objective function value** at optimality (follows from strong duality).

**What managerial intuition do you get from the Python solution?**

# Why is LP duality important?

1. We can create specialized computational algorithms (e.g., *simplex* algorithm versus *dual simplex* algorithm).

2. A primal problem with many constraints and only a few variables can be converted into a dual problem with a few constraints and many variables. Fewer constraints requires fewer computations in the simplex method.

3. It is the basis for obtaining (both analytically and computationally) solutions and algorithmic strategies when solving constrained ***nonlinear*** programs.

# Why is LP duality important?

It forms the basis of many approaches that incorporate uncertainty in optimization models.

– Robust optimization

# Next Class: Constrained Nonlinear Programs

Sometimes, an optimization problem must be formulated with non-linear constraints and/or objective functions. However, there are sometimes tricks that can be used to linearize the relations:

- Absolute value, minimax or maximin, minimizing the sum of deviations, floor/ceiling constraints.

In these cases, the problem can be easily solved. In other cases, the problem is nonlinear.

- Local vs. Global Optima: Convex functions, convex sets, convexity-preserving operations.
- Lagrange multipliers and the KKT conditions.
- Lagrangian duality for nonlinear programs.