Week 4: Introduction to Topics

Assignment Problem

The Assignment Problem involves finding the most efficient route between two points in a graph, typically represented as nodes connected by edges with associated weights. The objective is to minimize the total weight (distance, cost, etc.) required to traverse from a given source node to a destination node. This problem finds applications in logistics, transportation, network routing, and more.

In Excel Solver, you can model the Assignment Problem by defining decision variables that represent the assignment of tasks to agents. The objective function aims to minimize the total cost or maximize the total benefit of the assignments while ensuring that each task is assigned to exactly one agent and each agent is assigned only one task.

Assignment Problem Using Excel Solver:

- Define the objective as minimizing the total cost/time/effort.
- Set up decision variables representing whether a worker is assigned a specific task (binary: 0 or 1).
- Use constraints to ensure each task is assigned to exactly one worker and each worker receives at most one task.
- Solve using Solver with appropriate settings.

Integer Programming:

Integer programming (IP) extends linear programming by allowing variables to take only **integer values** rather than any real value. This means the variables cannot take fractional values such as 2.5 or 0.7, but can only be whole numbers such as 0, 1, 2, and so on. This adds a layer of complexity but opens up many real-world applications.

Integer programming problems arise in various real-world scenarios such as resource allocation, scheduling, network design, production planning, facility location and project selection where decisions need to be made among discrete options or when certain constraints necessitate integer solutions. By formulating problems as integer programming models, decision-makers can make more informed and optimized choices to address complex decision-making challenges.

There are two main types of integer variables:

- **Binary variables:** These can only take values of 0 or 1, often representing yes/no decisions like opening a store or building a factory.
- **Integer variables:** These can take any integer value, useful for decisions such as number of items produced, number of employees assigned, etc.

Using integer variables allows modelling problems such as:

- **Scheduling:** Deciding which days to open a shop or which jobs to assign to machines.

- **Routing:** Finding the shortest path with restrictions such as using only available roads.
- **Project selection:** Deciding which projects to invest in to maximize profit, considering budget and resource limitations.
- **Facility Location:** Where to install facilities such as service centres or warehouses, telecom towers.

Solving Integer programming (IP) problems:

- More complex than solving linear programming due to the discrete nature of integer variables.
- Standard linear programming methods using the Simplex will usually not yield integer valued solutions other than for network structured problems.
    - **Branch and bound:** Standard algorithm for solving integer programming problems that systematically explores possible integer solutions to find the optimal one. In many optimization software packages the algorithm is built-in to solve IP problems, including the EXCEL Solver.
    - **Specialized solvers:** Dedicated IP solvers such as CPLEX, GUROBI, and SCIP offer better performance and guarantees than general-purpose solvers.

While Excel Solver is primarily designed for linear programming, it can also handle **integer programming** with some limitations. Here's how you can solve problems involving binary and integer variables:

1. Define your problem:

- **Objective:** Clearly state what you want to minimize or maximize (e.g., total cost, projected profit).
- **Decision variables:** Identify variables representing your choices, clearly marking binary (0/1) and integer variables.
- **Constraints:** Formulate equations representing limitations and relationships between variables.

2. Set up your Excel spreadsheet:

- Enter your objective function in a cell.
- Define each decision variable in a separate cell.
- Use formulae to calculate other relevant values based on your variables.
- Set up constraints as inequalities in separate rows.

3. Use the Solver add-in:

- Go to the "Data" tab and click "Solver."
- Set the "Set Objective" to the cell containing your objective function.
- Choose "Min" or "Max" depending on your goal.
- Click "By Changing Variable Cells" and select the range of your decision variables.
- Add your constraints one by one.
- Click "Add" and enter the constraint formula in the "Cell Reference" field.
- Choose "<=" or ">=" for the constraint relationship.
- Optionally, select "Bin" for binary variables to ensure they only take 0 or 1 values.
- Click "Add" for each constraint.
- Click "Solve" to start the optimization process.

4. Analyze the results:

- Solver will display the optimal solution, including the values of your decision variables and the achieved objective value.
- Check if the solution meets all constraints and makes sense in your context.
- You can adjust the model and re-run Solver to explore different scenarios or find alternative solutions.

Mixed Integer Programming

Mixed Integer Programming (MIP) refers to problems where the variable values are mixed in nature, that is some variables can take continuous values while others can only take binary or integer values, as required. Here is a breakdown of its key aspects:

Types of Variables:

- **Continuous variables:** As in linear programming, these can take any real number within a defined range.
- **Binary variables:** Can only be 0 or 1, often representing on/off decisions like building a factory (1) or not (0).
- **General integer variables:** Can take any whole number within a specified range, useful for quantities like items produced or employees assigned.

To solve MIP problems using Excel Solver, you need to follow these steps:

1. **Formulate the Problem**: Clearly define the decision variables, objective function, and constraints of your MIP problem. Identify which variables need to be integers or binary.
2. **Set Up Your Excel Spreadsheet**:
    - Define cells for decision variables, objective function, and constraints.
    - Label cells appropriately to indicate the meaning of each variable, objective function, and constraint.
3. **Enter the Objective Function and Constraints**:
    - Enter the coefficients of the objective function and constraints into the appropriate cells.
    - Make sure to include integer and binary constraints as needed.
4. **Set Up Excel Solver**:
    - Go to the "Data" tab in Excel, and then click on "Solver" in the "Analysis" group.
    - In the Solver Parameters dialog box, set the objective function to minimize or maximize, and select the target cell containing the objective function.
    - Define the decision variable cells that Solver should adjust to find the optimal solution.
    - Enter any constraints in the "Subject to the Constraints" box.
    - Specify which variables should be integers or binary.
5. **Run Solver**: Click "Solve" in the Solver Parameters dialog box to let Excel Solver find the optimal solution.

Branch and Bound Method:

The branch and bound method is an effective algorithm for solving **integer linear programming (ILP)** problems, where some or all decision variables must be integers. It relies on two key principles:

1. Branching: This involves dividing the problem into smaller subproblems by adding constraints that force certain variables to take on specific range of values. Think of it like creating branches on a tree, where each branch represents a different possibility.

2. Bounding: For each subproblem, we obtain an upper/lower bound on the objective function value going down that branch. This upper/lower bound is commonly obtained by solving the corresponding **linear programming (LP) relaxation**, which is the same problem but with all variables allowed to be continuous. Subproblems whose upper bounds are worse than the current best integer solution can be discarded, as they cannot possibly lead to an improved solution.

Here is how the branch and bound method works in general:

1. **Solve the LP relaxation:** This gives an initial upper bound on the optimal solution value and can often be solved efficiently.
2. **Choose a branching variable:** Select a non-integer variable from the LP solution.
3. **Create branches:** Add two new constraints to the LP problem, one forcing the chosen variable to be less than or equal to the floor of its LP solution, and the other forcing it to be greater than or equal to the ceiling. This creates two new subproblems to be solved and the procedure is repeated.
4. **Current best solution:** If the solution to a subproblem is integer valued then save it as the current best solution.
5. **Update best solution:** If the solution to a subproblem is integer valued and better than the current best, update the current best.
6. **Discard fathomed subproblems:** If a subproblem's upper/lower bound is worse than the current best solution, discard it as it cannot lead to a better solution.
7. **Repeat steps 3-6:** Choose another subproblem to solve and repeat the process until all promising subproblems have been explored.
8. **Return the best solution found.**

The branch and bound method can be quite complex, with various strategies for choosing branching variables and prioritizing subproblems. However, it is a powerful tool for solving ILP problems.