Operations Research Interview Questions & Preparation

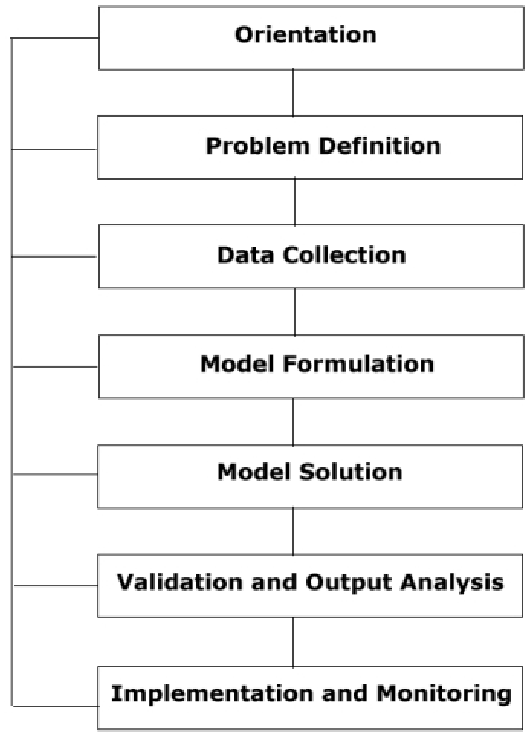
(ongoing)

Compiled by Tanmoy Das

Table of Contents

1 Basic of Operations Research

1. What are the steps of problem solving in OR?



https://sites.pitt.edu/~jrclass/or/or-intro.html

https://web.mit.edu/urban_or_book/www/book/chapter1/1.3.html

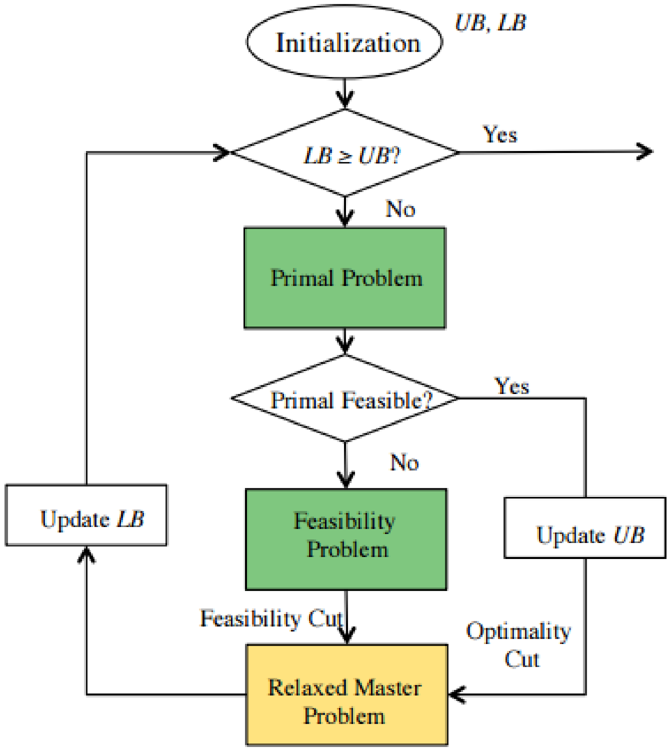
2. What do you do for infeasibility analysis?
3. Problem degeneration
4. Convex and concave functions
5. Mathematical programming formula (from verbal to math)
    a. Practise a few+++

2 Algorithm

2.1 Integer Programming

2.1.1 B&B

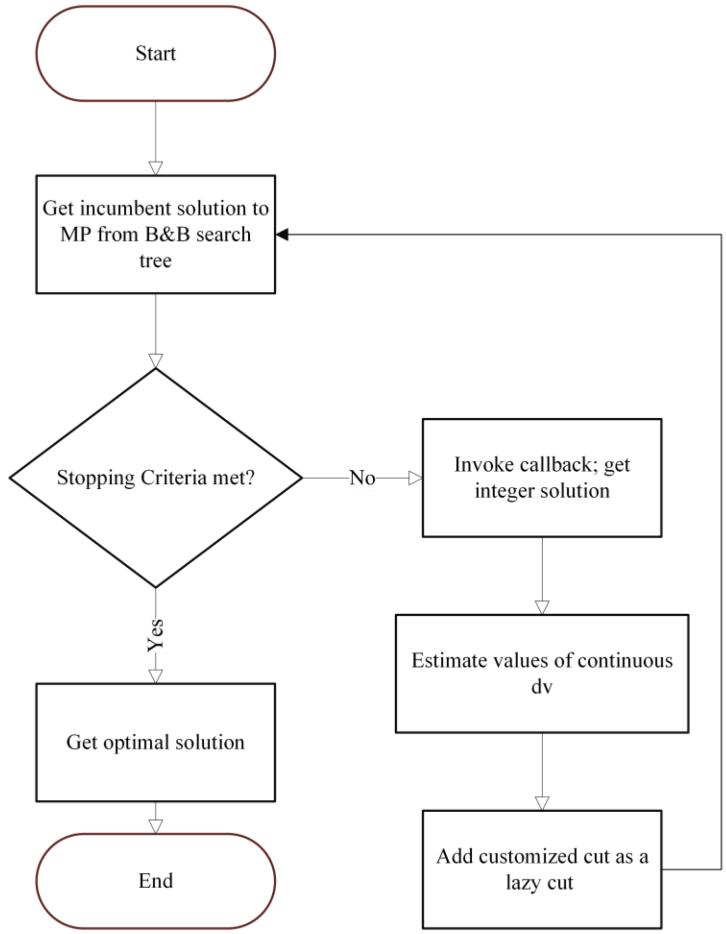
1. What are the steps in Benders Decomposition (UPS)



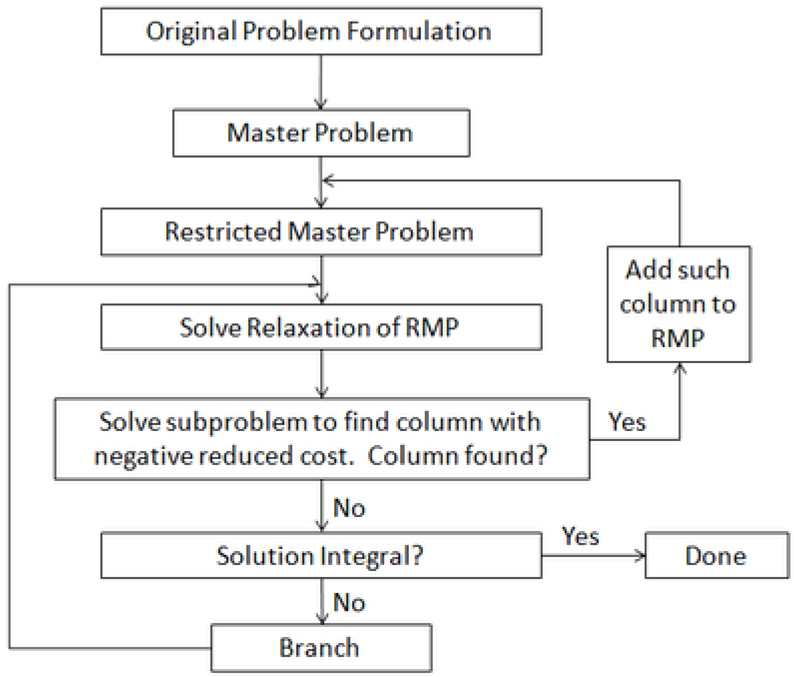
Finite convergence proof see to an optimum with a given tolerance is guaranteed if y is integer or f(.,y), g(.,y) are linear for all y in Y.

https://optimization.cbe.cornell.edu/index.php?title=Generalized_Benders_decomposition_%28GBD%29

2.1.2 Branch & Cut



2.1.3 B&P



https://www.wikiwand.com/en/Branch_and_price

2.1.4 Cutting plane

2.1.5 Column Generation

1. How to use column generation for VRP problem?
    a. Generic: Column generation or delayed column generation is an efficient algorithm for solving large linear programs.
    The overarching idea is that many linear programs are too large to consider all the variables explicitly. The idea is thus to start by solving the considered program with only a subset of its variables. Then iteratively, variables that have the potential to improve the objective function are added to the program.
    https://medium.com/@sean-patrick-kelley/how-to-implement-column-generation-for-vehicle-routing-bdb8027c957f

2.1.6 Other decomposition methods

decomposition methods? Benders, Column Generation, Branch and Bound, Branch and Cut, Branch and price, Dantzig wolf decomposition

Dantzig wolf decomposition

While there are several variations regarding implementation, the Dantzig–Wolfe decomposition algorithm can be briefly described as follows:

1. Starting with a feasible solution to the reduced master program, formulate new objective functions for each subproblem such that the subproblems will offer solutions that improve the current objective of the master program.
2. Subproblems are re-solved given their new objective functions. An optimal value for each subproblem is offered to the master program.
3. The master program incorporates one or all of the new columns generated by the solutions to the subproblems based on those columns' respective ability to improve the original problem's objective.
4. Master program performs *x* iterations of the simplex algorithm, where *x* is the number of columns incorporated.
5. If objective is improved, goto step 1. Else, continue.
6. The master program cannot be further improved by any new columns from the subproblems, thus return.

https://en.wikipedia.org/wiki/Dantzig%E2%80%93Wolfe_decomposition

2.2 Linear Programming

1. What are the different solution techniques for LPs?
    a. Graphical method, Simplex, interior point algorithms, and the Ellipsoid Method
    b.

2.2.1 Simplex

1. Optimality conditions in Simplex
    a. The entering variable in a maximization (minimization) problem is the non-basic variable having the most negative (positive) coefficient in the Z-row. The optimum is reached at the iteration where all the Z-row coefficient of the non-basic variables are non-negative (non-positive).
    b. The optimality condition is the reduced-cost condition. When reduced costs are positive, you have the optimum
    c.
2. 3 rules in converting primal to dual?
3. When to use primal vs dual?

Steps in SIMPLEX algorithm

I. Standard form
II. Introducing slack variables
III. Creating the tableau
IV. Pivot variables
V. Creating a new tableau
VI. Checking for optimality
VII. Identify optimal values

Step 1: Establish a given problem. (i.e.,) write the inequality constraints and objective function.

Step 2: Convert the given inequalities to equations by adding the slack variable to each inequality expression.

Step 3: Create the initial simplex tableau. Write the objective function at the bottom row. Here, each inequality constraint appears in its own row. Now, we can represent the problem in the form of an augmented matrix, which is called the initial simplex tableau.

Step 4: Identify the greatest negative entry in the bottom row, which helps to identify the pivot column. The greatest negative entry in the bottom row defines the largest coefficient in the objective function, which will help us to increase the value of the objective function as fastest as possible.

Step 5: Compute the quotients. To calculate the quotient, we need to divide the entries in the far right column by the entries in the first column, excluding the bottom row. The smallest quotient identifies the row. The row identified in this step and the element identified in the step will be taken as the pivot element.

Step 6: Carry out pivoting to make all other entries in column is zero.

Step 7: If there are no negative entries in the bottom row, end the process. Otherwise, start from step 4.

Step 8: Finally, determine the solution associated with the final simplex tableau.

https://byjus.com/maths/linear-programming/#:~:text=The%20linear%20programming%20problem%20can,and%20graphical%20method%20in%20detail.

Sensitivity analysis and reduced cost

2.2.2 Dual Simplex

2.3 NLP

1. Contraint optimisation (conditions, Lagrangien relaxation, Quasi-newton, Rampson)

2.4 MIP

1. What's the difference between LP & MIP?
    a. Continuous DV vs mix of integer and continuous variables
    b. MIP is difficult to solve compared to LP
2. What are the different solution techniques for MIPs?
    a.
3. ++

Logical transformation

Piecewise linear transformation

2.4.1 Exact Method

2.4.2 Heuristics

2.4.3 Meta-heuristics

2.4.3.1 *GA*

2.4.3.2 *PSO*

3 Problem Type

3.1 Knap-sack problem

3.2 Set Covering

$$
\begin{aligned}
\max \quad & \sum_j a_j y_j \\
\text{s.t} \quad & \sum_{i \in C_j} x_i \geq y_j \quad \forall j \\
& \sum_i x_i = k \\
& x_i, y_j \in \{0, 1\}
\end{aligned}
$$

https://optimization.cbe.cornell.edu/index.php?title=Set_covering_problem

https://en.wikipedia.org/wiki/Set_cover_problem

3.3 P-median

The “p-median” problem is a distance based optimization problem in which **p facilities** need to be located and assigned to the demand points such that each demand point is mapped to a single facility, and the sum of the weighted distance between all demand points and corresponding facilities is minimized.

3.4 MCLP

Maximal Covering Location Problem

$$
\begin{aligned}
\max \; & \sum_{i \in I} h_i y_i \\
s.t. \; & \\
& \sum_{j \in N_i} x_j \geq y_i, && i \in I \\
& \sum_{j \in J} x_j = p, && \\
& x_j \in \{0,1\}, && j \in J \\
& y_i \in \{0,1\}, && i \in I
\end{aligned}
$$

https://en.wikipedia.org/wiki/Maximum_coverage_problem

3.5 Location Allocation

1. Can you write a simple facility allocation model? When we use location-allocation modeling?

Verbal:

Objective: Min cost, $FixedCost_f \, .open_f$

St. number of facility <= 10

Allocation_f,c <= availability_f

3.6 Network Optimization

3.6.1 VRP formulation

3.6.2 TSP formulation

$$\min \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij} \colon$$

$$\begin{aligned}
x_{ij} &\in \{0,1\} && i,j = 1,\ldots,n; \\
u_i &\in \mathbf{Z} && i = 2,\ldots,n; \\
\sum_{i=1, i\neq j}^{n} x_{ij} &= 1 && j = 1,\ldots,n; \\
\sum_{j=1, j\neq i}^{n} x_{ij} &= 1 && i = 1,\ldots,n; \\
u_i - u_j + (n-1)x_{ij} &\leq n-2 && 2 \leq i \neq j \leq n; \\
2 \leq u_i &\leq n && 2 \leq i \leq n.
\end{aligned}$$

4 Computational Complexity

1. Please describe one **linear optimization problem/algorithm** and state its **computational complexity**.

Linear programming:

Linear programming is **an optimization technique for a system of linear constraints and a linear objective function**. An objective function defines the quantity to be optimized, and the goal of linear programming is to find the values of the variables that maximize or minimize the objective function.

Complexity

The complexity of an optimization algorithm depends on the following factors:

1- Number of iterations, 2- Number of individuals in the population, 3-complexity of the objective function, 4- If you sort the individuals, the complexity of sorting should be added

Example O(N*M*P*Q) where N is #1, M is #2, P is #3, Q is #4

The better estimation for #1 and #2 is computing the NFE(number of function evaluations) source: https://www.researchgate.net/post/How-can-I-calculate-the-computational-complexity-of-any-optimization-algorithm

The computational complexity, in general, depends on the optimization algorithm and the technique that you use. In some algorithms, the complexity can be measured by the time that the CPU needs to run the algorithm, others consider the computational complexity as the number of nested loops (for loops and others) per run and can be written as O(x), where x is your nested loops.

5 Uncertainty

5.1 Robust optimization

5.2 Stochastics Optimization

6 Large Scale Optimization

1. Tell me about large-scale optimization
2. How do you solve them?
3. What heuristics you will apply for large-scale?

7 Gurobi

1. How you set parameters of B&B
2. How to handle infeasibility?
    a. ComputeIIS
    b. How to manually check infeasibility?
3. Warm start
    a. How to use pretrained optimization model as initial sol

8 MCQs

1. If the feasible set of an optimization problem is unbounded, which of the following is true? (C3AI-T)
    a. No finite optimum point exists
    b. It has an infinite number of feasible points
    c. The existence of finite optimum points can not be assured
    d. None of the above.