CHAPTER ONE

COMBINATORIAL OPTIMIZATION PROBLEM

1.1 Introduction

Optimization Problem: In mathematics and computer science, an optimization problem (OP) is the problem of finding the best solution from all feasible solutions. OPs can be divided into two categories depending on whether the variables are continuous or discrete.

Combinatorial Optimization: In applied mathematics and theoretical computer science, **combinatorial optimization** (CO) is a topic that consists of finding an optimal object from a finite set of objects.

Combinatorial Optimization Problem: An OP with discrete variables is known as a **combinatorial optimization problem** (COP). In a COP, we are looking for an object such as an integer, permutation or graph from a finite set.

The aim of is to investigate the use of various optimization exacts and heuristics to solve the COP's.

1.2 Background

In many such problems, exhaustive search is not feasible. It operates on the domain of those optimization problems, in which the set of feasible solutions is discrete or can be reduced to discrete, and in which the goal is to find the best solution.

Related Topics for CO:

CO is a subset of mathematical optimization that is related to:

- Operations research.

- Algorithm theory and computational complexity theory.
- Artificial intelligence.
- Machine learning.
- Mathematics.
- Auction theory.
- Software engineering.

Applications for CO:

Applications for CO include, but are not limited to:

- Developing the best airline network of spokes and destinations.
- Deciding which taxis in a fleet to route to pick up fares.
- Determining the optimal way to deliver packages.

COP's can be viewed as searching for the best element of some set of discrete items; therefore, in principle, any sort of search algorithm or metaheuristic can be used to solve them. However, generic search algorithms are not guaranteed to find an optimal solution, nor are they guaranteed to run quickly (in polynomial time).

Specific Problems of COP:

The most specific problems of COP are [89]:

- Assignment problem.
- Closure problem.
- Constraint satisfaction problem.
- Cutting stock problem.
- Integer programming.
- Knapsack problem.
- Linear programming.
- Minimum spanning tree.

- Nurse scheduling problem
- Traveling salesman problem.
- Vehicle routing problem.
- Vehicle rescheduling problem.
- Weapon target assignment problem.

The aim of CO is to provide efficient techniques for solving mathematical and engineering related problems. These problems are predominantly from the set of NP-complete problems. Solving such problems requires effort (e.g., time and/or memory requirement) which increases dramatically with the size of the problem. Thus, for sufficiently large problems, finding the best (or optimal) solution with certainty is often infeasible. In practice, however, it usually suffices to find a "good" solution (the optimality of which is less certain) to the problem being solved. A subtle point to note is this algorithm designed to find "good" solutions to a problem might find the optimal solution however; it is infeasible to prove that the solution is in fact the optimal one.

1.3 CO Concept

Many problems arising in practical applications have a special discrete and finite nature, for examples to find the minimal value of COP: Shortest Path, Scheduling, Travelling Salesman Problem and many more.

Note that a large number of real-word planning problems called COP share the following:

1. They are optimization problems.
2. They are easy to state.
3. They have a finite usually very large number of feasible solutions.

4. The majority of these problems share the property that no polynomial method for their solution is known.

   Now we shall present an example of COP.

Example (1.1): The following problem is called the **Knapsack problem**. We are given an amount of C Euro and wish to invest it among a set of n options. Each such option i has cost c_i and profit p_i. The goal is to maximize the total profit.

Consider C = 100 and the following cost-profit table:

Option	Cost (c_i)	Profit (p_i)
1	100	150
2	1	2
3	50	55
4	50	100

Our choice of purchased options must not exceed our capital C. Thus the feasible solutions are {1}, {2}, {3}, {4}, {2,3}, {2,4}, {3,4}. Which is the best solution? We evaluate all possibilities and find that {3,4} give 155 altogether which maximizes our profit.

COP Formulation:

An instance I of a COP can formally be defined as a tuple $I = (U, P, val, extr)$ with the following meaning:

U : the solution space (on which *val* and S are defined),

P : the feasibility predicate,

val : the objective function $val : U \to R$,

extr : the extremum (usually max or min).

The feasibility predicates P induces a set:

S the set of feasible solutions: $S = \{X \in U : X \text{ satisfies } P\}$.

Our goal is to find a feasible solution where the desired extremum of *val* is attained. Any such solution is called an **optimum solution**, or simply an **optimum**. *U* and *S* are usually not given explicitly, but implicitly.

Let us investigate the problem in Example (1.1) in with this formalism.

$U = 2^{\{1,2,3,4\}}$

P ="total cost is at most C", i.e., $X \in S$ if $\sum_{i \in X} c_i \leq C$

$S = \{\{1\},\{2\},\{3\},\{4\},\{2,3\},\{2,4\},\{3,4\}\}$

$$val = \begin{cases} U \to R \\ X \mapsto \sum_{i \in X} p_i \end{cases}$$

extr = max.

The optimum solution here is {3,4} with value 155.