Vehicle Routing Problem

CS21B1023 SK ARIAN ISLAM , CS21B1078 E. SAILESWARA REDDY

April 2024

1 Introduction

This report defines our problem statement : Vehicle Routing. Here one vehicle has a certain amount of a product which costs different in different clusters of nodes.The vehicle has to maximise the profit he makes while keeping the distance covered under a budget.

2 Problem Definition

Under the following points we will define the problem.

2.1 State Configuration

The state of the problem is essentially the sequence of nodes covered from start and reaching back to the starting node(depot), i.e. essentially a subset of the Hamiltonian path , the total distance covered(cost) and the total profit made. The start configuration is a random path in some algorithms and in some it is the starting node(depot).The goal satisfying solution is the one we have at the end of iteration of the algorithm we have chosen.

Decision Variables:

Let:

x_{ij} be a binary variable indicating whether the vehicle travels from node i to node j
($x_{ij} = 1$ if the vehicle travels from node i to node j, and $x_{ij} = 0$ otherwise)
y_i be a binary variable indicating whether the vehicle visits node i
($y_i = 1$ if the vehicle visits node i, and $y_i = 0$ otherwise)

2.2 Constraints

1. Each node j is visited at most once (except depot, which is visited twice):

$$\sum_i x_{ij} = \begin{cases} 2 & \text{if } j = \text{depot} \\ 1 & \text{otherwise} \end{cases} \quad \forall j$$

2. Sub-tour elimination constraint:

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \leq |S| - 1 \quad \forall S \subset \text{Nodes}, |S| \geq 2$$

where $|S|$ represents the number of nodes in subset S.
3. Budget constraint:

$$\sum_i \sum_j d_{ij} \cdot x_{ij} \leq B$$

where B is the total budget and d_{ij} is the distance between node i and j.

2.3 Rules for State Change

The vehicle can visit any node,except depot until the content of the vehicle cannot be sold further or has been completely sold.

2.4 Objective Function

The Objective function is as follows:
Maximize: $\sum_i p_i \cdot x_i$ where $\exists i \in N$ (N is the set of all nodes)

3 General Algorithms

3.1 Depth-First Branch and Bound

This technique combines the power of depth first search with branch and bound. It systematically explores the solution space, identifying promising routes while efficiently pruning branches that lead to sub-optimal solutions.

Mathematical Formulation:

$$\text{Maximize:} \quad \sum_i \sum_j p_{ij} \cdot x_{ij}$$

Subject to:

$$x_{ij} \in \{0, 1\}$$

$$\text{Budget constraint:} \quad \sum_i \sum_j d_{ij} \cdot x_{ij} \leq B$$

Sub-tour elimination constraints

3.2 A* (A-star):

A* is a popular heuristic search algorithm that intelligently explores the search space by balancing between the cost of the path already traversed and the estimated cost to reach the goal. It can be tailored to incorporate domain-specific knowledge, such as product diversity and variable costs, to guide the search towards optimal or near-optimal solutions efficiently.

Mathematical Formulation:

$$F(n) = g(n) + h(n)$$

where $F(n)$ is the estimated total cost of the cheapest solution through node n, $g(n)$ is the cost of the path from the start node to node n, and $h(n)$ is the estimated cost of the cheapest path from node n to the goal.

3.3 Simulated Annealing (SA):

SA is a meta-heuristic optimization technique inspired by the annealing process in metallurgy. It involves iteratively exploring the solution space while allowing for occasional uphill moves to escape local optima. SA can be adapted to tackle the VRP by incorporating mechanisms to handle diverse products and variable costs, providing a robust approach to finding high-quality solutions.

Mathematical Formulation:

$$\text{Energy function: } E(S) = \frac{1}{\text{profit}(S)} \times \text{constant}$$

where S is a state and $\text{profit}(S)$ is the profit achieved by the state.

3.4 Genetic Algorithm (GA):

GA mimics the process of natural selection to iteratively evolve a population of candidate solutions towards optimal or near-optimal solutions. Through mechanisms such as crossover and mutation, GA can effectively explore the solution space, accommodating the complexities of diverse products and variable costs inherent in the VRP.

Mathematical Formulation:

$$\text{Population evolution: } P_{t+1} = \text{Selection}(P_t) \cup \text{Crossover}(P_t) \cup \text{Mutation}(P_t)$$

4 Conclusion

The VRP poses a complex optimization challenge, requiring efficient algorithms to maximize profit while managing costs. The techniques discussed, such as DFBB, A*, SA, and GA, offer diverse approaches to tackling this problem. By intelligently exploring solution spaces and adapting to dynamic environments, these algorithms provide effective solutions to the VRP.