VEHICLE ROUTING PROBLEM

PROBLEM STATEMENT

- We will be given a set of cities or service points and vehicles to service them.
- Single Depot & Multiple Depot with ***at least two vehicles***.
- For only one vehicle the problem will be reduced to TSP.
- We can also consider vehicle capacity (certain material) & customer requirements.
- **Problem:**

  what is the allocation to each vehicle?

  what is the route it should take?

OBJECTIVE

- So the problem can be categorized into two parts:

  Allocation Problem: Which set of cities goes to which vehicle.

  Routing Problem: Order in which the vehicle service, so the travel route is minimized.

- The objective of the problem is to:

  Minimize the transportation cost based on the global distance travelled.

  Minimize the number of vehicles needed to serve all customers.

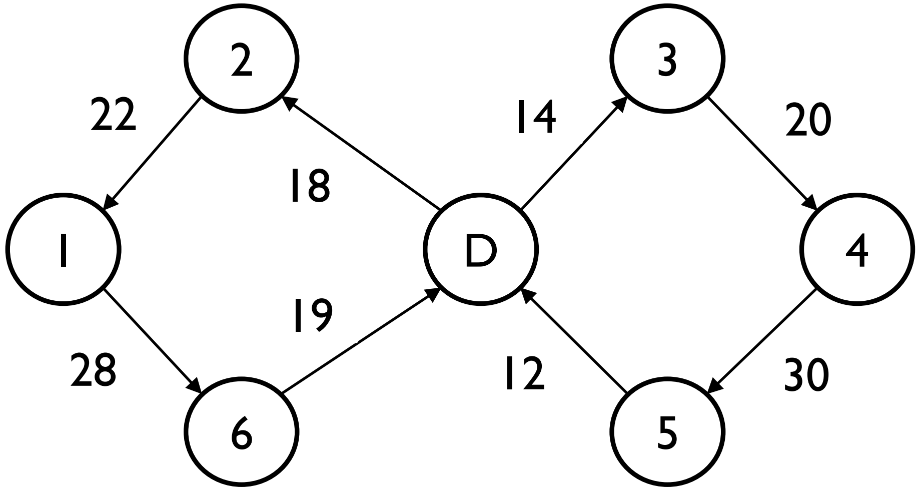
  Maximize a collected profit/score.

PROBLEM INSTANCE

- vehicle1: Route1: D – 3 – 4 – 5 – D = 14+20+30+12 = 76.
- Vehicle2: Route2: D – 2 – 1 – 6 – D = 18+22+28+19 = 87.

  Hence total distance covered = 76+87 = 163.
- Could be feasible solution.

Our objective is to minimize this cost.



CONSTRAINTS

- **Capacity**: $q_o[4,6,3,2,3,2]$ i.e., i-th index: customer i and $q_o[i]$ = customer requirement.

  Q: 10 means the vehicle can carry up to maximum 10 units of product.
- **Constraint**: And from now whenever we try to obtain a feasible solution, we need to look if the allocation satisfy the *capacity constraints*.
- Route1 total customer requirement is: 3+2+3 = 8.
- Route 2 total customer requirement is: 4+6+2 = 12 > 10.
- Hence the overall solution is not feasible now.

CONSTRAINTS (CONT.…)

- How many vehicles we need to meet the demand?

  Total demand = $\sum_{i=0}^{n} q_i$ = 4+6+3+2+3+2 = 20.

  So, the minimum vehicle required to meet the demand would be = $\left\lceil \frac{\sum_{i=0}^{n} q_i}{Q} \right\rceil = \frac{20}{10} = 2.$

Objective Function: Total transportation cost should be minimized and satisfies the capacity constraints.

SOLUTION

- **Brute Force Method**: First we allocate the cities to vehicle and try to solve the optimal route for each vehicle using TSP. We do it for each set of allocations and from that we take the total minimum distance as the optimal solution.

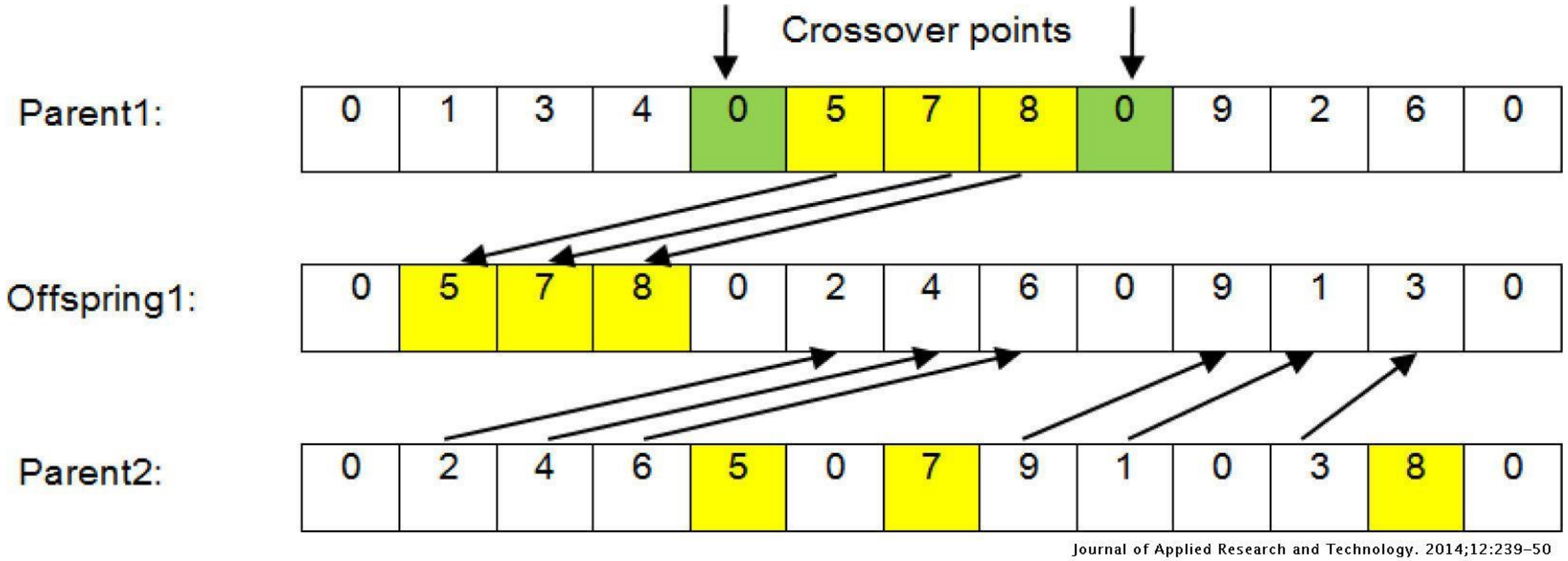
GENETIC ALGORITHM

- **Input**: Adjacency Matrix of the city map. Min vehicles: 4
- **Chromosome**: D – 1 – 3 – D – 3 – 4 – D – 5 – D – 6 – D. } 4 vehicles.
- **Problem**: D – 1 – 3 – D – 3 – 4 – D – 5 – 6 – D – D. } 3 vehicles.
- **Solution:** We will make Matrix[D,D] = ∞.

Initial Population ⟶ Selection ⟶ Crossover ⟶ Mutation ⟶ Mating Pool

[2]

CROSSOVER


Crossover points
Parent1:
0 1 3 4 0 5 7 8 0 9 2 6 0
Offspring1:
0 5 7 8 0 2 4 6 0 9 1 3 0
Parent2:
0 2 4 6 5 0 7 9 1 0 3 8 0
Journal of Applied Research and Technology. 2014;12:239-50

TS ALGORITHM

- **Basic Steps for TS (Tabu Search) Algorithm [1][4]**

Where, i, j = solution indexes; k = number of iterations; v∗= subset of solution; N(i, k) = neighbourhood solution ’i’ at iteration ’k’; f(i) = objective function value for solution ’i’

- **Step 1**: Select an initial solution i in S. Set i∗ = i and the Tabu List x.
- **Step 2**: Set a value for tabu list size x = x + 1 and a subset v∗ generates a solution in N(i, x).
- **Step 3**: Choose the best feasible and suitable move j in i∗ and set i = j.
- **Step 4**: If f(i∗) > f(i) then update the solution, set i∗ = i.
- **Step 5**: Update the Tabu List.
- **Step 6**: If the Stopping Criterion is met stop and report the solution. Else go to Step 2.