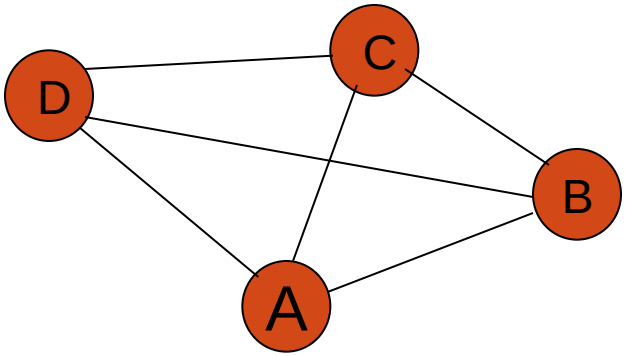
Brute force & Genetic Algorithms

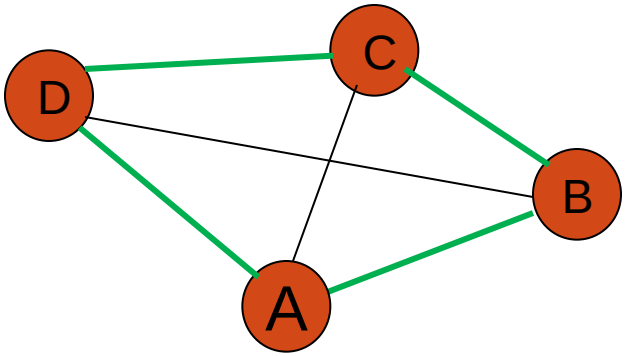
The Traveling Salesman Problem (TSP)

Find a tour that visits **each city exactly once** and that **minimizes** the total distance.



The Traveling Salesman Problem (TSP)

TSP is the problem of finding a tour that visits each city exactly once and that minimizes the total distance.



Applications of TSP

- Vehicle Routing
- Robotics
  - Navigation
  - Arm usage
- Order picking from warehouses.

TSP is more complicated than it may appear!

If you just start from city and keep going to the city nearest to it, you can get a bad solution.

(n-1)! Combinations to check for all possibilities.

n = 17 , (n-1)! $= 2.092279 \times 10^{13}$

n = 26 , (n-1)! $= 4.0329146 \times 10^{26}$

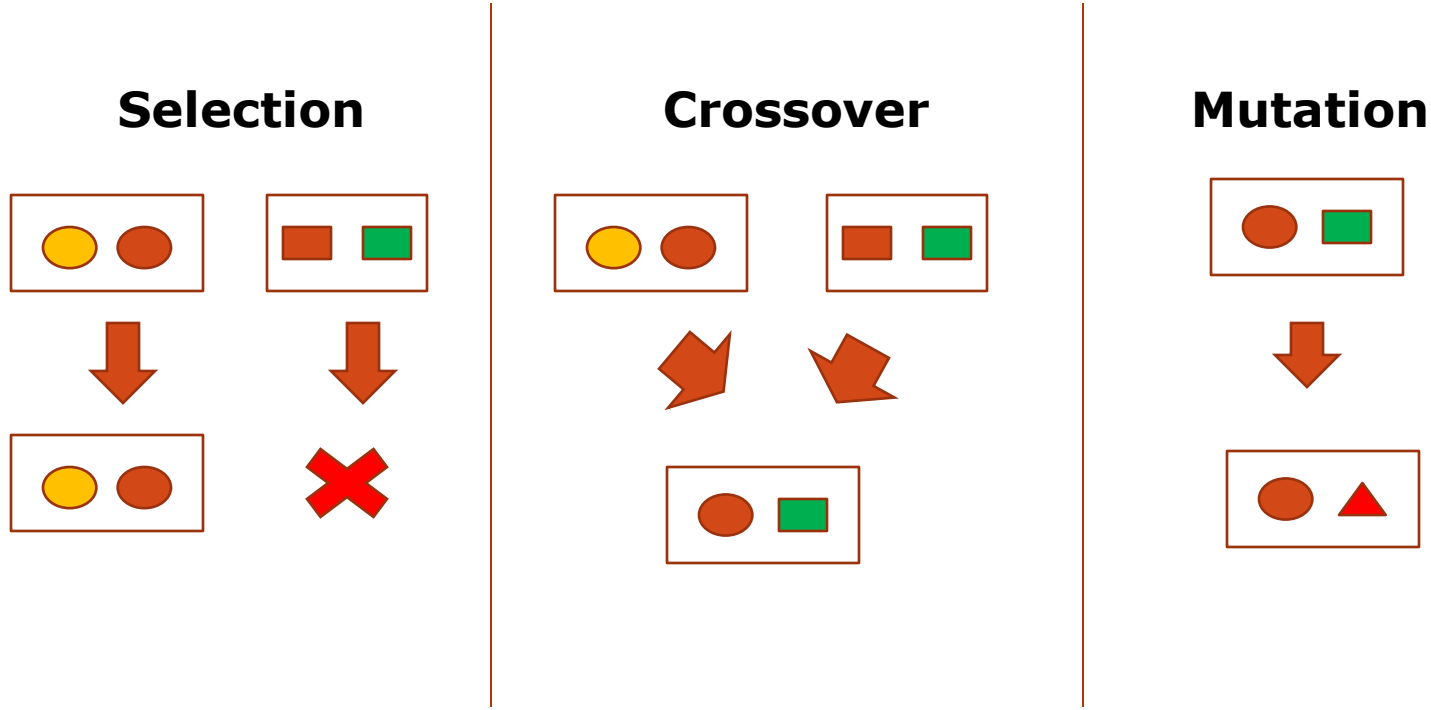
Solutions for TSP

- Branch & Bound Algorithms
- Heuristics Approaches
- Genetic Algorithm

Genetic Algorithm (GA)

- inspired by biological processes:

  selection, crossover and mutation



Demo

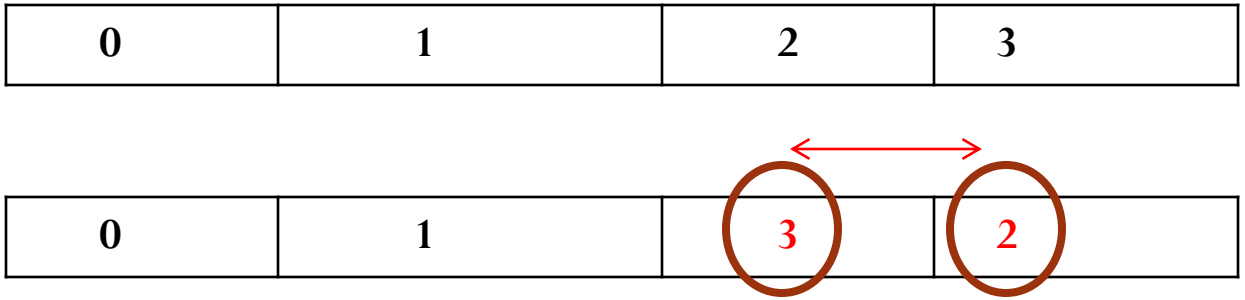
- Goal: min. route visiting every city only once.
- Cities = 17
- Initial population = 500
- Generations = 1000
- Correct Answer = 2085
- (17 - 1)! = 20922789888000 **(trillion)**

Representation

- Each city mapped to 0 – 17
- A route : 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
- Other representations
  - 00000 00001 ..... …… ……..01111 10000
  - 0 1 2 3 4 5 6 7 8 9 A B……. ……. F 10

Mutation

- Swapping random 2 points

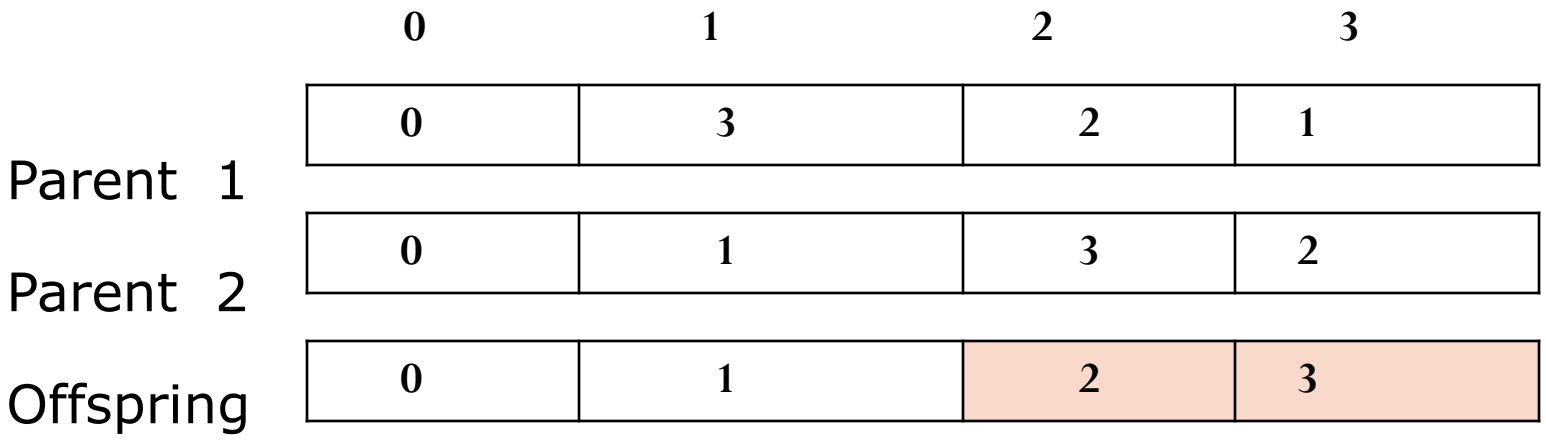


Selection

Kill 20% of the worst performing population

Crossover Operation

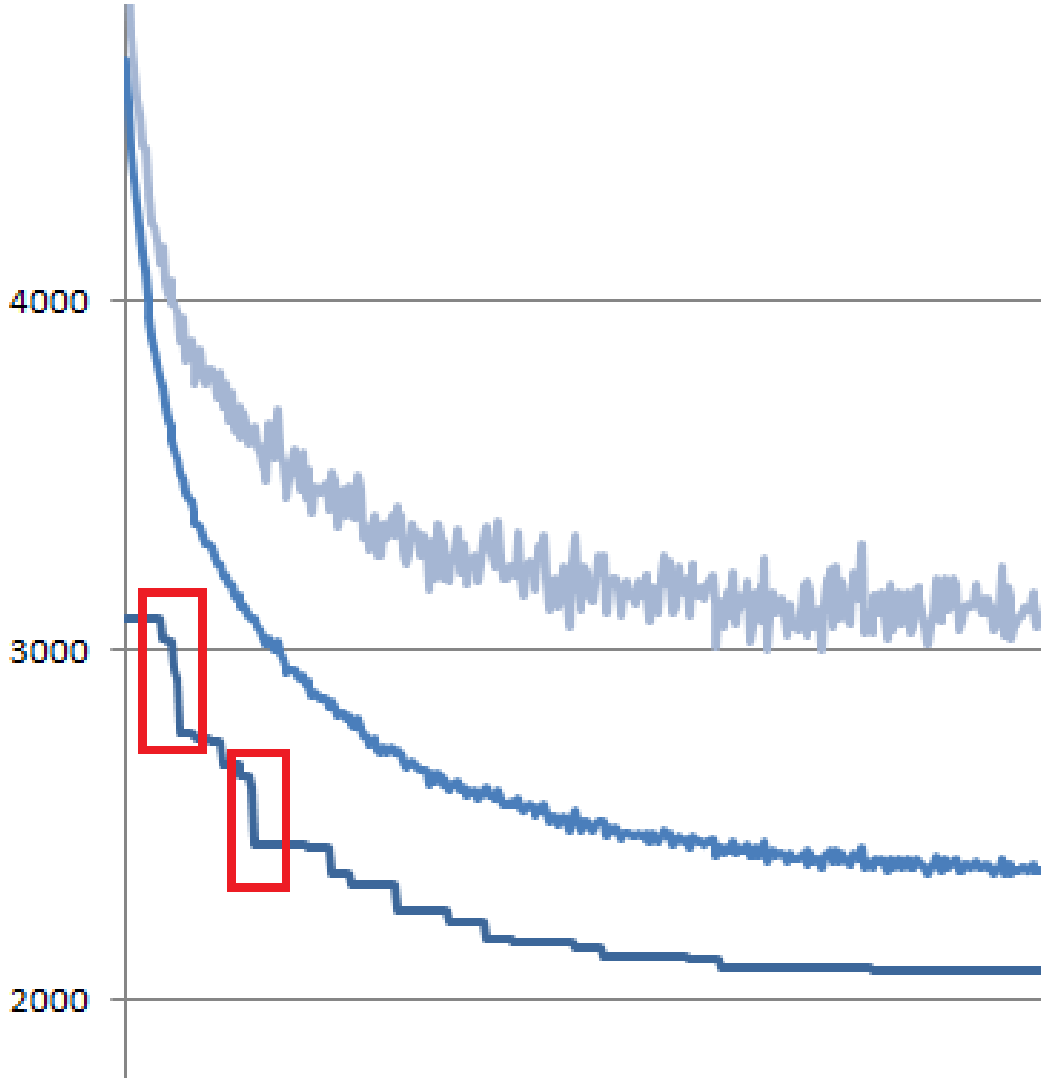
route is encoded as string



	Positions	
City	Parent 1	Parent 2
0	0	0
1	3	1
2	2	3
3	1	2

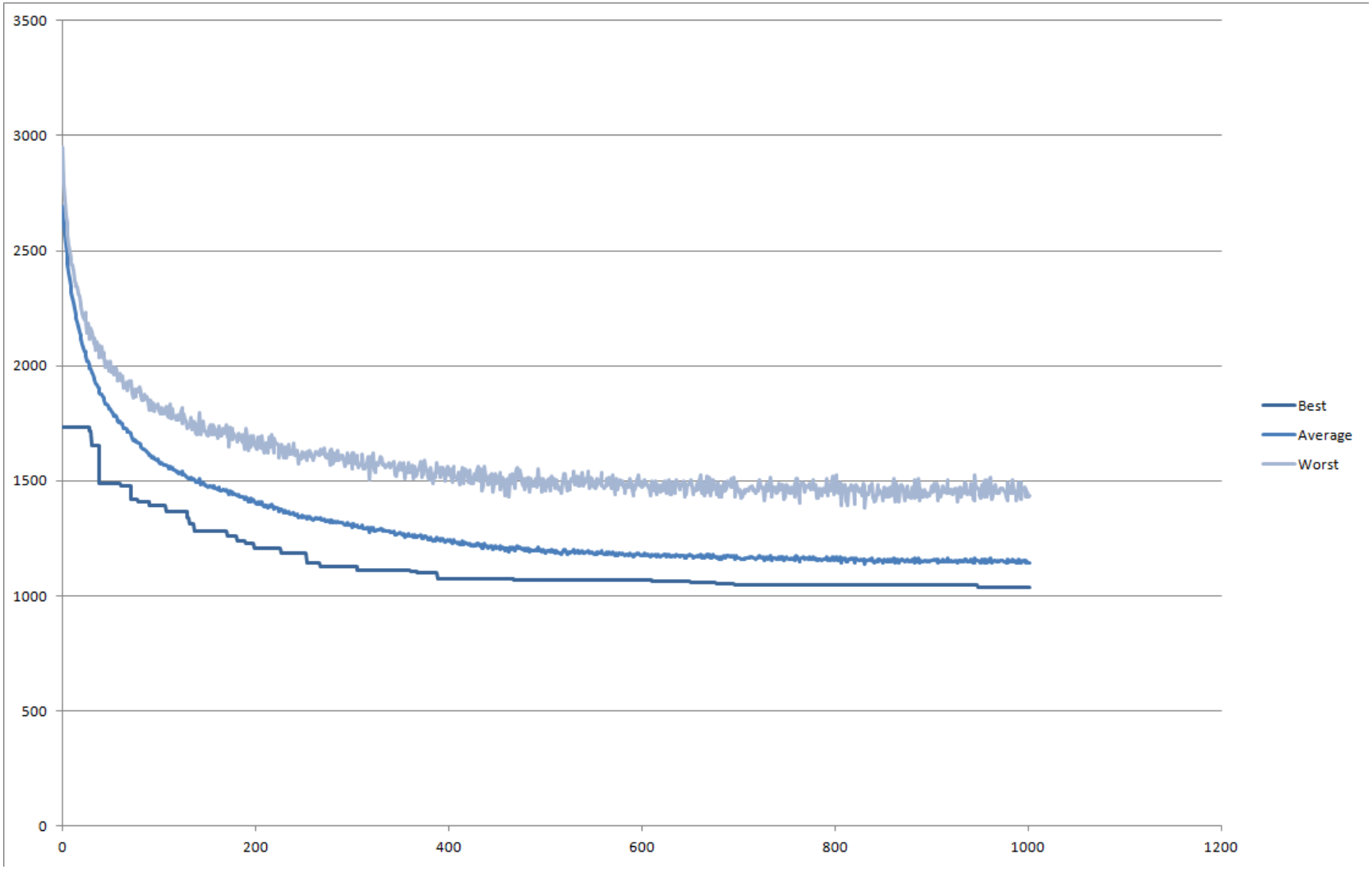
Evolutionary Improvements

- Areas where evolutions improved the solution



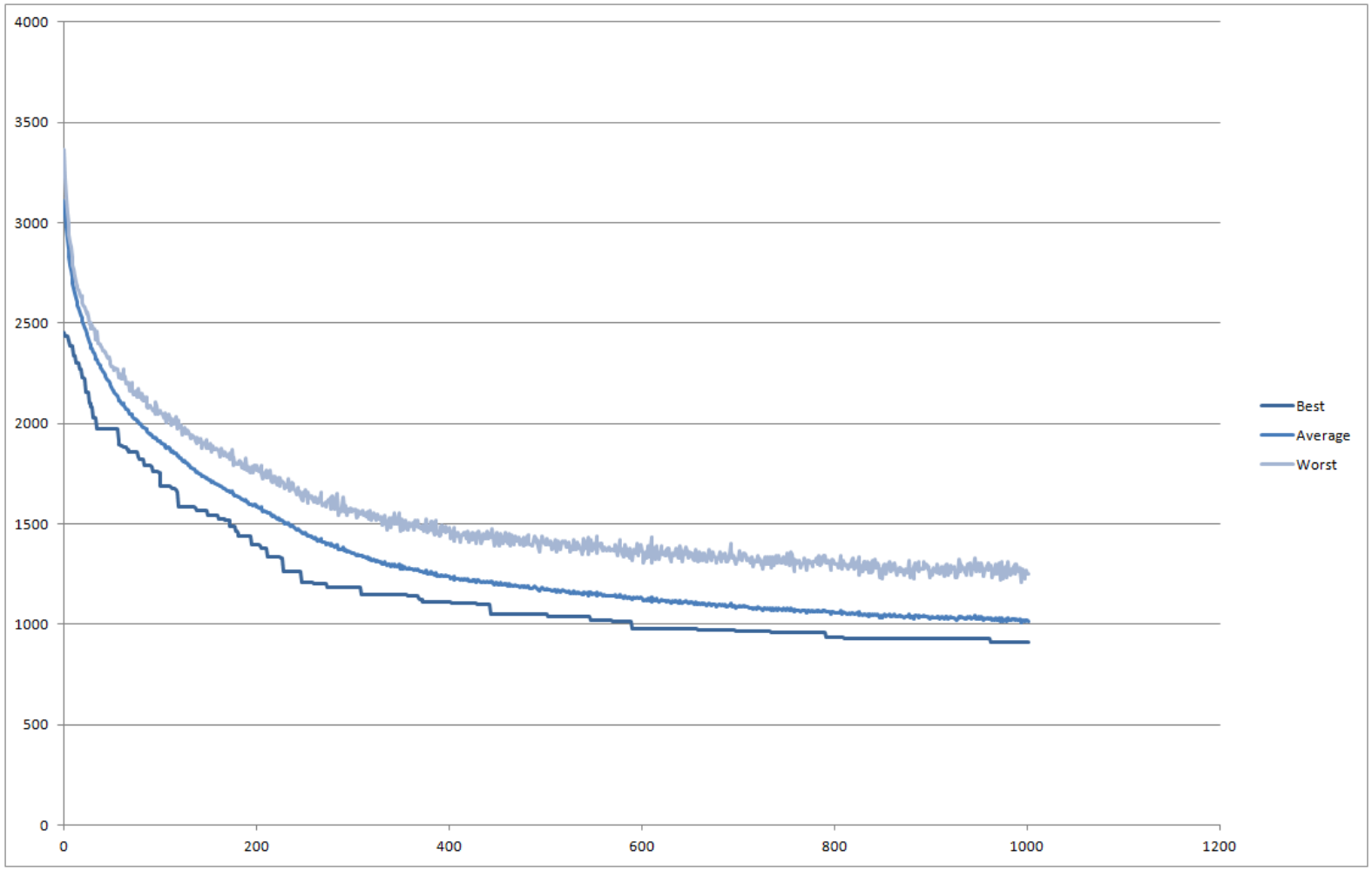
26 Cities

- Answer: 937, Our Result: 1036



42 Cities

- Answer: 669, Our Result:914



Lessons Learned

- Importance of a **good** representation :
  uniqueness, optimization, validity, etc

- Being **practical**:
  Theoretical mathematics is ‘elegant’ –
  but sometimes not suitable for real world applications

- Room for **creativity** (algorithm design and implementation)

Thank You

- Dr. Sohan Dharmaraja
- Dr. Sanjeewa
- Dr. Jagath
- Dr. Oliver
- Dr. Yudi