**KSP-[GA-TS]**

4. For T10/T8 to be comparable with 1.25(10/8), the algorithm's complexity should have been linear, but in our case, the backtracking algorithm has an exponential complexity of 2^n, therefore the ratio T10/T8(approx. 5.54) is closer to 2^10/2^8. This means the backtracking algorithm is not feasible for larger datasets because of its exponentially growing execution time.

5. Because the neighborhood search algorithm does not guarantee that it will find the best solution and it actually gets stuck when encountering local optimas, both Sns-8 and Sns-10 are worse than the solutions found by the backtracking algorithm using the same datasets.

6. The ratio between Tns-10 and Tns-8 is approximately 1.47 and 2.61 for Tns-100/Tns-50. Even when compring Tns-10/Tns-8 with Tb-10/Tb-8, we can see that the time of execution grows slower when using larger datasets for Neighborhood Search than for Backtracking. To have an idea of how much slower, we have to think that Tb-100/Tb-50 would be approximately 2^50 and compare it with Tns-100/Tns-50 = 2.61. Cosidering the above, the Neighborhood Search algorithm is feasible for larger datasets in terms of execution time, but the solution won't be the best one, it would actually be a local optima.

7. Because of its execution time, the Neighborhood Search algorithm is a better choice than Backtracking for larger datasets, even if it fails to find the best solution and it stops at a local optima. For small datasets, Backtracking is suitable because it ensures the best solution is found, but because of its exponential complexity, it is not feasible for larger datasets.

10. In general, the Genetic Algorithm found the better results for each dataset. The backtracking algorithm would always find the best solution, but it is not feasible for larger datasets due to its exponential complexity and it was only run for dataset 8 and 10. Also Tabu search did a better job at finding a solution than Neighborhood Search, because it also takes into consideration solutions belonging to a neighborhood which are worse than the best solution found so far, therefore it does not get stuck at the first local optima encountered as the Neighborhood Search algorithm does.

11. Tabu Search had a slightly better time performance than the Genetic Algorithm, but we can say that both algorithms perform well on large datasets, the chosen number of iterations having the biggest impact on the execution time of both algorithms.

12. Although the Neighborhood Search had the best time performance of the three algorithms, the solutions foud using NS were worse than the solutions found by the other two algorithms. As we have seen previously, Tabu Search was slightly faster overall than the Genetic Algorithm, but the latter obtained better solutions.

13. A large population size combined with a small number of iterations, as well as a small population size combined with a large number of iterations had a negative impact on the solution found, the best solutions being found when using a middle-sized population combined with a medium but relatively high number of iterations. Crossover should be the main producer of new solutions because it combines couples of solutions with good genes, therefore the crossover rates should be high compared with the mutation rates, because a mutation is more likely to deteriorate the fitness function of an individual and cloning prevents the exploration of new individuals.

14. The tabu tenure is highly dependent on the dataset size. A large tenure for a small dataset could get the algorithm stuck, because there could be no valid moves left. The tabu tenure should always be set to a reasonable value with respect to the dataset size.

15. Conclusions:

Backtracking:

- only one that guarantees optimal solution

- high complexity, feasable only for small datasets

Neighborhood search:

- low complexity

- fast convergence to a local optima

- gets stuck there

Tabu search:

- similar to neghborhood search but forbids past t moves (t = tabu tenure)

- has a short-term memory

- this introduces an exploration behaviour

Genetic algorithm:

- keeps best solutions by cloning them

- uses mutations to explore the solution set

- crossover to combine solutions and seek better ones

- slightly higher complexity than Neighborhood Search and Taboo Search but with better results