

COS 314 Assignment 1

Keelan Matthews – u21549967

Technical Specification

Objective

The aim of this assignment is to implement the Iterated Local Search and Tabu Search algorithms and evaluate their performance on the given datasets. Our objective is to optimize these algorithms to obtain as many optimal or near-optimal solutions as possible, while minimizing the computation time.

Iterated Local Search Algorithm

The Iterated Local Search Algorithm adopts the best-fit algorithm throughout its lifespan. The algorithm starts with the generation of the initial solution, which involves sorting the values in non-increasing order and packing them into the bins via the best-fit algorithm. Then, for a specified number of iterations, a new solution is created by perturbing the previous solution. Half of the values from a bin are removed and placed in new bins, one by one, using the best-fit algorithm.

Following this, a local search is executed on the new solution by removing items from bins and placing them into a new bin if it fits. Finally, the bin count of the new solution is compared to the bin count of the best solution, and it is updated if the bin count of the new solution is smaller than that of the best solution.

Tabu Search Algorithm

The Tabu Search Algorithm uses the same perturbation and initial solution algorithms as the Iterated Local Search Algorithm to ensure that the search space is consistent and comparable between the two algorithms. Unlike the Iterated Local Search Algorithm, the Tabu Search Algorithm does not use a local search algorithm after perturbation. Instead, it stores the perturbed solution in a table and only considers solutions that are not already in the table, as the ones already stored are considered taboo since they have already been visited.

In the Tabu Search Algorithm, the fitness value of the bin count is compared between the new solution and the best solution, but only if the new solution is not already in the table. If the new solution is smaller than the best solution, it is updated. However, if the new solution is in the table, the algorithm skips it and continues for a specified number of iterations.

Evaluation Metrics

Problem Set	ILS			Tabu		
	Opt.	Opt.-1	Sum	Opt.	Opt.-1	Sum
Falkenauer_T (80)	0	20	80	0	20	80
Falkenauer_U (80)	6	26	80	6	26	80
Scholl_1 (720)	549	112	720	550	111	720
Scholl_2 (480)	316	119	480	313	122	480
Scholl_3 (10)	0	8	10	0	5	10
Schwerin_1 (100)	15	85	100	12	88	100
Schwerin_2 (100)	41	59	100	42	58	100
Hard28 (28)	5	23	28	5	23	28
Waescher (17)	2	15	17	2	15	17
Total (1615)	934	467	1615	930	468	1615

Problem Set	Average times (s)	
	ILS	Tabu
Falkenauer_T	0.064	0.058
Falkenauer_U	0.043	0.041
Scholl_1	0.013	0.011
Scholl_2	0.032	0.027
Scholl_3	2.906	2.535
Schwerin_1	0.007	0.005
Schwerin_2	0.009	0.006
Hard28	0.043	0.039
Waescher	0.106	0.062
Total (s)	3.223	2.784

Results

After analyzing the results obtained from both the Iterated Local Search and Tabu Search algorithms, it's evident that each algorithm has its own unique strengths and weaknesses.

In Table 1, we observe that Iterated Local Search generated more optimal solutions than Tabu Search across all datasets. Although both algorithms produced a similar number of near-optimal solutions, Iterated Local Search computed four more optimal solutions than Tabu Search. However, it's worth noting that the difference between the two algorithms is relatively small, and therefore, the choice of the preferred algorithm should also consider other factors such as computational resources and time.

In Table 2, we can see that Tabu Search performed much more efficiently than Iterated Local Search, with a shorter total execution time. This is a critical factor to consider in applications where time is important. Moreover, while Tabu Search generated fewer optimal solutions compared to Iterated Local Search, the difference is still relatively small, and therefore, the Tabu Search algorithm remains a viable option in situations where efficiency and speed are critical factors.

Conclusion

In conclusion, the choice of the algorithm to use should depend on the specific requirements of the problem being solved. If generating optimal solutions is the top priority, then Iterated Local Search is the preferred algorithm. However, if computational efficiency and time are critical factors, then Tabu Search is the better option.