ECE457A Assignment 2 Question 3

By: Swapan Shah & Xiaotong Zhang

Solution Description

All initialization is done before the tabu search is run. The flow and distance numbers are copied over into respective matrices. An initial solution is set up randomly by initializing an array of length 20, with each index number as its location, and the actual value as the "department". The tabu search will go through all possible exchanges, or swaps, of the elements within the solution and determine the most optimal solution that does is not tabu or can be accepted by the aspiration criteria. The tenure length is initially set up as 10. If using dynamic tenure, a random length between 10 and 15 is used, changing every 50 iterations. The tabu list is set up using a 20x20 array that keeps track of the moves already made, where the columns are the locations that the specific element has occupied previously and the rows are the element value themselves.

In order to force diversity, a frequency list is implemented that keeps track of the different exchanges that have been made, where the exchanges are represented by the pair values, directly correlating to the indices in the array. Whenever the cost is calculated, the frequency of the solution is also accounted for, as to force the algorithm to explore different neighbors and neighborhoods that haven't been explored nearly as much.

The main aspiration criteria used includes keeping track of the lowest cost so far, not just within a specific iteration. Regardless if a solution is tabu or not, as long as it is the most optimal solution so far it will be accepted.

The simplified search algorithm is:

1) Set up initial solution
2) Check if termination conditions are met for that solution
3) If not, generate the appropriate neighbors depending on the set up (all or partial)
4) Determine the best exchange cost (depending on aspiration criteria)
5) Go back to step 2 and repeat until termination conditions have been met

<u>Perform the following changes on your TS code (one by one) and compare the results.</u>

1) Change the initial starting point (initial solution) 10 times

Table 1: Summary of different initial starting points

Initial Solution	**Number of Iterations**
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]	1556
[14, 4, 7, 5, 17, 1, 8, 3, 15, 6, 11, 16, 12, 2, 10, 9, 13, 18, 19, 20]	2858
[17, 5, 18, 15, 19, 6, 7, 10, 20, 2, 13, 1, 12, 8, 4, 9, 3, 14, 11, 16]	91
[3, 18, 2, 15, 19, 6, 14, 7, 20, 4, 13, 10, 12, 8, 11, 9, 5, 1, 17, 16]	732
[4, 18, 16, 2, 19, 17, 12, 3, 11, 7, 14, 13, 10, 9, 5, 6, 1, 20, 15, 8]	3202
[5, 20, 13, 2, 17, 1, 3, 19, 8, 16, 7, 15, 11, 18, 12, 14, 6, 4, 9, 10]	219
[13, 19, 1, 6, 15, 18, 5, 7, 2, 8, 14, 9, 12, 20, 3, 10, 11, 17, 17, 4]	672
[17, 12, 20, 13, 3, 10, 14, 9, 19, 18, 15, 2, 8, 6, 5, 7, 11, 4, 1, 16]	3807
[15, 20, 10, 13, 1, 6, 19, 4, 2, 12, 8, 17, 7, 5, 11, 18, 3, 9, 14, 16]	5902
[14, 9, 12, 11, 8, 4, 10, 17, 19, 3, 16, 2, 6, 1, 13, 7, 18, 20, 5, 15]	3334

Looking at Table 1, it can be clearly seen that varying the initial solution clearly effects the time it takes to reach the final solution. Depending on the initial solution and how close it is to the most optimal in terms of number of swaps necessary, it will change by either taking more iterations or less. The neighborhood in which the search started in will also affect the optimal solution that can be found, as multiple ones exist.

2) Change the tabu list size – smaller and larger than your original choice

By changing the tabu list size to be greater than the original choice, the amount of iterations necessary increases, with everything else equal. This may be due to the fact that it takes many more iterations for the same swap to be allowed, or the tabu list to be updated to allow a specific switch. As such, more options are explored and a lower solutions often found sooner. Similar reasoning can be used for lowering the tabu list size. By changing the tabu list size to be smaller than the original choice, the amount of iterations necessary increases as well, with everything else equal. With a smaller tabu list, the likelihood of looping through very similar solutions is much higher, thus increasing the number of iterations it may take to reach the optimal solution.

Original

```
Initial Solution:
04 | 18 | 16 | 02 | 19 | 17 | 12 | 03 | 11 | 07 | 14 | 13 | 10 | 09 | 05 | 06 | 01 | 20 | 15 | 08 |
Total Cost 1721
Final Solution:
17 | 05 | 07 | 01 | 06 | 19 | 15 | 20 | 08 | 13 | 04 | 02 | 12 | 11 | 16 | 18 | 14 | 10 | 03 | 09 |
Final Cost: 1285
Total # of Iterations: 3202
END SEARCH
```

Smaller tabu size (8)

```
Initial Solution:
04 | 18 | 16 | 02 | 19 | 17 | 12 | 03 | 11 | 07 | 14 | 13 | 10 | 09 | 05 | 06 | 01 | 20 | 15 | 08 |
Total Cost 1721
Final Solution:
13 | 05 | 16 | 09 | 17 | 20 | 08 | 11 | 19 | 04 | 07 | 15 | 12 | 14 | 02 | 06 | 01 | 03 | 10 | 18 |
Final Cost: 1349
```

Note: This solution is not optimal (within 10,000 iterations).

```
Total # of Iterations: 10000
END SEARCH
```

Larger tabu size (12)

```
Initial Solution:
04 | 18 | 16 | 02 | 19 | 17 | 12 | 03 | 11 | 07 | 14 | 13 | 10 | 09 | 05 | 06 | 01 | 20 | 15 | 08 |
Total Cost 1721
Final Solution:
```

```
09 | 03 | 10 | 14 | 18 | 16 | 11 | 12 | 02 | 04 | 13 | 08 | 20 | 15 | 19 | 06 | 01 | 07 | 05 | 17 |
Final Cost: 1285
Total # of Iterations: 2296
END SEARCH
```

3) Change the tabu list size to a dynamic one – an easy way to do this is to choose a range and generate a random uniform integer between this range every so often (i.e., only change the tabu list size infrequently)

By using a random uniform integer between 10 and 15 every 10 iterations, the number of iterations decreases. Depending on the neighborhood that is being explored and the number of iterations already passed in finding the solution, varying the tabu size can dramatically change the amount of iterations required to find the solution.

Original

```
Initial Solution:
04 | 18 | 16 | 02 | 19 | 17 | 12 | 03 | 11 | 07 | 14 | 13 | 10 | 09 | 05 | 06 | 01 | 20 | 15 | 08 |
Total Cost 1721
Final Solution:
17 | 05 | 07 | 01 | 06 | 19 | 15 | 20 | 08 | 13 | 04 | 02 | 12 | 11 | 16 | 18 | 14 | 10 | 03 | 09 |
Final Cost: 1285
Total # of Iterations: 3202
END SEARCH
```

With Dynamic Tabu Tenure Size

```
Initial Solution:
04 | 18 | 16 | 02 | 19 | 17 | 12 | 03 | 11 | 07 | 14 | 13 | 10 | 09 | 05 | 06 | 01 | 20 | 15 | 08 |
Total Cost 1721
Random Tabu Size: 13
Random Tabu Size: 14
Random Tabu Size: 11
Random Tabu Size: 12
Random Tabu Size: 13
Random Tabu Size: 10
Random Tabu Size: 13
Random Tabu Size: 10
Random Tabu Size: 10
Random Tabu Size: 12
Random Tabu Size: 11
Final Solution:
06 | 01 | 07 | 05 | 17 | 13 | 08 | 20 | 15 | 19 | 16 | 11 | 12 | 02 | 04 | 09 | 03 | 10 | 14 | 18 |
Final Cost: 1285
Total # of Iterations: 518
END SEARCH
```

4) Add one or more aspiration criteria such as best solution so far, or best solution in the neighborhood, or in a number of iterations

By adding aspiration criteria of allowing a lower costing solution, even if is “tabu” to be accepted, the number of iterations required to reach the solution decreased. More neighbors can be explored more quickly, allowing a lower solution to be found more quickly.

Original

```
Initial Solution:
04 | 18 | 16 | 02 | 19 | 17 | 12 | 03 | 11 | 07 | 14 | 13 | 10 | 09 | 05 | 06 | 01 | 20 | 15 | 08 |
Total Cost 1721
Final Solution:
17 | 05 | 07 | 01 | 06 | 19 | 15 | 20 | 08 | 13 | 04 | 02 | 12 | 11 | 16 | 18 | 14 | 10 | 03 | 09 |
Final Cost: 1285
Total # of Iterations: 8136
END SEARCH
```

With Aspiration

```
Initial Solution:
04 | 18 | 16 | 02 | 19 | 17 | 12 | 03 | 11 | 07 | 14 | 13 | 10 | 09 | 05 | 06 | 01 | 20 | 15 | 08 |
Total Cost 1721
Final Solution:
06 | 01 | 07 | 05 | 17 | 13 | 08 | 20 | 15 | 19 | 16 | 11 | 12 | 02 | 04 | 09 | 03 | 10 | 14 | 18 |
Final Cost: 1285
Total # of Iterations: 280
END SEARCH
```

5) Use less than the whole neighborhood to select the next solution

By limiting the neighborhood to only being able to swap elements from position 1 to 10, the solution requires many more iterations in order to find the optimal solution. The optimal solution is often skipped over due to the randomness, and heavily dependent on the random generator. By limiting the neighborhood to only a fraction of the exchanges, it increases the number of iterations as compared to a random exchange generator.

Original

```
Initial Solution:
03 | 18 | 02 | 15 | 19 | 06 | 14 | 07 | 20 | 04 | 13 | 10 | 12 | 08 | 11 | 09 | 05 | 01 | 17 | 16 |
Total Cost 1329
Final Solution:
09 | 03 | 10 | 14 | 18 | 16 | 11 | 12 | 02 | 04 | 13 | 08 | 20 | 15 | 19 | 06 | 01 | 07 | 05 | 17 |
Final Cost: 1285
Total # of Iterations: 732
END SEARCH
```

With Partial Neighborhood

```
Initial Solution:
03 | 18 | 02 | 15 | 19 | 06 | 14 | 07 | 20 | 04 | 13 | 10 | 12 | 08 | 11 | 09 | 05 | 01 | 17 | 16 |
Total Cost 1329
Final Solution:
10 | 13 | 06 | 03 | 09 | 18 | 12 | 20 | 02 | 14 | 05 | 08 | 07 | 15 | 19 | 17 | 01 | 11 | 04 | 16 |
Final Cost: 1356
Note: This solution is not optimal (within 10,000 iterations).
Total # of Iterations: 10000
END SEARCH
```

6) Add a frequency based tabu list and/or aspiration criteria (designed to encourage the search to diversify)

To diversity the search, a frequency tabu list is used to keep track of the already made exchanges of positions and elements. This frequency is added to the cost to decrease the chance that a frequent move is chosen as the optimal solution at a specific iteration. Depending on the initial solution, this could either speed up or significantly slow down the search, depending on the closeness of the new elements that require exchanging for the optimal solution.

Original

```
Initial Solution:
12 | 01 | 15 | 04 | 09 | 11 | 13 | 08 | 03 | 17 | 02 | 18 | 20 | 06 | 14 | 10 | 07 | 05 | 16 | 19 |
Total Cost 1721
Final Solution:
17 | 05 | 07 | 01 | 06 | 19 | 15 | 20 | 08 | 13 | 04 | 02 | 12 | 11 | 16 | 18 | 14 | 10 | 03 | 09 |
Final Cost: 1285
Total # of Iterations: 852
END SEARCH
```

With Frequency Based Tabu List

Initial Solution:

12 | 01 | 15 | 04 | 09 | 11 | 13 | 08 | 03 | 17 | 02 | 18 | 20 | 06 | 14 | 10 | 07 | 05 | 16 | 19 |

Total Cost 1721

Final Solution:

06 | 01 | 07 | 05 | 17 | 13 | 08 | 20 | 15 | 19 | 16 | 11 | 12 | 02 | 04 | 09 | 03 | 10 | 14 | 18 |

Final Cost: 1285

Total # of Iterations: 546

END SEARCH