Mark Demore II, 2d Lt

CSCE686 - Dr. Lamont

Spr 2020 - Homework 7

1. One potential representation of the VCP and the neighboring structure is an adjacency matrix, where the value in the position *A(i,j)* is 1 if *vi* and *vj* are connected, and 0 if they are not. So for any *i*, the neighbors of vertex *vi* are all *j* for which *A(i,j)* is 1. If the vertex *i* is in the covering set, the value of *vi* is set to 1, else it is 0. This can easily be applied to the independent set problem, since all vertices, *vi*, that are of value 0 make up the independent set. An easy incremental way to evaluate neighbors is to jump to their row in the adjacency matrix by setting the value *i* in *A(i,j)* to the first neighbor, and then the next and so on.
2. The representation from part A can easily be applied to a simulated annealing (SA) algorithm, by looking at neighbors. The algorithm can start with an initial solution, and an alternate solution can be generated from one of its neighbors. If the cost of the alternate solution is less than the initial solution, the alternate solution replaces the current solution. Otherwise, the annealing probability is calculated and used to decide if we keep the current solution or adopt the alternate.

Severity: O(1)

1. Move customer ck from route Ri to the last customer in route Rj. For

Route Ri, the new route goes from customer ck-1 directly to ck+1. Ordering for all other customers remains the same in both routes.

Severity: O(n)

2: Customer ck is removed from route Ri, which links customer ck-1 to ck+1. Calculate the new route cost for Rj if ck is placed between two nodes Cn and Cn+1 in Rj, iterating through all of route Rj. ck is inserted where cost is minimized for the new route, Rj.

Severity: O(n2)

3: Remove customer ck from route Ri and link customer ck-1 to ck+1. For route Rj, rerun the algorithm to minimize total cost of the route with the new set of customers.

1: Every item in the tabu list is the full list of every route.

2: Every item in the tabu list is a set of the nodes that are in every route, but doesn’t include the optimal route for each set of nodes.

3: Every item in the tabu list is a set of the customer that was moved, the route it was moved from, and the route it was moved to. The last move can be reversed.

1. The algorithm below shows an application of tabu search for the Maximum Independent Set (MIS) problem. This makes use of a swap between neighbors and adds the previous vertex to the tabu set.

1: Input: A graph G, Itersmax (maximum allowed iterations per run)

2: Output: The largest independent set S∗ found.

3: S ← Initialization() /\* Generate a feasible independent set S\*/

4: S∗ ← S /\* S∗ records the largest independent set found so far \*/

5: f∗ ← f(S) /\* f∗ records the cardinality of S∗ \*/

6: Initialize tabu list /\* Initialize the tabu list\*/

7: for iters ← 1 to Itersmax do

8: if there exists an eligible intensification move then

9: S ← IntensificationStep(S) /\* Apply (k, 1)-swap (k ≤ 1) to improve solution S\*/

10: if f(S) > f∗ then

11: S∗ ← S, f∗ ← f(S)

12: end if

13: else

14: S ← DiversificationStep(S) /\* Apply (k, 1)-swap (k > 1) to perturb solution S\*/

15: end if

16: Update tabu list

17: end for

18: return S∗

**References**

[1] <https://www.geeksforgeeks.org/vertex-cover-problem-set-1-introduction-approximate-algorithm-2/>

[2] [https://en.wikipedia.org/wiki/Vertex\_cover](https://en.wikipedia.org/wiki/Vertex_cover#:~:text=The%20minimum%20vertex%20cover%20problem%20is%20the%20optimization%20problem%20of,cover%20in%20a%20given%20graph.&text=The%20vertex%20cover%20problem%20is%20an%20NP%2Dcomplete%20problem%3A%20it,point%20for%20NP%2Dhardness%20proofs.)

[3] <http://www.info.univ-angers.fr/pub/hao/papers/JinHaoEAAI2014.pdf>

[4] Xu, X., & Ma, J. (2006). An efficient simulated annealing algorithm for the minimum vertex cover problem. *Neurocomputing*, *69*(7-9 SPEC. ISS.), 913–916. <https://doi.org/10.1016/j.neucom.2005.12.016>

[5] <https://en.wikipedia.org/wiki/Neighbourhood_(graph_theory)>

[6] <https://en.wikipedia.org/wiki/Tabu_search>