Tabu Search and Automorphic Equivalence

Sachin Kulkarni

03,08/03/2006

1 Introduction

Tabu Search is one of the local search techniques. It is developed by Fred Glover.

1.1 Idea of Tabu Search

Local search employs the idea that a given solution S may be improved by making small changes. Following is a general algorithm for a heuristic local search:

1. Take an arbitrary solution
2. Check how good the solution is
3. Change the solution a bit by moving a solution x to x_1, which is in its neighborhood. Choose a solution, which decreases the value of an objective funtion.

Problem with some strategies is to escape from local minima where the search cannot find any further neighborhood solution that decreases the objective function value. Different strategies have been proposed to solve this problem. One of the most efficient strategies is tabu search. Tabu search allows the search to explore solutions that do not decrease the objective function value only in those cases where these solutions are not forbidden. Tabu Search is especially good for the problem of sequenceing or scheduling.
Consider an example of Iron ore, which has a filtering scheme and in the end it produces steel. It has a set of filters. This is problem of filter sequencing. One can apply tabu search to find the optimal filteirng sequence.

1.2 Algorithm

Following is the Tabu Search algorithm. Given is a set of numbers and an objectibe function defined to give an objective value to a sequence of these numbers.

1. Take an arbitrary initial solution/sequence. Calculate the objective function over the sequence, $f(S_i)$
2. Choose any one pair of numbers to swap with each other. It generates a solution in the neighborhood of current solution. Compute the improvement of the objective function over the current value. There are nCk such pairs to be considered. Calculate the values for all such pairs.
3. Sort them according to their improvement over the current solution.
4. Pick up the pair providing the best improvement. A solution picked in the round i is not considered for the next 3 rounds.
5. Continue this procedure till the stopping condition is not met.

2 Application of tabu search

This section gives the description of how tabu search can be applied to the problem of creating structurally equivalent blocks from the given set of nodes.

2.1 Problem Definition

Here we are given the number of groups to be formed, minimum variance of dissimilarity within the group, number of elements in the graph.
Consider the example of 12 elements to be divided into 4 groups. Following diagram gives the random initial configuration of taub search with three boundaries defined.

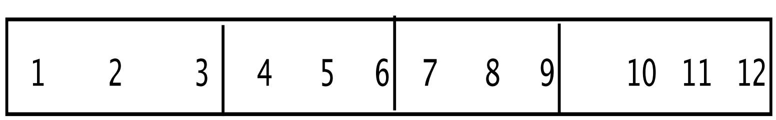

Figure 1: Application of Tabu Search to find structurally equivalent groups

Here following restrictions are defined on the swapping of the elements:

1. Two boundaries shouldn't be swapped as it doesn't change the group formation.
2. Swapping a number with a number within the same group is not allowed as it also doesn't change the group formation.
3. Two boundaries shouldn't be side by side or at the begining or at the end.

2.2 Mapping the problem to Tabu Search

Here we need to define the objective function of tabu search as following:

$$\sum_{i=1}^{4} varianceOfDissimilarity_i \tag{1}$$

where number of groups is 4 and $varianceOfDissimilarity_i$ is variance of dissimilarity within the group i.

2.2.1 Neighborhood

In this problem, a neighborhood is defined as the solution, in which a member changes from one block to another block. $(i, j)1$ is done by changing element 1 from ith block to jth block.

2.3 Space Complexity

There are K blocks and n elements. So space complexity is $O(K^2 * n)$.

3 Automorphic Equivalence

Two vertices u and v of a labeled graph G are automorphically equivalent if all the vertices can be re-labeled to form an isomorphic graph with the labels of u and v interchanged. Two automorphically equivalent vertices share exactly the same label-independent properties. There is hierarchy of three equivalence concepts. Automorphic Equivalence is stricter definition than Regular Equivalence and Structural Equivalence is stricter than both Automorphic Equivalence and Regular Equivalence.
Regular Equivalence holds when

- Two nodes have similar neighbors (i.e. their neighbors are also regular equivalent)

Automporphic Equivalence holds when above condition is true and also the following condition.

- Graphs can become equivalent by subgraph swapping.

More intuitively, nodes are automorphically equivalent if we can permute the graph in such a way that exchanging the two nodes has no effect on the distances among all the nodes in the graph.

3.1 Calculation of Automorphic Equivalence

Given a network, we want to find the automorphically equivalent nodes and cluster the equivalent nodes. Following steps are followed to do this:

1. Calculate all the pairwise distances dis_{ij}.
2. Define vector V_i of size n, where kth element of the vector v is $dist_{ik}$.
3. sort v_i with ascending values of geodesic distances.
4. Form a matrix of such n vectors, where ith column represents vector v_i. Here each vector v_i represents profile of node i with respect to its distance with its neighbors.
5. Using this matrix, calculate the amount of similarity between all the pairs of nodes by calculating dissimilarity between vectors.
6. This way matrix $SimMatrix$ is formed, on which clustering is applied to form automorphically equivalent groups.

References

[1] http://www.idsia.ch/ monaldo/tabusearch.html

[2] Introduction to Social Networks by Robert A. Hanneman