**Assignment 2**

**Implemetation of Local Search Techniques**

**B.Tech CSE VII Semester, Winter 2018**

**Co-ordinator: U. A. Deshpande**

**Date: 12th September 2018**

**Hard Deadline: 25th September 2018 (R1, R2, R3, R4 – 2.00 pm onwards)**

**Submit on email id:** [**uad.ai.assignment@gmail.com**](mailto:uad.ai.assignment@gmail.com) **with roll no, name and assignment number in subject line. (Viva in CSE 105 – contact Mansi A. Radke)**

A conference has n papers accepted. Our job is to organize them in a best possible schedule. The schedule has p parallel sessions at a given time. Each session has at the most k papers (i.e. n <= p\*k). We first define the characteristics of a good schedule. A schedule is an assignment of each of the papers to the parallel sessions. For any good schedule most people should feel no conflict about which session to attend. That is,

1. all papers in one session should be closely related to each other.

AND

1. all papers in parallel sessions should be as far away as possible to avoid conflict.

Let us assume we are given a function representing the distance between two papers: d(p1,p2), such that d is between 0 and 1. We can similarly define a similarity between two papers s(p1,p2) = 1-d(p1,p2). Now we can define the goodness of a schedule as follows: **Sum(similarities of all pairs of papers in a session) + C.Sum(distances of all pairs of papers in parallel sessions)**. The constant C trades off the importance of semantic coherence of one session versus reducing conflict across parallel sessions. Our goal is to find a schedule with the maximum goodness.

Hint: 1. Use Simulated Annealing with Random Restarts

2. Refer to the following link: https://github.com/topics/local-search

|  |
| --- |
| **Test case:** |
|  |
|  |  |
|  |  |
|  |  |
|  | n = 4  C = 1  k = 2  p = 2 |
|  | 0 0.4 0.8 1 |
|  | 0.4 0 0.6 0.7 |
|  | 0.8 0.6 0 0.3 |
|  | 1 0.7 0.3 0 |

The last input if 4X4 matrix which indicates the distance values as number of papers is total of 4.