Local Search for scheduling

Ebraheem Kashkoush

May 15, 2020

0.1 Abstract

In this project we use Local Search to solve scheduling problime, the problem is :
Scheduling on restricted uniformly related machines.
Input:An integer number of machines m $\geq$ 2. A set of n jobs J={1,2,...,n} where job j has an integer processing time $P_j > 0$ Machines speeds $s_i \in \{1, 2, 4\}$ fori= 1,2,...,m
Goal:Find a partition (assignment) of the jobs into non-empty subsets $I_1, I_2, ..., I_m$
Objective:...
.

Note: it's not the best way to solve the problem using Local Search but i use this problem to build example of using Local Search we can get better soloution by using compination of B&B and Local Search for example.
in my soloution some time it enter to local minimum for that reason we will discuss on the future work how we can solve this problem and also how to optimize the algorthim run time.

0.2 algorthim

0.2.1 initial soloution

- first we read the input files,we have two inputs the first one for tasks each line in input file has task time value ,and the second file for machines each line have the speed of new machine.
  we enter the input into two vectors 1- vector<Node> J , and 2- vector<machin> M ,where J hold tasks (Node hase two value : index and task time) ,and M hold machines
  each machine has :
  int speed :- hold machine speed {1,2,4}
  int TasksTime; int index; std::map¡int,Node¿ Tasks;