Approximation Algorithms Lecture 4

Contents

1 Recap

2 Content

2.1 Load Balancing / Parallel Machine Scheduling

Given m identical machines and n processes/jobs on them, consider an assignment of jobs to machines. The maximum load (sum of times taken by jobs assigned to a machine) of the assignment is called the *makespan* of the assignment. Find an assignment with the minimum makespan.

Greedy algorithm (list-scheduling)

- Assign the job to the machine which is the least loaded.

Bad example

Consider the example where we have 100 instances of a job that takes time 1, and one instance of a job that takes time 10, with a total of 10 machines. Here $\mathbf{OPT} = 11$, but the greedy algorithm gives 20. Note that we can get such an example for any number of machines, with the ratio as $\frac{m+1}{2m}$.

Claim 0.1

This algorithm has $\mathbf{ALG} \leq 2\mathbf{OPT}$.

Proof. Consider the machine j corresponding to the makespan. Let p_k be the time taken by the job k which is the last job on machine j, and let T be the load on this machine before adding this job. By the choice of which machine is chosen, we have $load_i \geq T$ at the time when we added job k, for all i. Then we have $\frac{\sum_{i=1}^{n} p_i}{m} \geq \frac{\sum_{i=1}^{n} load_i}{m} \geq T$. Note that $\mathbf{OPT} \geq \frac{\sum_{i=1}^{n} p_i}{m} \geq T$. Now also note that $\mathbf{OPT} \geq \max_i p_i \geq p_k$, so we have $\mathbf{OPT} \geq p_k$.

By the definition of k, we have $p_k + T$ as the makespan. So we have $\mathbf{OPT} \geq 2\mathbf{ALG}$, so we are done. □

2.2 Vertex Cover

Given an undirected unweighted graph $G = (V, E)$, a vertex cover is a set $S \subseteq V$ such that each edge e in E has at least one endpoint in S. Find the minimum vertex cover in a given graph G.

Matching

A matching in a graph $G = (V, E)$ is a subset of edges which are independent (two edges are independent if they don't share an endpoint).

Maximal matching

A matching is maximal if its size can't be increasing by adding another edge.

Claim 0.2

Minimum vertex cover is at least as large as any matching.

Proof. Note that for any edge in the matching, at least one vertex is needed in the vertex cover. Hence the size of any vertex cover is at least the size of any matching. □

Algorithm

1. Find a maximal matching in G.
2. For every matched edge, put both endpoints of the edge in the solution S.

Claim 0.3

S is a vertex cover.

Proof. Suppose there is an edge e which is not adjacent to any vertex in S. We claim that we can add this edge to the matching, which will contradict the maximality. Note that S is the set of vertices which are adjacent to at least one edge in the matching. Since e is not incident on any vertex in S, there is no edge in the matching that shares an endpoint with e, whence we are done. □

Claim 0.4

$|S| \leq 2|\mathbf{OPT}|$.

Proof. Note that $|\mathbf{OPT}|$ is the size of the minimum vertex cover. Then we have $|\mathbf{OPT}| \geq |M|$. But we have $|S| \leq 2|M|$, which gives us $|S| \leq 2|M| \leq 2|\mathbf{OPT}|$, as needed. □