The Design and Analysis of Algorithms

Lecture 22 Approximation Algorithm III

Zhenbo Wang

Department of Mathematical Sciences, Tsinghua University

Content

LP Rounding: Vertex Cover

Generalized Load Balancing

Weighted Vertex Cover

Def. Given a graph $G = (V, E)$, a vertex cover is a set $S \subseteq V$ such that each edge in E has at least one end in S.

- *Weighted Vertex Cover.* Given a graph G with vertex weights, find a vertex cover of minimum weight.

$$
\begin{aligned}
\min \sum_{i \in V} w_i x_i & & \\
s.t. x_i + x_j \geq 1 & \quad \forall (i, j) \in E, & \text{(ILP)} \\
x_i \in \{0, 1\} & \quad \forall i \in V. &
\end{aligned}
$$

Weighted Vertex Cover: LP Relaxation

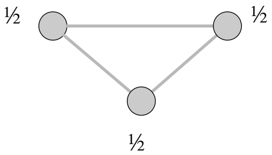
- The LP relaxation of weighted vertex cover:

$$
\begin{aligned}
&\min \sum_{i \in V} w_i x_i && && \\
&s.t. x_i + x_j \geq 1 && \forall (i,j) \in E, && \text{(LP)} \\
&x_i \geq 0, && \forall i \in V. &&
\end{aligned}
$$

Obs. Optimal value of (LP) is $\leq$ optimal value of (ILP).

Pf. LP has fewer constraints.

Note. LP is not equivalent to vertex cover.



Q. How can solving LP help us find a small vertex cover?

A. Solve LP and round fractional values.

Weighted Vertex Cover: LP Rounding Algorithm

Lemma 1

If x^* is optimal solution to (LP), then $S = \{i \in V : x_i^* \geq 1/2\}$ is a vertex cover whose weight is at most twice the min possible weight.

Pf. [S is a vertex cover]

Consider an edge $(i,j) \in E$.

Since $x_i^* + x_j^* \geq 1$, either $x_i^* \geq 1/2$ or $x_j^* \geq 1/2$,
$\Rightarrow (i,j)$ covered.

Pf. [S has desired cost]

Let S^* be optimal vertex cover. Then

$$\sum_{i \in S^*} w_i \geq \sum_{i \in V} w_i x_i^* \geq \sum_{i \in S} w_i x_i^* \geq \frac{1}{2} \sum_{i \in S} w_i. \ \square$$

Weighted Vertex Cover and Inapproximability

Theorem 2

The rounding algorithm is a 2-approximation algorithm.

Pf. Lemma + fact that LP can be solved in poly-time. □

Theorem 3 (Dinur-Safra, Ann. Math., 2004)

If P ≠ NP, then no ρ-approximation for weighted vertex cover for any $\rho < 1.3606$ (even if all weights are 1).

On the Hardness of Approximating Minimum Vertex Cover

Irit Dinur* Samuel Safra†

May 26, 2004

Abstract

We prove the Minimum Vertex Cover problem to be NP-hard to approximate to within a factor of 1.3606, extending on previous PCP and hardness of approximation technique. To that end, one needs to develop a new proof framework, and borrow and extend ideas from several fields.

Weighted Vertex Cover: Inapproximability

Theorem 4 (Khot-Regev, J. Comput. Syst. Sci., 2008)

Based on Unique Game Conjecture, vertex cover is hard to approximate within $2 - \epsilon$ for any $\epsilon > 0$.

Vertex Cover Might be Hard to Approximate to within $2 - \varepsilon$

Subhash Khot * Oded Regev †

Abstract

Based on a conjecture regarding the power of unique 2-prover-1-round games presented in [Khot02], we show that vertex cover is hard to approximate within any constant factor better than 2. We actually show a stronger result, namely, based on the same conjecture, vertex cover on k-uniform hypergraphs is hard to approximate within any constant factor better than k.

Generalized Load Balancing

Input. Set of m machines M; set of n jobs J.

- For each job there is just a subset of machines to which it can be assigned.
- Job j must run contiguously on an authorized machine in $M_j \subseteq M$.
- Each machine can process at most one job at a time.

Def. Let $J(i)$ be the subset of jobs assigned to machine i.

The *load* of machine i is $L_i = \sum_{j \in J(i)} t_j$.

Def. The *makespan* is the maximum load on any machine $L = \max_i L_i$.

Def. *Generalized Load balancing.* Assign each job to a machine to minimize makespan.

Generalized Load Balancing: ILP

ILP formulation

x_{ij} is time of machine i spends processing job j.

$$
\begin{aligned}
& \min \ L \\
& s.t. \sum_i x_{ij} = t_j && \forall j \in J, \\
& \quad \sum_j x_{ij} \le L && \forall i \in M, \\
& \quad x_{ij} \in \{0, t_j\} && \forall j \in J \text{ and } i \in M_j, \\
& \quad x_{ij} = 0 && \forall j \in J \text{ and } i \notin M_j.
\end{aligned} \tag{ILP}
$$

Generalized Load Balancing: Relaxation

LP relaxation

$$
\begin{aligned}
&\min \ L \\
&s.t. \sum_i x_{ij} = t_j && \forall j \in J, \\
&\quad \sum_j x_{ij} \leq L && \forall i \in M, \\
&\quad x_{ij} \geq 0 && \forall j \in J \text{ and } i \in M_j, \\
&\quad x_{ij} = 0 && \forall j \in J \text{ and } i \notin M_j.
\end{aligned}
\qquad \text{(LP)}
$$

Generalized Load Balancing: Lower Bounds

Lemma 5

The optimal makespan $L^* \geq \max_j t_j$.

Pf. Some machine must process the most time-consuming job. □

Lemma 6

Let L be optimal value to the LP. Then, optimal makespan $L^* \geq L$.

Pf. LP has fewer constraints than IP formulation. □

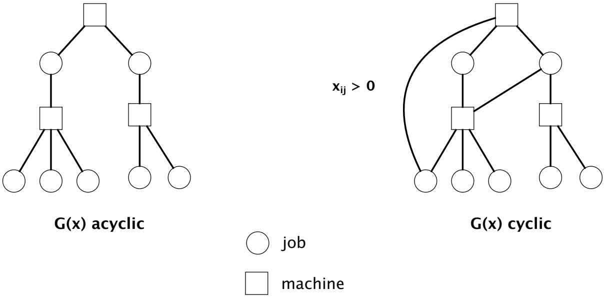
Generalized Load Balancing: Structure of LP Solution

Lemma 7

Let x be solution to LP. Let $G(x)$ be the graph with an edge between machine i and job j if $x_{ij} > 0$. Then there exists x such that $G(x)$ is acyclic.

- If LP solver doesn't return such an x, can transform x into another LP solution where $G(x)$ is acyclic.

Pf. (deferred)



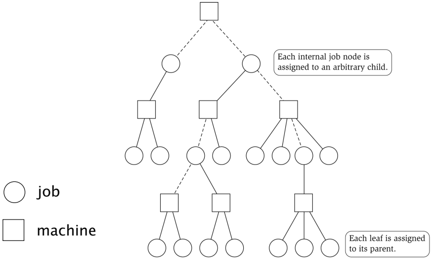
Generalized Load Balancing: Rounding

- *Rounded solution.* Find LP solution x where $G(x)$ is a forest.
- Root forest $G(x)$ at some arbitrary machine node r.
- If job j is a leaf node, assign j to its parent machine i.
- If job j is not a leaf node, assign j to any one of its children.

Lemma 8

Rounded solution only assigns jobs to authorized machines.

- If job j is assigned to machine i, then $x_{ij} > 0$. LP solution can only assign positive value to authorized machines. □



Generalized Load Balancing: Analysis

Lemma 9

If job j is a leaf node and machine $i = parent(j)$, then $x_{ij} = t_j$.

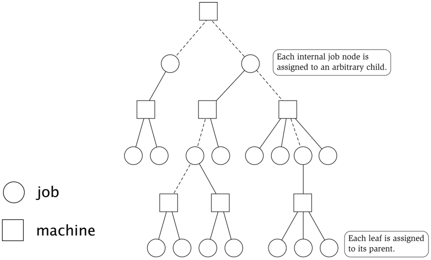
Pf. Since i is a leaf, $x_{ij} = 0$ for all $j \neq parent(i)$.

LP constraint $\sum_i x_{ij} = t_j$ guarantees $x_{ij} = t_j$. $\square$

Lemma 10

At most one non-leaf job is assigned to a machine.

Pf. The only possible non-leaf job assigned to i is $parent(i)$. $\square$



Generalized Load Balancing: Analysis

Theorem 11

Rounded solution is a 2-approximation.

Pf. Let $J(i)$ be the jobs assigned to machine i.

By lemma 10, the load L_i on machine i has two components:

- leaf nodes:

$$\sum_{j \in J(i),\ j \text{ is a leaf}} t_j = \sum_{j \in J(i),\ j \text{ is a leaf}} x_{ij} \leq \sum_{j \in J} x_{ij} \leq L \leq L^*$$

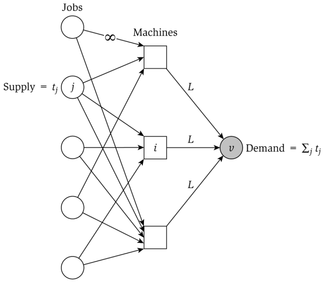
- parent: $t_{parent(i)} \leq L^*$.

Thus, the overall load $L_i \leq 2L^*$. □

Generalized Load Balancing: Flow Formulation

Flow formulation of LP.

$$
\begin{array}{ll}
\sum_i x_{ij} = t_j & \forall j \in J \\
\sum_j x_{ij} \leq L & \forall i \in M \\
x_{ij} \geq 0 & \forall j \in J \text{ and } i \in M_j \\
x_{ij} = 0 & \forall j \in J \text{ and } i \notin M_j
\end{array}
$$



Obs. Solution to feasible flow problem with value L are in 1-to-1 correspondence with LP solutions of value L.

Generalized Load Balancing: Structure of Solution

Lemma 7

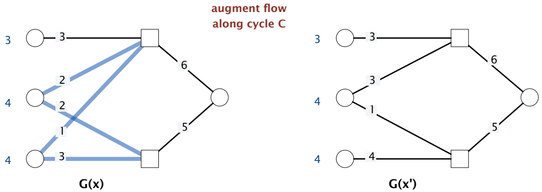
Let (x, L) be solution to LP. Let $G(x)$ be the graph with an edge between machine i and job j if $x_{ij} > 0$. We can find another solution (x', L) such that $G(x')$ is acyclic.

Pf. Let C be a cycle in $G(x)$.

Augment flow along the cycle C.

At least one edge from C is removed (and none are added).

Repeat until $G(x')$ is acyclic.



Conclusions

- *Running time.* The bottleneck operation in our 2-approximation algorithm is to solve LPs with $mn + 1$ variables.
- *Remark.* Can solve LP using flow techniques on a graph with $m + n + 1$ nodes: binary search to find L^*, and find feasible flow if it exists.
- *Extensions: unrelated parallel machines.* [Lenstra-Shmoys-Tardos, Math. Program., 1990]
  - Job j takes t_{ij} time if processed on machine i.
  - 2-approximation algorithm via LP rounding.
  - If $P \neq NP$, then no ρ-approximation exists for any $\rho < 3/2$.

Mathematical Programming 46 (1990) 259–271 259
North-Holland

APPROXIMATION ALGORITHMS FOR SCHEDULING UNRELATED PARALLEL MACHINES

Jan Karel LENSTRA
Eindhoven University of Technology, Eindhoven, The Netherlands, and Centre for Mathematics and Computer Science, Amsterdam, The Netherlands

David B. SHMOYS and Éva TARDOS
Cornell University, Ithaca, NY, USA

Homework

- Read Chapter 11 of the textbook.
- Exercises 8 & 9 in Chapter 11.