Scheduling on Unrelated Parallel Machines

Approximation Algorithms,

V. V. Vazirani

Book Chapter 17

Nicolas Karakatsanis, 2008

Description of the problem

Problem 17.1 (Scheduling on unrelated parallel machines)

Given a set J of jobs, a set M of machines, and for each $j \in J$ and $i \in M$, $p_{ij} \in Z^+$ the time taken to process job j on machine i, the problem is to schedule the jobs on the machines so as to minimize the *makespan*, i.e., the maximum processing time for any machine. We will denote the number of jobs by n and the number of machines by m

Unrelated machines
No relation between the processing times of a job on the different machines

Identical machines
Each job j has the same running time, say p_{ij}, on each of the machines
Chapter 10 $\rightarrow$ Minimum makespan scheduling (admits PTAS)

Uniform machines
The processing time for job j on machine i is $p_{ij} = p_j / s_i$, s_i is the speed of machine i

Parametric Pruning in an LP setting

Integer Program

minimize t

subject to
$$\sum_{i \in M} x_{ij} = 1, \qquad j \in J$$
$$\sum_{j \in J} x_{ij} p_{ij} \le t, \qquad i \in M$$
$$x_{ij} \in \{0,1\}, \quad i \in M, \quad j \in J$$

Indicator variable x_{ij}
denotes whether job j is scheduled on machine I

Makespan t
Objective is to minimize makespan t

First constraint set
Ensures each job is scheduled on one of the machines

Second constraint set
Ensures each machine has processing time of at most t

Unbounded integrality gap

Example 17.2
Suppose we have only one job, which has a processing time of m on each of the m machines. Clearly, the minimum makespan is m.
However the optimal solution to the LP-relaxation is to schedule the job to the extent of $1/m$ on each machine, leading to an objective function value of 1.
Therefore : Integrality gap = m

Parametric Pruning in an LP setting

- "Unfair" advantage given to linear relaxation
  - Integer program automatically sets $x_{ij} = 0$ if $p_{ij} > t$
  - Linear relaxation allowed to set nonzero values: $0 \le x_{ij} \le 1$
- Possible solution
  - Issue following constraint to the linear relaxation:
  - $\forall i \in M, j \in J$: if $p_{ij} > t$ then $x_{ij} = 0$
  - However, this is NOT a linear constraint
- Actual solution provided by parametric pruning technique
  - Parameter $T \in Z^+$ is our guess for a lower bound on optimal makespan
  - Define $S_T = \{(i, j) \mid p_{ij} \le T\}$
  - Define a family of linear programs $LP(T)$ one for each value of $T \in Z^+$

Parametric Pruning in an LP setting

$LP(T)$ uses the variable x_{ij} for $(i,j) \in S_T$ and asks if there is a feasible, fractional schedule of makespan $\leq T$ using the restricted possibilities

$$\sum_{i:(i,j)\in S_T} x_{ij} = 1, \qquad j \in J$$

$$\sum_{j:(i,j)\in S_T} x_{ij} p_{ij} \leq T, \qquad i \in M$$

$$x_{ij} \geq 0, \qquad (i,j) \in S_T$$

Properties of extreme point solutions

Initial step of the algorithm

Perform appropriate binary search to find T^*

T^* is the smallest value of T such that *LP(T)* has a feasible solution.

Clearly T^* is a lower bound on OPT : $T^* \leq OPT$

LP-rounding algorithm

round an extreme point solution to $LP\left(T^*\right)$ to find a schedule having makespan $\leq 2T^*$

Extreme point solutons to *LP(T)* provide many useful properties

Properties of extreme point solutions

Lemma 17.3

Any extreme point solution to *LP(T)* has at most $n+m$ nonzero values

Proof

Let $r = |S_T|$ represent the number of variables on which *LP(T)* is defined.
A fesible solution to *LP(T)* is an extreme point solution iff $(\Leftrightarrow)$
it corresponds to setting r linearly constraints of *LP(T)* to equality
At least $r-(n+m)$ of these constraints must be chosen from the third set of constraints (i.e. $x_{ij} \geq 0$)
Therefore at least $r-(n+m)$ variables of x_{ij} must be set to 0
So, at most $r-(n+m)$ variables are nonzero
Conclusion:
Any extreme point solution has at most $n+m$ nonzero variables

Properties of extreme point solutions

Let $\boldsymbol{x}$ be an extreme point solution to *LP(T)*.
Job j is *integrally set in* $\boldsymbol{x}$ if it is entirely assigned to one machine.
Otherwise job j is *fractionally set in* $\boldsymbol{x}$

Corollary 17.4

Any extreme point solution to *LP(T)* must set at least $n-m$ jobs integrally

Proof

Let $\boldsymbol{x}$ be an extreme point solution to *LP(T)*, and let α and β be the number of jobs that are integrally and fractionally set by $\boldsymbol{x}$, respectively
Each job of the later kind is assigned to at least 2 machines and therefore results in at least 2 nonzero entries in $\boldsymbol{x}$. Hence we get
$\alpha+\beta=n$ and $\alpha+2\beta\leq n+m$
Therefore, $\beta\leq m$ and $\alpha\geq n-m$

Define $G=(J,M,E)$ to be the bipartite graph on vertex set $J\cup M$ such that $(j,i)\in E$ iff $x_{ij}\neq 0$. Let $F\subset J$ be the set of jobs that are fractionally set in $\boldsymbol{x}$, and let H be the subgraph of G induced on vertex set $F\cup M$. A matching in H will be called a *perfect matching* if it matches every job $j\in F$

Scheduling algorithm

Algorithm 17.5 (Scheduling on unrelated parallel machines)

1. By a binary search in the interval $[\alpha/m, \alpha]$, find the smallest value of $T \in Z^+$ for which *LP(T)* has a feasible solution. Let this value be T^*
2. Find an extreme point solution, say $\boldsymbol{x}$, to *LP(T)*.
3. Assign all integrally set jobs to machines as in $\boldsymbol{x}$
4. Construct graph H and find a perfect matching M in it (e.g. using the procedure of Lemma 17.7)
5. Assign fractionally set jobs to machines according to matching M

Additional properties of extreme point solutions

A connected graph on vertex set V is a *pseudo-tree* if it contains at most $|V|$ edges.
A graph is a *pseudo-forest* if each of its connected components is a pseudo-tree.

Lemma 17.6

Graph G is a pseudo-forest

Proof
We will show that the number of edges in each connected component of G is bounded by the number of vertices in it. Hence, each connected component is a pseudo-tree.
Consider a connected component G_c. Restrict *LP(T)* and $\boldsymbol{x}$ to the jobs and machines of G_c only, to obtain $LP_c(T)$ and x_c. Let $x_{\bar{c}}$ represent the rest of $\boldsymbol{x}$.
The important observation is that x_c must be an extreme point solution to $LP_c(T)$.
Suppose that this is not the case. Then, x_c is a convex combination of two feasible solutions to $LP_c(T)$. Each of these, together with $x_{\bar{c}}$ form a feasible solution to *LP(T)*. Therefore, $\boldsymbol{x}$ is a convex combination of two feasible solutions to *LP(T)*, leading to a contradiction
Now, applying Lemma 17.3, we get that G_c is a pseudo-tree

Additional properties of extreme point solutions

Lemma 17.7

Graph H has a perfect matching

Proof
Graph H is also a pseudo-forest.
In H, each job has a degree of at least 2. So, all leaves in H must be machines. Kepp matching a leaf with the job it is incident to, and remove them both from the graph. (at each stage all leaves must be machines.)
I the end we will be left with even cycles (since we started with a bipartite graph).
Match off alternate edges of each cycle.
This gives a perfect matching in H.

Additional properties of extreme point solutions

Theorem 17.8

Algorithm 17.5 achieves an approximation guarantee of factor 2 for the problem of scheduling on unrelated parallel machines.

Proof
Clearly, $T^* \leq OPT$, since *LP(OPT)* has a feasible solution. The extreme point solution ***x*** to $LP(T^*)$ has a fractional makespan of $\leq T^*$.
Therefore, the restriction of ***x*** to integrally set jobs has a (integral) makespan of $\leq T^*$. Each edge (i, j) of *H* satisfies $p_{ij} \leq T^*$.
The perfect matching found in *H* schedules at most one extra job on each machine.
Hence the total makespan is $\leq 2T^* \leq 2 \cdot OPT$.
Finally the algorithm clearly runs in polynomial time