Scheduling

R. Chandrasekaran

March 24, 2020

Scheduling Summer 1996 R. Chandrasekaran

1. Approximation to m-Machine Flow Shops

Vector Summation in Banach Space and polynomial Algorithms for Flow Shops and Open Shops, by S. Sevest'janov, Mathematics of Operations Research, 20, (1995), pp. 90-103

A Vector-Sum Theorem and its Application to improving Flow Shop Guarantees by Imre Barany, Mathematics of Operations Research, vol 6 #3, August 1981, pp. 445-452.

The original form of this theorem is known as *Compact Vector Summation* (CVS) thoerem due to Steinitz (1913) where instead of $\frac{3}{2}d$, was a function $\phi(d)$. Sevestyanov (1978) has d instead of $\frac{3}{2}d$ with slightly different complexity. The discussion here is from Barany's paper.

Theorem 1 *For a finite set $V = \{v_1, v_2, ..., v_n\} \subseteq R^d$ with*

$$\begin{aligned} \sum_{i=1}^{n} v_i &= 0 \\ \|v_i\| &\leq 1 \qquad \forall i \end{aligned}$$

where $\|x\|$ is any norm for x [here we use $\|x\| = \max_{j=1}^{d} |x_j|$] there is a permutation $i_1, i_2, ..., i_n$ of the set $\{1, 2, .., n\}$ such that

$$\max_{1 \leq k \leq n} \left\| \sum_{j=1}^{k} v_{i_j} \right\| \leq \frac{3}{2} d$$

This permutation can be found in $O(n^2 d^3 + nd^4)$ steps.

First we will show the application of this result to get approximate solution to flowshops. Consider a m-machine n-job flow shop. Let $t_{i,j}$ represent the processing time of job i on machine j. Suppose $t_{i,j} \leq K$ for all i and j. Let $M_j = \sum_{i=1}^{n} t_{i,j} \leq nK$ for all j; and $M = \max_{j=1}^{m} M_j \leq nK$. From this input, we can modify the data so that we get new processing times $t'_{i,j}$ that satisfy:

$$\begin{aligned} t_{i,j} &\leq t'_{i,j} \leq K \\ \sum_{i=1}^{n} t'_{i,j} &= M \qquad \forall j \end{aligned}$$

and this can be done in polynomial time. An example is shown below:

$t =$

3	4	0	5
2	2	2	2
3	2	1	4

$M = 12; M_1 = 12; M_2 = 8; M_3 = 10; K = 5$. Its corresonding t' looks like:

$t' =$

3	4	0	5
5	3	2	2
5	2	1	4

Theorem 2 *There exists a permutation schedule for which the finish time T satisfies the inequalties:*

$$M \leq T \leq M + (m-1)[\frac{3m-1}{2}]K$$

Moreover this schedule can be obtained in $O(n^2m^3 + nm^4)$ steps.
Proof. *(Using Theorem above): First we construct a fictitious problem with same number of jobs and machines for which the processing times are gien by $t'_{i,j}$ where*

$$\begin{aligned} t_{i,j} &\leq t'_{i,j} \leq K \text{ for all } i, j \\ \sum_{i=1}^{n} t'_{i,j} &= M \text{ for all } j \end{aligned}$$

This is easy to in $O(nm)$ steps. Given a permutation $i_1, i_2, ..., i_n$ of jobs, the finish time for this schedule is given by

$$T = \max_{1=k_0<k_1<k_2<...<k_m=n} \left[\sum_{j=1}^{m-1} \sum_{s=k_j}^{k_j+1} t_{i_s,j+1} \right]$$

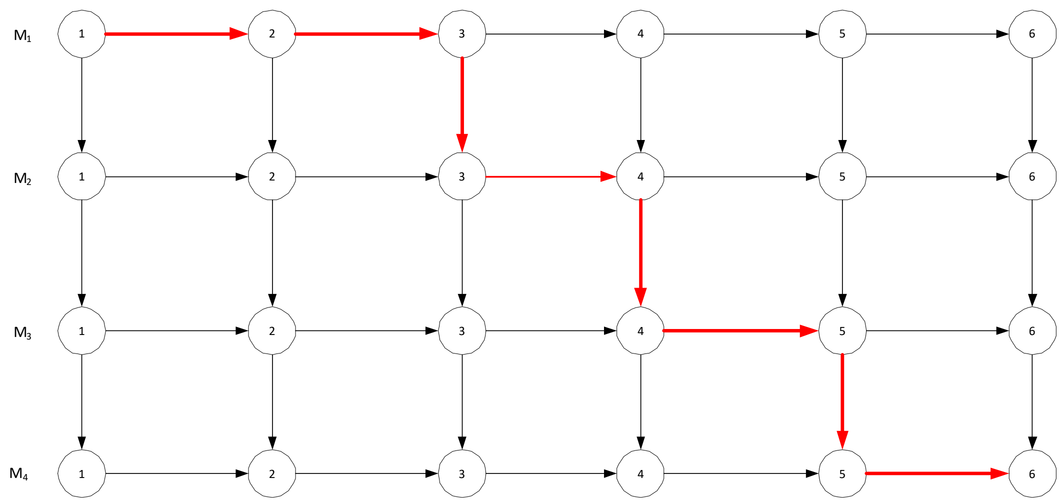
Hence

$$T' = \max_{1=k_0<k_1<k_2<...<k_m=n} \left[\sum_{j=1}^{m-1} \sum_{s=k_j}^{k_j+1} t'_{i_s,j+1} \right]$$

Clearly $M \leq T \leq T'$. ■

Given a permutation $(i_1, i_2, ..., i_n)$ of jobs and a corresponding permutation schedule, we get a precedence graph that is shown below for permutation

(1, 2, .., 6):



Please note the horizontal edges depict the permutation schedule and the vertical ones depict the order in which jobs must go through machines in the flowshop. All horizontal edges go to the right and all vertical ones go down. The longest path from top left to bottom right gives us the length of the schedule where each node has weight corresponding the processing time of the job on the machine. One such path is indicated in red. This is what is used in the formula for length of the permutation schedule with $(i_1, i_2, ..., i_n)$ and processing times $t_{i,j}$:

$$T = \max_{1=k_0 \leq k_1 \leq ... \leq k_m = n} \left[\sum_{j=0}^{m-1} \sum_{s=1}^{k_j} t_{i_s, j+1} \right]$$

Similarly for $t'_{i,j}$ we get T' for the same permutation:

$$T' = \max_{1=k_0 \leq k_1 \leq ... \leq k_m = n} \left[\sum_{j=0}^{m-1} \sum_{s=1}^{k_j} t'_{i_s, j+1} \right]$$

Since $t_{i,j} \leq t'_{i,j} \forall i, j; M = \max_j M_j$ where $M_j = \sum_{i=1}^{n} t_{i,j}$ it follows that

$$M \leq T \leq T'$$

The next equation needs some explanation: We first state it and show it for the example above with $m = 4; n = 6$.

$$T' = M + \max[\sum_{j=1}^{m-1} t'_{i_{k_j}, j+1} + \sum_{j=1}^{m-1} \sum_{s=1}^{k_j} (t'_{i_s, j} - t'_{i_s, j+1})]$$

$$\leq \quad M + (m-1)K + \max[\sum_{j=1}^{m-1} \sum_{s=1}^{k_j} (t'_{i_s,j} - t'_{i_s,j+1})]$$

where max is taken over $1 = k_0 \leq k_1 \leq ... \leq k_m = n$. In the above diagram for the red path $1 = k_0; k_1 = 3; k_2 = 4; k_3 = 5; k_4 = 6 - n$. Assume the permutation is $(1, 2, ..., n)$. Hence $i_s = s$ in the above equation. Now we concentrate on the expression $\sum_{j=1}^{m-1} \sum_{s=1}^{k_j} (t'_{i_s,j} - t'_{i_s,j+1})]$. Expanded it looks like:

$i \rightarrow$ / $j \downarrow$	$k_0 = 1$	2	$k_1 = 3$	$k_2 = 4$	$k_3 = 5$	$k_4 = 6$
1	$t'_{1,1} - \mathbf{t}'_{1,2}$	$t'_{2,1} - \mathbf{t}'_{2,2}$	$t'_{3,1} - \mathbf{t}'_{3,2}$			
2	$\mathbf{t}'_{1,2} - \mathbf{t}'_{1,3}$	$\mathbf{t}'_{2,2} - \mathbf{t}'_{2,3}$	$\mathbf{t}'_{3,2} - \mathbf{t}'_{3,3}$	$t'_{4,2} - \mathbf{t}'_{4,3}$		
3	$\mathbf{t}'_{1,3} - t'_{1,4}$	$\mathbf{t}'_{2,3} - t'_{2,4}$	$\mathbf{t}'_{3,3} - t'_{3,4}$	$\mathbf{t}'_{4,3} - t'_{4,4}$	$t'_{5,3} - t'_{5,4}$	

All terms in bold cancel out and we get this equals:

$i \rightarrow$ / $j \downarrow$	$k_0 = 1$	2	$k_1 = 3$	$k_2 = 4$	$k_3 = 5$	$k_4 = 6$
1	$t'_{1,1}$	$t'_{2,1}$	$t'_{3,1}$			
2				$t'_{4,2}$		
3	$-t'_{1,4}$	$-t'_{2,4}$	$-t'_{3,4}$	$-t'_{4,4}$	$t'_{5,3} - t'_{5,4}$	

$[\sum_{j=1}^{m-1} t'_{i_{k_j},j+1}$ looks like:

$i \rightarrow$ / $j \downarrow$	$k_0 = 1$	2	$k_1 = 3$	$k_2 = 4$	$k_3 = 5$	$k_4 = 6$
1		$t'_{3,2}$				
2			$t'_{4,3}$			
3				$t'_{5,4}$		

and the sum of these two looks like:

$i \rightarrow$ / $j \downarrow$	$k_0 = 1$	2	$k_1 = 3$	$k_2 = 4$	$k_3 = 5$	$k_4 = 6$
1	$t'_{1,1}$	$t'_{2,1}$	$t'_{3,1}$			
2			$t'_{3,2}$	$t'_{4,2}$		
3	$-t'_{1,4}$	$-t'_{2,4}$	$-t'_{3,4}$	$t'_{4,3} - t'_{4,4}$	$t'_{5,3}$	

Consider the expression for $M' = M = M'_4 = t'_{1,4} + t'_{2,4} + t'_{3,4} + t'_{4,4} + t'_{5,4} + t'_{6,4}$. Adding to the above we get:

$i \rightarrow$ / $j \downarrow$	$k_0 = 1$	2	$k_1 = 3$	$k_2 = 4$	$k_3 = 5$	$k_4 = 6$
1	$t'_{1,1}$	$t'_{2,1}$	$t'_{3,1}$			
2			$t'_{3,2}$	$t'_{4,2}$		
3				$t'_{4,3}$	$t'_{5,3}$	
4					$t'_{5,4}$	$t'_{6,4}$

which is the expression $\sum_{j=0}^{m-1}\sum_{s=1}^{k_j} t'_{i_s,j+1}$. Hence the result follows. The last inequality in the equation comes from the fact $t'_{i,j} \leq K \forall i,j$.

Now to transform this so that we can apply the first thereom, we let:

$$v_i = [(t'_{i,1} - t'_{i,2}), (t'_{i,2} - t'_{i,3}), ..., (t'_{i,m-1} - t'_{i,m})] \in \mathbf{R}^{m-1}; i = 1, 2, ..., n$$

$$\sum_{i=1}^{n} v_i = [(\sum_{i=1}^{n} t'_{i,1} - \sum_{i=1}^{n} t'_{i,2}), (\sum_{i=1}^{n} t'_{i,2} - \sum_{i=1}^{n} t'_{i,3}), ..., (\sum_{i=1}^{n} t'_{i,m-1} - \sum_{i=1}^{n} t'_{i,m})]$$

But $\sum_{i=1}^{n} t'_{i,j} = M \forall j$. Hence $\sum_{i=1}^{n} v_i = 0 \in \mathbf{R}^{m-1}$. Since $t'_{i,j} \leq K \forall i,j$; it follows that $\left|t'_{i,j} - t'_{i,j+1}\right| \leq K \forall i,j$.

Moreover,

$$\max[\sum_{j=1}^{m-1}\sum_{s=1}^{k_j}(t'_{i_s,j} - t'_{i_s,j+1})] \leq (m-1)\max_{k=1}^{n}\left\|\sum_{s=1}^{k} v_{i_s}\right\|$$

with $\|x\| = \max_i |x_i|$ which is a norm called the $\infty-$ norm. The first theroem states there is a permutation $i_1, i_2, ..., i_n$ for which

$$\max_{k=1}^{n}\left\|\sum_{s=1}^{k} v_{i_s}\right\| \leq \frac{3(m-1)}{2}K$$

which in turn implies

$$M \leq T \leq T' \leq M + (m-1)(\frac{3(m-1)}{2})K$$

The last term in the above is independent of n the number of jobs. Hence, as the number of jobs increases, this algorithm tends to optimal solution. What remains is a proof of the first theorem which we give now. I am following Barany's paper closely.

Thereom: For a finite set $V = \{v_1, v_2, ..., v_n\} \sqsubseteq \mathbf{R}^d$ with $\sum_{i=1}^{n} v_i = 0; \|v_i\| \leq 1; i = 1, 2, ..., n$ there is a permutation $i_1, i_2, ..., i_n$ such that $\max_{k=1}^{n}\left\|\sum_{s=1}^{k} v_{i_s}\right\| \leq \frac{3}{2}d$. Moreover, this permutation can be found in $O(n^2d^3 + nd^4)$ steps.

Proof: Let

$$\sum_{j=1}^{n} \gamma_j v_j = 0$$

We call the vector γ as a *linear dependence*. Given the conditions of the theorem, $\gamma = (0, 0, ..., 0) \in \mathbf{R}^n$ and $\gamma = (1, 1, ..., 1) \in \mathbf{R}^n$ are both linear dependences. The algorithm produces a finite sequence of linear dependences $\gamma^0, \gamma^1, ..., \gamma^p$. For $i = 1, 2, ..., p$ let

$$\begin{aligned} A_i &= \{j : \gamma_j^i = 1\} \\ B_i &= \{j : 0 < \gamma_j^i < 1\} \\ C_i &= \{j : \gamma_j^i = 0\} \end{aligned}$$

The set of linear dependences will satisfy the following relations:

(a) $0 \leq \gamma_j^i \leq 1; i = 1, 2, ..., p; j = 1, 2, ..., n$

(b)$\sum_{j=1}^{n} \gamma_j^i v_j = 0; i = 1, 2, ..., p$

(c)$|B_i| \leq d; i = 1, 2, ..., p$

(d)$A_{i+1} \supset A_i; i = 1, 2, ..., p-1 \Leftrightarrow$ the set of components that are equal to 1 in γ^{i+1} includes those in γ^i and there is at least one more in γ^{i+1}.

(e)$A_p = \{1, 2, ..., n\} \Leftrightarrow \gamma^p = (1, 1, ..., 1) \in \mathbf{R}^n$

(f)$|B_{i+1} \cup A_{i+1} \backslash A_i| \leq 2d; i = 1, 2, ..., p-1 \Leftrightarrow$ the number of positive elements in γ^{i+1} is no more than $2d$ larger than the numebr of 1's in γ^i.