# The Parallel Machine Scheduling Problem with Path-like Constraints

Yann Hendel, Wieslaw Kubiak
Memorial University, Canada, Yan.Hendel@gmail.com, WKubiak@mun.ca

## 1 Presentation of the problem

We are given a set of different jobs $\{J_1, \ldots, J_m\}$. Each job $J_i$ has $c_i$ copies. All these job copies have to be scheduled on $F$ parallel machines. We break with the usual assumption that any subset of jobs can be processed on any machine. Instead, we assume that only some subsets of jobs may be processed on the same machine. These subsets are defined by all $s$-$t$ paths in an acyclic graph $G = (\{s, t\}, N, A)$, where each arc $i$ represents a job $J_i$ with a processing time $p_i$ and a path from the source $s$ to the sink $t$ of this graph represents a feasible set of jobs that can be executed on the same machine. We search for a schedule, if any exists, which minimizes the makespan.

Our motivation to study this scheduling problem comes from the following network flow context. Given an acyclic directed graph $G = (\{s, t\}, N, A)$, a cost (or *length*) function $p : A \to \mathbb{Z}_{\geq 0}$, a capacity function $c : A \to \mathbb{Z}_{>0}$ and a value $F$ of flow in $G$. Let $f : A \to \mathbb{Z}_{\geq 0}$ with $f_i \leq c_i$ be a flow in $G$ with its value $F$. The *decomposition of the flow $f$ into paths* is a multi-set $D_f$ of $F$ paths (in principle, not all different) such that by pushing a unit of flow on each of these paths we obtain the flow $f$ of $G$. Let $\mathbb{D}_f$ be the set of all the *decompositions of the flow $f$ into paths* and let $D_f \in \mathbb{D}_f$. We denote by $\mu_\pi$ the value of the total flow along the $s$-$t$ path $\pi$ in the decomposition $D_f$. Then, a decomposition $D_f$ is equivalent to a set of pairs $(\pi, \mu_\pi)$ such that $\sum \mu_\pi = F$. Notice that now the $\pi$'s are all different. For each path $\pi$, we denote by $p(\pi)$ its cost, that is the sum of costs of its arcs. Let $\Lambda = \max_{\pi \in D_f} p(\pi)$. We consider flows $f$ of value $F$ that maximize $\sum_{i \in A} f_i$ and such their decompositions $D_f$ that minimize $\Lambda$. We refer to this problem as SLP (for shortest longest path). If the capacity function $c$ obeys the flow conservation law, then there is a flow which saturates every arc. Consequently, the network flow problem becomes the scheduling problem with the number of machines equal the total capacity of the arcs leaving the source $s$.

This scheduling problem generalizes the well-known identical parallel scheduling to minimize makespan, $P||C_{\max}$. The generalization is shown in Figure 1. The instance of SLP showed in Figure 1 has $n$ jobs with positive processing times, there is a single copy of each of these jobs, and $n$ dummy jobs with negligible processing times, there are $k - 1$ copies of each of these jobs. Moreover, $F = k$. The two numbers on each arc in Figure 1 denote respectively its capacity and its processing time. The flow conservation law is met be the arc capacities. Solving SLP for this instance is equivalent to solving $Pk||C_{\max}$ since an $s - t$ path can take any subset of upper arcs. Therefore, each upper arc must be assigned to a path or equivalently to a machine so as to minimize the makespan.

The SLP is related in the literature to the length-bounded maximum flow problem (see Baier [1] for an overview). In this latter problem, one seeks to maximize the flow with the constraint that there is a decomposition of this flow such that the length of its paths are bounded by a constant. It should be noted that the optimal solution of SLP may not yield a maximal flow.

In this talk, we shall assume that the solution to the scheduling problem exists by assuming that the flow function $f$ is given and $f = c$. It remains to find the decomposition $D_f$ which minimizes $\Lambda$ (we denote this problem by $\text{SLP}^f$). We address the following cases of this problem: when the value of the flow is bounded by a constant $k$ (that is there are $k$ parallel machines), we denote

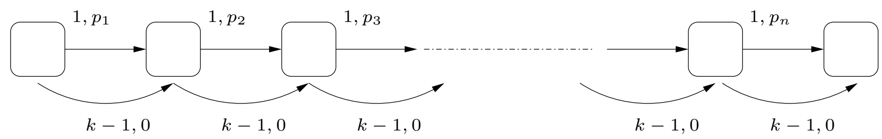


Figure 1: The SLP equivalent to $Pk||C_{\max}$

this problem $\text{SLP}^f_{\text{BFk}}$, when the lengths on the arcs are bounded by a polynomial of the input size, $\text{SLP}^f_{\text{BC}}$, and when the flow is unitary on each arc (i.e. there is a unique copy) $\text{SLP}^f_{\text{UF}}$.

## 2 Main results

In the first part of the presentation, we address some computational complexity issues, in particular the question whether the $\text{SLP}^f$ problem is in the class NP. We give a short certificate for the case when $F$ is bounded by a polynomial of the input size (the $\text{SLP}_{\text{BFk}}$ belongs to this case) and therefore this particular case is proven in NP. We then propose some pathologic instances where any flow decomposition of $f$ that minimizes $\lambda$ in the general $\text{SLP}^f$ problem has an exponential number of different paths. These instances provide some evidence that $\text{SLP}^f$, though $NP$-hard in the strong sense, may not actually be in NP. However, for the case where the cost are bounded, that is ($\text{SLP}^f_{\text{BFk}}$), we provide a second certificate for which the existence of a solution to $\text{SLP}^f_{\text{BFk}}$ can be answered in polynomial time. Thus $\text{SLP}^f_{\text{BFk}}$ is proven in NP.

We then show that $\text{SLP}^f$ remains NP-Hard in the strong sense even for $\text{SLP}^f_{UF}$. However, $\text{SLP}^f_{\text{BFk}}$ is proven ordinary NP-hard. Next, we show that $\text{SLP}^f_{\text{BF2}}$ is equivalent to $P2||C_{\max}$ and we propose a pseudopolynomial time dynamic programming algorithm for $\text{SLP}^f_{\text{BFk}}$. This algorithm generalizes the well-known algorithm for $Pk||C_{\max}$ (see [2]). The time complexity of the dynamic programming algorithm is of $O(n\Delta^k)$, where $\Delta$ is the length of the longest path in the graph. For the unitary costs, the algorithm runs in polynomial time, its complexity is $O(nm^k)$. Finally, we use this dynamic programming algorithm to develop an FPTAS which runs in $O(\frac{k^k n^{k+1}}{\epsilon^k})$ for any $\epsilon > 0$ for the $\text{SLP}^f_{\text{BFk}}$ problem.

Finally, we present open problems and possible directions for future research.

## References

[1] G. Baier (2003). Flows with path restrictions. *PhD thesis, T.U. Berlin*

[2] M.H. Rothkopf (1966). Scheduling independent tasks on parallel processors. *Management Sci.*12, 437-447.