Directed acyclic graph with processing time on Vertices

ILP Solver

Vince Popp, Walker Williams, Sayantan Datta, Brandon Beckwith

Monday, March 16, 2020

ITCS 6115 / 8115 : Spring 2020

Problem Statement

Input: A directed acyclic graph of tasks $G = (V, E)$ with processing time on the vertices (tasks) p_i. A number of processors m.

Output: A mapping of tasks to processors $\pi(i) \in \{1, \ldots, m\}$. A start time on tasks $\sigma(i) \geq 0$. A task cannot start until all its predecessor tasks are complete. Once a task i starts running, it will run for p_i time until completion. Two tasks running on the same processor cannot be running on the same time.

Metric: Minimize the time of completion of the last task.

The Problem

- Example:
  - Assembly line

ILP Formulation: Objective fn

- Objective Function:
  - Minimize the time of completion of the last task

    $\textbf{min}\, Z$

Constraints

1. $Z - c_i \geq 0$, and $\forall i \in V$
2. $c_i = \sigma_i + p_i$, time for completion for a task
3. $\sigma_i = \sigma(i)$ is the starting time for a process
4. Starting time of each process needs to be greater than zero
    4.1 $\sigma_i \geq 0$, and $\forall i \in V$

Constraints(2)

3 A task cannot start until all its predecessor tasks are complete.

$$c_i \leq \sigma_j \ \forall i, j \in V \text{ such that } \quad i \in succ(j)$$
$$\sigma_i + p_i \leq \sigma_j \ \forall i, j \in V \text{ such that } \quad i \in succ(j)$$
$$\sigma_j - \sigma_i \geq p_i \ \forall i, j \in V \text{ such that } \quad i \in succ(j)$$

Constraints(3)

4 Once a task i starts running, it will run for p_i time until completion.

4.1 If a processor has a task i, at a particular time $\sigma(i)$, then the processor has the same task i at the time $\sigma(i) + p_i$

4.2

$$-\sum_{k=t_{ij}}^{t_{ij}+p_i} x_{ijk} \geq -p_i$$

4.3

$$x_{ijk} = \begin{cases} 1 & \text{if task i is allocated to processor } j \text{ at time } k \\ 0 & \text{otherwise} \end{cases}$$

4.4

$$t_{ij} = \begin{cases} \sigma_i & \text{time task i starts in processor j} \\ 0 & \text{otherwise} \end{cases}$$

Constraints(3a)

1. All processors have finished

Complexity Summary

1. $\mathcal{O}(2^{|V|})$, for all possible subsets of tasks that can be assigned to each processor.

ILP Validation

1. Because we required that all tasks and their predecessors are complete, we know that by the end the ILP will always complete every task and find the minimal finish time.