1.1 Problem Definition

In general, a Job-Shop problem is a set of Machines and Jobs. The Jobs consist of multiple Tasks. Each Task needs to be processed for some duration on a specific Machine.

The goal is to find a schedule for all the Jobs.

For the purposes of this project, an instance of a Job-Shop problem is defined by:

- the set of Machines $\mathbb{M}$, with ids $m \in [1, \mathcal{M}]$, where $\mathcal{M} = |\mathbb{M}|$,
- the set of Jobs $\mathbb{J}$, with ids $j \in [1, \mathcal{J}]$, where $\mathcal{J} = |\mathbb{J}|$,
- the set of Tasks $\mathbb{T}$, with ids $t \in [1, \mathcal{T}]$, where $\mathcal{T} = |\mathbb{T}|$,
- a function $f_j: \mathbb{T} \to [1, \mathcal{J}]$, assigning Tasks to their Jobs,
- a function $f_m: \mathbb{T} \to [1, \mathcal{M}]$, mapping Tasks to Machines,
- a Task duration function $f_d: \mathbb{T} \to \mathbb{N}$,
- a relation R_p over $\mathbb{T}$, which defines a strict total order for all the subsets $\{x \in \mathbb{T} \mid f_j(\mathrm{x}) = \mathrm{i}\}$ with $i \in [1, \mathcal{J}]$

1.2 Additional Notation

- $\mathfrak{J}_{\mathscr{k}} = \left|\{x \in \mathbb{T} \mid f_j(x) = \mathscr{k}\}\right|$ for $\mathscr{k} \in \mathbb{J}$, the number of Tasks mapped to Job $\mathscr{k}$
- $\mathfrak{M}_\ell = |\{x \in \mathbb{T} \mid f_m(x) = \ell\}|$ for $\ell \in \mathbb{M}$, the number of Tasks mapped to Machine ℓ,
- $T_{ij} \in \mathbb{T}$ denotes the j-th Task in the strict total order of Job i as given by R_p, where $i \in [1, \mathcal{J}]$ and $j \in [1, \mathfrak{J}_i]$

1.3 Optimization Goal

A schedule is a function $\mathcal{S}: \mathbb{T} \to \mathbb{N}$, which defines the completion time for each Task.

The makespan of a schedule is defined as: $\max\limits_{x \in \mathbb{T}}\big(\mathcal{S}(x)\big)$.

The fitness function for a given schedule returns the makespan for that schedule.

The optimization goal is to minimize the fitness function.

Optimization occurs with the following additional constraints:

1. $\forall a, b \in \mathbb{T}\Big(f_j(a) \neq f_j(b) \Longrightarrow \big((a, b) \notin R_p \wedge (b, a) \notin R_p\big)\Big)$
2. $\forall i \in [1, \mathcal{J}]\ \forall j \in [1, \mathfrak{J}_i]\left(\mathcal{S}(T_{ij}) \geq \begin{cases} \mathcal{S}(T_{i(j-1)}) + f_d(T_{ij}) & \text{if } j \geq 2 \\ f_d(T_{ij}) & \text{if } j = 1 \end{cases}\right)$
3. $\forall i, \mathscr{k} \in [1, \mathcal{J}]\ \forall j \in [1, \mathfrak{J}_i]\ \forall \ell \in [1, \mathfrak{J}_{\mathscr{k}}]\Big(\big(f_m(T_{ij}) = f_m(T_{\mathscr{k}\ell})\big) \Longrightarrow \Big(\big(\mathcal{S}(T_{ij}) \leq \mathcal{S}(T_{\mathscr{k}\ell}) - f_d(T_{\mathscr{k}\ell})\big) \vee \big(\mathcal{S}(T_{\mathscr{k}\ell}) \leq \mathcal{S}(T_{ij}) - f_d(T_{ij})\big)\Big)\Big)$

The constraints can be summarized as:

1. There are no precedence relations between Task of different Jobs.
2. The schedule fulfills the precedence relation of each Job and the Tasks are done in sequential order.
3. There is no processing overlap for Tasks on the same Machine.

The requirement for all Tasks to be scheduled, is captured implicitly by the definition of $\mathcal{S}$ ($\mathbb{T}$ is the domain of $\mathcal{S}$).