Problems for the course
"Implementation of algorithms in software"
March 2021.

A *partition* (or an assignment) of (the elements of) a set (of jobs, or items, or vertices) Y into sets $J_1, J_2, \ldots, J_m$ must satisfy $\bigcup_{1 \le i \le m} J_i = Y$, and for any $1 \le i_1 < i_2 \le m$, $J_{i_1} \cap J_{i_2} = \emptyset$ (every element of Y belongs to exactly one subset). For most scheduling problems (all problems below belong to this class), the value m is fixed in advance. For some problems it is not fixed (for example, bin packing). We will use a partition to describe a schedule (without specifying the exact times allocated to the jobs), where J_i is the set of jobs of machine i. A different way to define a schedule or assignment for the set of jobs J is a function $A : J \to \{1, 2, \ldots, m\}$. In this case we let $J_i = \{j \in J | A(j) = i\}$.

Proposed problems

1. **Fixed number scheduling on identical machines.**

   Input: A set of jobs consisting of $m \cdot n$ jobs $J = \{1, 2, \ldots, mn\}$, where job j has an integer processing time $p_j > 0$. An integer number of (identical) machines $m \ge 2$.

   Goal: Find a partition (assignment) of the jobs to the (m) machines, $I_1, I_2, \ldots, I_m$, such that for any odd value of i $(i = 1, 3, \ldots)$, it holds that $|I_i \cap J| = n$ (the number of jobs for these machines is the average number).

   Objective:: Maximize $\min_{1 \le i \le m} \sum_{j \in I_i} p_j$.

2. **Scheduling on restricted uniformly related machines.**

   Input: An integer number of machines $m \ge 2$. A set of n jobs $J = \{1, 2, \ldots, n\}$, where job j has an integer processing time $p_j > 0$. An integer $s \ge 1$, and machine speeds $s_i \in \{1, s\}$ for $i = 1, 2, \ldots, m$.

   Goal: Find a partition (assignment) of the jobs into non-empty subsets, $I_1, I_2, \ldots, I_m$.

   Objective:: Minimize $\max_{1 \le i \le m} \sum_{j \in I_i} \frac{p_j}{s_i}$.

3. **Scheduling with even machines and jobs.**

   Input: An integer number of (identical) machines $m \ge 2$. A set of n jobs $J = \{1, 2, \ldots, n\}$, where job j has an integer processing time $p_j > 0$.

   Goal: Find a partition (assignment) of the jobs into non-empty subsets, $I_1, I_2, \ldots, I_m$, where for every job j such that the job index j is divisible by 3, its machine index i is even. For example, job 1 can be assigned to any machine but job 3 cannot be assigned to machine 1 (and to machines $3, 5, \ldots$ if they exist).

Objective:: Minimize $\max_{1\leq i\leq m} \sum_{j\in I_i} p_j$.

4. **Scheduling on favorite machines to minimize makespan.**

   Maybe a bit harder than other problems.

   Input: A set of n jobs $J = \{1, 2, \ldots, n\}$, where job j has an integer size $p_j > 0$, an integer number of (identical) machines $m \geq 2$. Job j also has a machine index i_j $(1 \leq i \leq m)$ such that its processing time on machine i_j is p_j, and its processing time on any machine $i \neq i_j$ is $2 \cdot p_j$ (that is, larger by a factor of 2).

   Goal: Find an assignment σ of the jobs to the m machines, $J_1, J_2, \ldots, J_m$. Let $C_i(\sigma)$ be defined as follows: $C_i(\sigma) = \sum_{j\in J, \sigma(j)=i, i_j\neq i} 2 \cdot p_j + \sum_{j\in J, \sigma(j)=i, i_j=i} p_j$ (this is the completion time of machine i). That is, for each machine, its completion time is the sum of processing time of its jobs, where for j such that this machine is its favorite, the processing time is p_j and otherwise, $2 \cdot p_j$.

   Objective:: Minimize $\max_{1\leq i\leq m} C_i(\sigma)$. That is, the maximum completion time of any machine.

5. **Scheduling with types to minimize the makespan.**

   Input: An integer number of (identical) machines $m \geq 2$. A set of n jobs $J = \{1, 2, \ldots, n\}$, where every job j has an integer processing time $p_j > 0$ and a positive integer type $t_j \in \{1, 2, 3, \ldots, 7\}$.

   Goal: Find a partition of the jobs of to the machines, $J_1, J_2, \ldots, J_m$, such that every subset has jobs of at most three types (for any i there are three values $k_{i_1}, k_{i_2}, k_{i_3} \in \{1, 2, 3, \ldots, 7\}$, such that if $j \in J_i$, then $t_j \in \{k_{i_1}, k_{i_2}, k_{i_3}\}$).

   Objective:: Minimize $\max_{1\leq i\leq m} \sum_{j\in J_i} p_j$.

6. **Scheduling with special jobs on identical machines to minimize the makespan.**

   Input: A set of n jobs $J = \{1, 2, \ldots, n\}$, where job j has an integer processing time $p_j > 0$. An integer number of (identical) machines $m \geq 2$ (where $n \geq 2m$).

   Jobs $1, 2, \ldots, 2m$ are called special, all other jobs are called regular.

   Goal: Find a partition (assignment) of the jobs to the (m) machines, $I_1, I_2, \ldots, I_m$, such that for any i $(1 \leq i \leq m)$, $I_i \cap \{1, \ldots, 2m\} = 2$ (every machine has exactly two special jobs).

   Objective:: Minimize $\max_{1\leq i\leq m} \sum_{j\in I_i} p_j$.

7. **Scheduling with divisible cardinality sets on identical machines to minimize the makespan.**

   Input: A set of n jobs $J = \{1, 2, \ldots, n\}$, where job j has an integer processing time $p_j > 0$. An integer number of (identical) machines $m \geq 2$.

   Goal: Find a partition (assignment) of the jobs to the (m) machines, $I_1, I_2, \ldots, I_m$, such that for any i $(1 \leq i \leq m)$, $|I_i|$ is either in $\{1, 2, 3, 4, 5\}$ or it is divisible by 3 (for each machine, its number of jobs it is at most 5 or it is an integer multiple of 3).

   Objective:: Maximize $\min_{1 \leq i \leq m} \sum_{j \in I_i} p_j$.

8. **Scheduling with odd cardinality sets on identical machines to minimize the makespan.**

   Input: A set of n jobs $J = \{1, 2, \ldots, n\}$, where job j has an integer processing time $p_j > 0$. An integer number of (identical) machines $m \geq 2$.

   Goal: Find a partition (assignment) of the jobs to the (m) machines, $I_1, I_2, \ldots, I_m$, such that for any i $(1 \leq i \leq m)$, $|I_i|$ is odd (for each machine, the number of jobs received by it is not divisible by 2).

   Objective:: Maximize $\min_{1 \leq i \leq m} \sum_{j \in I_i} p_j$.

9. **Scheduling with mostly divisible cardinality sets on identical machines to minimize the makespan.**

   Input: A set of n jobs $J = \{1, 2, \ldots, n\}$, where job j has an integer processing time $p_j > 0$. An integer number of (identical) machines $m \geq 2$.

   Goal: Find a partition (assignment) of the jobs to the (m) machines, $I_1, I_2, \ldots, I_m$, such that for any i $(1 \leq i \leq m)$, $|I_i|$ is either in $\{1, 3\}$ or it is divisible by 2 (for each machine, the number of jobs received by it is at most 4 or it is an integer multiple of 2).

   Objective:: Minimize $\max_{1 \leq i \leq m} \sum_{j \in I_i} p_j$.

10. **Scheduling with three types on identical machines to minimize makespan.**

    Input: Three sets of jobs, each consisting of n jobs $J_1 = \{1, 2, \ldots, n\}$, $J_2 = \{n+1, n+2, \ldots, 2n\}$, $J_3 = \{2n+1, 2n+2, \ldots, 3n\}$,, where job j has an integer processing time $p_j > 0$. An integer number of (identical) machines $m \geq 2$.

    Let the type of j be denoted by $t_j \in \{1, 2, 3\}$, where $t_j = 1$ if $1 \leq j \leq n$, $t_j = 2$ if $n+1 \leq j \leq 2n$, and $t_j = 3$ if $2n+1 \leq j \leq 3n$.

Goal: Find a partition (assignment) of the jobs to the (m) machines, $I_1, I_2, \ldots, I_m$, such that for any pair i,ℓ ($1 \le i \le m$, $\ell = 1, 2, 3$), $I_i \cap J_\ell \ne \emptyset$ (every machine has at least one job of each type).

Objective:: Maximize $\min_{1\le i\le m} \sum_{j\in I_i} p_j$.

11. **Fair scheduling with two types on identical machines to minimize makespan.**

    Input: Two sets of jobs, consisting of n and $2n$ jobs, respectively, $J_1 = \{1, 2, \ldots, n\}$, $J_2 = \{n+1, n+2, \ldots, 3n\}$, where job j has an integer processing time $p_j > 0$. An integer number of (identical) machines $m \ge 2$.

    Let the type of j be denoted by $t_j \in \{1, 2\}$, where $t_j = 1$ if $1 \le j \le n$, and $t_j = 2$ if $n+1 \le j \le 3n$.

    Goal: Find a partition (assignment) of the jobs to the (m) machines, $I_1, I_2, \ldots, I_m$, such that for any i ($1 \le i \le m$), $|I_i \cap J_1| \ge 1$ and $|I_i \cap J_2| \ge 2$ (every machine has at least one job of the first type and at least two jobs of the second type).

    Objective:: Minimize $\max_{1\le i\le m} \sum_{j\in I_i} p_j$.

12. **Almost equitable scheduling.**

    Input: An integer number of (identical) machines $m \ge 2$. A set of $m \cdot n$ jobs $J = \{1, 2, \ldots, mn\}$, where job j has an integer processing time $p_j > 0$.

    Goal: Find a partition (assignment) of the jobs into non-empty subsets, $I_1, I_2, \ldots, I_m$, where $|I_i| \in \{n-1, n, n+1\}$ for all values of i.

    Objective:: Maximize $\min_{1\le i\le m} \sum_{j\in I_i} p_j$.

13. **Almost equitable pair scheduling.**

    Input: An even integer number of (identical) machines $m \ge 2$. A set of $m \cdot n$ jobs $J = \{1, 2, \ldots, mn\}$, where job j has an integer processing time $p_j > 0$.

    Goal: Find a partition (assignment) of the jobs into non-empty subsets, $I_1, I_2, \ldots, I_m$ (where $\bigcup_{1\le i\le m} I_i = J$, for any $1 \le i \le m/2$, $|I_{2i-1} \cup I_{2i}| \le 2n+1\}$).

    Objective:: Minimize $\max_{1\le i\le m} \sum_{j\in I_i} p_j$.

14. **Scheduling with cardinality constraints.**

    Input: An integer number of (identical) machines $m \ge 2$. A set of n jobs $J = \{1, 2, \ldots, n\}$, where job j has an integer processing time $p_j > 0$. An integer parameter k.

    Goal: Find a partition (assignment) of the jobs into non-empty subsets, $I_1, I_2, \ldots, I_m$, where for any $1 \le i \le m-1$, $|I_i| \le k$ (this holds for all machine except for possibly the last one).

    Objective:: Maximize $\min_{1\le i\le m} \sum_{j\in I_i} p_j$.