MISTA 2013 challenge: the Multi-Mode Resource-Constrained Multi-Project Scheduling Problem

1 Problem description

The subject of the MISTA 2013 challenge is the multi-mode resource-constrained multi-project scheduling problem (MRCMPSP). Multiple projects have to be scheduled simultaneously taking into account the availability of local and global resources. The MRCMPSP has a large practical relevance, including examples in the construction and production sector.

Projects and activities

A set $\mathcal{P}$ of n projects has to be scheduled, under the restriction of several time and resource constraints. The projects are identified by their index $i \in \mathcal{P}$, $i = \{0, \ldots, n-1\}$. Each project $i \in \mathcal{P}$ consists of a set of non-preemptive jobs or activities J_i. For each activity $j \in \{1, \ldots, |J_i|\}$ of project i, a start time $s_{ij} \geq 0$ has to be determined. The first and last activities of the projects ($j = 0$, resp. $j = |J_i| + 1$) are dummy activities, which have a zero duration and no resource requirements. Each project $i \in \mathcal{P}$ has a release date r_i, i.e. the earliest time at which the activities of project i can start ($s_{i0} \geq r_i$).

Local and global resources

With each project $i \in \mathcal{P}$ a set $L_i = \{1, \ldots, |L_i^\rho|, |L_i^\rho| + 1, \ldots, |L_i^\rho| + |L_i^\nu|\}$ of resources are associated, where the subset $L_i^\rho = \{1, \ldots, |L_i^\rho|\}$ denotes the local renewable resources and $L_i^\nu = \{|L_i^\rho| + 1, \ldots, |L_i^\rho| + |L_i^\nu|\}$ the non-renewable resources. Renewable resources have a fixed capacity per time unit, while the non-renewable resources have a fixed capacity for the whole project duration. Each resource $l \in L_i$ has a fixed capacity of c_{il}.

Finally there exists a set G^ρ of global renewable resources that are shared among the projects. Similar to the local resources, the availability of the global resources is limited by c_g^ρ, $g \in G^\rho$. There are no global non-renewable resources.

Execution modes

For each activity $j \in J_i$, $i \in \mathcal{P}$, one or more execution modes are available. The execution mode in which an activity is performed, determines both the time required to complete the activity, as well as the specific resources requirements.

Let $M_{ij} = \{1, \ldots, |M_{ij}|\}$ be the set of possible modes in which activity $j \in J_i$, $i \in \mathcal{P}$, can be performed, and d_{ijm} the processing time of activity $j \in J_i$ in mode $m \in M_{ij}$. Moreover, r^{ρ}_{ijml}, r^{ν}_{ijml} and r^{ρ}_{ijmg} respectively define the consumption of local renewable, non-renewable and global renewable resources when activity $j \in J_i$ is processed in mode $m \in M_{ij}$.
Note that a feasible schedule must always adhere to the following resource constraints:

$$\sum_{j \in J_i} \sum_{m \in M_{ij}} x_{ijt}\, y_{ijm}\, r^{\rho}_{ijml} \leq c_{il} \qquad \forall i \in \mathcal{P}, l \in L^{\rho}_i, t \in [0,T] \qquad (1)$$

$$\sum_{j \in J_i} \sum_{m \in M_{ij}} y_{ijm}\, r^{\nu}_{ijml} \leq c_{il} \qquad \forall i \in \mathcal{P}, l \in L^{\nu}_i \qquad (2)$$

$$\sum_{i \in \mathcal{P}} \sum_{j \in J_i} \sum_{m \in M_{ij}} x_{ijt} y_{ijm}\, r^{\rho}_{ijmg} \leq c_g \qquad \forall g \in G^{\rho}, t \in [0,T] \qquad (3)$$

$$\sum_{m \in M_{ij}} y_{ijm} = 1 \qquad \forall i \in \mathcal{P}, j \in J_i \qquad (4)$$

$$x_{ijt} \in \{0,1\} \qquad \forall i \in \mathcal{P}, j \in J_i, m \in M_{ij}, t \in [0,T] \qquad (5)$$

$$y_{ijm} \in \{0,1\} \qquad \forall i \in \mathcal{P}, j \in J_i, m \in M_{ij} \qquad (6)$$

where, binary variables x_{ijt} indicate whether a particular activity $j \in J_i$ is performed at time $t \in [0,T]$, and binary variables y_{ijm} denote whether activity $j \in J_i$ runs in mode $m \in M_{ij}$. T is an upper bound on the time horizon of the scheduling problem.

Precedence constraints
Certain activities may require the completion of other tasks prior to their own start. In such a case, if activity $j \in J_i$ must precede activity $j' \in J_i$, a precedence relation $j \prec j'$ is defined. Feasible schedules must obey all precedence relations, i.e.

$$\sum_{m \in M_{ij}} s_{ij} + y_{ijm} d_{ijm} \leq s_{ij'} \qquad if\ j \prec j' \qquad (7)$$

1.1 Objectives

The objective is to find a feasible schedule with respect to constraints [1-7] while minimizing the total project delay (TPD) (Section 1.1.1) and the total makespan (TMS) (Section 1.1.2). The two objectives are considered in lexicographical order, the TPD is the primary objective, while the TMS is used as a tie breaker.

1.1.1 Total Project Delay (TPD)

The project delay is defined as the difference between the Critical Path Duration (CPD), a theoretical lower bound on the earliest finish time of the project, and the actual project duration (makespan). More formally, the Project Delay (PD_i) for a project $i \in \mathcal{P}$ is defined as:

$$PD_i = MS_i - CPD_i$$

where MS_i, the makespan of project i, is:

$$MS_i = f_i - r_i \quad with \; f_i = s_{i(|J_i|+1)}.$$

and CPD_i, the Critical Path Duration[1] of project i, is:

$$CPD_i = EF_{i(|J_i|+1)}$$

where EF_{ij} is the earliest finish time of activity $j \in J_i$, which is recursively defined as:

$$\begin{aligned} EF_{i0} &= 0 && \forall i \in \mathcal{P} \\ EF_{ij} &= \max_{j' \in pred(j)} EF_{ij'} + \min_{m} d_{ijm} && \forall i \in \mathcal{P}, j > 0 \end{aligned}$$

In the above definition, $pred(j)$ denotes the predecessor set of activity j, i.e. $pred(j) = \{j' | j' \prec j\}$. Finally, the total project delay is expressed as:

$$TPD = \sum_{i=1}^{n} PD_i$$

1.1.2 Total Makespan (TMS)

The Total Makespan TMS is the duration of the whole multi-project schedule.

$$TMS = \max_{i \in \mathcal{P}}(f_i) - \min_{i \in \mathcal{P}}(r_i)$$

1.2 Instance input file format

All problem inputs will be integer valued, and will use the following file format. Each problem instance will be defined by $n+1$ files, one global file and n project files.

Global file

The global file contains the number of projects, the release dates, the path to the project files, and the global resource capacities. Values are space separated and should respect the following order:

[1]Values for the critical path duration of the projects will be given in the input file.

```
Number of projects
Release date project 0
Critical path duration project 0
Path to project file of project 0
Release date project 1
Critical path duration project 1
Path to project file of project 1
...
Release date project n-1
Critical path duration project n-1
Path to project file of project n-1
Number of resource types
Global resource capacities (-1 is not global)
```

Project files
Each project is defined in a separate file. The relative path to this file is given in the global file (e.g. *j20.mm/j2010_1.mm*). A project file contains the number of activities (including the dummy activities), the precedence relations between the activities, the execution modes, and the local resource capacities. The files use the PSPLIB[2] MRCPSP file format.

Important note: the global resources always overwrite the local resources and their capacities. For example, consider the following global resource capacities (16 -1 -1 -1) and local resource capacities (RR1=14 RR2=18 NR1=60 NR2=68), then there exists one global renewable resource with capacity 16, one local renewable resource with capacity 18 and two non-renewable resources with capacities 60 and 68. The local renewable resource with capacity 14 can be ignored.

1.3 Solution output format

A solution for the MRCMPSP is defined as follows.
For each activity j of project i,

- the selected mode m_{ij},
- and the start time s_{ij},

must be given.
An example is given in Table 1.

1.4 Solution validator

A validator, to check if a produced solution is valid or not, and to compute the total objective cost, will be available for download at the challenge website.

[2]PSPLIB benchmark website: `http://129.187.106.231/psplib/`

Project i	Activity j	Mode m_{ij}	Start time s_{ij}
0	0	2	0
0	1	1	5
...	...	...	...
1	0	0	4
1	1	0	8
...	...	...	...

Table 1: Solution output format example

Java is required to run the validator. The usage is:

```
java -jar MRCMPSP-validator.jar global_problem_file schedule_file
```

This solution validator will be used during the challenge to evaluate produced solutions. Note that the aim is to check the solution, not the instance; so the solution validator assumes the instance is valid.

1.5 Example

An example MRCMPSP instance and a corresponding feasible solution can be found on the challenge website. The instance has 2 projects and 10 jobs per project.