Common data representation

The plain text file (.txt) is used for the storage of instances and all the data for one instance are included in one file, consisting only of integer numbers. The file consists of four main sections, separated by an empty line. No other empty lines are allowed (to ease the file reading). The four sections are separated as follows:

1) Definition of resources
2) Definition of operations, called activities in RCPSP
3) Definition of products, using the NTNA structure
4) Definition of orders

The setup times are not considered, the time-lags (both positive and negative) are included, but they will be generated only for the RCPSP instances. Part types, bills of material and toolgroups are not explicitly given in the data file, but can be used for inner data representation during the generation and solving the IPPS problems. Therefore, the IPPS instances will have the specific structure while the RCPSP is considered in the general form. The data structure is the same for both types of problems.

1. Resources

The section dedicated to the resources consists of two lines. On the first line, the number of resources is given and on the second line, the capacity is defined for each resource:

```
numberOfResources
capacity1 capactiy2 ...
```

For the IPPS problem, all capacities will be equal to one, i.e. only unary resources are considered.

2. Operations ≡ Activities

The term *operation* for the IPPS will be equal to the term *activity* for the RCPSP. The section for definition of operations is introduced by the line containing only the total number of operations, or more precisely operation types since each operation type can be a part of more products, or even included more times in the same product definition. Then each line is dedicated to the definition of one operation type:

```
numberOfOperationTypes
operationId1 numberOfModes definitionOfModes releaseTime duedate deadline priority
operationId2 numberOfModes definitionOfModes releaseTime duedate deadline priority
(...)
```

where each mode in `definitionOfModes` contains resource Id, resource demand and processing time, i.e.:

```
resourceId1 demand1 processingTime1 resourceId2 demand2 processingTime2 ...
```

Each operation thus has Id, release time, duedate, deadline, priority and it has to be assigned to one of the specified resources. For each resource, the demand and the processing time can be different. Release time, duedate and deadline will be not considered for the IPPS problems. More precisely, those data will be included in the file, but ignored during the reading if the IPPS problem is considered.

3. Products

Each product is represented by one NTNA instance and all the NTNA instances are then considered to be in parallel. The definition of products starts with the line where the number of products is specified. The definition of each product starts with the line containing its Id and the number of nodes of the NTNA instance. Then each line contains node Id, operation Id, part type Id (specific for IPPS instances), definition of in/out label, successors and additional time-lags for one node of the product NTNA:

```
numberOfProducts
productId1 numberOfNodes
nodeId1 operationId partTypeId inLabel outLabel successorsCount successorNodeId1 timeLag1
(continued) numberOfAdditionalTimeLags nodeId1 timeLag1...
nodeId2 operationId partTypeId...
(...)
productId2 numberOfNodes
(...)
```

The NTNA instance can be constructed based on the given in/out labels and list of successors for each node. The additional time-lags are not included in the NTNA structure. The NTNA instance has to be in the nested form, i.e. both parallel and alternative branchings can be arbitrary nested one in each other, but no other interaction among branchings is allowed. In other words, each edge that starts in the node of some branching has to end in the node of the same branching.
Each node corresponds to one operation type (defined in the previous section) and it can have an arbitrary number of successors, referenced by Ids of nodes (not operation types!). Each operation type can be included more times within one product. For the IPPS problem, each product can be internally defined using BOMs and other specific structures. The only additional attribute for such a purpose is a part type Id, which is ignored for the RCPSP instances.
The number of additional time-lags for the IPPS will be always zero and the time-lags related to the precedence relations (defined by successors for each node) will be also equal to zero and the time distances for precedence related operations will be derived in run the time based on the assignment to the resources (and thus with fixed processing times).

4. Orders

The section with orders definition again starts with the line containing the number of orders and then each line represents one order:

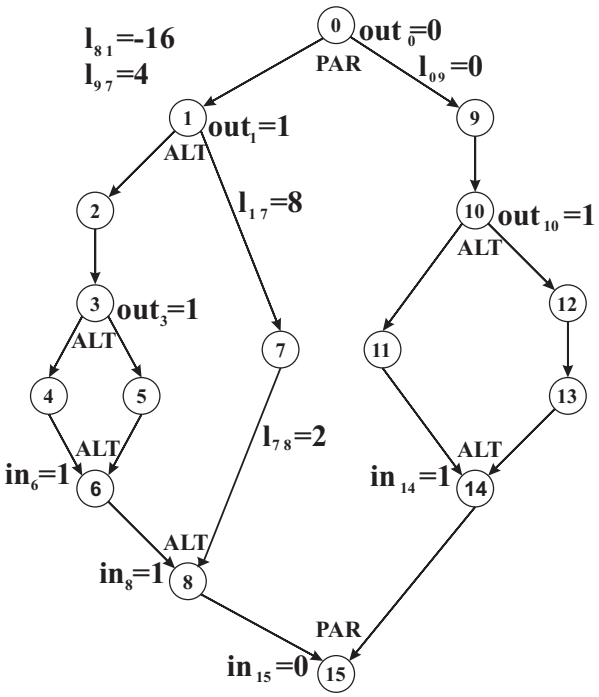
```
numberOfOrders
orderId1 productId releaseTime duedate priority
orderId2 productId releaseTime duedate priority
(...)
```

Each order has its Id, it has assigned one product (by Id), release time, duedate and priority. For the RCPSP problem, there will be always only one order with zero release time and priority while the duedate will be sufficiently high number.

5. Examples

RCPSP

The representation of the RCPSP instance depicted in Figure 1a is shown in Figure 1b.



(a) Example of the RCPSP instance

```
3
4 2 2

16
0 1 0 1 2 0 10 25 1
1 1 0 4 5 1 8 22 5
2 1 1 2 3 0 10 24 4
3 1 2 1 6 2 21 31 1
4 1 1 2 1 6 14 38 1
5 1 1 2 7 6 14 38 5
6 1 0 0 0 0 0 0 0
7 1 2 1 4 6 18 29 2
8 1 0 3 5 0 25 60 1
9 1 1 2 3 0 6 23 3
10 1 0 2 4 4 15 40 1
11 1 2 1 5 12 31 41 4
12 1 1 1 3 15 28 50 3
13 1 1 1 4 18 33 50 4
14 1 2 1 6 0 30 45 1
15 1 2 1 3 20 50 60 1

1
0 16
0 0 0 0 0 2 1 3 9 0 0
1 1 0 0 1 2 2 5 7 8 0
2 2 0 0 0 1 3 2 0
3 3 0 0 1 2 4 3 5 5 0
4 4 0 0 0 1 6 3 0
5 5 0 0 0 1 6 7 0
6 6 0 1 0 1 8 0 0
7 7 0 0 0 1 8 4 0
8 8 0 1 0 1 15 7 1 1 -16
9 9 0 0 0 1 10 0 1 7 4
10 10 0 0 1 2 11 0 12 4 0
11 11 0 0 0 1 14 6 0
12 12 0 0 0 1 13 3 0
13 13 0 0 0 1 14 4 0
14 14 0 1 0 1 15 0 0
15 15 0 0 0 0 0

1
0 0 0 1000 0
```

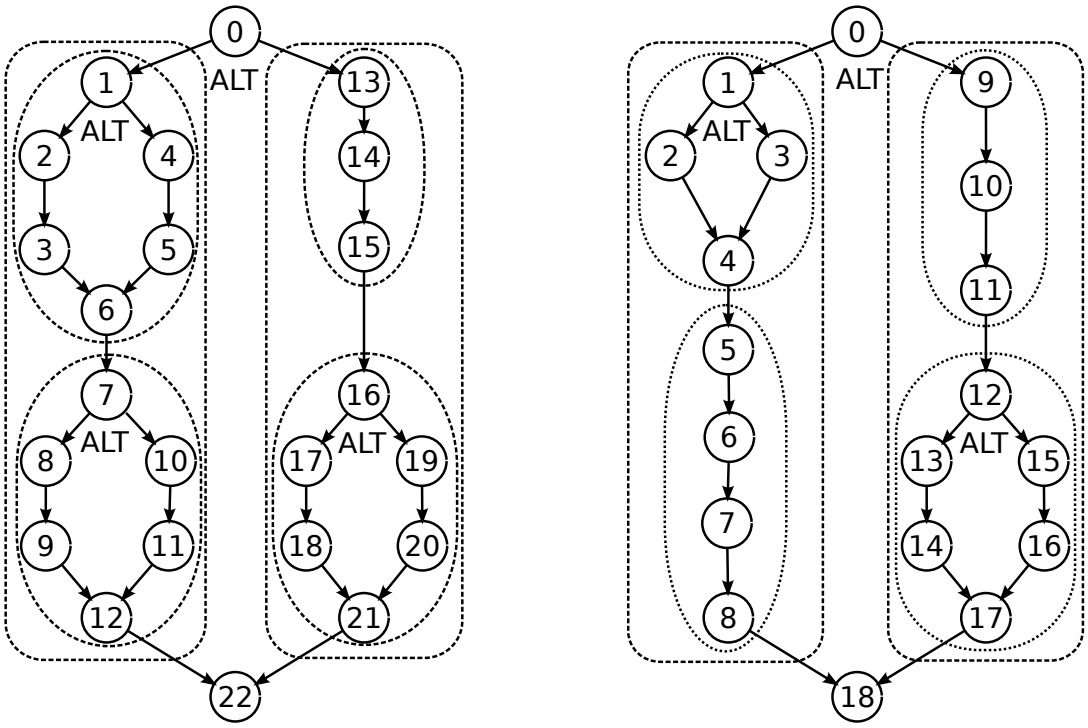
(b) Example of the data file content

Figure 1: Example of the instance and the data representation

IPPS

The representation of products for an example IPPS instance is given in Figure 2. The corresponding data file is shown in Figure 3. There are six resources(machines) one of which is for the dummy operations. Such operations have always zero processing time and zero resource demand.

There are nine operation types where operation type (8) is used for dummy operations. The release times and priorities of all operations are set to zero and the duedates and deadlines are set to `MaxNumber` defined in the settings file. Consequently, all those parameters do not influence the IPPS instance at all.



(a) Product 1 (b) Product 2

Figure 2: NTNA representations of two defined products

There are two products, each containing two different bills of material. Therefore, two separate NTNA instances are defined, see Figure 2a and Figure 2b. The first level of alternatives (starting at node 0) represents the alternative between two BOMs. The second level (e.g. node 1 for product 1 and node 12 for product 2) is for alternative routes of the parts.

At the end, all three orders are specified. The full data file content (with some commentaries added) is the following:

```
6 %Resources
1 1 1 1 1 1

9 %Operation types
0 1 0 1 1 0 1000 1000 0
1 2 1 1 6 2 1 6 0 1000 1000 0
2 2 1 1 4 2 1 5 1000 1000 0
3 2 3 1 7 4 1 8 1000 1000 0
4 2 1 1 5 2 1 7 1000 1000 0
5 2 1 1 4 2 1 6 1000 1000 0
6 2 1 1 5 2 1 7 1000 1000 0
7 2 1 1 7 2 1 4 1000 1000 0
8 1 5 0 0 0 1000 1000 0

2 %Products
1 21
0 8 0 0 1 2 1 0 13 0 0
1 8 1 0 1 2 2 0 4 0 0
2 4 1 0 0 1 3 0 0
3 3 1 0 0 1 6 0 0
4 3 1 0 0 1 5 0 0
5 4 1 0 0 1 6 0 0
6 8 1 1 0 1 7 0 0
7 8 2 1 2 8 0 10 0 0
8 4 2 0 0 1 9 0 0
9 3 2 0 0 1 12 0 0
10 3 2 0 0 1 11 0 0
11 4 2 0 0 1 12 0 0
12 8 2 1 0 1 22 0 0
13 8 3 0 0 1 14 0 0
14 2 3 0 0 1 15 0 0
15 8 3 0 1 1 16 0 0
16 8 1 0 0 1 17 0 0
```

(continued)

```
17 4 1 0 0 1 18 0 0
18 3 1 0 0 1 21 0 0
19 3 1 0 0 1 20 0 0
20 4 1 0 0 1 21 0 0
21 8 0 0 1 0 0 0 0 0

2 18
0 8 0 0 1 2 1 0 9 0 0
1 8 4 0 1 2 2 0 3 0 0
2 2 4 0 0 1 4 0 0
3 4 4 0 0 1 4 0 0
4 8 4 1 0 1 5 0 0
5 8 5 0 0 1 6 0 0
6 2 5 0 0 1 7 0 0
7 2 5 0 0 1 8 0 0
8 8 5 0 0 1 18 0 0
9 8 3 0 0 1 10 0 0
10 2 3 0 0 1 11 0 0
11 8 3 0 0 1 12 0 0
12 8 1 0 1 2 13 0 15 0 0
13 4 1 0 0 1 14 0 0
14 3 1 0 0 1 17 0 0
15 3 1 0 0 1 16 0 0
16 4 1 0 0 1 17 0 0
17 8 1 1 0 1 18 0 0
18 8 0 1 0 0 0 0 0 0

3 %Orders
1 1 0 32 4
2 2 0 13 5
3 2 0 13 3
```

Figure 3: Data file for the IPPS instance