

iMOPSE solution validator

Case study

Marek E. Skowroński, Paweł B. Myszkowski
m.e.skowronski@pwr.edu.pl ; pawel.myszkowski@pwr.edu.pl
Wrocław University of Technology

November 24, 2014

1 Introduction

This document presents the case study of using the iMOPSE validator tool.

2 Sample project definition and schedule

The case study would be presented on the basis of the validation of the file containing 10 tasks, 3 resources, 4 precedence relations and 3 skill types. The definition file structure is as follows:

```
File name: 10_3_5_3.def
Creation date: Thu Oct 30 17:39:39 CET 2014
Website: http://imopse.ii.pwr.edu.pl/
Reference: Myszkowski P. B., Skowronski M. E., Olech L., Oslizlo K.,
Hybrid Ant Colony Optimization in solving Multi-Skill Resource-Constrained Project Scheduling Problem,
Soft Computing, DOI: DOI 10.1007/s00500-014-1455-x
```

```
=====
General characteristics:
```

```
Tasks: 10
```

```
Resources: 3
```

```
Precedence relations: 4
```

```
Number of skill types: 3
```

```
=====
ResourceID  Salary  Skills
1      56.0   Q1: 0   Q2: 1
2      53.6   Q2: 2   Q0: 1
3      28.9   Q0: 1   Q1: 0
```

```
=====
TaskID  Duration  Skill  Predecessor IDs
1      37  Q2: 1
2      36  Q2: 2
3      21  Q0: 1
4      23  Q1: 0
5      36  Q0: 1
6      13  Q2: 1
7      13  Q1: 0   4 5
8      37  Q0: 1
9      36  Q2: 1   7
10     19  Q1: 0   3
```

3 Validation

Following solution (Fig. 3) gives positive validation results:

```
Hour Resource assignments (resourceID-taskID)
1 1-1 2-2 3-3
23 3-4
38 2-5
39 1-6
75 3-7 2-8
89 1-9 3-10
```

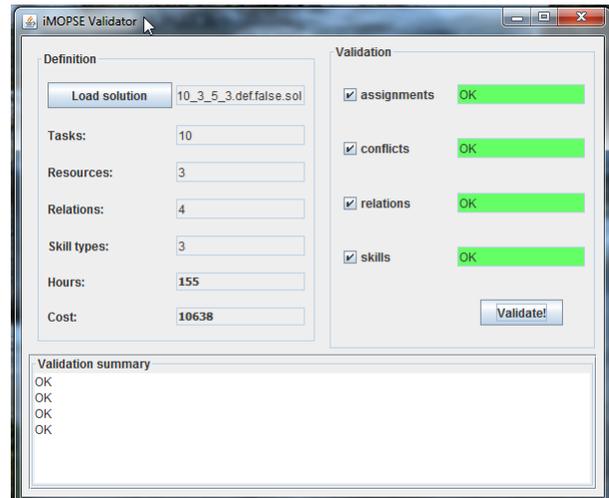


Fig. 3 Succeeded validation.

3.1 Assignments validation

This type of validation is used to check whether all tasks have some resource assigned. It is not about checking the resource capabilities of performing given task, only checking if a task has any resource assigned.

If some task-to-resource assignment would be removed from the solution, the validator returns fail of the assignments validation. In the example below (Fig. 3.1) the assignment of the task 8 has been removed. It caused not only the assignments validation failure but also the skills validation and the precedence relations validation failure, because removing assignments made start time and the resource assigned properties reset to the default value (-1).

```
Hour Resource assignments (resourceID-taskID)
1 1-1 2-2 3-3
23 3-4
38 2-5
39 1-6
75 3-7
89 1-9 3-10
```

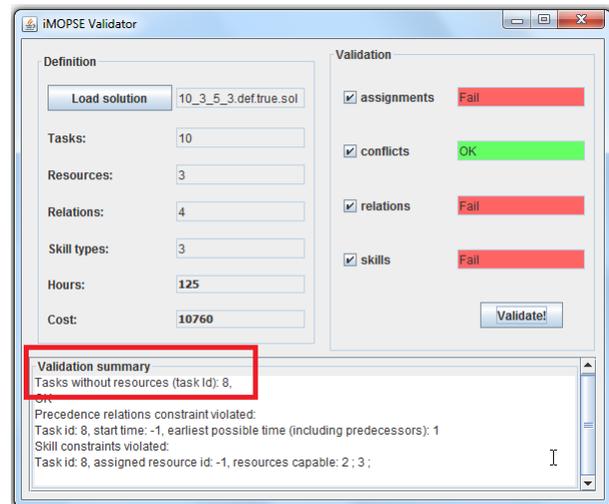


Fig. 3.1.1 Wrong assignments validation failure.

3.2 Conflicts validation

To check whether any resource has to perform more than one task in given time unit, the conflict validation is launched. Following example (Fig. 3.2.1) presents case when the resource 1 has to take care of the tasks 1 and 2 in

the same time unit, as those tasks have been scheduled to start by the same resource in the same timestamp.

```

Hour Resource assignments (resourceID-taskID)
1 1-1 1-2 3-3
23 3-4
38 2-5
39 1-6
75 3-7 2-8
89 1-9 3-10
    
```

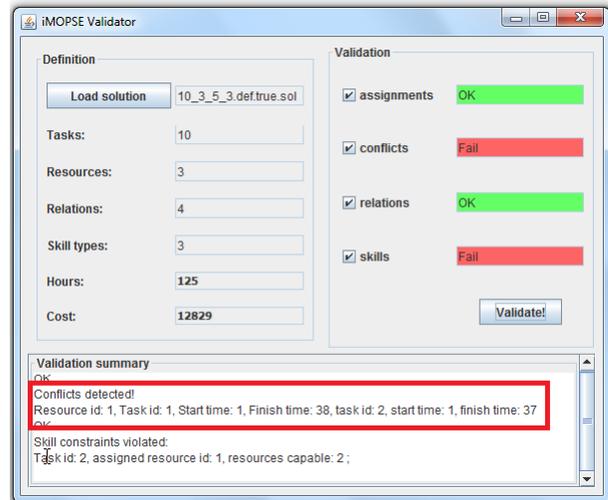


Fig. 3.2.1 Conflicts validation failure (many tasks assigned to the same resource and set the same start time).

Tasks do not have to start in the same time unit. A conflict also appears when a resource has more than one resource assigned in given time unit but their start times are different. In the example (Fig. 3.2.2): the task 1 starts in the hour 1, but the task 2 starts in the hour 5. The task 1 lasts 37 hours, therefore if the task 2 is set to start earlier than the hour 38 (in case when assigned to the same resource as the task 1), then the conflict appears.

```

Hour Resource assignments (resourceID-taskID)
1 1-1 3-3
5 1-2
23 3-4
38 2-5
39 1-6
75 3-7 2-8
89 1-9 3-10
    
```

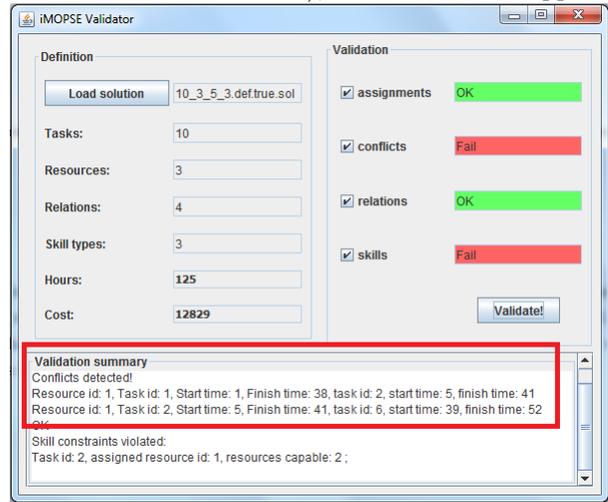


Fig. 3.2.2 Conflicts validation failure (many tasks assigned to the same resource and have overlapping start time).

3.3 Skills validation

Skills validation is used to check whether any task has capable resource assigned. The capabilities are validated by checking the skills owned by a resource and a skill required by given task. It is not enough for a resource to have required skill type, but it also needs to have this skill type on the familiarity level no lower than required.

Following example presented (Fig. 3.3.1) the skills validation failure when a task has assigned a resource that does not have required skill type. The task 4 has been assigned to the resource 2. However, the task 4 requires the

skill Q1, while the resource 2 disposes only the skill Q0 and Q2.

Hour Resource assignments (resourceID-taskID)

```

1 1-1 2-2 3-3
23 2-4
38 2-5
39 1-6
75 3-7 2-8
89 1-9 3-10

```

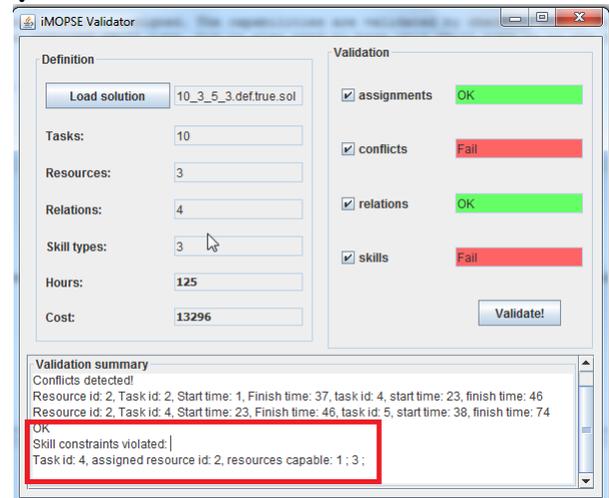


Fig. 3.3.1 Skills validation failure (task assigned to a resource not containing required skill type).

Another example presented (Fig. 3.3.2) skills validation fail when a task has assigned a resource that has required skill type but on the lower familiarity level than required. The task 2 requires the skill Q2 at the familiarity level equal to 2, while the assigned resource (1) has the required skill type Q2 on the lower familiarity level (1). Therefore, it cannot be assigned to the task because it could not cope performing it.

Hour Resource assignments (resourceID-taskID)

```

1 1-1 1-2 3-3
23 3-4
38 2-5
39 1-6
75 3-7 2-8
89 1-9 3-10

```

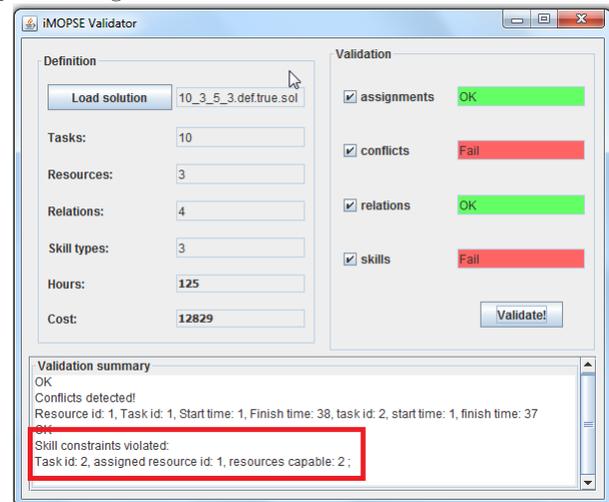


Fig. 3.3.2 Skills validation failure (a task assigned to a resource containing required skill type but on the lower than required familiarity level).

3.4 Precedence relations validation

To validate if some task starts too early than it is possible (when it has some predecessors) the precedence relations validation is used. If any task has some predecessors, it cannot start before all of its predecessors would be finished.

Following example (Fig. 3.4.1) presents precedence relations validation fail when some task has start time set earlier than its predecessors would be finished. The task 9 should not start before the task 7 would be finished, while the task 7 is set to start in hour 75. he task 7 last 13 hours, therefore the task 9 can start in hour 89 (75 + 13 = 88) or later. Setting start time of the task 9 to hour 85 means that precedence relation between the task 7

and 9 would be violated.

```

Hour Resource assignments (resourceID-taskID)
1 1-1 2-2 3-3
23 3-4
38 2-5
39 1-6
75 3-7 2-8
85 1-9 3-10
    
```

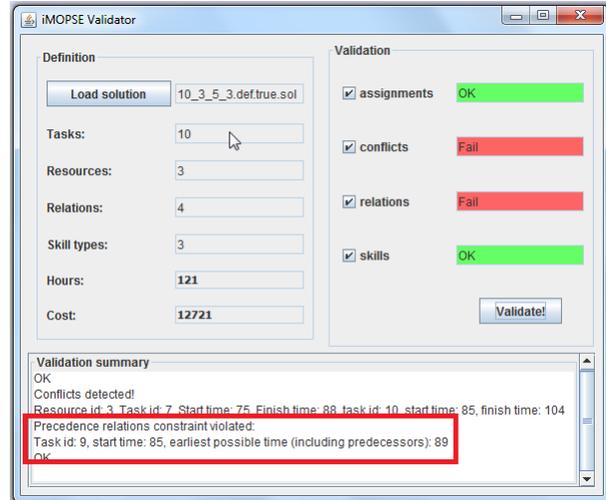


Fig. 3.4.1 Precedence relations validation failure (start hour set to the hour earlier than the predecessor’s finish hour).

Another example (Fig 3.4.2) presents a situation when a task has many predecessors and only some of precedence relations related with this task are satisfied. The task 7 has two predecessors: 4 and 5. The task 4 lasts 23 hours, while the task 5 needs 36 hours. If the task 4 is set to start in 23 and it would be only one predecessor of the task 7, then the task 7 can start in the hour 47. However, the task 7 has two predecessors: the task 4 and the task 5. Therefore, it is needed to consider all precedence relations with the task 7. Hence, the task 7 cannot start before 75, because the task 5 starts in the hour 38 and lasts 36 hours (finishes in 74). Setting the start time of the task 7 earlier than 75 (i.e. 72) would cause violating of the precedence relations between the task 7 and its predecessor – the task 5.

```

Hour Resource assignments (resourceID-taskID)
1 1-1 3-3
5 1-2
23 3-4
38 2-5
39 1-6
72 3-7
75 2-8
89 1-9 3-10
    
```

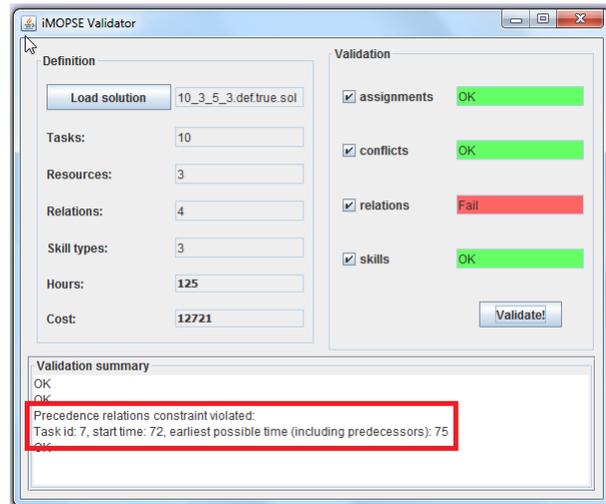


Fig. 3.4.2 Precedence relations validation failure (start hour set to the hour earlier than one of the predecessor’s finish hour).