



HELMUT SCHMIDT
UNIVERSITÄT

Universität der Bundeswehr Hamburg

**HELMUT-SCHMIDT-UNIVERSITÄT
UNIVERSITÄT DER BUNDESWEHR HAMBURG
LEHRSTUHL FÜR BETRIEBSWIRTSCHAFTSLEHRE,
INSBES. LOGISTIK-MANAGEMENT
Prof. Dr. M. J. Geiger**

**Arbeitspapier / Research Report
RR-15-03-01 · March 2015 · ISSN 2192-0826**

MMLIBchecker

A checker software for multi-mode resource-constrained
project scheduling problem (MRCPSP) solution files

Martin Josef Geiger

Contents

1	Introduction	1
2	Solution file format	1
3	Program settings and solution checking	2
3.1	Retrieving the instance data	2
3.2	Drag and drop operations	2
3.3	Filename conventions	2
4	Output of the checker	3
	References	5

Abstract

Debugging can be a tedious task. We know this from experience. With the program described in this paper, we try to help fellow researchers when implementing novel solution techniques for the multi-mode resource-constrained project scheduling problem (MRCPSP). The program supports in particular the MMLIB-instances of VAN PETEGHEM and VANHOUCKE (MMLIB50, MMLIB100, and MMLIB+).

1 Introduction

MMLIBchecker is a small program that helps with the verification of solutions files for the MMLIB50, MMLIB100, and MMLIB+ instances of VAN PETEGHEM and VANHOUCKE [3]. The instances have been created on the basis of a thorough investigation of previously existing benchmark datasets. We believe that future research will be applied to them, especially as they contain several features that have been missing in the older PSPLIB [2] and Boctor [1] datasets: more than three execution modes (all efficient and therefore non-removable), more renewable and more non-renewable resources, to name a few.

Note that the MMLIB-instances may currently be retrieved from
<http://www.projectmanagement.UGent.be>

2 Solution file format

Solution files must be plain text files. Each activity of a project must be described in a separate line. Three values must be given for each activity: (1) the ID (1 being the smallest index), (2) the selected mode (1 being the smallest index), (3) the starting time (an integer value). The values are separated with a space in between.

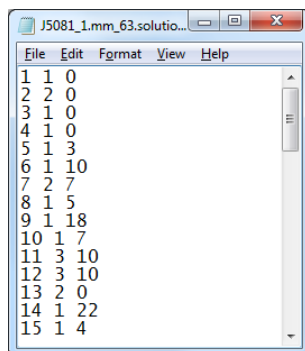


Figure 1: Example of a solution file

Figure 1 depicts the first fifteen lines of a solution file. The first ‘dummy’ activity of the project has index 1, which is fully in line with the MMLIB-datasets.

3 Program settings and solution checking

3.1 Retrieving the instance data

Obviously, the MMLIBchecker-program must be able to load the data of the MMLIB-problem instances from somewhere. We have chosen to integrate this by adding a text-box to the main window, described as ‘Model folder:’. Figure 2 shows this functionality in the top-left corner of the main application window (the default path is C:\MMLIB\).

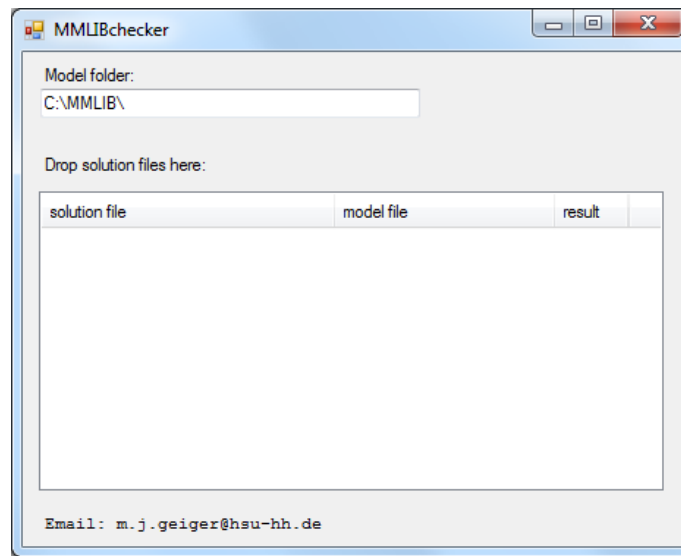


Figure 2: MMLIBchecker

It is recommended to extract all 4320 MMLIB-instances in a single folder.

3.2 Drag and drop operations

Solution files can be dropped into the white ‘listview’ of the main window. The program will immediately start checking the received solution files. Multiple solution files can be dropped at once.

3.3 Filename conventions

It is required that the solution file contains the filename of the problem file, starting from the leftmost character. For example, the checker will extract Jall100_1.mm from a file named Jall100_1.mm_my_solution.txt.

4 Output of the checker

If the checker finds the solution file to be feasible, the line in the output box is highlighted with a light green color. Figure 3 gives an example of the main window after a feasible solution has been dropped and checked. Note that the actual objective function value (makespan) of this solution is not displayed/highlighted. Knowing that all instances possess a last ‘dummy’-operation, the makespan of the schedule is the starting time of this last activity and therefore given in plain text in the solution file.

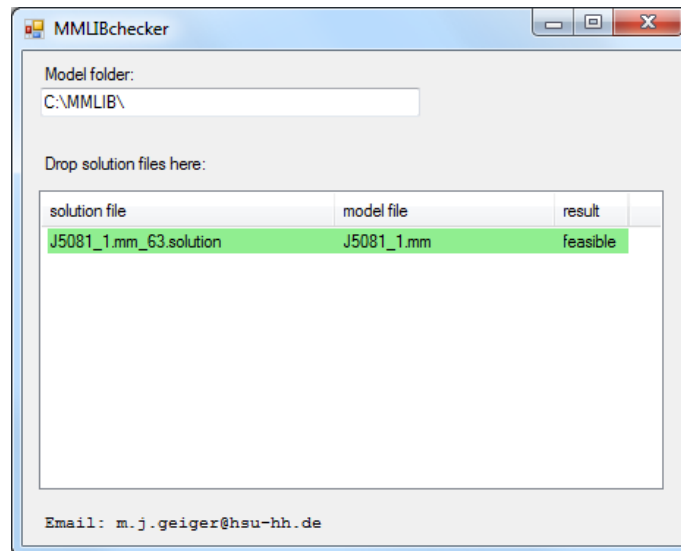


Figure 3: MMLIBchecker has found a feasible solution

In case of an infeasible solution, a detailed description of the identified problems is presented to the user. This protocol can later be re-displayed by clicking on the corresponding item (filename).

Figure 4 depicts the output for an infeasible solution file.

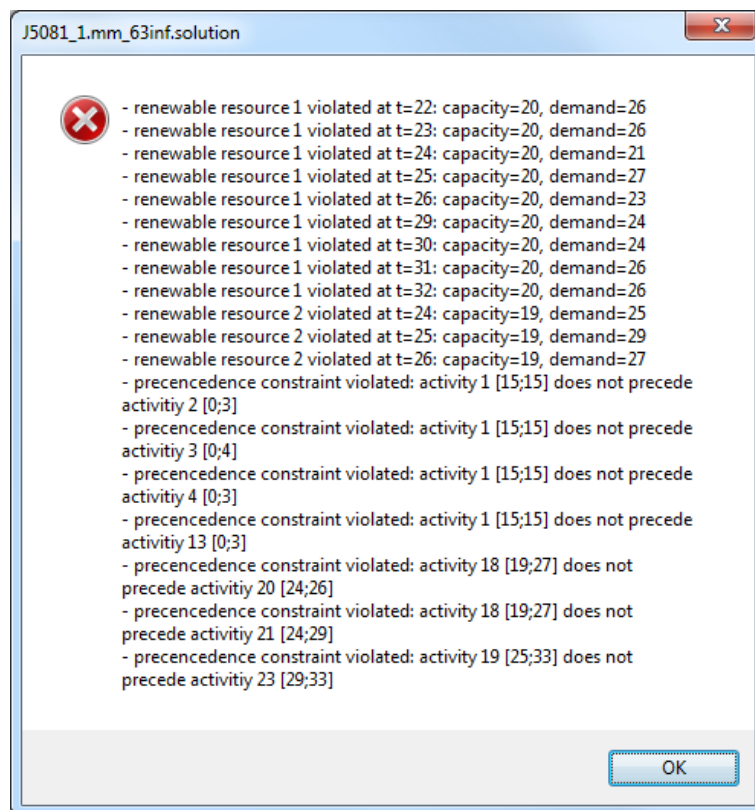


Figure 4: MMLIBchecker has detected an infeasible solution

References

- [1] F. Boctor. Heuristics for scheduling projects with resource restrictions and several resource-duration modes. *International Journal of Production Research*, 31:2547–2558, 1993.
- [2] Rainer Kolisch, A. Sprecher, and Andreas Drexl. Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, 41:1693–1703, 1995.
- [3] Vincent Van Peteghem and Mario Vanhoucke. An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. *European Journal of Operational Research*, 235:62–72, 2014.