For the baseline, we try to generate a uniform sample from the entire state space.

There are multiple variables.

1. We can vary the how courses are divided over the lessons.
2. We can vary the distribution of the lessons over the “roomslots”.
3. We can vary the distribution of the students over the lessons

ad1) If all the lectures, tutorials and labs are scheduled once, there are 72 lessons. However, each tutorial and lab has a maximum number of students per lesson.

Also, we can calculate the number of students per course.

To be able to fit students in the maximum number of students per lesson, tutorials and labs have to be split into multiple lessons. There is a minimum of 129 lessons if we state that all students should be scheduled in the lessons for their courses, taking into account the maximum of students per lesson.

The maximum number of lessons is equal to the maximum number of “roomslots”,which is 145.

For the baseline, we have done the following:

* we have used the number of 129 lessons. This simplifies the solution. Later on, we can optimize the distribution of courses over lessons in iterative steps, or generate random solutions including 129-145 lessons.
* we have randomized the distribution of the lessons over the schedule.
* we have used an even distribution of the number of students over the lessons. This simplifies the solution. Later on, we can still move students between different classes in iterative steps, or generate random solutions with varying number of student numbers per lesson.
* we have randomized the distribution of the students over the lessons.

We have run the random algorithm 50.000 times and saved only the valid schedules. This is a box plot of the valid results.

The minimum number of points is 578, the average is 848. Because 50.0000 is a very small portion of the total number of possible results, this is only an indication of what can be a good solution. There is not enough time to de a representative sample. Using some simple rules, we have already found solutions with less than 200 malus points. The chance that we will find a “good” solution using random schedules is limited. We can however still use the random results as a basis for iterative algorithms.

