Read-me For The Creation of Test Data GUI

Isak Bohman
MAI, Linköping University

January 18, 2016

1 Introduction

The following document outlines the means by which one may utilize a specialized interface for the creation of test data sets, to be used when trying out new scheduling algorithms.

2 The Interface

Figure 1: Window 4. This is the interface where all data parameters for test data are specified, and where test data are created and saved.

In order to run the interface, one must first open the file GUI.m in MATLAB. Upon launching the interface, via MATLAB's run function, a number of windows are immediately displayed. All except one of these windows is empty upon launch, as they are meant to display test data when they have been created. The window that is in focus will be our cynosure at this moment.
Next, we will turn our attention to the selector "Mode Selector". This box determines which parameters that are possible to change when specifying which sort of data to create. As an interlude, we will explain the options which are common to all modes.

- The box "#Data sets" defines the number of copies of a given data set which are to be created. The data sets are copies in the sense that they are created using the same basic settings, yet each one will

be unique due to stochastics in the creation of the data. For the project at hand, ten copies were made for every setting, in order to study the stability of the algorithms used.

- The button "Create test data" is the button of interest when one has decided upon which parameter values to use for a particular data set. Upon pressing this button, an interval of time will pass, which will be small for a low number of tasks, data sets and dependencies, or one may need to wait several minutes if a more demanding data type is selected. Then, if the algorithm finishes in due time, barring for memory overflow, the results will be displayed graphically in windows 1-3. If more than one data set is created, the last data set will be the one being exhibited.

- The button "Save test data" is used for saving test data once at least one data set has been created. The test data files are saved at a prespecified location relative to the program folder. Each test data set is saved in a separate folder, which will contain four files. One file, DependencyMatrix.dat, contains information pertaining to data parameters for the dependency matrix as well as the dependency attributes of the data set. Another file, TimelinesSolution.dat, contains all information relating to the placement of tasks, i.e. a timeline solution as well as timeline attributes. There is also a file specifying the data parameters used for creating the data set. This file contains numerical values quantifying standard deviations, expected values, number of tasks, etc. for the data set. Since the actual number of tasks in a data set is stochastic, the number of tasks value in this file will be merely an approximation of actuality. One file is AMPL adapted, with the idiosyncrasies this entails (such as the fact that all test data must be saved in a single file, using a certain syntax). This file will not be of interest unless one uses AMPL for solving the problem that the data set represents.

- The button "Launch testing GUI" launches a rudimentary version of a preliminary GUI that was to be used for running tests of the algorithms. However, since a testing GUI was done by an other project member whom I have had little collaboration with, this didn't materialize in the anticipated fashion.

- The button "Make and save multiple test data" is a button specifically crafted for the need of creating a series of test data sets in an automated way, without the painstaking input of a human. As of current, pushing this button will create a series of 520 test data sets, whose degree of difficulty varies in integer steps of 5, from 0 to 60. The number of tasks, and thus dependencies, varies nonlinearly as a polynomial of the fourth degree, as a function of the level of difficulty. These parameters are specified in the code, and might be of interest to implement directly in the GUI.

At last, we turn to the different modes of data creation which are available to us.

2.1 The Manual Mode

This is the default mode, and settings most specific to this mode are marked in green. The other modes described below depend upon settings which may be tampered with in this mode. In addition to the green areas, several other parameters may be adjusted: The approximate number of tasks may be adjusted as one wishes, the number of time-lines may be arbitrarily chosen, the number of time steps for each time-line may be chosen as well as the total number of dependencies. In addition to this, the following settings, bounded by green boxes, may be changed.

- Level of occupancy, which is the proportion of time on all time-lines which is occupied by tasks, i.e. the average workload per processing unit. The lowest setting is 5 % and the highest is 95 %.

- The check-box "Utilize dependency chains" is selected if one wants to use dependency chains when creating a data set. When this option is selected, a large portion of all dependencies will be between tasks located on the same time-line, and not very far apart, as a dependency chain involves 3-10 tasks, where only one of the dependencies between these tasks is between two different time-lines. It may be noted that, for smaller data sets, there is a high degree of variability in the number of tasks which cross time-lines compared to the number that do not. When this check-box is selected, dependencies will also tend to be between tasks which are close together in time, which is deemed to be a more realistic scenario than that of haphazardly chosen dependencies.

- The check-box "Constrain dependencies to stay within a single timeline" is self-explanatory. When this check-box is selected, no dependencies will cross into different time-lines.
- The check-box "Rectify dependency probabilities based on task lengths" will alter the probability of a task partaking in a dependency by making it proportional to the length of the task, instead of all tasks having equal probability regardless of their length, as is the case when this box is not selected.
- A task spacing distribution may be chosen according to one's wishes. By default, the chi-squared setting is used as this distribution exhibits nice properties, such as non-negativity (unlike the normal distribution) while having a fairly fat tail. If the normal distribution setting is instead chosen, the expected value for the number of tasks (which is stochastic) will not be correct, with an error escalating when one increases the standard deviation due to the fact that a larger portion of the distribution will fall below zero, which is not allowed.
- The standard deviation slider located below is used for specifying the standard deviation for the distribution specified previously. The minimum level is 0.05, while the maximum level is 0.95. This setting is not highly interesting, as this only provides a linear scaling factor for the different kinds of intervals of interest.
- For the task spacing distribution, one may not select the expected value for the length of each task directly, due to the fact that this is already determined by the number of tasks selected, together with the level of occupancy.
- The task length distribution works completely analogously to the task spacing distribution.
- The minimum time between dependent tasks distribution is fairly self-explanatory. One may adjust both the mean as well as the standard deviation of this distribution in a straight-forward manner.
- The maximum time between dependent tasks distribution is completely analogous to the one above.
- The minimum starting time distribution as well as the deadline distribution are also analogous.
- Task distribution across time-lines is the distribution which generates the number of tasks for each time-line. Here, it is also not possible to change the expected value for this distribution, as this is already determined by the number of tasks to be allocated.
- Two other settings, which are grayed out, were originally intended to be included. The spatial distribution of tasks setting would adjust dependency probability based upon how far two time-lines are from each other, while the temporal distribution would make it more likely that two close tasks will be in a dependency than two tasks which are far apart.
- In a future version, due to suggestions from the supervisor, it may be possible to change the proportion of 25-50-25 % distribution for long, medium length, and short intervals, respectively.

2.2 The Simplified Mode

This mode was originally intended to work as the platform for creating test data for the project. However, it was deemed to contain too many settings to be feasible.

2.3 The Further Simplified Mode

The Further Simplified mode is the mode that was created specifically for the demands of this project. Upon selecting this mode, all green and red fields are greyed out, except for two of the red ones, which become blue. In this mode, there are scant possibilities to customize data, as most settings are already predetermined. However, one may choose to either use continuous or discrete difficulties. The former option is the default setting, and allows for smoothly increasing the level of difficulty until one obtains the desired parameters, while the latter option gives the user only three different difficulty settings to choose from. Regarding the settings which are prescribed, the following may be said.

- The option "Utilize dependency chains" is activated. It may be noted that the implementation of dependency chains was finalized after significant testing (or rather, versions of algorithms adapted to certain kinds of test data) had already been completed on the part of the Tabu group; hence, this option is of no consequence for the furtherance of this current project. However, for future versions of test data, one should aspire to utilize dependency chains, which is why this option has been activated.
- The option "Constrain dependency chains to stay within a single time-line" is not activated, since, according to the problem description, dependency chains shall be utilized. Dependency chains jump from time-lines, which is in conflict with the activation of this option. Activating this option is tantamount to a linear up-scaling of a single-time-line problem, with the scaling factor equal to the number of time-lines to be created.
- The option "Rectify dependency probabilities based on task lengths" is activated. What this means, is that the probability of a task partaking in a dependency with any other task is proportional to the length of the task. This is reasonable since tasks which take a long time complete are more likely to be highly important.

Regarding the four options available in the "Further Simplified Mode" box, the following may be uttered.

- Choice of distributions, standard deviations and expected values are the same for most settings.
- High dependencies. This option will (nominally) put the number of dependencies as four times the number of tasks. Evidently, this will create a quite intricate dependency structure between tasks, as we already for the basic case require that all tasks partake in at least one dependency.
- High density. In this case, the level of average occupancy for time-lines is 70 %, which is significantly higher than the value of 40 %, which is used for the other cases.
- Many time-lines. In this case, the number of time-lines being employed is increased from a maximum of 5 to a maximum of 30, which is the upper limit. Clearly, it is still required that there is at least one task per time-line, which will impose a restriction on data sets with very few tasks.
- Standard settings. Here, all of the settings we previously tampered with are restored to normal. This means a level of occupancy of 40 %, the number of time-lines is at most 5, the number of dependencies per task is nominally 2.
- The two greyed-out alternatives, "Long attributes intervals" and "Long dependency intervals", were originally devised in order to facilitate a test series where the lengths of these intervals could be varied. However, the group decided that this will not be tested.

2.4 Window 1

Figure 2: Window 1. Here, we see all time-lines and tasks displayed in an intuitive way.

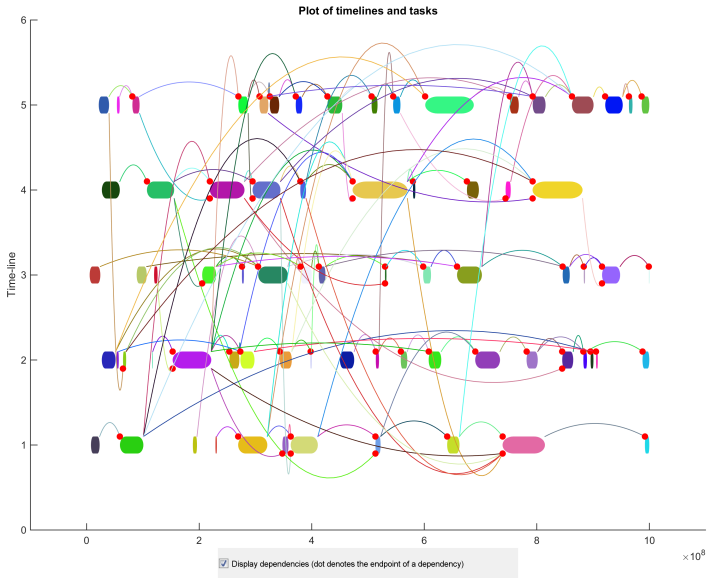
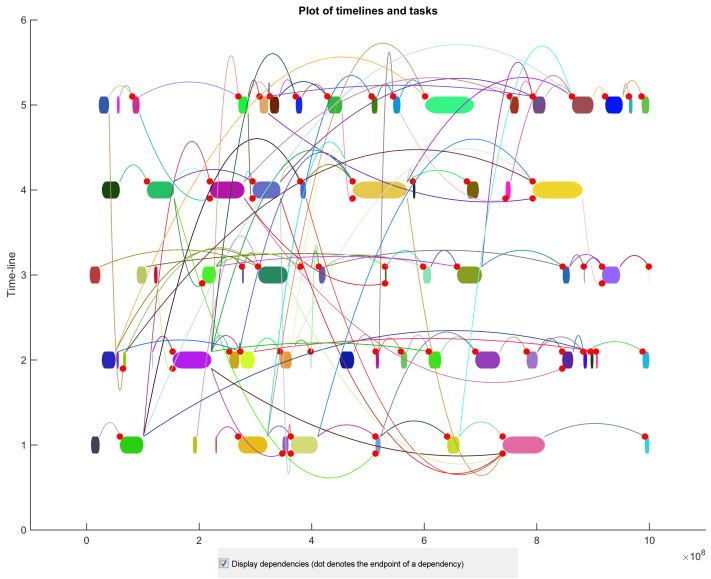


Figure 3: Window 1. Here, dependencies are displayed using second-degree splines and red dots to mark the beginning of the second task in the dependency.



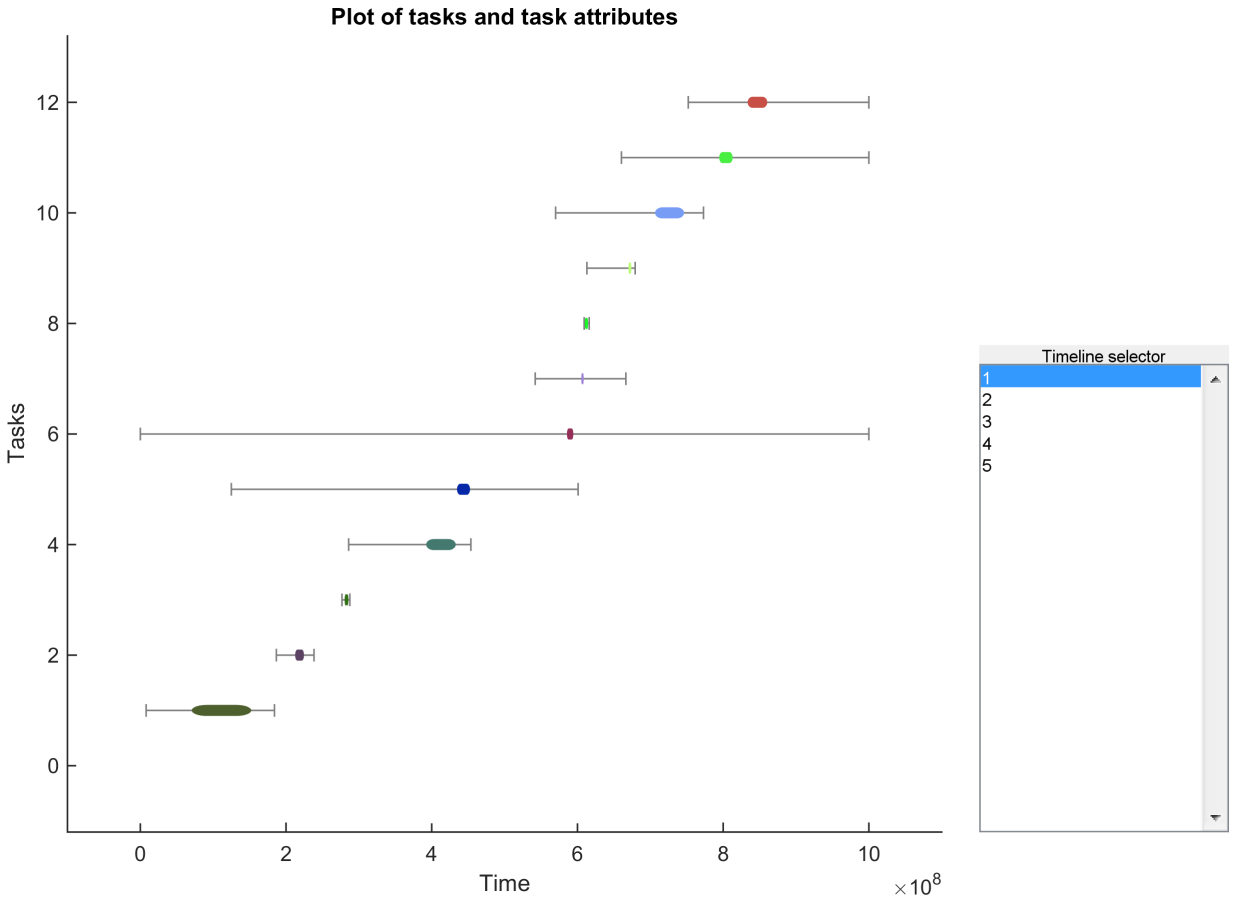
Window 1 was created in order to validate the feasibility of the test data generator, and has been useful for finding bugs in the program. Upon generating test data, a number of coloured boxes appear in this window. These boxes represent the tasks that have been created, and the position of as well the length of each task is clearly illustrated - as long as the size of the data set is not immense. In this figure, one has the option of displaying dependencies between tasks using a small check-box located below the plot. If this check-box is selected, dependencies between tasks will be displayed using second-degree splines. The curves go from the end of the first task in the dependency to the start of the second task in the dependency, and takes

into account the time-line of each task in a dependency. If the data set is large, it will take a while for the computer to render all of the different lines. One may also notice that near the end of the dependency spline, there is a red dot. This is a feature which was mainly of interest in the stage of debugging the program, as it would reveal if there are dependencies going in the wrong way (i.e. the order of tasks in a dependency is reversed, compared to what is acceptable).

2.5 Window 2

In Window 2, it may be necessary to explain what is at display. Here, task intervals as well as task lengths are displayed for all tasks on a single time-line, which may be selected using the "Timeline selector". This feature shows the structure of allowed task distributions across time-lines, and has been used to verify that the proportion of long task intervals is about 25 %, that the number of medium length intervals is about 50 % and that the amount of short intervals is about 25 %.

Figure 4: Window 2. Here, all tasks on a timelines are displayed, and the corresponding admissible intervals of placement.



2.6 Window 3

Window 3 is slightly more complicated than the previous one. In this figure, we display all dependencies in a single plot. A coloured box is used to display the first task in a dependency, and we clearly see the starting time as well as the length of this task. To the right of this task, we see a grey interval, containing a red dot. The grey box indicates the admissible starting time of the second task in the dependency, given that the first task actually starts at the time it does in the solution which has been created. The placement of the

grey interval is thus relative to the placement of the first task, as demanded by the problem specification. The red dot displays the actual starting time of the second task in the dependency, and it must lie on the grey interval if the program is to be bug-free. One may notice that the time-lines of tasks in dependencies are not displayed here. If one wants to see this, one must consult Window 1 instead.

Figure 5: Window 3. Here, all dependencies and their intervals are displayed.