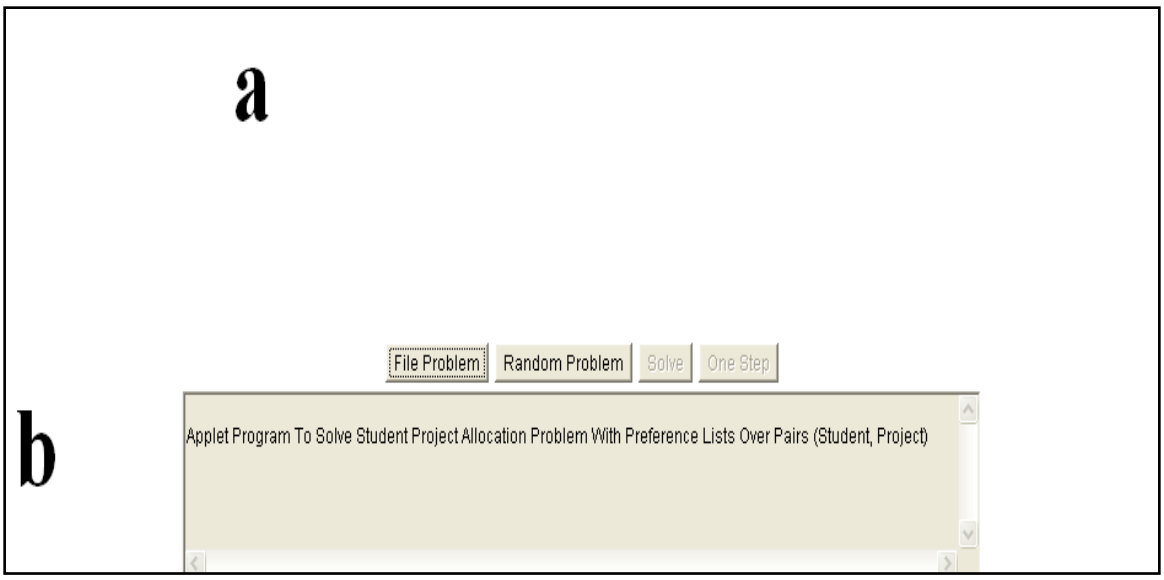

Figure 5.2: The program start window, part “a” is the visual panel and part “b” is the utility panel.

The first part is a visual panel on which preference lists of students or lecturers are drawn. The second part is utility panel which consists of four buttons and text filed; this panel is shown in details in figure 5.3. File problem button is used to display an instance of SPA-(s, p) model sorted in text file. Random problem button is used to create an instance of SPA-(s, p) not sorted in file. Solve button is used to begin solving the instance without stopping. One step button is used to solve one step and stop after that step. In the text area some sentences are written to clarify the current step.

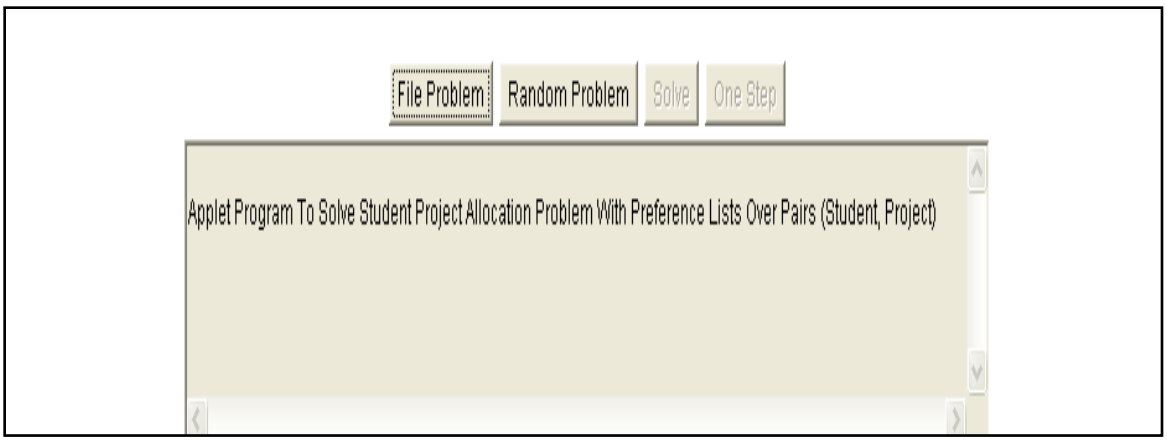

Figure 5.3: The program utility panel.

5.3.1 The Creation of the Instances

In the applet program the yellow color means that student is free that is, he has no an assignments with any project that preferred, while the yellow color means that, the lecturers or the projects are under-subscribed. The green color is used to represent a primary assignment between students and projects. When project or lecturer becomes full or over-subscribed they colored orange or red respectively. Clicking the file problem button or the random problem button an instance is drawing on the visual panel, firstly, names of the students and the lecturers are appeared on the panel without their preference lists, and projects which are offered by the same lecturer are linked to him by lines as figure 5.4 show.

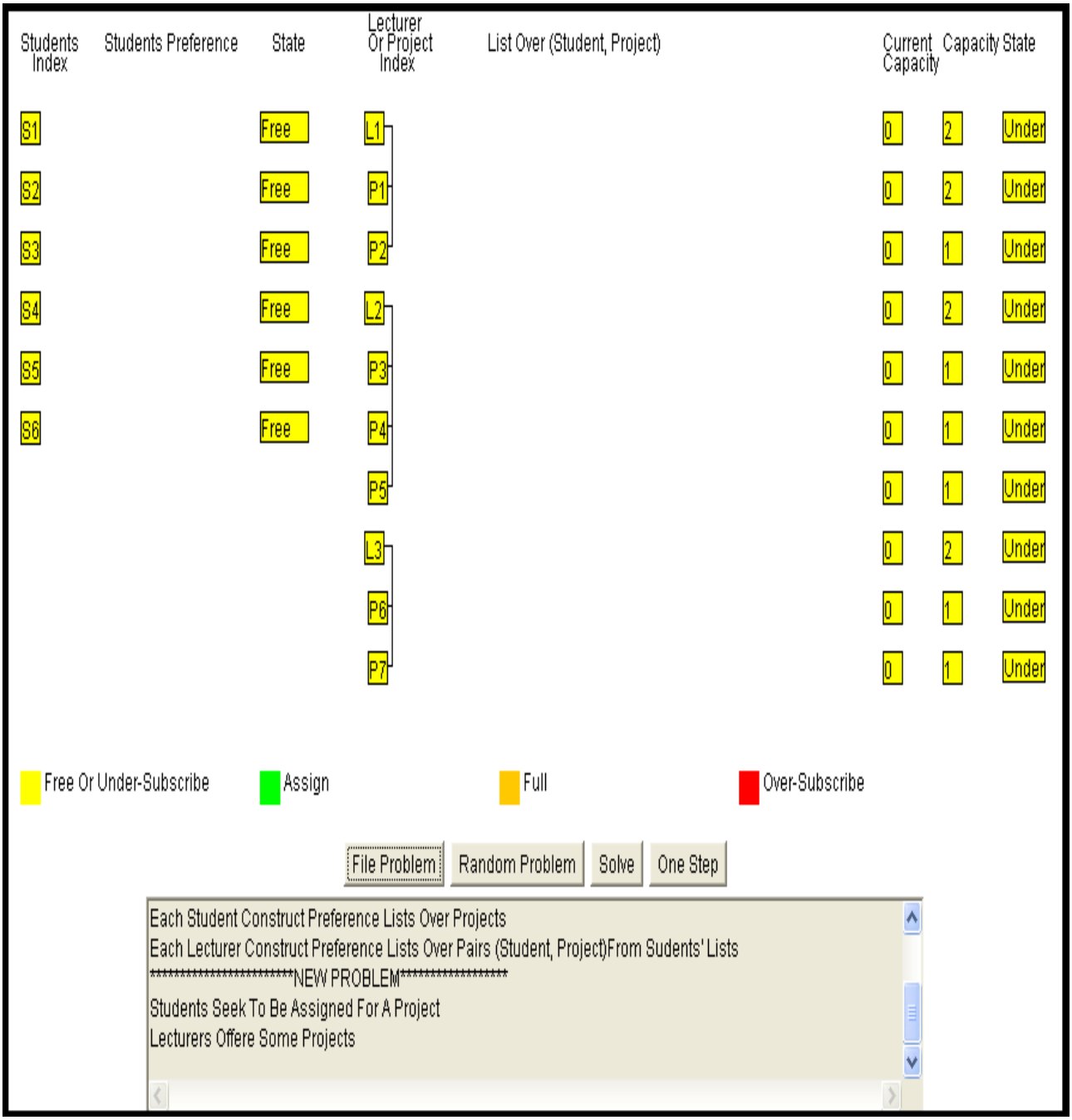

Figure 5.4: The program window after clicking “File Problem” button.

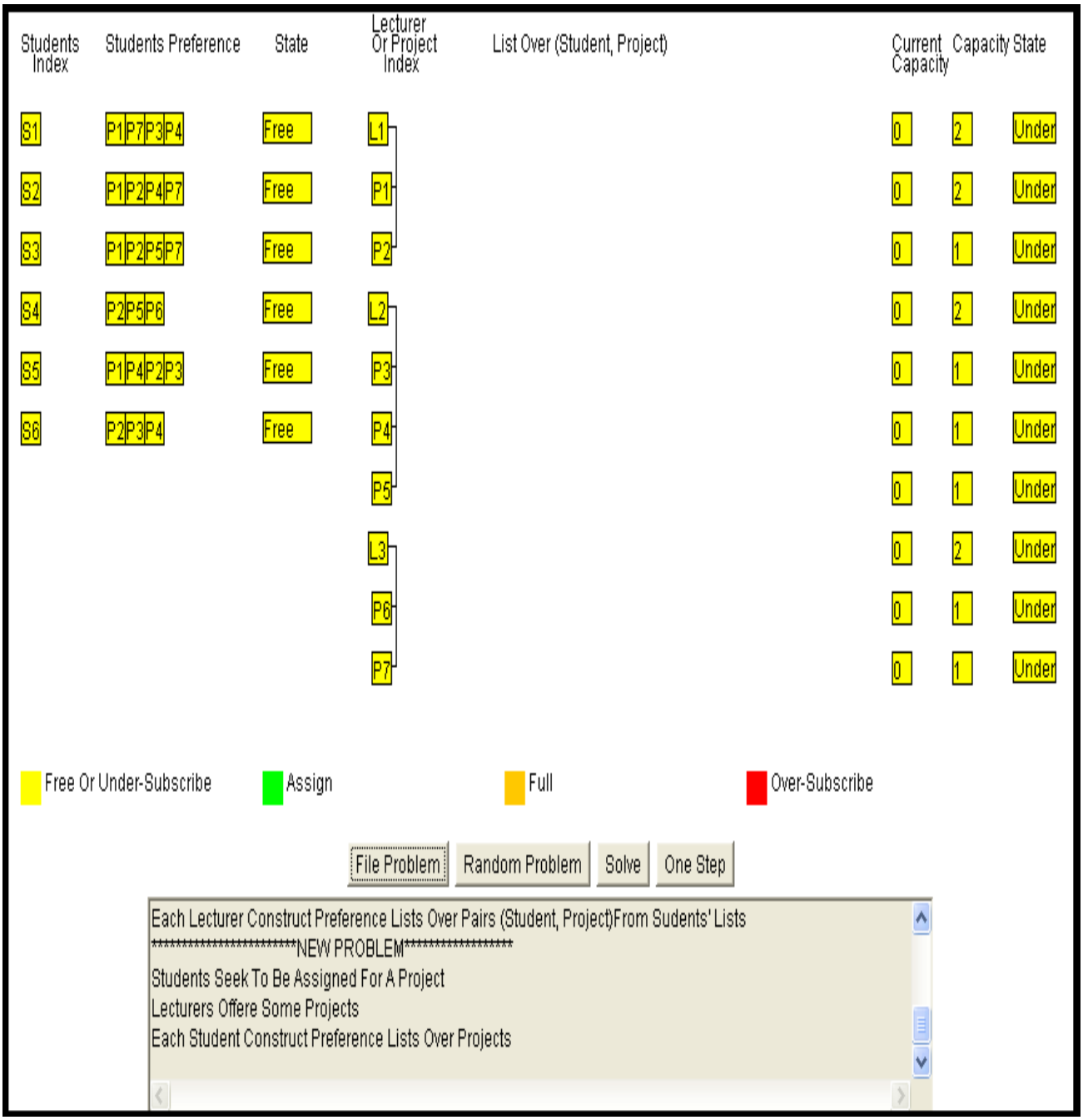

Figure 5.5: The program window after clicking “File Problem” button.

After that each student creates his preference list over projects as we see in figure 5.5, this preference list is presented as continuous boxes each of them contain an entity from the student preference list. Figure 5.6 show the visualization of the preference list $A_1 : p_1\ p_7\ p_3\ p_4$ of the student s_1 from

the given instance in figure 5.1, also after the preference list the status of the student is displayed. If the student is free the word “Free” is written in yellow box and if the student is assigned then the word “Assigned” is written in green box.

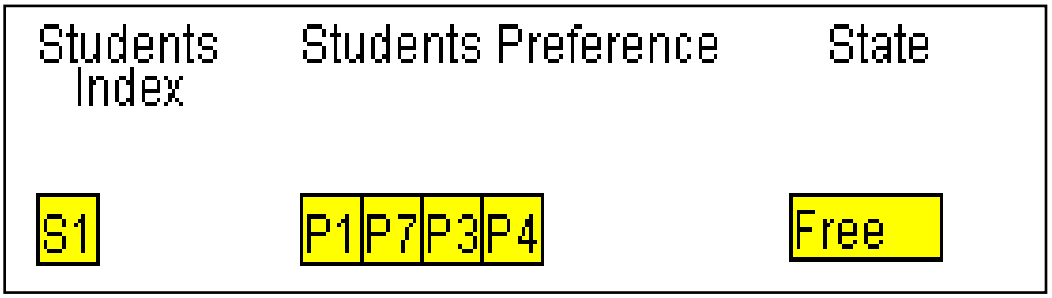

Figure 5.6: The visualization of student preference list.

When all students supply their preference lists, the lecturers start to construct their preference lists over pairs, that lists is displayed on the visual panel as we see in figure 5.8. For each project $p_j \in P_k$ a preference list L_k^j is created from the preference list of the lecturer l_k who offers that project. The preference list of each lecturer is continuous boxes each of these boxes holds one pair also each lecturer is connected to the preference lists of his projects as we see in figure 5.7.

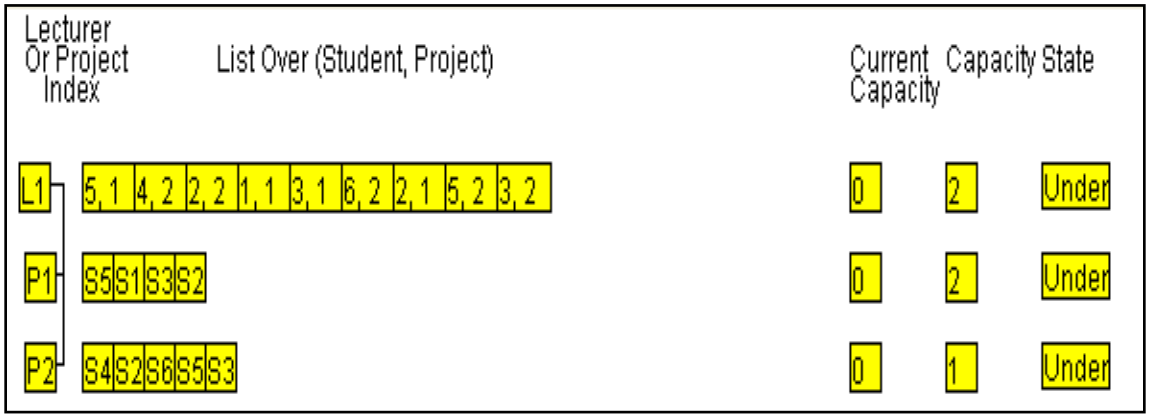

Figure 5.7: The visualization of lecturer l_1 preference list and his projects’ preference lists.

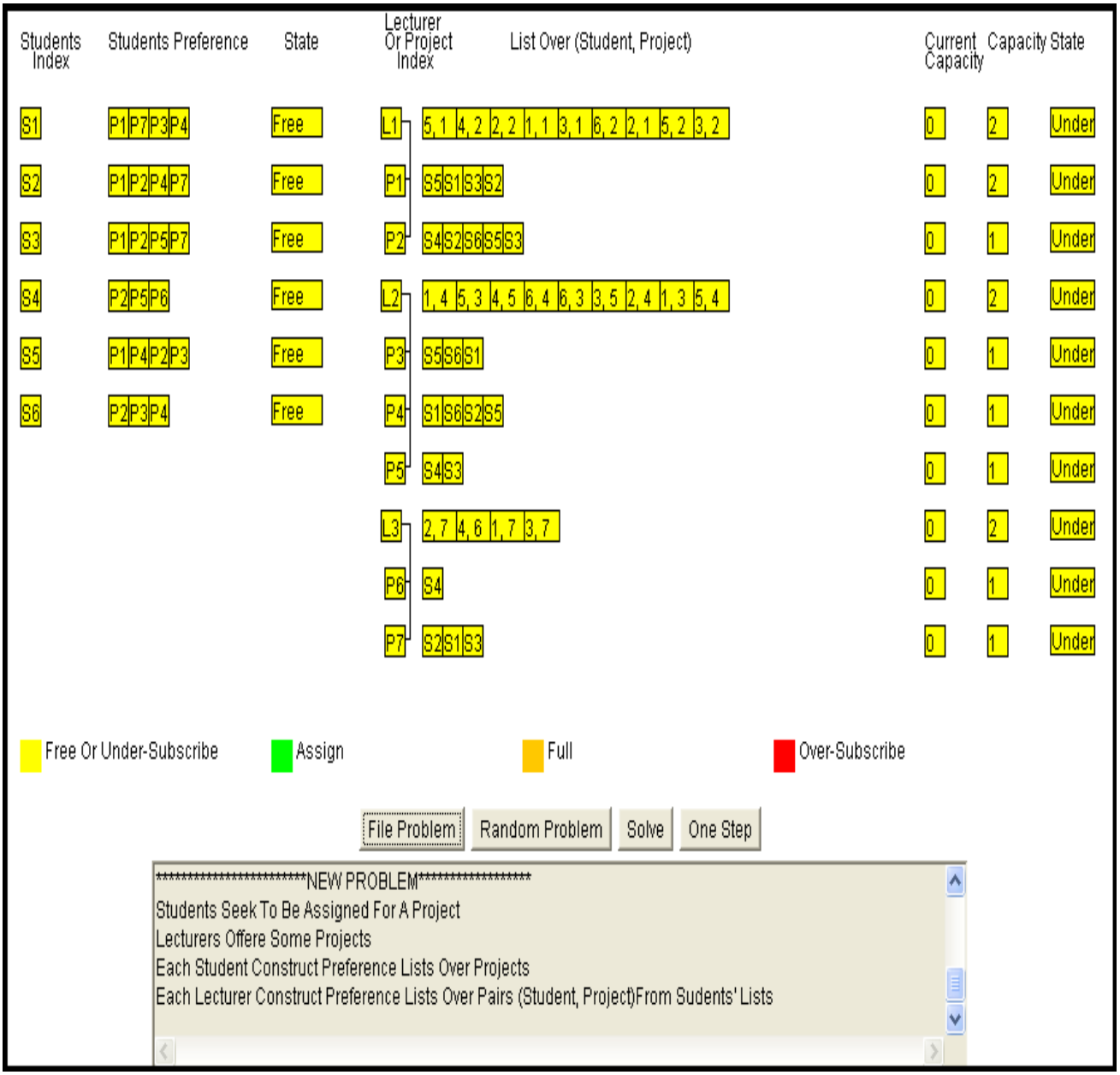

Figure 5.8: The visualization of the complete instance.

Figure 5.8 contains a number of columns, the first, named student index which contains a list of students, and to the left there is the second column shows the students' preference lists, followed by the third column on the right refers to the current state of the student in terms of whether it is linked with a project or he is free. The fourth column contains the index of the lecturers and their offered projects, that followed by the fifth column

contains the lecturers' preference lists over the student-project pairs. In the far right there are other three columns, one refers to the provisional capacity during the execution of the algorithm and the second refers to the actual capacity of lecturers and projects, the third column, which stands at the extreme right refers to the moment state of the projects and the lecturers in terms of whether they are linked to either one of the students or they are still free.

5.3.2 The Algorithm Tracing

The execution of the algorithm is consists of a number of iterations. There are three operations can be done in each iteration these operations are Assignment Operation, Deletion Operation, and Break Operation. In the following we explain how these operations were displayed in the program.

- *Assignment*: In this process the student choose the first project in his preference list over projects, this choice results a correlation between student and professor who offered the project, the choice shows through flashing the boxes of the student, the project and the lecturer in the panel and the choice colors these boxes with green color and linking the student to the project by an edge as we see in figure 5.9. The flashing refers to the provisional assignment. Finally, states and capacities of the student, the project and the lecturer are updated, see figure 5.10.

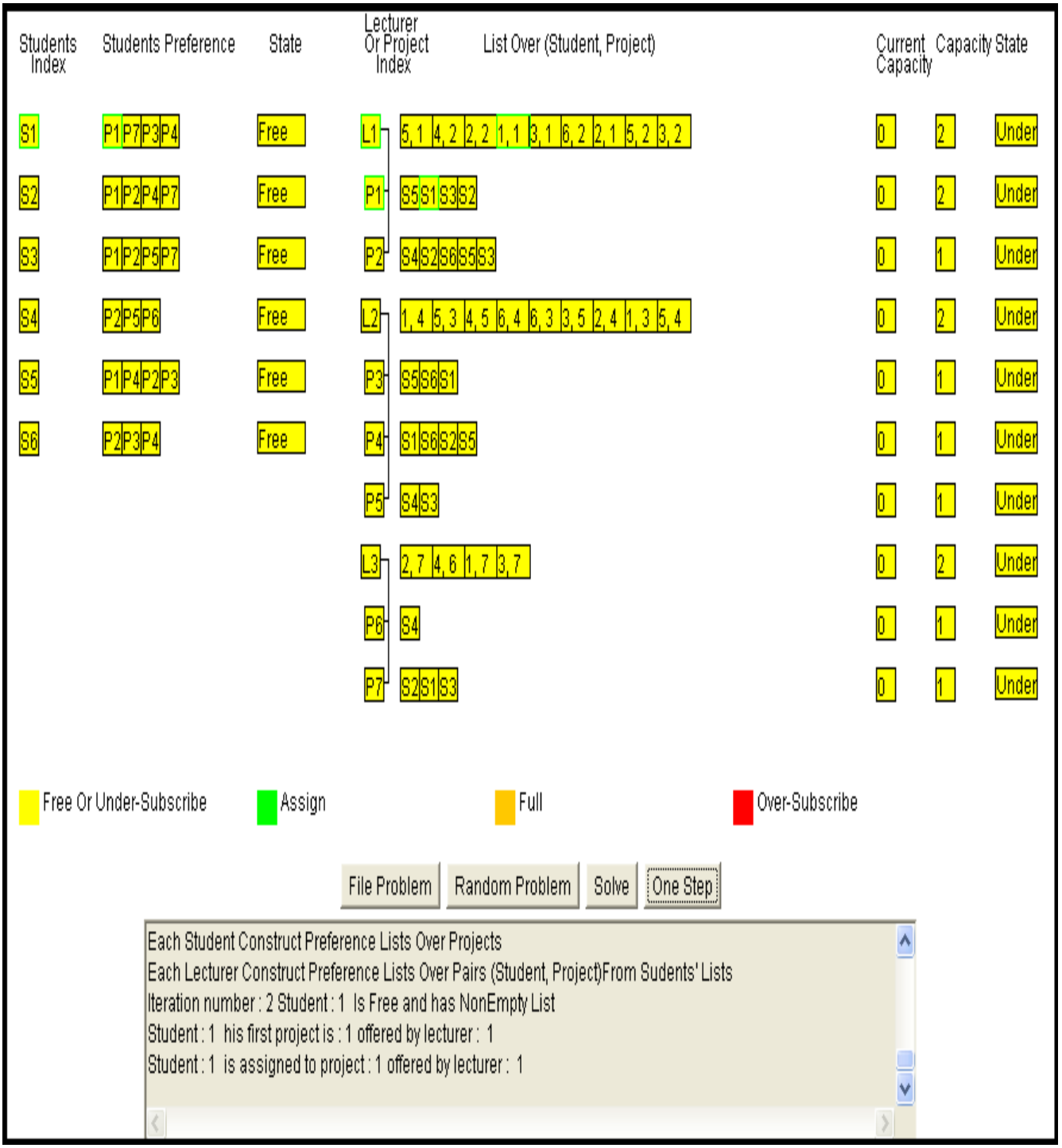

Figure 5.9: Student s_1 chooses the first project in his preference.

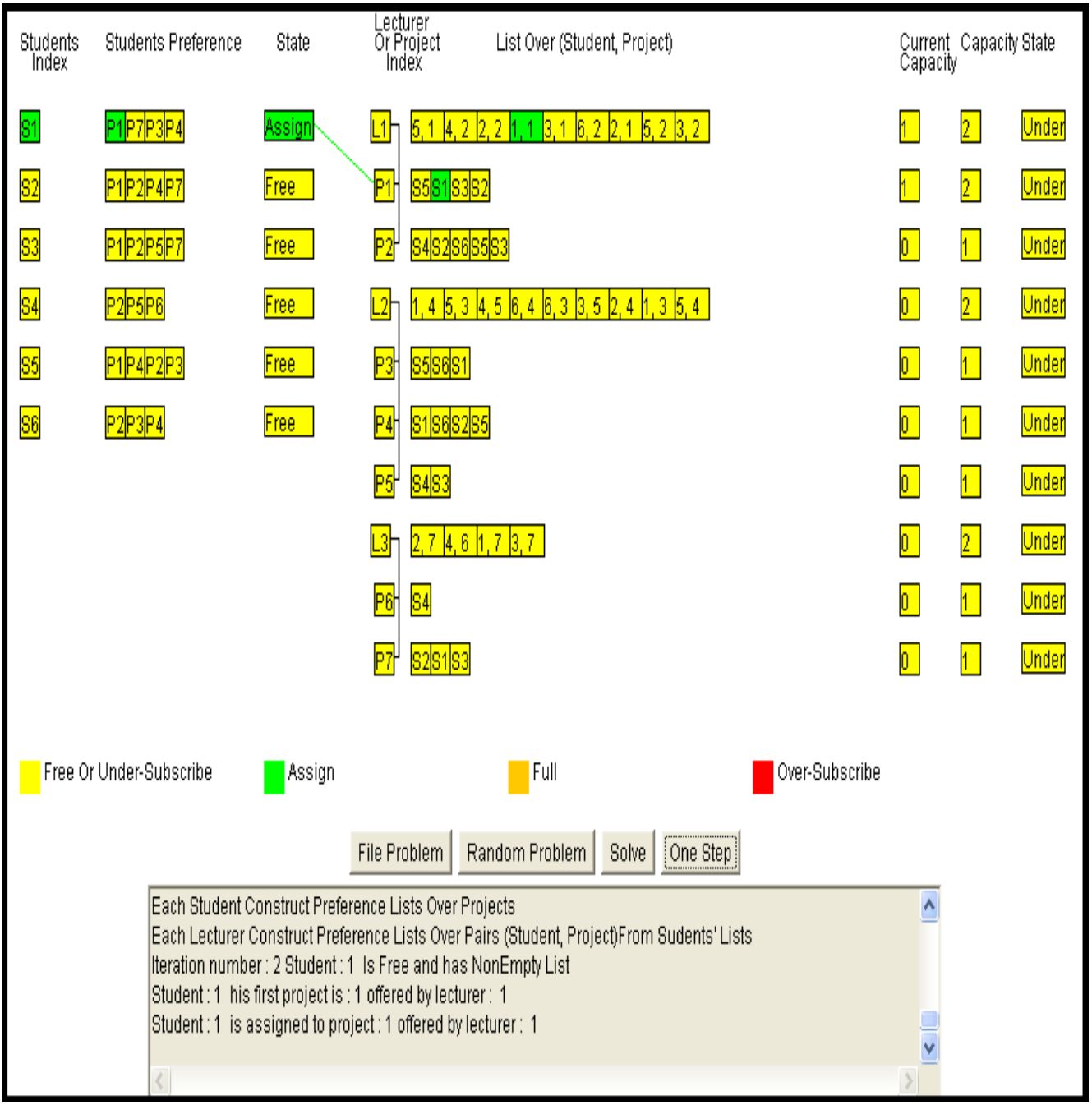

Figure 5.10: Student s_1 applied to project p_1, these lead to provisional assignments between them.

- *Deletion*: During the execution of the algorithm, any lecturer or project may become full capacity. In this case, entries are possibly deleted from the students' preference lists, and from the projected preference lists of

lecturers. Let (s_v, p_u) is the worst pair assigned to l_k and the successor pair (s_i, p_j) is deleted because l_k becomes full at the same time, the student s_i has to delete from the projected preference list of the lecturer l_k. In this case the applet colors the box that contains the name of this lecturer or the project with orange color and deletes all the unwanted pairs or students from the preference lists. In figure 5.11; the lecturer l_1 and project p_1 become full so their corresponding names are colored orange, the pairs (s_5, p_2) and (s_3, p_2) are the successors to worst pair (s_2, p_1) assigned to lecturer l_1 so this two pairs are deleted from the preference list of lecturer l_1 and the students s_3 and s_5 are deleted from preference list of project p_2 and at the same time project p_2 is deleted from preference lists of those students. Figure 5.12 displays the preference lists after deleting the first pair (s_5, p_2). Then figure 5.13 shows preference lists after deleting the two pairs (s_5, p_2) and (s_3, p_2) from the lecturer's preference list. Also the students s_3 and s_5 are deleted from the preference list of the project p_2 and the project p_2 is deleted from preference lists of those students.

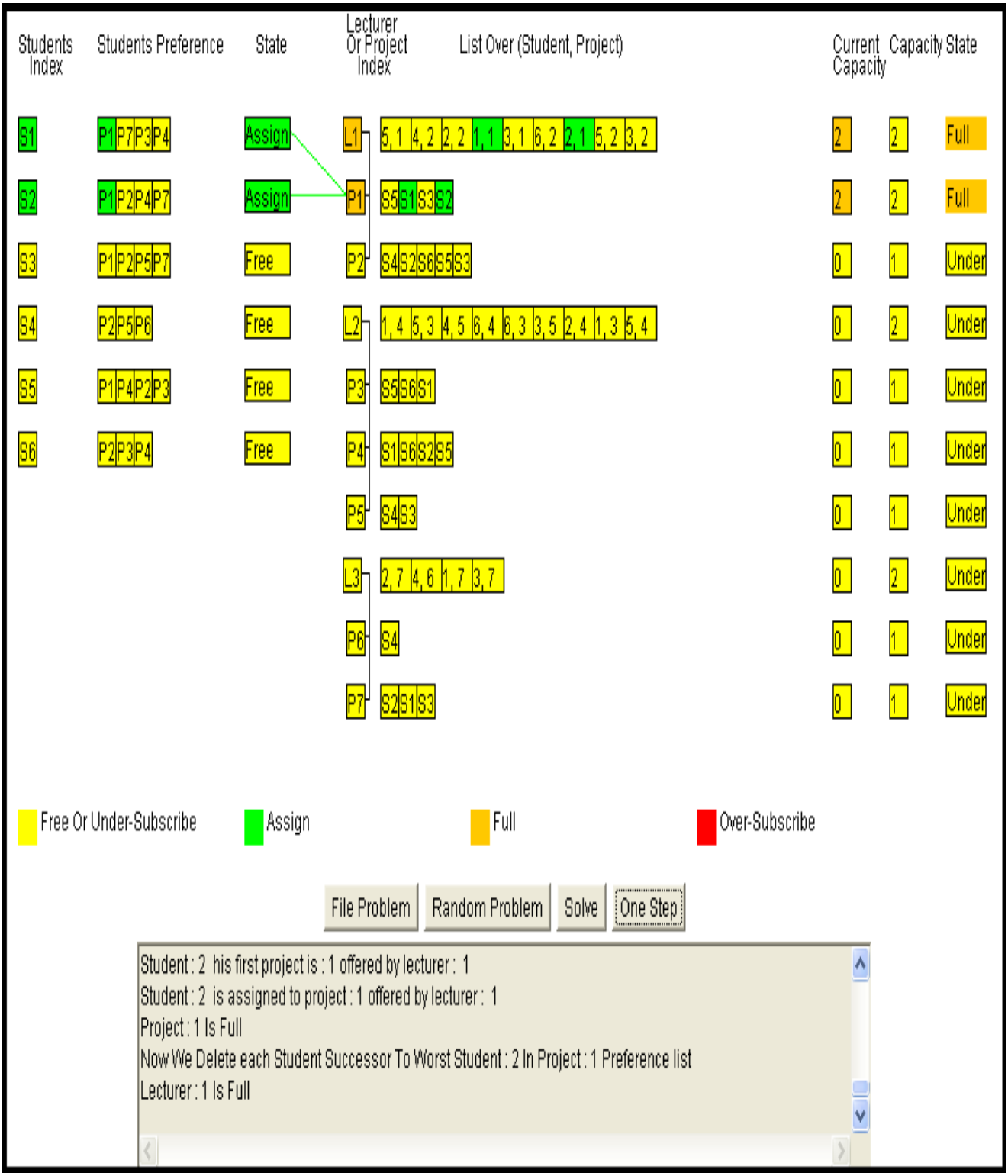

Figure 5.11: The lecturer l_1 and project p_1 become full and their corresponding names are colored orange.

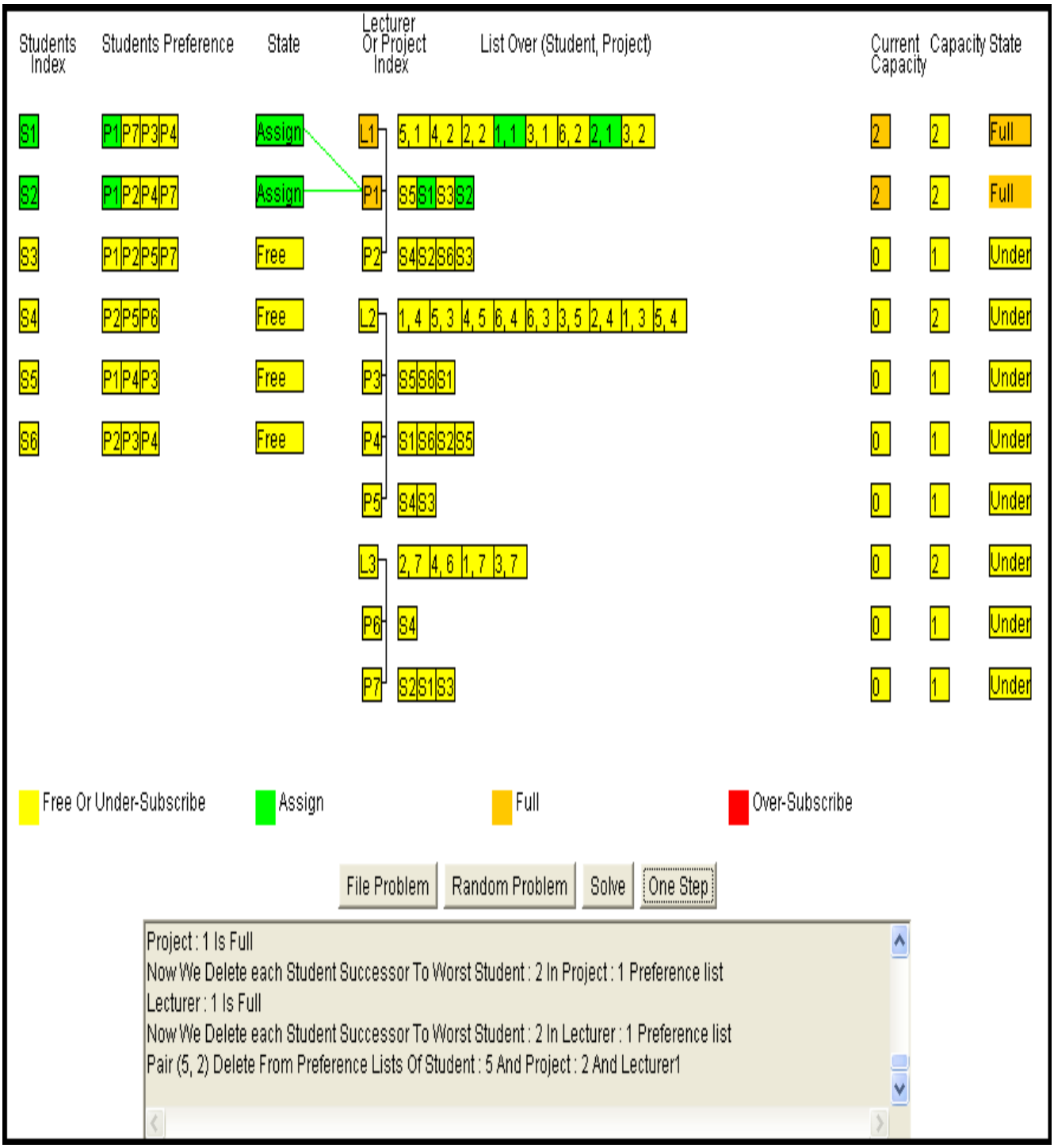

Figure 5.12: The preference lists after deleting the first pair (s_5, p_2).

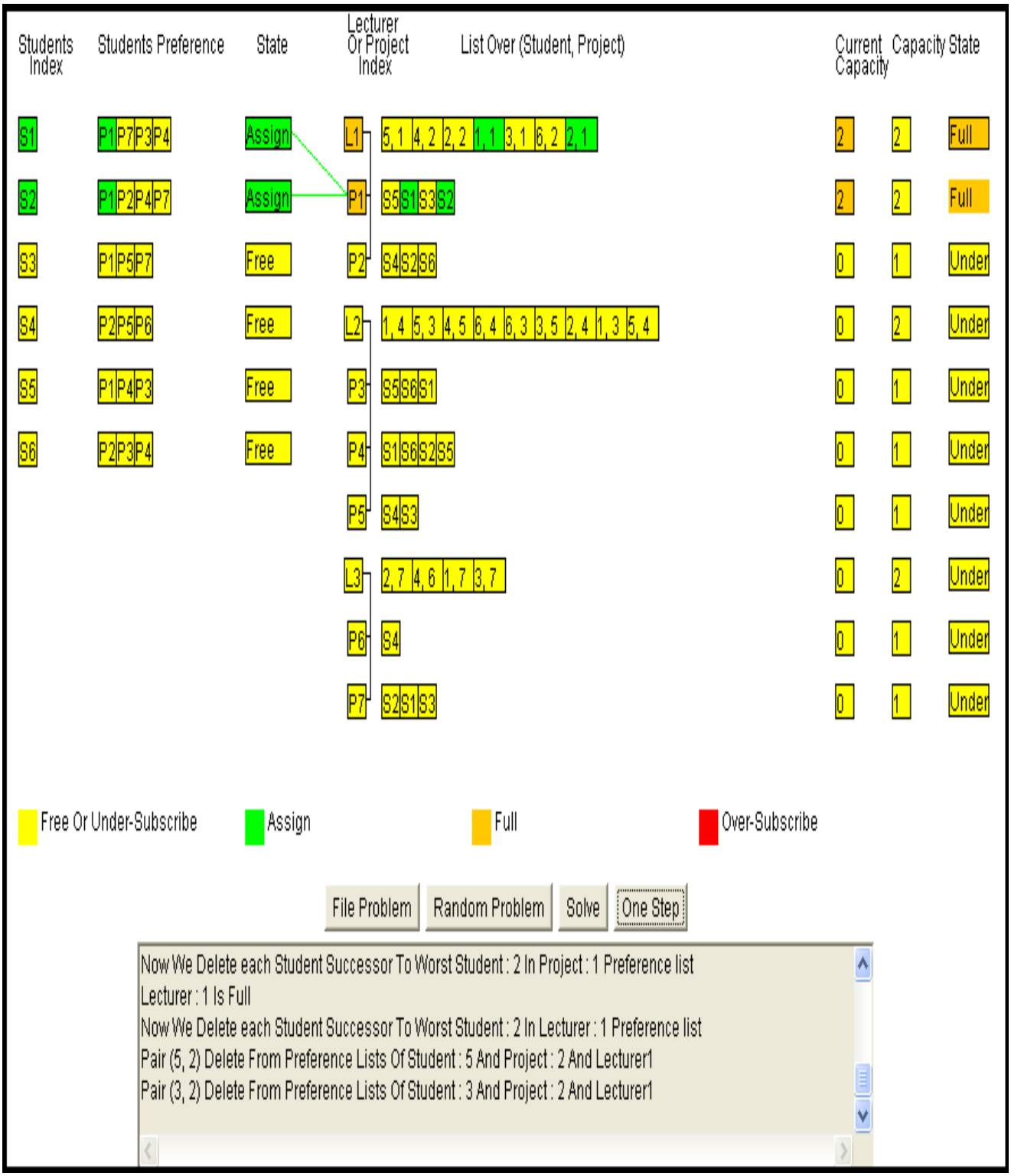

Figure 5.13: The preference lists after deleting the two pairs (s_5, p_2) and (s_3, p_2).

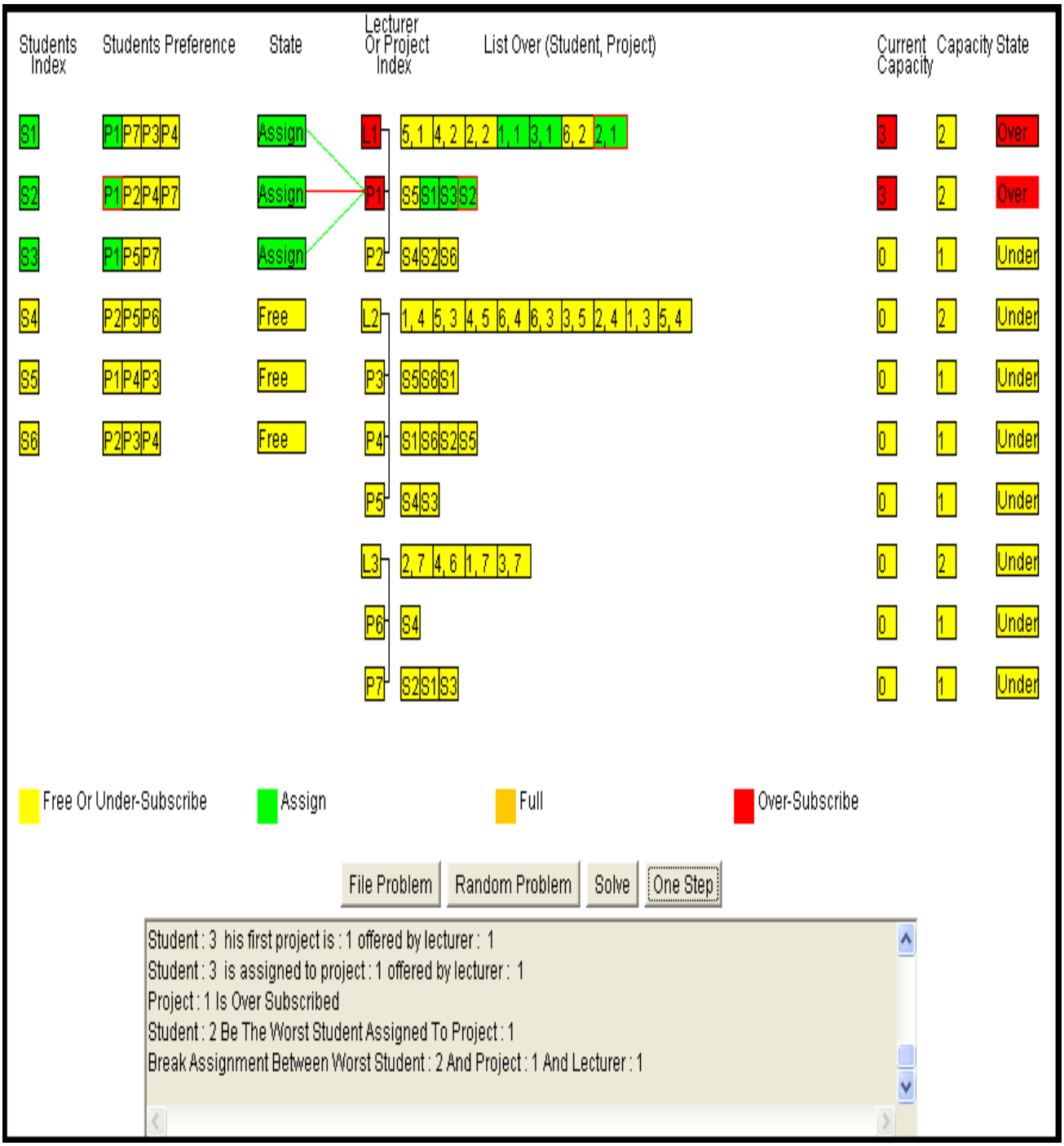

Figure 5.14: lecturer l_1 and project p_1 are over-subscribed, the worst student s_2 that was assigned to the project p_1 is selected and the edge between them is flashing.

- Break: A free student is assigned to the first project in his preference list. This leads to a provisional assignment between students, projects and lecturers, this assignment can be broken later when a project or a lecturer becomes over-subscribed and their boxes are colored red. As figure 5.14 shows, the project p_1 becomes over-subscribed and the student s_2 is selected to break his assignment to the project p_1 at this moment the edge between them is flashing before breaking the assignment between the student s_2 and the project p_1. Figure 5.15 show the preference lists after breaking the assignment between the student s_2 and the project p_1.

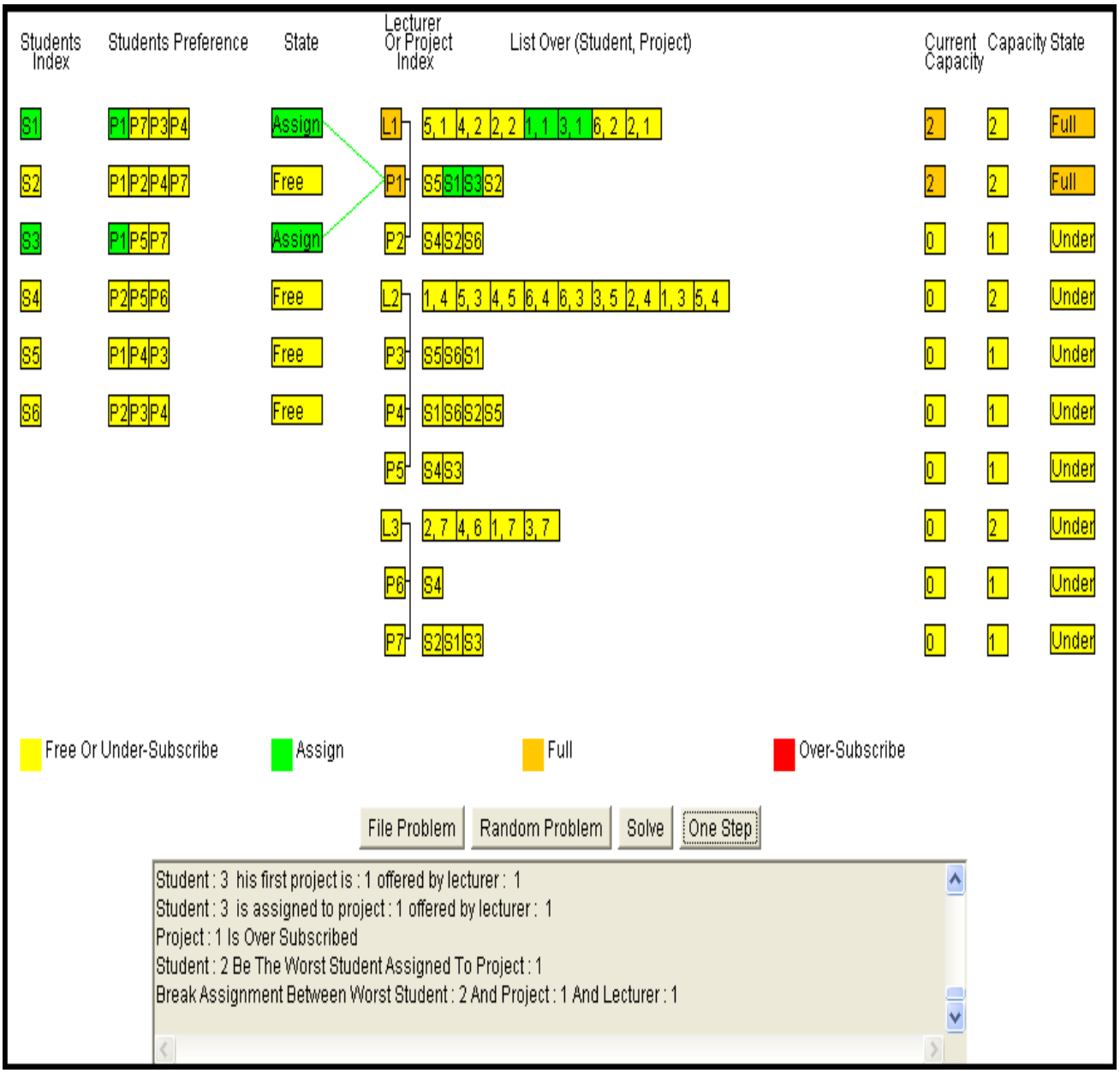

Figure 5.15: preference lists after break assignment between the student s_2 and the project p_1.

Each iteration loop includes a free student applying to the first project on his/her preference list over the projects. After a number of iterations bounded by the overall length of the student preference lists, each student is assigned to at most one project and the assigned pairs constitute the stable

match. The stable match is displayed as green boxes in the panel, the stable match is written in the lower part of Figure 5.16, the name of the student and the name of the best possible wishes project in his/her preference list.

Figure 5.16: the stable matching.