**ISE328 – Algorithms and Optimization Methods**

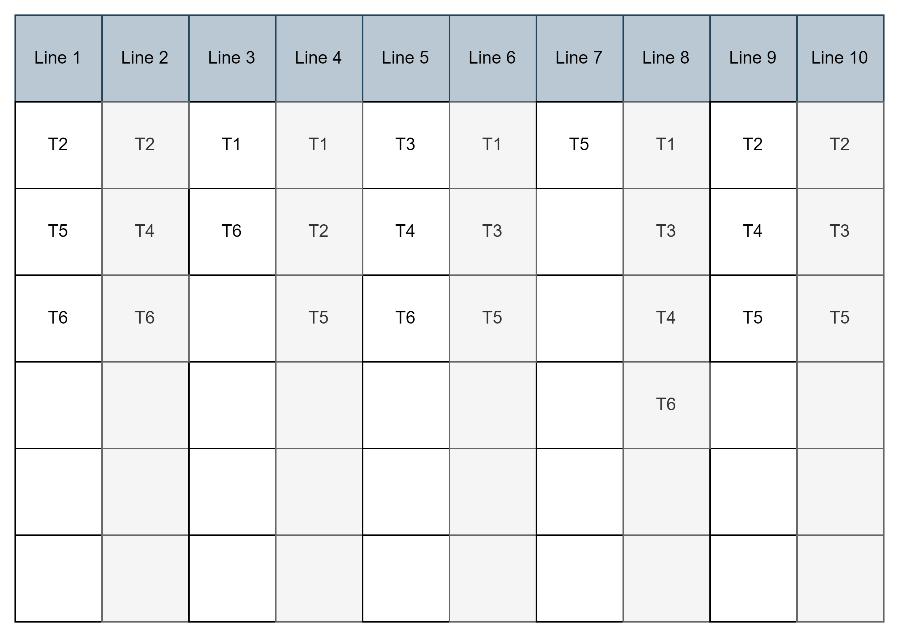
**2023-2024 Spring Semester**

**Homework #2 Report**

**Converting Test Case Selection Problem into ILP Problem:**

In the test case selection problem, our main goal is to find minimum number of test cases that checks all the code lines. So, in ILP, our main goal is to minimize number of test cases. So, we can directly say that our objective function is T1+T2+T3+T4+…. +TN. And it is a minimization problem. Where T1……TN E {0,1}. So each one of the test cases can be either included in the solution (1) or not included in the solution (0).

While converting the constraints (subject to) part of the Test Case Selection Problem, the main goal is to make sure that each line is checked by one of the chosen test cases. So, while converting it into the ILP problem, each constraint should be about each line, whether it is tested by test case or not. So, my solution is to create a two-dimensional array that stores Lines number and which test cases are testing that line. Fallowing image illustrates the two-dimensional arrow that I have created.



The image on the left side illustrates the instance given on the homework document. I have stored lines numbers and which Test cases checks that line.

While converting the problem into ILP, I have used this table to make my job easier. We should include all lines in the problem and make sure that they are solved. So, for example for the line 1, we should have T2 or T5 or T6 (or we could pick more than 1). Which converts into T2+T5+T6>0, (at least one of the test cases should be in the solution in order to line 1 to be tested by the test cases). For each line, my code creates these constraints.

“G” Value in the cplex means it is bigger or equal to that right hand side, so , instead of T2+T5+T6>0, I have used T2+T5+T6>=1 which means the same thing (at least one of them should be included in the solution).

Generalizing this technique to all of the lines, I have created the constraints with this way. (T1……TN E {0,1} constraint still apply in this section). In the end for this instance, my code created fallowing constraints:

T2+T5+T6>=1

T2+T4+T6>=1

T1+T6>=1

T1+T2+T5>1

T3+T4+T6>=1

T1+T3+T5>=1

T5>=1

T1+T3+T4+T6>=1

T2+T4+T5>=1

T2+T3+T5>=1

With this way (creating constraints based on the line), I ensured that no line left untested.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Code working with the homework instance.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Code working with the given instance:

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu siyah, karanlık içeren bir resim

Açıklama otomatik olarak oluşturuldu