# Discrete Optimization - Assignment 2

## Theoretical part

First, we start by clarifying the notation. We are given element and sets where . Further we are given a cost function defined for each . The ILP problem is:

Solving the LP-relaxation yields a potentially fractional solution, . We then let be the sets that are picked in the fractional solution. That is, . Note that for due the constrains the problem; if than one could reduce the primal variable and still satisfy all constrains for edges that is covered by

Secondly ,we turn to the question. Let be the set picked in an iteration of the random rounding where sets have probabilities are . As argued in Vazirani, the chance that any element is not covered is:

:

Then, we use Fréchet inequalities as we are not sure that we can assume independence between two elements being covered by (we guess that we cannot):

Thus, we know that the probability of at least half the sets is not covered is:

Thirdly, we turn to the cost. We know that:

Since . Next, we need that the integrality gap of LP relaxation is the harmonic series, , as argued in Vazirani on page 111:

Finally, we apply the Markov inequity to get:

Thus, we get that by the union bound:

## Implementation

We have printed a table with times at the end of this document

### 2.1

Note that we made an implementation with the CPLEX java API to avoid having to export the results

## 2.2

From an efficiency standpoint in random rounding, we 1) only look at sets with a primal variable greater than zero in the LP relaxation and 2) we only consider sets not yet included in any iteration. Lastly, we set as in slide 27

Related to the theoretical question, we see that the product of the LP relaxation objective value and the integer solution is far from the final objective value of the random rounding. This intuitively makes sense given that the harmonic series plus the inverse of the exponential number is small. To illustrate, we note that the harmonic series take on values greater than 5 when the . Thus for all instances:

We would assume that the union collections so do not differ much in cost. Why? Because we are likely to pick the same sets the collection as we use the same probability of including a given set. I.e. high probability sets will likely occur in the each collection so the union should not differ much

## 2.3

We choose to use the primal-dual schema in chapter 15

## Performance table

Below is a table of the result and running time. Running times are reported as “Algorithm running time” / “Algorithm running time” + “time for loading the instance.” All running times are computed when solely running the program with the particular algorithm to avoid problems with caching distorting the comparison. Note that we compute a mean, minimum and maximum for random rounding over 200 samples. The reported running time is for the first solution in the random rounding column:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Random rounding | | | | Rounding | | Primal-dual | | Cplex exact | | Clpex LP-relaxation | |
| Instance | Running time | Min | Max | Mean | Running time | Value | Running time | Value | Running time | Value | Running time | Value |
| scpa3 | 108/152 | 415 | 487 | 451.36 | 109/135 | 490 | 2.24/27 | 448 | 346/373 | 232 | 113/139 | 228.00 |
| scpc3 | 173/216 | 480 | 580 | 529.26 | 125/163 | 587 | 2.42/39 | 463 | 850/892 | 243 | 122/155 | 234.54 |
| scpnrf1 | 320/495 | 39 | 62 | 51.25 | 321/484 | 78 | 2.02/160 | 43 | 12,165/12,308 | 14 | 314/450 | 8.89 |
| scpnrg5 | 322/466 | 467 | 541 | 513.80 | 309/416 | 621 | 3.93/105 | 412 | - | - | 290/385 | 148.23 |