

Info: Whenever it is asked to implement something with python-mip you can also use any other similar library and/or programming language of your choice. Just shortly ask in Discrod to clarify if this is fine for me. Also, if not specified in detail, for all programming exercises generate some interesting test cases to run the program and output solutions and objective values in a reasonable format.

1 Oracle-based algorithm for (continous) budgeted interval uncertainty

1.1 Continous budgeted interval uncertainty

Prove that the min-max robust optimization problem with continous budgeted uncertainty can be solved by solving 2 nominal problems.
(Do not use the results on locally budgeted uncertainty to prove this.)

1.2 Implementation

Implement both the algorithm for continous and discrete budgeted interval uncertainties and test them for two combinatorial optimization problems of your choice.

2 Locally budgeted uncertainty

2.1 Implementation 1

Implement the oracle based algorithm for locally budgeted uncertainty that needs 2^K many executions of the nominal problem. Compare it to the implementations of section 1.2.

2.2 Selection

Give a formal prove for the Theorem on slide 14 about the selection problem.

2.3 Knapsack

Extend the results for selection to the knapsack problem to obtain a pseudo-polynomial algorithm for knapsack.

2.4 Implementation Knapsack

Implement the compact model (slide 9), the oracle based algorithm (slide 10) and the pseudo-polynomial algorithm for knapsack (exercise 2.3). Generate relatively large random instances and compare the running times of the three appraoches.