Algorithm Engineering – Exercise 5

Team 3 (Dennis Weiss, Lisa Dörr, Khiem That Ton)

1. Branching Algorithm

We implemented the algorithm presented in the lecture which solves the optimization problem directly by branching on merging two vertices of a P_3 and branching on deleting the edge between them (and setting it to forbidden).

We further considered solving the connected components of the graph separately. We added the data reductions *Heavy Non-Edge*, *Heavy Edge Single End* and *Heavy Edge Both Ends* and the *Closed Neighborhood* rule, which we described in our third milestone. We employed the following upper and lower bounds.

1.1. Upper Bound

As described in Algorithm 1 we firstly run the greedy closed neighborhood heuristic as described in the previous lecture for n_{greedy} iterations and choose the best solution found and then run Simulated Annealing for n_{SA} iterations with initial temperature T_{start} on that solution.

Algorithm 1 Upper Bound Algorithm

procedure UPPERBOUND(G)
    $(\hat{G}, \hat{k}) \leftarrow$ (**null**, ∞)
    for $1 \leqslant i \leqslant n_{greedy}$ **do**
        $(G', k') \leftarrow$ Randomized Greedy Closed Neighborhood Heuristic on G
        if $k' < \hat{k}$ **then**
            $(\hat{G}, \hat{k}) \leftarrow (G', k')$
        end if
    end for
    $(\hat{G}, \hat{k}) \leftarrow$ Simulated Annealing on $\hat{G}$ for n_{SA} iterations with initial temperature T_{start}
return $\hat{k}$
end procedure

1.2. Lower Bounds

- *Edge-Disjoint P_3's Lower Bound:*

$$LB = \sum_{(u,v,w) \in \mathcal{P}} \min(s(u,v), s(v,w), -s(u,w))$$

with $\mathcal{P}$ being a list of edge-disjoint P_3's and $s(u,v)$ being the weight of the vertex pair (u,v). $\mathcal{P}$ is chosen by sorting the list of all P_3's of the graph by the smallest absolute weight of any of the three edges in descending order and iteratively adding a P_3 to $\mathcal{P}$ if it is edge-disjoint with all P_3's added to $\mathcal{P}$ so far.

- *Relaxed ILP Lower Bound* from lecture 5

2. Analysis

Since the computation of the *Relaxed ILP Lower Bound* is much slower than the *Edge-Disjoint P_3's Lower Bound*, we used only the latter for our baseline algorithm.

In Figure 1 the ratios between the solution of the upper bound (Section 1.1) and *Edge-Disjoint P_3's* lower bound (Section 1.2) to the exact solution for all problem instances that were solved within a $30s$ time limit are depicted. The upper bound had an average ratio to the exact solution of 1.020 and for the lower bound the average ratio was 0.749. It can be concluded that our upper bound is much tighter than our lower bound and thus the lower bound is the main bottleneck of the algorithm's performance.

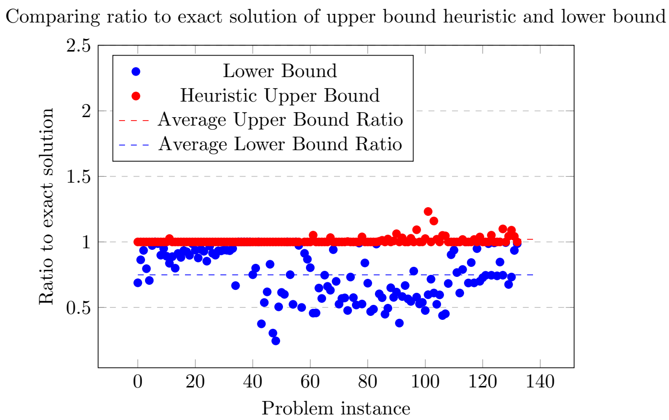

Figure 1: Tightness of upper and lower bound

In Figure 2 we compare the ratios to the exact solution of different versions of the lower bound. The *Greedy Lower Bound* selects the edge-disjoint subset of P_3's as in Section 1.2. The *Exact Solution Lower Bound* selects the edge-disjoint subset of P_3's that maximizes the lower bound. This problem reduces to a MAXIMUM-WEIGHT INDEPENDENT SET problem, which we solved using an ILP solver. From the results of this experiment we conclude that the heuristic of sorting the list of P_3's and then iteratively selecting the P_3's is already close to optimally selecting an edge-disjoint subset of P_3's. This implies that there is no great use in further optimizing the heuristic

Param.	Description	Value range	Value found by SMAC
p_{CC}	probability to split the connected components into separate graphs in each recursive step	$[0, 1]$	0.1122
p_{LP-LB}	probability to compute the relaxed LP lower bound in each recursive step	$[0, 1]$	0.6040
p_{LB}	probability to compute the *Edge-Disjoint* P_3*'s* lower bound in each recursive step	$[0, 1]$	0.8695
n_{greedy}	Runs of the greedy heuristic of the upper bound	$\{2^0, 2^1, \ldots, 2^{10}\}$	2^{10}
n_{SA}	Iterations of the Simulated Annealing process of the upper bound	$\{2^6, 2^7, \ldots, 2^{16}\}$	2^{14}
T_{start}	Initial temperature used for the Simulated Annealing process of the upper bound	$[0.1, 10]$	8.5962

Table 1: Parameters defined for the configuration optimization process

of selecting the P_3's. To obtain a tighter lower bound it is required to use a fundamentally different lower bound, such as the *Relaxed ILP Lower Bound*.

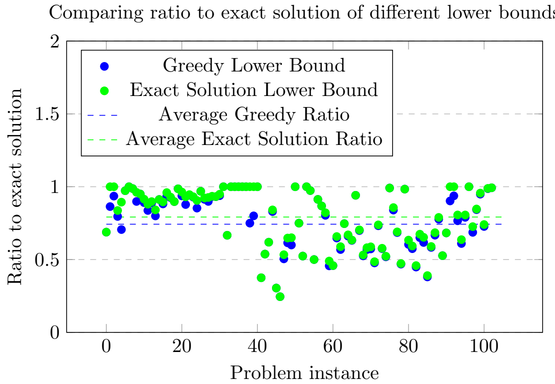

Figure 2: Comparison of different lower bounds

In Figure 3 we compare the running times between versions of the algorithm with and without using the data reductions developed in our third milestone. The data reductions have no significant effect on the average running time and do not provide a consistent improvement of the algorithm in our experiments.

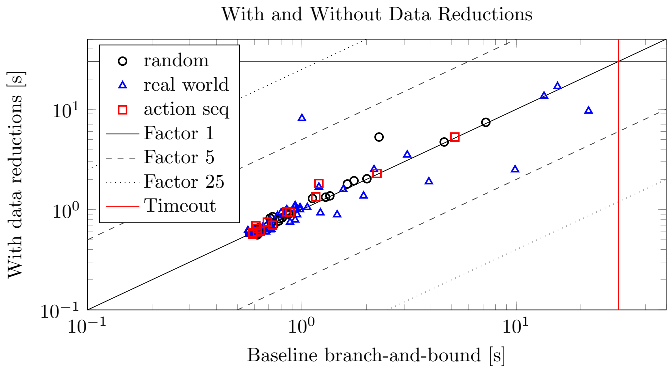

Figure 3: Comparison of running times between algorithm versions with and without use of data reductions

Furthermore, we investigated the effect of splitting the graph into its connected components. We could not observe an improvement by this strategy. In both cases 117 of the publicly available problem instances could be solved within a time limit of $5min$.

3. Optimizing Algorithm Parameters

We employed SMAC [1] to automatically optimize parameters of our algorithm. The parameters defined with their value ranges and the resulting parameters can be seen in Table 1. We used the PAR-10 value with a timeout of $180s$ per problem instance as the value to optimize. We ran the optimizer for 36 hours on 3 parallel instances.

4. Final Evaluation

In Figure 4 we compare the running times of this milestone's algorithm (with the parameters found by SMAC), as in Table 1, our third milestone's solver and the ILP solver (IBM ILOG CPLEX). We conclude that our new solver outperforms our previous search tree algorithm and comes close to the performance of the ILP formulation.

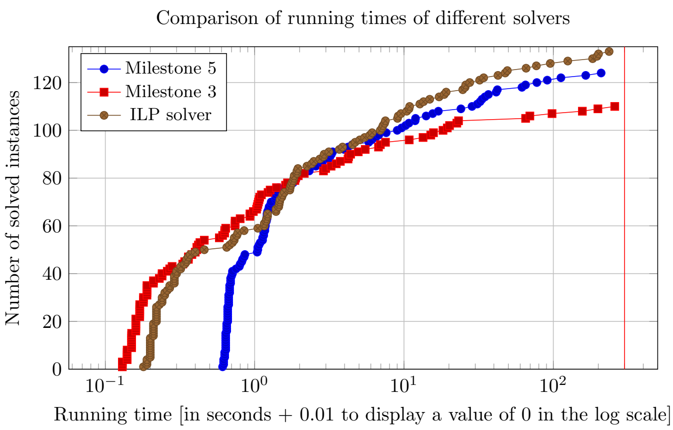

Figure 4: Comparison of this milestone's algorithm with parameter values found by SMAC, our milestone 3 algorithm and the ILP solver

References

[1] F. Hutter, H. H. Hoos, K. Leyton-Brown, Sequential model-based optimization for general algorithm configuration, in: Proc. of LION-5, 2011, p. 507–523.