A Comparative Study of SMT and MILP for the Nurse Rostering Problem

Alvin Combrink[1], Stephie Do[1], Kristofer Bengtsson[2], Sabino Francesco Roselli[1], Martin Fabian[1]

Appendix: Details on the Experimental Evaluation

A. Problem A

Problem A is inspired by [1] and [2], that detail typical NRPs. Instances are generated from a given number of shifts and staff members. Shifts from Table I are added sequentially, from top to bottom, to the first day. Once the end of the table is reached, the same is done for the next day, and so on, until the desired total is achieved. Similarly, staff members from Table II are added sequentially, cycling through the list repeatedly, until the problem contains the desired number of staff members.

This problem includes six daily shifts: 2 day, 2 evening, and 2 night shifts, each lasting 9 hours with equal workload. Shifts overlap with adjacent shifts. All shifts require the *Nurse* qualification (N), while one shift also requires *Administration* (A). Each set of personnel includes one person with both qualifications and another with half the desired workload. Overlapping shifts are disallowed for all personnel. The constraints for Problem A are detailed in Table III, along with the GC used to model them and the values given to the input parameters.

Instances of the problem are created for the number of shifts ranging from 50–300, equivalent to approximately 8–50 days in the schedule, and the number of staff members from 10–40. The solvers are given a time limit of 1 hour to either find a solution or determine that the problem is infeasible. If they fail to return a result within this time frame, the instance is considered unsolved.

Figures 1a and 1b show the solving times for Z3 and Gurobi, respectively. Each square represents one instance of the problem with a certain number of shifts and staff members, seen on each axis. The unsatisfiable instances are shown with white hatching and a pixel is completely white if the solver was terminated due to the time limit. As can be seen, Z3 was not able to solve all instances within the time limit while Gurobi could. The instances in the top-left region, with many shifts and few staff members, appear to be harder to solve and thus require more time for both solvers. Here, Gurobi was faster and managed to solve them before the time limit. More interesting is the bottom-right region where the instances have many staff members but few shifts. Here, unsatisfiable instances occur which Gurobi appears to decide equally, if not faster, than the satisfiable instances in the same area. Z3, on the other hand, was not able to decide most of these within the time limit.

Figure 2a compares solver performance using the logarithmic quotient $\log_{10}(\text{Gurobi time} / \text{Z3 time})$. Positive values indicate Z3 is faster, while negative values indicate Gurobi is faster. Black regions represent the cases where Z3 failed to terminate within the time limit but Gurobi succeeded. Figure 2b shows the same instance matrix but clarifies which solver is faster (or equal within 5%). The results reveal distinct performance patterns: Gurobi outperforms Z3 in regions with fewer staff members relative to shifts (top left) and fewer shifts relative to staff members (bottom right). In the remaining areas, such as near the center where the problems are not as constrained, Z3 generally finds solutions faster than Gurobi.

[1]Division of Systems and Control, Department of Electrical Engineering, Chalmers University of Technology, Göteborg, Sweden
[2]Research and Technology Development, Group Trucks Operations, Volvo AB Göteborg, Sweden {kristofer.bengtsson@volvo.se}

TABLE I: Set of shifts for problem A, with each respective type, start hour, duration, workload (WL) in hours, and required qualifications.

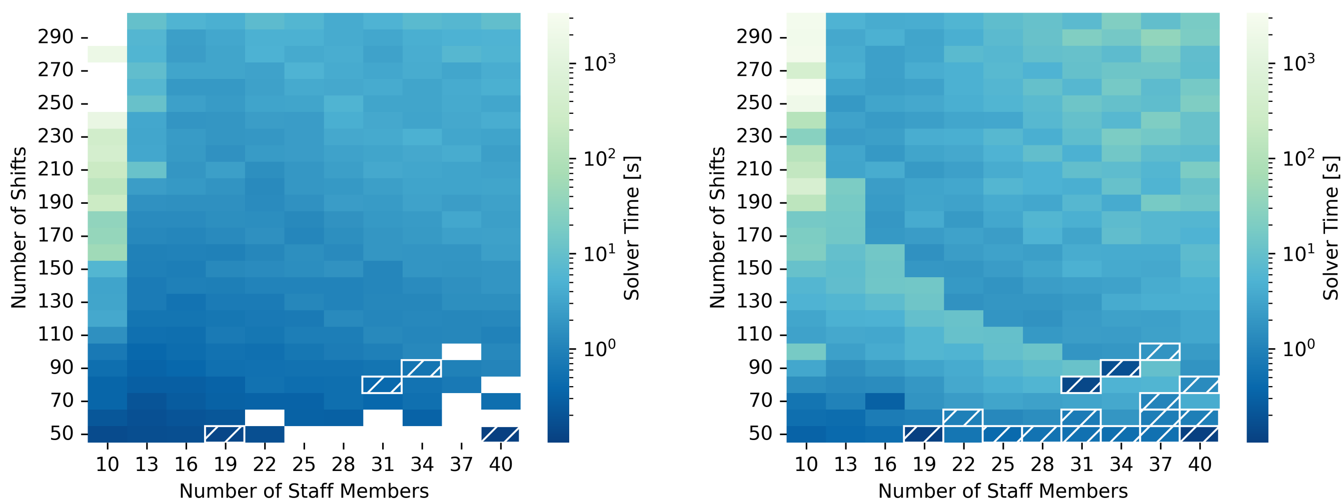
Type	Start	Dur.	WL	Req.Qual.
Nurse D1	06:00	9:00	9	N, A
Nurse D2	06:00	9:00	9	N
Nurse E1	14:00	9:00	9	N
Nurse E2	14:00	9:00	9	N
Nurse N1	22:00	9:00	9	N
Nurse N2	22:00	9:00	9	N

TABLE II: Set of staff members for problem A, with each respective desired workload in hours, and qualifications.

ID	Desired WL	Qualifications
1	100	N, A
2	100	N
3	100	N
4	50	N

TABLE III: Constraints for problem A.

1.	All shifts must be assigned at least one person; **GC1**: $\widetilde{\mathcal{S}} = \mathcal{S}$, $\widetilde{\mathcal{P}} = \mathcal{P}$, $x = y = 0$.
2.	The 1st staff member cannot be assigned to any shift that starts on days 3–5 of the schedule; **GC1**: $\widetilde{\mathcal{S}} = \mathcal{S}^{sd}(\mathcal{S}, \{3,4,5\})$, $\widetilde{\mathcal{P}} = \{1\}$, $x = y = \lvert\mathcal{S}^{sd}(\mathcal{S}, \{3,4,5\})\rvert$.
3.	Everyone must be qualified for the shifts that they are assigned to; **GC2**: $\widetilde{\mathcal{S}} = \mathcal{S}$, $\widetilde{\mathcal{P}} = \mathcal{P}$, $x = y = 0$.
4.	No one should be assigned to shifts that overlap in time unless they are allowed; **GC3**: $\widetilde{\mathcal{P}} = \mathcal{P}$, $x = y = 0$.
5.	At least 50% of the shifts assigned to the 1st staff member must be of the types *Nurse D1* and *Nurse D2*; **GC4**: $\widetilde{\mathcal{S}} = \mathcal{S}^{t}(\mathcal{S}, \{\textit{Nurse D1, Nurse D2}\})$, $\widetilde{\mathcal{P}} = \{1\}$, $u = 0.5$, $v = 1$.
6.	If the 1st staff member is assigned to any shift starting on days 1 or 2, the 4th and 7th staff members may not be assigned to any shifts on those days; **GC5**: $\widetilde{\mathcal{S}}_1 = \widetilde{\mathcal{S}}_2 = \mathcal{S}^{sd}(\mathcal{S}, \{1,2\})$, $\widetilde{\mathcal{P}}_1 = \{0\}$, $\widetilde{\mathcal{P}}_2 = \{4,5\}$, $x = y = 0$.
7.	If someone has been assigned a day shift, they cannot be assigned a night shift on the same day; $\forall j \in \mathcal{P}$, $\forall d \in \mathcal{D}$: **GC5**: $\widetilde{\mathcal{S}}_1 = \mathcal{S}^{sd}(\mathcal{S}, d) \cap \mathcal{S}^{t}(\mathcal{S}, \{\textit{Nurse D1, Nurse D2}\})$, $\widetilde{\mathcal{S}}_2 = \mathcal{S}^{sd}(\mathcal{S}, d) \cap \mathcal{S}^{t}(\mathcal{S}, \{\textit{Nurse N2, Nurse N2}\})$, $\widetilde{\mathcal{P}}_1 = \widetilde{\mathcal{P}}_2 = \{j\}$, $x = y = 0$.
8.	No one can be assigned to shifts for more than 6 days in a row; **GC6**: $\widetilde{\mathcal{S}} = \mathcal{S}$, $\widetilde{\mathcal{P}} = \mathcal{P}$, $x = 0$, $y = 6$.
9.	If someone has been assigned to shifts for 4–6 days in a row, they must be off 3 days before and after; **GC7**: $\widetilde{\mathcal{S}} = \widetilde{\mathcal{S}}_1 = \widetilde{\mathcal{S}}_2 = \mathcal{S}$, $\widetilde{\mathcal{P}} = \mathcal{P}$, $x = 4$, $y = 6$, $n = m = 3$.
10.	All consecutive days of shift assignments must be of the same type; **GC8**: $\widetilde{\mathcal{S}} = \mathcal{S}$, $\widetilde{\mathcal{P}} = \mathcal{P}$.
11.	All personnel workloads must be within 30% of the expected workload, with regards to all shifts; **GC9**: $\widetilde{\mathcal{S}} = \mathcal{S}$, $\widetilde{\mathcal{P}} = \mathcal{P}$, $v = 0.3$.



(a) Solving times using Z3. White indicates that the time limit was reached. White hatching means the solver returned that the problem is unsatisfiable.

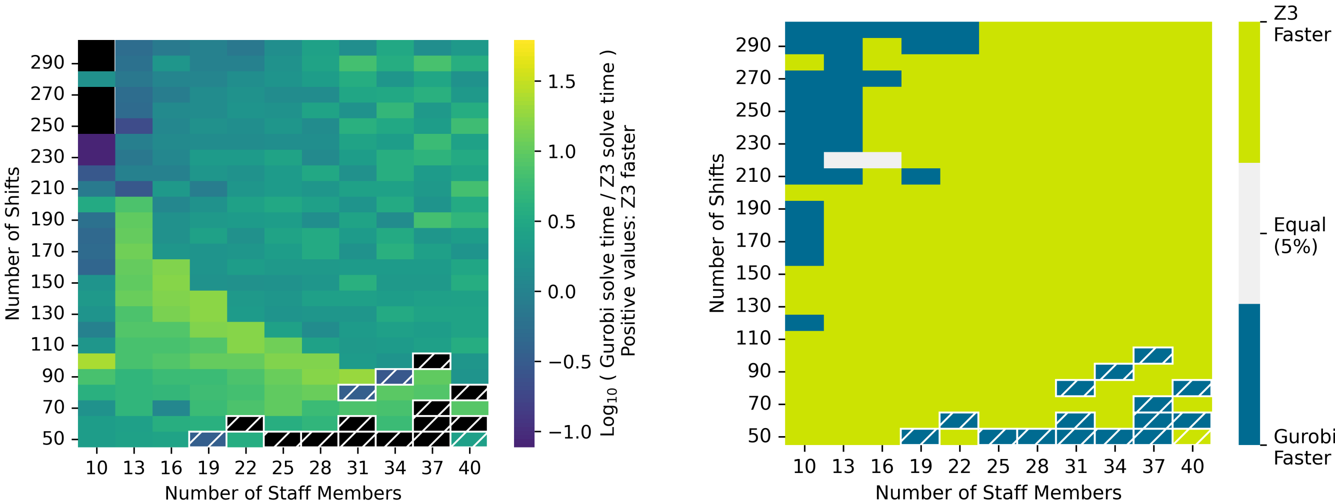
(b) Solving times using Gurobi. White hatching means the problem is unsatisfiable.

Fig. 1: Each respective solver's solving times for Problem A.

To further explore this, we ran the same experiments but without constraint 11, which bounds assigned workloads to near desired workloads. The results are shown in Figures 3a and 3b, where all previously infeasible problems are now feasible and both solvers are able to terminate within the time limit. We additionally ran the same experiments when including *only* constraints 1 and 11, requiring all shifts to be assigned someone and assigned workloads to be near desired workloads. The infeasible instances (when including all constraints) remained infeasible now with only constraints 1 and 11, confirming that the infeasibility is due to the combination of those two constraints. Thus, many shifts relative to staff members puts assigned workloads closer to their upper bounds while few shifts relative to staff members puts assigned workloads closer to their lower bounds. This suggests that Gurobi is faster to process the more tightly constrained instances than Z3.

B. Problem B

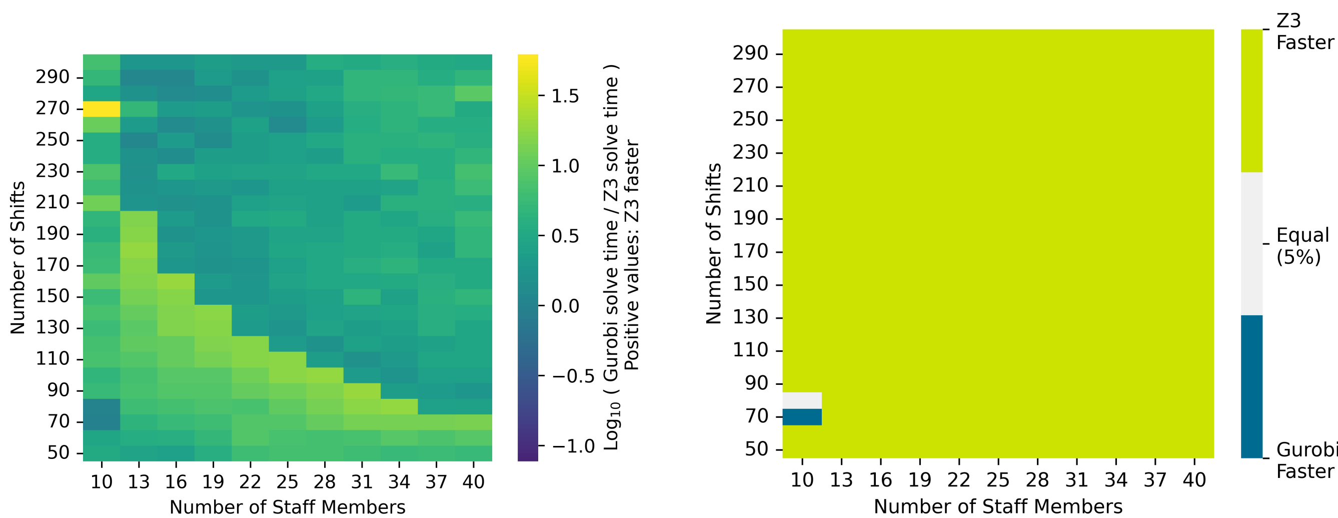
Problem B is larger and attempts to mirror a more complex scheduling problem in a hospital ward. Inspiration for the constraints come from the real-world case presented in [3]. The staff set is constant, containing 29 staff members with a mix of qualifications and desired workloads (Table IV). The number of desired days decides the set of shifts; the top 19 shifts in Table V are added to each day in the schedule. An additional *Admin* shift occurs every second day, starting on the first day of the schedule. The constraints are given in Table VI. Staff members are allowed to be assigned to any of the following shift type pairs simultaneously if they are qualified for them: *Nurse D-C.Nurse D*, *Nurse E-C.Nurse E*, *Doctor D-Doctor S1*,



(a) A comparison of the solving times between Z3 and Gurobi for each problem instance. Black means Z3 was not able to find a solution within the time limit while Gurobi could. White hatching means the problem is infeasible.

(b) A comparison between Z3 and Gurobi based on solving times for each problem instance. White hatching means the instance is infeasible.

Fig. 2: Comparative analysis of Z3 and Gurobi performance on Problem A instances.



(a) A comparison of the solve times between Z3 and Gurobi for each problem instance, after constraint 11 has been removed.

(b) A comparison between Z3 and Gurobi based on solve times for each problem instance, after constraint 11 has been removed.

Fig. 3: Comparative analysis of Z3 and Gurobi performance on Problem A instances where constraints 11 has been removed.

Doctor D-Doctor S2, and *C.Nurse D-Admin* .

Figure 4 shows the solving times for each solver over 1–17 days included in the scheduling problem, with a time limit of 5 hours. Z3 is able to find a solution faster than Gurobi in all instances. None of the solvers where able to find solutions to problems including more than 16 days within the time limit.

References

[1] Haroldo G Santos et al. "Integer programming techniques for the nurse rostering problem". In: *Annals of Operations Research* 239.1 (2016), pp. 225–251.

[2] Edmund K Burke et al. "A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem". In: *European Journal of Operational Research* 188.2 (2008), pp. 330–341.

[3] Alvin Combrink and Stephie Do. "Automatic shift scheduling for healthcare personnel using satisfiability modulo theory". MA thesis. Electrical Engineering, 2021. URL: `https://hdl.handle.net/20.500.12380/304237`.

TABLE IV: Set of staff members for problem B, with each respective desired workload in hours, and qualifications.

ID	Desired WL	Qualifications
1-3.	100	N, CN
4.	100	N, A
5-14.	100	N
15.	75	N
16-17.	50	N
18.	100	D, S1, S2
19-20.	100	D, S1
21-22.	100	D, S2
23.	75	D, S2
24-26.	100	D
27.	50	D
28.	100	A

TABLE V: Set of shifts for problem B, with each respective type, start hour, duration, workload (WL) in hours, and required qualifications.

	Type	Start	Dur.	WL	Req.Qual.
1-5.	Nurse D	06:00	9:00	9	N
6-9.	Nurse E	14:00	9:00	9	N
10-11.	Nurse N	22:00	9:00	7	N
12-13.	Doctor D	06:00	10:00	10	D
14.	Doctor E	12:00	10:00	10	D
15.	Doctor N	19:00	9:00	7	D
16.	Doctor S1	05:00	9:00	9	D, S1
17.	Doctor S2	05:00	9:00	9	D, S2
18.	C.Nurse D	06:00	10:00	10	CN
19.	C.Nurse E	16:00	8:00	8	CN
20.	Admin *	08:00	5:00	5	A

* Occurs every second day, starting on the first day.

TABLE VI: Constraints for problem B.

1. All shifts must be assigned at least one person; **GC1**: $\widetilde{\mathcal{S}} = \mathcal{S}$, $\widetilde{\mathcal{P}} = \mathcal{P}$, $x = y = 0$.
2. Everyone must be qualified for the shifts that they are assigned to; **GC2**: $\widetilde{\mathcal{S}} = \mathcal{S}$, $\widetilde{\mathcal{P}} = \mathcal{P}$, $x^{=,y} = 0$.
3. No one should be assigned to overlapping shifts that they are not allowed to work at the same time; **GC3**: $\widetilde{\mathcal{P}} = \mathcal{P}$, $x = y = 0$.
4. At least 10% of the shifts assigned to person 1 (1st staff cycle) must be of the types *Nurse D* and *Nurse E*; **GC4**: $\widetilde{\mathcal{S}} = \mathcal{S}^t(\mathcal{S}, \{\textit{Nurse D, Nurse E}\})$, $\widetilde{\mathcal{P}} = \{1\}$, $u = 0.1$, $v = 1$.
5. If someone is assigned to a *Doctor D*, *Doctor S1* or *Doctor S2* shift, they cannot be assigned to a *Doctor N* shift on the same day; $\forall j \in \mathcal{P}$, $\forall d \in \mathcal{D}$: **GC5**: $\widetilde{\mathcal{P}}_1 = \widetilde{\mathcal{P}}_2 = \{j\}$, $\widetilde{\mathcal{S}}_1 = \mathcal{S}^t(\mathcal{S}, \{\textit{Doctor D, Doctor S1, Doctor S2}\}) \cap \mathcal{S}^{sd}(\mathcal{S}, d)$, $\widetilde{\mathcal{S}}_2 = \mathcal{S}^t(\mathcal{S}, \{\textit{Doctor N}\}) \cap \mathcal{S}^{sd}(\mathcal{S}, d)$.
6. If someone is assigned to a *Nurse D* or *C.Nurse D* shift, they cannot be assigned to a *Nurse N* or *C.Nurse E* shift on the same day; $\forall j \in \mathcal{P}$, $\forall d \in \mathcal{D}$: **GC5**: $\widetilde{\mathcal{P}}_1 = \widetilde{\mathcal{P}}_2 = \{j\}$, $\widetilde{\mathcal{S}}_1 = \mathcal{S}^t(\mathcal{S}, \{\textit{Nurse D, C.Nurse E}\}) \cap \mathcal{S}^{sd}(\mathcal{S}, d)$, $\widetilde{\mathcal{S}}_2 = \mathcal{S}^t(\mathcal{S}, \{\textit{Nurse N, C.Nurse E}\}) \cap \mathcal{S}^{sd}(\mathcal{S}, d)$.
7. If someone is assigned to an *Admin* shift, they cannot be assigned to a *Nurse E*, *Nurse N*, *C.Nurse E* or *Doctor N* shift on the same day or on the day before; $\forall j \in \mathcal{P}$, $\forall d \in \mathcal{D}$: **GC5**: $\widetilde{\mathcal{P}}_1 = \widetilde{\mathcal{P}}_2 = \{j\}$, $\widetilde{\mathcal{S}}_1 = \mathcal{S}^t(\mathcal{S}, \{\textit{Admin}\}) \cap \mathcal{S}^{sd}(\mathcal{S}, d)$, $\widetilde{\mathcal{S}}_2 = \mathcal{S}^t(\mathcal{S}, \{\textit{Nurse E, Nurse N, C.Nurse E, Doctor N}\}) \cap \big(\mathcal{S}^{sd}(\mathcal{S}, d-1) \cup \mathcal{S}^{sd}(\mathcal{S}, d)\big)$.
8. No one can be assigned to shifts for more than 6 days in a row; **GC6**: $\widetilde{\mathcal{S}} = \mathcal{S}$, $\widetilde{\mathcal{P}} = \mathcal{P}$, $x = 1$, $y = 6$.
9. No one can be assigned to night/evening shifts for more than 3 days in a row; **GC6**: $\widetilde{\mathcal{S}} = \mathcal{S}^t(\mathcal{S}, \{\textit{Nurse E, Nurse N, Doctor E, Doctor N}\})$, $\widetilde{\mathcal{P}} = \mathcal{P}$, $x = 1$, $y = 3$.
10. If someone has been assigned to shifts for 3-5 days in a row, they must be off 2 days before and after; **GC7**: $\widetilde{\mathcal{S}} = \widetilde{\mathcal{S}}_1 = \widetilde{\mathcal{S}}_2 = \mathcal{S}$, $\widetilde{\mathcal{P}} = \mathcal{P}$, $x = 3$, $y = 5$, $n = m = 2$.
11. If someone has been assigned to shifts for 6 days in a row, they must be off 3 days before and after; **GC7**: $\widetilde{\mathcal{S}} = \widetilde{\mathcal{S}}_1 = \widetilde{\mathcal{S}}_2 = \mathcal{S}$, $\widetilde{\mathcal{P}} = \mathcal{P}$, $x = 6$, $y = 6$, $n = m = 3$.
12. All consecutive days of shift assignments, excluding *Admin* shifts, must be of the same type; **GC8**: $\widetilde{\mathcal{S}} = \mathcal{S} \setminus \mathcal{S}^t(\mathcal{S}, \textit{Admin})$, $\widetilde{\mathcal{P}} = \mathcal{P}$.
13. All workloads for nurses must be within 60% of the expected workload, with respect to to all nurse shifts; **GC9**: $\widetilde{\mathcal{S}} = \mathcal{S}^t(\mathcal{S}, \{\textit{Nurse D}, \textit{Nurse E}, \textit{Nurse N}, \textit{C.Nurse D}, \textit{C.Nurse E}\})$, $\widetilde{\mathcal{P}} = [1\,..\,17]$, $v = 0.6$.
14. All workloads for doctors must be within 60% of the expected workload, with respect to to all doctor shifts; **GC9**: $\widetilde{\mathcal{S}} = \mathcal{S}^t(\mathcal{S}, \{\textit{Doctor D}, \textit{Doctor E}, \textit{Doctor N}, \textit{Doctor S1}, \textit{Doctor S2}\}$, $\widetilde{\mathcal{P}} = [18\,..\,27]$, $v = 0.6$.

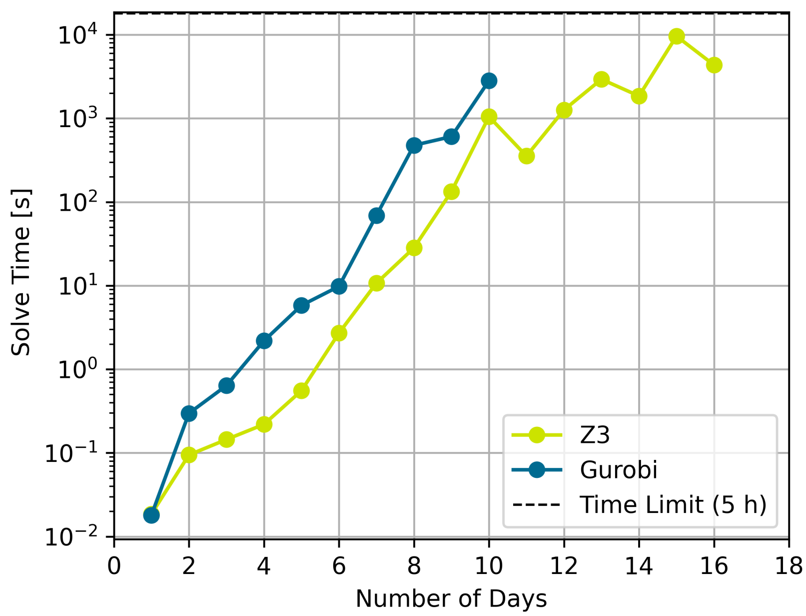


Fig. 4: Solving times for both Z3 and Gurobi over the number of days included in the schedule for Problem B. The number of days ranges from 1–17. If no data point is shown then the time limit of 5 hours was reached.