

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

Alvin Combrink¹, Stephie Do¹, Kristofer Bengtsson², Sabino Francesco Roselli¹, Martin Fabian¹

Abstract—The effects of personnel scheduling on the quality of care and working conditions for healthcare personnel have been thoroughly documented. However, the ever-present demand and large variation of constraints make healthcare scheduling particularly challenging. This problem has been studied for decades, with limited research aimed at applying Satisfiability Modulo Theories (SMT). SMT has gained momentum within the formal verification community in the last decades, leading to the advancement of SMT solvers that have been shown to outperform standard mathematical programming techniques.

In this work, we propose generic constraint formulations that can model a wide range of real-world scheduling constraints. Then, the generic constraints are formulated as SMT and MILP problems and used to compare the respective state-of-the-art solvers, Z3 and Gurobi, on academic and real-world inspired rostering problems. Experimental results show how each solver excels for certain types of problems; the MILP solver generally performs better when the problem is highly constrained or infeasible, while the SMT solver performs better otherwise. On real-world inspired problems containing a more varied set of shifts and personnel, the SMT solver excels. Additionally, it was noted during experimentation that the SMT solver was more sensitive to the way the generic constraints were formulated, requiring careful consideration and experimentation to achieve better performance. We conclude that SMT-based methods present a promising avenue for future research within the domain of personnel scheduling.

I. INTRODUCTION

Personnel scheduling, or *rostering*, is crucial in healthcare. Services depend heavily on human resources scheduled in shifts to meet highly variable and continuous demand. This requires 24/7 coverage while accounting for staff fatigue, workload balancing, matching the many staff qualifications with shift requirements, and so on. These stringent conditions make healthcare scheduling particularly challenging. Evidence shows that effective rostering has an impact on efficiency and workplace conditions, as poor scheduling can impair alertness and cognitive performance [1, 2], and increase risks of self-injury or commuting accidents [2, 3]. However, predictable schedules that consider staff needs can mitigate these risks [4, 5].

There exists a myriad of variants of the *Nurse Rostering Problem* (NRP) [6] and *Physician Scheduling Problem* (PSP) [7], with wide-ranging variations in the number of

shifts, personnel, and constraints. For example, [8] uses the common three shifts per day (Morning, Evening and Night), [9] extends this to four (Day, Early, Late and Night) while [10] includes up to 15 duties per day. Comprehensive accounts of rostering variants and methods can be found in e.g. [7, 11, 12, 13]. The majority of the rostering approaches reviewed in [7] use mathematical programming techniques. However, there is an apparent shift toward combining exact methods with heuristics to leverage performance gains at the cost of optimality guarantees. It is therefore interesting to evaluate if performance gains can be found over standard mathematical programming techniques without losing such guarantees.

Satisfiability Modulo Theories (SMT) has gained momentum within the formal verification community in the last two decades, driving advancements in SMT solvers such as Z3 [14], MathSAT5 [15], and OpenSMT [16]. By extending propositional satisfiability to incorporate decision procedure theories involving predicate logic and quantifiers, SMT benefits from increased expressibility while retaining some of the performance of SAT solvers. The viability of SMT solvers has been demonstrated. Notably, Z3 yielded competitive results [17] compared to the commercial Mixed Integer Linear Program (MILP) solver Gurobi [18] on the job-shop scheduling problem. Similar results were shown on the rotating workforce problem when comparing SMT with two techniques based on network-modeling and constraint programming [19]. *Weighted SMT* was used to address the NRP with both hard and soft constraints by weighting constraints in the objective function [20]. However, no comparisons were made with other methods.

Of the research-developed NRP methods, only around 30% are implemented in practice [21]. A likely explanation for this is the large variability in the types of constraints that a real-world healthcare department places on its rosters, and the difficulty to model these constraints. Many methods build upon too simplified models to be applicable to real-world problems [11]. Formulating *generic constraints* is one way to address this. Perhaps one of the earliest works implementing generic constraints is presented in [22]. More recently, generic constraints for the PSP were formulated with both hard and soft constraints [23]. It was shown that even a relatively few generic constraints could model a moderate range of requirements by applying their model to two different wards.

The contributions of this paper build upon the work in [24] in cooperation with the Queen Silvia Children's Hospital in Gothenburg, Sweden. First, we propose a set of generic constraints that can model a wide range of real-world

This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

¹Division of Systems and Control, Department of Electrical Engineering, Chalmers University of Technology, Göteborg, Sweden {combrink, rsabino, fabian}@chalmers.se

²Research and Technology Development, Group Trucks Operations, Volvo AB Göteborg, Sweden {kristofer.bengtsson@volvo.se}

scheduling constraints. Second, we formulate the generic constraints as both SMT and MILP problems, and perform comparisons of the respective state-of-the-art solvers, Z3 and Gurobi. Experimental results highlight strengths for each solver; Gurobi excels on highly constrained instances near or beyond the border of infeasibility, while Z3 performs better on less constrained instances. In conclusion, which solver to use depends on the underlying problem.

II. PROBLEM DEFINITION

The goal of scheduling in this context is to find an assignment of *staff members* to *shifts* while fulfilling all constraints. Shifts are given, and only one staff member can be assigned to a shift.

Let the *personnel set* $\mathcal{P} = [1..n_p]$ contain the indexes corresponding to each staff member. A staff member $j \in \mathcal{P}$ has a *desired workload* $dw_j \in \mathbb{R}$, which is the desired amount of workload that they are assigned during the scheduling period. Additionally, a staff member has a set of *qualifications* $Q_j \subseteq \mathcal{Q}$, where \mathcal{Q} is the set of all qualifications, e.g. “Administrator” or “Nurse”.

Similarly, let the *shift set* $\mathcal{S} = [1..n_s]$ contain indexes for each shift. A shift $i \in \mathcal{S}$ has a *shift type* t_i representing a category that the shift is included in, e.g. “Day Shift” or “Anesthesiologists Shift”; *start day* $sd_i \in \mathcal{D}$ where $\mathcal{D} = [1..T]$ contains an index for each day of the T -day long scheduling period; a *duration* $d_i \in \mathbb{R}$; a *workload* $w_i \in \mathbb{R}$ representing how burdensome a shift is, which is not necessarily equal to the duration; and a set of *required qualifications* $RQ_i \subseteq \mathcal{Q}$, where staff member $j \in \mathcal{P}$ is qualified for shift $i \in \mathcal{S}$ if $RQ_i \subseteq Q_j$. Let

$$QP_i = \{j \in \mathcal{P} \mid RQ_i \subseteq Q_j\}$$

be the set of *qualified personnel* for shift $i \in \mathcal{S}$.

In many real-world hospital wards, it is permissible for certain shifts which overlap in time to be assigned to the same staff member. This occurs in cases where, for example, the tasks of the shifts are compatible so that one person can handle both, but they are scheduled separately due to qualification requirements or staffing flexibility. We refer to these as *overlapping shifts*. For n shifts, there will be $2^n - n - 1$ number of overlapping combinations. Each combination is designated an index in the *overlapping combinations set* $\mathcal{O} = [1..n_o]$. An *overlapping combination* $k \in \mathcal{O}$ corresponds to a set of overlapping shifts $\mathcal{S}_k \subseteq \mathcal{S}$ and a set of personnel \mathcal{P}_k . Every person in \mathcal{P}_k must be qualified for every shift in \mathcal{S}_k , i.e. $\mathcal{P}_k \subseteq \cap_{i \in \mathcal{S}_k} QP_i$, and are *allowed* to be assigned to all these shifts at once. We distinguish between being *allowed* and being *assigned* to \mathcal{S}_k ; \mathcal{P}_k specifies which staff are allowed to be assigned to \mathcal{S}_k , while constraints determine if and to what degree this is respected. For example, a staff member that is unwilling to work overlapping shifts may not be included in \mathcal{P}_k . However, whether the staff member is assigned to \mathcal{S}_k or not, despite not being allowed, depends on the constraints.

The number of overlapping combinations grows rapidly with the number of overlapping shifts. Therefore, we make

a simplifying assumption that if someone is allowed to be assigned to all shifts in an overlapping combination, then they are allowed to be assigned to any subset of those shifts.

A. Generic Constraints

Generic constraint (GC) formulations take input parameters of the forms $\tilde{\mathcal{P}} \subseteq \mathcal{P}$, $\tilde{\mathcal{S}} \subseteq \mathcal{S}$, $x, y \in \mathbb{N}$, and $u, v \in \mathbb{R}$.

- GC1** The number of shifts in $\tilde{\mathcal{S}}$ that are not assigned at least one person from $\tilde{\mathcal{P}}$ must be greater than or equal to x and less than or equal to y .
- GC2** The number of shifts in $\tilde{\mathcal{S}}$ that have been assigned an unqualified person from $\tilde{\mathcal{P}}$ must be greater than or equal to x and less than or equal to y .
- GC3** The number of times some person $j \in \tilde{\mathcal{P}}$ is assigned to all shifts in a set of overlapping shifts $k \in \mathcal{O}$ for which they are not allowed to be assigned to ($j \notin \mathcal{P}_k$) must be greater than or equal to x and less than or equal to y .
- GC4** For each person in $\tilde{\mathcal{P}}$, the fraction of their assigned workload from shifts in $\tilde{\mathcal{S}}$, if they are assigned to any at all, must be greater than or equal to u and less than or equal to v .
- GC5** If any person in $\tilde{\mathcal{P}}_1$ is assigned to a shift in $\tilde{\mathcal{S}}_1$, then the number of assignments of people in $\tilde{\mathcal{P}}_2$ to shifts in $\tilde{\mathcal{S}}_2$ must be greater than or equal to x and less than or equal to y .
- GC6** The number of consecutive days of shifts in $\tilde{\mathcal{S}}$ assigned to a person in $\tilde{\mathcal{P}}$ must be greater than or equal to x and less than or equal to y .
- GC7** For a person in $\tilde{\mathcal{P}}$ that is assigned to shifts in $\tilde{\mathcal{S}}$ for x to y consecutive days, the n days immediately before must be without assignments to shifts in $\tilde{\mathcal{S}}_1$ and m days immediately after must be without assignments to shifts in $\tilde{\mathcal{S}}_2$.
- GC8** If a staff member in $\tilde{\mathcal{P}}$ is assigned to a shift in $\tilde{\mathcal{S}}$, then, on the day immediately before the assigned shift’s start day, the staff member must either be assigned to a shift in $\tilde{\mathcal{S}}$ with the same shift type or not be assigned to any shift in $\tilde{\mathcal{S}}$.
- GC9** For a staff member in $\tilde{\mathcal{P}}$, the workload that they are assigned to from shifts in $\tilde{\mathcal{S}}$ divided by their desired workload must be within v percent of the expected workload ratio. The expected workload ratio is the total workload of all shifts in $\tilde{\mathcal{S}}$ divided by the total desired workload of all staff members in $\tilde{\mathcal{P}}$.

See Section IV for examples of constraints that these GCs can model. We consider only hard constraints, therefore, this is a constraint satisfaction problem.

III. FORMULATIONS

This section presents the SMT and MILP formulations of GC1–GC3. For brevity, the remaining formulations are available at our repository¹. The goal is to find an assignment to all variables such that all constraints are satisfied.

¹<https://github.com/Adcombrink/NRP-GenericConstraints>

A. SMT Formulation

For each person $j \in \mathcal{P}$ and shift $i \in \mathcal{S}$, the Boolean variable $a_{j,i}$ denotes if j is assigned to i . Every instance of a GC formulation (with specific parameter values) adds constraints to the problem. Therefore, in the following formulations, all parameters should be interpreted as specific for a GC instance. We begin by defining useful notation. The *Boolean to Binary* operator maps *True* $\rightarrow 1$ and *False* $\rightarrow 0$,

$$\oplus \omega = 1 \text{ if } \omega \text{ is true, } 0 \text{ otherwise.}$$

Boolean Summation extends this to count the number of *True* evaluations for a function over a set Ω ,

$$\bigoplus_{\omega \in \Omega} f(\omega) = \sum_{\omega \in \Omega} \oplus f(\omega).$$

We now formulate the constraints. Only one staff member can be assigned to a shift, formulated in (1). **GC1** is formulated in (2) and **GC2** in (3),

$$\left(\bigoplus_{j \in \mathcal{P}} a_{j,i} \right) \leq 1, \quad \forall i \in \mathcal{S}, \quad (1)$$

$$x \leq \left(\bigoplus_{i \in \tilde{\mathcal{S}}} \bigwedge_{j \in \tilde{\mathcal{P}}} \neg a_{j,i} \right) \leq y, \quad (2)$$

$$x \leq \left(\bigoplus_{i \in \tilde{\mathcal{S}}} \bigvee_{j \in \tilde{\mathcal{P}} \setminus QP_i} a_{j,i} \right) \leq y. \quad (3)$$

For **GC3**, we define the set \mathcal{O}_j^{NA} containing every $k \in \mathcal{O}$ with 2-sized \mathcal{S}_k to which staff member j is not allowed to be assigned,

$$\mathcal{O}_j^{NA} = \left\{ k \in \mathcal{O} \mid |\mathcal{S}_k| = 2, j \notin \mathcal{S}_k \right\}, \quad \forall j \in \tilde{\mathcal{P}}. \quad (4)$$

Thus, we constrain the sum over all staff members in $j \in \tilde{\mathcal{P}}$ and overlapping combinations in $k \in \mathcal{O}_j^{NA}$ where j is assigned to all shifts in \mathcal{S}_k :

$$x \leq \sum_{j \in \tilde{\mathcal{P}}} \left(\bigoplus_{k \in \mathcal{O}_j^{NA}} \bigwedge_{i \in \mathcal{S}_k} a_{j,i} \right) \leq y. \quad (5)$$

B. MILP Formulation

The MILP formulation uses 0-1-variables: $a_{j,i} = 1$ if person j is assigned to shift i , 0 otherwise. Besides these, additional auxiliary 0-1-variables δ are used in most GC formulations, which should be interpreted as specific to each GC instance. First, only one person can be assigned to a shift,

$$\sum_{j \in \mathcal{P}} a_{j,i} \leq 1, \quad \forall i \in \mathcal{S}. \quad (6)$$

GC1 is formulated in (8) and uses auxiliary variables $\delta_i, \forall i \in \tilde{\mathcal{S}}$, which are modeled in (7) using the big-M method ($M \geq$

$|\tilde{\mathcal{P}}|$) such that $\delta_i = 1$ if no one in $\tilde{\mathcal{P}}$ is assigned to shift i , 0 otherwise,

$$1 - \delta_i \leq \sum_{j \in \tilde{\mathcal{P}}} a_{j,i} \leq M(1 - \delta_i), \quad \forall i \in \tilde{\mathcal{S}}, \quad (7)$$

$$x \leq \sum_{i \in \tilde{\mathcal{S}}} \delta_i \leq y. \quad (8)$$

GC2, formulated in (10), uses auxiliary variables $\delta_i, \forall i \in \tilde{\mathcal{S}}$, which are modeled in (9) using the big-M method ($M \geq |\tilde{\mathcal{P}}|$) such that $\delta_i = 1$ if at least one unqualified person in $\tilde{\mathcal{P}}$ is assigned to shift i ,

$$\delta_i \leq \sum_{j \in \tilde{\mathcal{P}} \setminus QP_i} a_{j,i} \leq M\delta_i, \quad \forall i \in \tilde{\mathcal{S}}, \quad (9)$$

$$x \leq \sum_{i \in \tilde{\mathcal{S}}} \delta_i \leq y. \quad (10)$$

Finally, **GC3** in (12) uses \mathcal{O}_j^{NA} from (4). Auxiliary variables $\delta_{k,j}, \forall k \in \mathcal{O}_j^{NA}, \forall j \in \tilde{\mathcal{P}}$, are modeled in (11) such that $\delta_{k,j} = 1$ if j is assigned to all shifts in \mathcal{S}_k , 0 otherwise,

$$2\delta_{k,j} \leq \sum_{i \in \mathcal{S}_k} a_{j,i} \leq \delta_{k,j} + 1, \quad \forall k \in \mathcal{O}_j^{NA}, \forall j \in \tilde{\mathcal{P}}, \quad (11)$$

$$x \leq \sum_{j \in \tilde{\mathcal{P}}} \sum_{k \in \mathcal{O}_j^{NA}} \delta_{k,j} \leq y. \quad (12)$$

IV. EXPERIMENTAL EVALUATION

We compare the SMT and MILP formulations using the solvers Z3 (version 4.8.12.0) and Gurobi (version 9.1.2), respectively, on two different types of rostering problems. As Gurobi uses multi-core processing by default, this was enabled for Z3. Apart from that, both solvers use default settings without customizations. All experiments are run on a 2020 MacBook Air, 8-core Apple M1, 16 GB RAM, macOS Monterey 12.0.1. See our repository for additional details¹.

A. Problem A

Problem A is inspired by [8] and [9], which 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 number of shifts 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). In Table II, one person has both qualifications and another has 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 and input parameters used to model them.

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 member types for problem A, with each respective desired workload in hours, and qualifications.

Type	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**: $\tilde{\mathcal{S}} = \mathcal{S}$, $\tilde{\mathcal{P}} = \mathcal{P}$, $x = y = 0$.
2. The 1st staff member cannot be assigned to any shift that starts on days 3–5 of the schedule; **GC1**: $\tilde{\mathcal{S}} = \mathcal{S}^{sd}(\mathcal{S}, \{3, 4, 5\})$, $\tilde{\mathcal{P}} = \{1\}$, $x = y = |\mathcal{S}^{sd}(\mathcal{S}, \{3, 4, 5\})|$.
3. Everyone must be qualified for the shifts that they are assigned to; **GC2**: $\tilde{\mathcal{S}} = \mathcal{S}$, $\tilde{\mathcal{P}} = \mathcal{P}$, $x = y = 0$.
4. No one should be assigned to shifts that overlap in time unless they are allowed to; **GC3**: $\tilde{\mathcal{P}} = \mathcal{P}$, $x = y = 0$.
5. At least 50% of the shifts assigned to the 1st staff member must be of the types *Nurse D1* and *Nurse D2*; **GC4**: $\tilde{\mathcal{S}} = \mathcal{S}^t(\mathcal{S}, \{Nurse D1, Nurse D2\})$, $\tilde{\mathcal{P}} = \{1\}$, $u = 0.5$, $v = 1$.
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**: $\tilde{\mathcal{S}}_1 = \tilde{\mathcal{S}}_2 = \mathcal{S}^{sd}(\mathcal{S}, \{1, 2\})$, $\tilde{\mathcal{P}}_1 = \{0\}$, $\tilde{\mathcal{P}}_2 = \{4, 5\}$, $x = y = 0$.
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**: $\tilde{\mathcal{S}}_1 = \mathcal{S}^{sd}(\mathcal{S}, d) \cap \mathcal{S}^t(\mathcal{S}, \{Nurse D1, Nurse D2\})$, $\tilde{\mathcal{S}}_2 = \mathcal{S}^{sd}(\mathcal{S}, d) \cap \mathcal{S}^t(\mathcal{S}, \{Nurse N2, Nurse N2\})$, $\tilde{\mathcal{P}}_1 = \tilde{\mathcal{P}}_2 = \{j\}$, $x = y = 0$.
8. No one can be assigned to shifts for more than 6 days in a row; **GC6**: $\tilde{\mathcal{S}} = \mathcal{S}$, $\tilde{\mathcal{P}} = \mathcal{P}$, $x = 0$, $y = 6$.
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**: $\tilde{\mathcal{S}} = \tilde{\mathcal{S}}_1 = \tilde{\mathcal{S}}_2 = \mathcal{S}$, $\tilde{\mathcal{P}} = \mathcal{P}$, $x = 4$, $y = 6$, $n = m = 3$.
10. All consecutive days of shift assignments must be of the same type; **GC8**: $\tilde{\mathcal{S}} = \mathcal{S}$, $\tilde{\mathcal{P}} = \mathcal{P}$.
11. All personnel workloads must be within 30% of the expected workload, with regards to all shifts; **GC9**: $\tilde{\mathcal{S}} = \mathcal{S}$, $\tilde{\mathcal{P}} = \mathcal{P}$, $v = 0.3$.

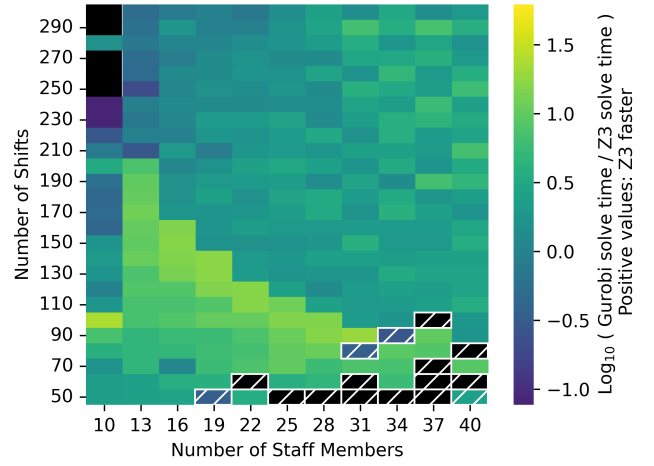


Fig. 1: A comparison of the solving times between Z3 and Gurobi for each problem instance. Black means Z3 was not able to find a solution within the time limit while Gurobi could. White hatching means the problem is infeasible.

Instances of the problem are created for the number of shifts ranging from 50–300, equivalent to approximately 8–50 scheduled days, 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.

Figure 1 compares solver performance using the logarithmic quotient $\log_{10}(\text{Gurobi time} / \text{Z3 time})$ across varying numbers of staff members and shifts. Positive values indicate Z3 is faster, while negative values indicate Gurobi is faster. Black regions represent cases where Z3 failed to terminate within the time limit but Gurobi succeeded. For no instances did Z3 terminate within the time limit but not Gurobi. White hatching denotes infeasible instances. Figure 2 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). It was found that relaxing constraint 11, which bounds assigned workloads to near desired workloads, made all infeasible problems feasible and lead to both solvers terminating within the time limit. 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 instances which are more tightly constrained by constraint 11. In the remaining areas, such as near the center where the problems are not as constrained, Z3 generally finds solutions faster than Gurobi.

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 [24].

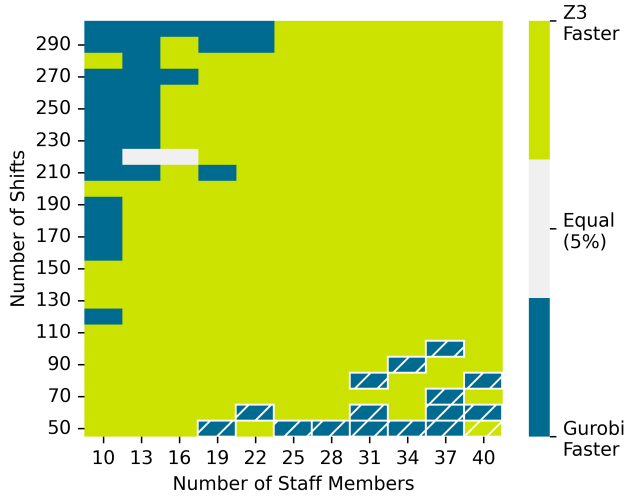


Fig. 2: A comparison between Z3 and Gurobi based on solving times for each problem instance. White hatching means the instance is infeasible.

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

The staff set is constant, containing 29 staff members with a mix of qualifications and desired workloads (Table IV). The number of desired days in the schedule 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.

The constraints are given in Table VI. Parameter values are deferred to our repository¹. *Nurse D-C.Nurse D*, Staff members are allowed to be assigned to any of the following shift type pairs simultaneously: *Nurse E-C.Nurse E*, *Doctor D-Doctor S1*, *Doctor D-Doctor S2*, and *C.Nurse D-Admin*.

Figure 3 shows the solving times for each solver over 1–17 days included in the scheduling problem, with an arbitrary time limit of 5 hours. Z3 is able to find a solution faster than Gurobi in all instances. None of the solvers were able to find solutions to problems including more than 16 days within the time limit.

V. DISCUSSIONS & CONCLUSIONS

The GCs were formulated to maximize each respective solver’s performance. In this regard, Z3 showed more sensitivity than Gurobi. Integer-only formulations for GC4 and

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**.
2. Everyone must be qualified for the shifts that they are assigned to; **GC2**.
3. No one should be assigned to overlapping shifts that they are not allowed to work at the same time; **GC3**.
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**.
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; **GC5**.
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; **GC5**.
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; **GC5**.
8. No one can be assigned to shifts for more than 6 days in a row; **GC6**.
9. No one can be assigned to night/evening shifts for more than 3 days in a row; **GC6**.
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**.
11. If someone has been assigned to shifts for 6 days in a row, they must be off 3 days before and after; **GC7**.
12. All consecutive days of shift assignments, excluding *Admin* shifts, must be of the same type; **GC8**.
13. All workloads for nurses must be within 60% of the expected workload, with respect to to all nurse shifts; **GC9**.
14. All workloads for doctors must be within 60% of the expected workload, with respect to to all doctor shifts; **GC9**.

GC9 (achieved by scaling and rounding) lead to significant improvements on solving times for Z3 while negligibly affecting Gurobi. This suggests that Z3 leverages efficient integer-based solution strategies when available. Moreover, Z3 showed marked sensitivity to implementation details, with explicit arithmetic expressions sometimes outperforming built-in functions despite their high-level equivalence. These performance variations highlight that, when using Z3, careful consideration of the constraint implementation is essential. For practitioners, this implies that some experimentation with alternative formulations may be necessary to achieve better performance, particularly for complex scheduling problems like those encountered in healthcare environments.

Although this work focuses exclusively on hard constraints, real-world schedules rely heavily on soft constraints. As a next step, adopting the approach in [24] by weighting GCs would introduce soft constraints while leveraging their

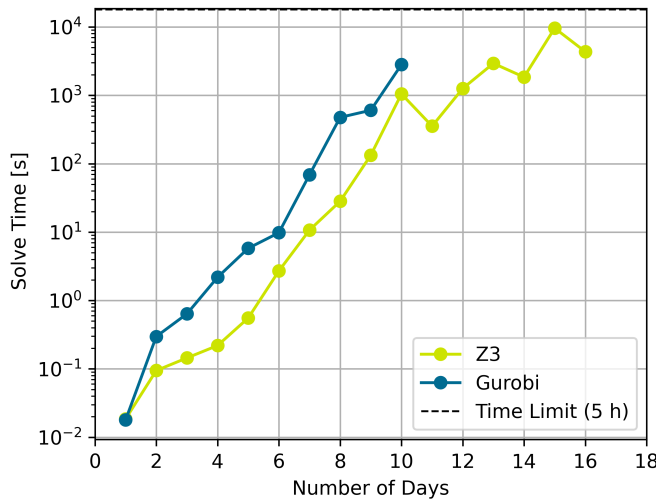


Fig. 3: Solving times for both Z3 and Gurobi over the number of days included in the schedule for Problem B. The number of days ranges from 1–17. If no data point is shown then the time limit of 5 hours was reached.

expressiveness. Furthermore, hybrid approaches which take advantage of in-domain knowledge, heuristics, local search techniques, etc., offer several promising directions toward improved performance and scalability.

To conclude, this article presented a set of generic constraints that can model a wide range of rostering constraints. Furthermore, SMT and MILP formulations were made and used to model two sets of problems, one based on those found in the literature and one based on a real-world setting. A comparison of the state-of-the-art SMT and MILP solvers Z3 and Gurobi, respectively, was then done on the problem sets. Results showed how each solver’s performance depends on the problem; Gurobi appears to excel on infeasible or highly constrained problems, while Z3 performs better on more relaxed problems. Therefore, the specific problem to solve determines which solver and underlying framework (SMT/MILP) is more suitable. Connecting back to this article’s original aim of exploring SMT for the nurse rostering problem, we argue that SMT offers a promising direction for future research. Z3’s faster solving times on the real-world based problem B in Section IV-B suggest that SMT-based methods could provide value to real-world healthcare service providers, and further the work toward more sustainable and efficient healthcare.

REFERENCES

- [1] Saranea Ganesan et al. “The impact of shift work on sleep, alertness and performance in healthcare workers”. In: *Scientific Reports* 9.1 (2019), pp. 1–13.
- [2] Shantha MW Rajaratnam, Mark E Howard, and Ronald R Grunstein. “Sleep loss and circadian disruption in shift work: health burden and management”. In: *Medical Journal of Australia* 199 (2013), S11–S15.
- [3] Laura K Barger et al. “Neurobehavioral, health, and safety consequences associated with shift work in safety-sensitive professions”. In: *Current Neurology and Neuroscience reports* 9.2 (2009), pp. 155–164.
- [4] Joan Williams et al. “Stable scheduling study: Health outcomes report”. In: *Available at SSRN 4019693* (2022).
- [5] Janet L Wilson. “The impact of shift patterns on healthcare professionals”. In: *Journal of Nursing Management* 10.4 (2002), pp. 211–219.
- [6] Brenda Cheang et al. “Nurse rostering problems—a bibliographic survey”. In: *European Journal of Operational Research* 151.3 (2003), pp. 447–460.
- [7] Melanie Erhard et al. “State of the art in physician scheduling”. In: *European Journal of Operational Research* 265.1 (2018), pp. 1–18.
- [8] Haroldo G Santos et al. “Integer programming techniques for the nurse rostering problem”. In: *Annals of Operations Research* 239.1 (2016), pp. 225–251.
- [9] 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.
- [10] Edmund Burke et al. “A memetic approach to the nurse rostering problem”. In: *Applied Intelligence* 15.3 (2001), pp. 199–214.
- [11] Edmund K Burke et al. “The state of the art of nurse rostering”. In: *Journal of Scheduling* 7.6 (2004), pp. 441–499.
- [12] Andreas T Ernst et al. “Staff scheduling and rostering: A review of applications, methods and models”. In: *European Journal of Operational Research* 153.1 (2004), pp. 3–27.
- [13] Jorne Van den Bergh et al. “Personnel scheduling: A literature review”. In: *European Journal of Operational Research* 226.3 (2013), pp. 367–385.
- [14] Leonardo De Moura and Nikolaj Bjørner. “Z3: An efficient SMT solver”. In: *Tools and Algorithms for the Construction and Analysis of Systems*. Ed. by C. R. Ramakrishnan and Jakob Rehof. Springer, 2008, pp. 337–340.
- [15] Alessandro Cimatti et al. “The MathSAT5 SMT solver”. In: *Tools and Algorithms for the Construction and Analysis of Systems*. Ed. by Nir Piterman and Scott A. Smolka. Springer, 2013, pp. 93–107.
- [16] Roberto Bruttomesso et al. “The OpenSMT solver”. In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2010, pp. 150–153.
- [17] Sabino Francesco Roselli, Kristofer Bengtsson, and Knut Åkesson. “SMT solvers for job-shop scheduling problems: Models comparison and performance evaluation”. In: *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 547–552.
- [18] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2024. URL: <https://www.gurobi.com>.
- [19] Christoph Erking and Nysret Musliu. “Personnel Scheduling as Satisfiability Modulo Theories.” In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 2017, pp. 614–621.
- [20] Carlos Ansótegui et al. “Solving weighted CSPs with meta-constraints by reformulation into satisfiability modulo theories”. In: *Constraints* 18.2 (2013), pp. 236–268.
- [21] Deborah L Kellogg and Steven Walczak. “Nurse scheduling: from academia to implementation or not?” In: *Interfaces* 37.4 (2007), pp. 355–369.
- [22] Mihoko Okada. “An approach to the generalized nurse scheduling problem-Generation of a declarative program to represent institution-specific knowledge”. In: *Computers and Biomedical Research* 25.5 (1992), pp. 417–434.
- [23] Louis-Martin Rousseau, Gilles Pesant, and Michel Gendreau. “A general approach to the physician rostering problem”. In: *Annals of Operations Research* 115.1 (2002), pp. 193–205.
- [24] Alvin Combrink and Stephanie Do. “Automatic shift scheduling for healthcare personnel using satisfiability modulo theory”. MA thesis. Electrical Engineering, 2021.