Solving a physician rostering problem

Tatiana C. Meister, Toni I. Wickert, Luciana S. Buriol
Institute of Informatics, Federal University of Rio Grande do Sul - UFRGS
Porto Alegre, Brazil
tcmeister@inf.ufrgs.br, tiwickert@inf.ufrgs.br, buriol@inf.ufrgs.br

***Abstract*—Scheduling the activities for a group of physicians in a hospital is a recurring activity that impacts both the health-related costs and the quality of services provided by these professionals. Mismanagement of scheduling could lead to the use of excessive overtime to achieve health demands, increasing the costs. Moreover, physicians assigned to a stressful task over long periods may result in a reduction in their performance and cause a sense of injustice towards other physicians.**

The use of algorithms makes scheduling more agile and with better results when compared to the manual execution. This work addresses a physician rostering problem applied to the Hospital de Clínicas de Porto Alegre (HCPA), in Brazil. Instances were generated based on data provided by the hospital and approached employing the Coin-OR (CBC) solver and a VNS heuristic. Both strategies were selected to evaluate the performance of an exact integer programming and a heuristic in different instances, classified by demand and availability of physicians.

***Index Terms*—Physician rostering problem, Heuristics, VNS, CBC**

I. Introduction

Scheduling in hospitals consists of defining which shifts and areas each professional will work over a planning horizon, such as a week or a month. The automatic generation of the timetables for physicians is a recurring task that has gained attention in recent years, due to the increased demand and costs of health services [1]. Unlike the nurse scheduling, the assignment of tasks for physicians takes into account, in addition to the legal aspects and employment contracts, a series of individual agreements that these employees have with their working hospitals [2].

Defining a schedule, in most hospitals, is still a manual activity performed by professionals with more experience, occupying a good part of their time in an activity not directly related to their specialty [3]. Moreover, in many cases, manual scheduling generates poor results due to the exponential number of assignments among physicians to be analyzed.

The use of automated tools to generate the timetables allows professionals to be released to work directly in their specialty, also providing better quality solutions. This work aims to analyze the use of an exact method against a heuristic algorithm for the physician rostering problem applied to Hospital de Clínicas de Porto Alegre (HCPA), in Brazil, for different types of instances.

The remainder of the paper is organized as follows. Section II presents the literature review about the problem. Section III defines the physician rostering problem with its constraints and its input data. Section V shows the characteristics of the instances based on the HCPA, as the four different groups considered and the implemented generator. Section IV presents the aspects concerning the VNS heuristic, like the construction of the initial solution and the details regarding the implemented neighborhoods. Section VI shows the computational results and the Section VII concludes the analyses.

II. Literature Review

The work of [5] presents a review of 60 relevant publications about physician scheduling, evaluating common characteristics, and analyzing the increased research over the decades. The authors divided the problem into three categories:

- *Workforce problems:* those that are looking for an optimal number of employees, taking into account future demands and types of service prioritization strategies;
- *Allocation problems:* those that aim to identify the optimal assignments for the physicians in a planning horizon;
- *Rescheduling problems:* those that deal with unforeseen day-to-day absences, requiring a repair strategy to restore the feasibility of an already existing solution.

The authors of [6] focus on the optimization of the use of human resources due to the increased demand for health services without the proper growth of resources. The scheduling problem uses characteristics of the surgical area at the "Spedali Civili di Brescia" hospital, in northern Italy, and two mathematical formulations are proposed for the reconstruction phase of an Adaptive Large Neighborhood Search (ALNS) heuristic. The computational experiments employed instances based on real cases provided by the hospital with a planning horizon of 15 days. The results of the ALNS heuristic was compared with a model solved by Gurobi solver, demonstrating better performance compared against the heuristic.

The work of [7] provides a flexible mixed-integer linear formulation allowing the user to determine, for each constraint, which will be used and what is the impact in the final cost of the solution. For the tests, the authors used the CPLEX solver, applied to the case of the Hematology Center of the University Hospital in Rome, Italy. The results were compared with the manual scheduling done by the hospital, taking into account a planning horizon of four months.

The study conducted by [8] implements an integer linear formulation with execution by CPLEX solver. The main concern is to retain professionals through an approach that gives greater importance to the quality of work for physicians. The authors use the current case of a Singaporean government hospital as

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

a basis for the definition of constraints and the generation of instances.

[8] proposed three mathematical models that complement each other. The first model seeks to minimize the number of shifts and areas that have not reached the minimum demand of physicians. The second model aims to reduce the difference between the scheduling of the first model and the optimal scheduling of each physician, based on their preferences and workload. Finally, the third model seeks to balance the workload of the physicians. For small instances, the combination of the mathematical models solved by CPLEX achieved the optimal solutions. However, a local search heuristic was implemented and used to reach feasible solutions within an acceptable time limit for larger instances.

The work [9] addressed the reduction of the patient's care fragmentation, which tries to prevent more than one physician from following the patient during their stay at the hospital. The research was inspired by a real-case from the Auckland hospital in New Zealand and proposed a mathematical model with the constraints based on this hospital. Most of the publications about physician scheduling give priority for the hospital demands to each area and day shift, without distinguishing the treatment of the patients.

[10] presents a physician scheduling problem based on the constraints of five different hospitals in Montreal, Canada. To solve the problem, the authors applied aa tabu search and an integer programming. The constraints are divided into four categories:

- *Supply and demand constraints:*, which addresses hospital requirements in each area;
- *Workload constraints:* evaluate the number of worked hours by each professional according to their employment contract;
- *Fairness constraints:* seek to equalize the assignments among all physicians;
- *Ergonomic constraints:* seek to increase the well-being at work, by reducing allocations that are non-preferential.

The authors of [12] present the application of an integer formulation, using the CPLEX solver. The problem considers a planning horizon of one month and the characteristics of the anesthesiology department of a hospital in Berlin, Germany. The scheduling process at the hospital, which was done manually, was replaced by the mathematical model with the aid of tools that allowed daily adjustments in the assignments generated. The quality of a solution is focused on overtime reduction, physician assignment preferences, and fairness criteria.

The problem of the Hospital de Clínicas de Porto Alegre (HCPA) has already been studied by [11], where an integer programming formulation was proposed and solved by CPLEX and a Fix and Optimize Matheuristic (F&O). The MIP solver solved to optimality small-sized instances in a few seconds, while for large-sized ones a heuristic method generated the best results.

The same characteristics and instances of the problem of [11] were used by [13]. The authors proposed a Late Acceptance Hill Climbing heuristic (LAHC). The results present similar results to the F&O heuristic, with slightly better quality for the large-sized instances.

The physician rostering problem presented in this paper can be classified as an allocation problem since there is no intention to identify an optimal staff, nor to deal with rescheduling. As an allocation problem, the objective is to identify an optimal group of assignments with the existing medical staff.

We continue the analysis of the HCPA problem with an updated version of the constraints. An instance generator based on the real characteristics of the referred hospital that allows users to change parameters according to their necessity of analysis is implemented and used for the evaluating of the results.

III. Physician Rostering Problem Definition

The physician rostering problem consists of determining to which shifts and areas a group of physicians will perform over a planning horizon, respecting a set of constraints. The constraints are divided into two groups:

- *Hard constraints:* must be respected. Otherwise, the solution will turn invalid;
- *Soft constraints:* are desirable to be obeyed. When violated, a cost is added to the objective function.

The constraints may change according to the hospital, given different characteristics among the hospitals. Such as its policies, the labor laws of the country, the employment contracts, and the preferences of the physicians. The objective of the problem is to find the assignments for all physicians considering a planning horizon of one month. A shift and a hospital area define an assignment for a physician on a day. The allocation suffers different penalties according to the day of the week, preferences of each physician and hospital policies.

An instance of the problem contains the following information:

- A set of holidays that must be considered as non-business days.
- A set of areas with a minimum and a maximum number of physicians for each day and shift.
- A set of physicians, where for each physician are defined:
  - A monthly workload;
  - An ideal number of hours to be scheduled on non-business days;
  - An indication of which hospital areas the physician is allowed to work;
  - A set of non-preferential areas;
  - A set of fixed assignments, determining the day, shift and area to work;
  - A set of day/shifts that the physician must not be assigned due to occasional absences or vacation;
  - A set of non-preferential days/shifts.

Considering an instance with two days to be scheduled, a hospital with ten physicians and two areas, Table II shows

TABLE I
EXAMPLE OF A PHYSICIAN SCHEDULING (INFORMATION)

Shift	Area	Day 1		Day 2	
		Min	Max	Min	Max
Early	A1	2	3	2	2
Late	A1	1	2	1	2
Night	A1	1	1	1	1
Early	A2	2	3	2	2
Late	A2	1	2	1	2
Night	A2	1	1	1	1

TABLE II
EXAMPLE OF A PHYSICIAN SCHEDULING (FINAL SCHEDULE)

iPhys	Day 1			Day 2		
	Early	Late	Night	Early	Late	Night
P1	A1	-	-	A1	-	-
P2	A1	-	-	-	A1	-
P3	-	A2	-	A2	-	-
P4	-	-	A2	-	-	A2
P5	A2	-	-	A2	-	-
P6	A2	-	-	-	A2	-
P7	-	A1	-	A1	-	-
P8	-	-	A1	-	-	A1

a valid schedule according to the hard constraints described below. The physicians $P1$-$P6$ have permission in both areas, while the physicians $P7$-$P8$ have permission only in area $A1$. The minimum and maximum demand per day/shift/area are demonstrated in Table I. For both tables, the columns Min and Max represents the minimum and the maximum demand of physicians in the day/shift/area, respectively. The columns $Early$, $Late$ and $Night$ represent the area assigned to this day/shift for each physician in the early, late, and night shifts, respectively.

The *hard constraints* are described below, and all must be obeyed. Otherwise, the solution becomes infeasible. In the version of 2018, presented by [13], the hard constraint $H5$ did not exist, and $H4$ had only considered all-day absences.

- **H1 – Minimum demand**
  For each day/shift/area, the number of assigned physicians must be greater or equal to the minimum demand.
- **H2 – Maximum demand**
  For each day/shift/area, the number of assigned physicians must be less or equal to the maximum demand.
- **H3 - Authorization in the areas**
  Physicians cannot be assigned to areas where they do not have permission.
- **H4 - Absences**
  Physicians cannot be assigned to a hospital area on a shift of the day when they are absent.
- **H5 - Fixed assignments**
  Physicians must be assigned to a day/shift/area in which they have a fixed assignment.
- **H6 - Shifts on working days**
  On a business day, physicians can only be assigned one shift (early, late, or evening).
- **H7 - Shifts and areas on non-business days**
  On a non-business day (Saturdays, Sundays, and holidays), a physician can only be assigned on the night shift or both day shifts (early and late) and in the same area.
- **H8 - Invalid shift successions**
  Physicians assigned in the night shift cannot be assigned in the early or late shifts of the immediately following day.

The quality of a solution - its cost - is directly related to the SOFT CONSTRAINTS violations described below. Each soft constraint has an associated penalty weight that allows sorting them by priority. All the soft constraints are evaluated simultaneously, which means that more than one may be penalized the same assignment for a physician.

- **S1 - Minimum working hours**
  This penalty occurs when the number of assigned hours is below the physician's workload.
- **S2 - Maximum working hours**
  This penalty occurs when the number of assigned hours is above the physician's workload.
- **S3 - Minimum working hours on non-business days**
  This penalty occurs when the number of assigned hours on non-business days is different between day shifts (early and late) and night shifts.
- **S4 - Maximum working hour on non-business days**
  This penalty occurs when the number of assigned hours on non-business days is above the ideal hours.
- **S5 - Balancing day and night working hours on non-business days**
  This penalty occurs when the number of assigned hours on non-business days is different between day shifts (early and late) and night shifts.
- **S6 - Incomplete weekend**
  This penalty occurs when the physician is assigned on only one day of the weekend (Saturday or Sunday, but not both).
- **S7 - Maximum number of working weekends**
  This penalty is applied when the number of working weekends exceeds two. For a weekend to be considered as worked, at least one of the days (Saturday or Sunday) must be assigned to a shift and a hospital area.
- **S8 - Maximum number of consecutive night assigned**
  This penalty occurs when the physician has more than three consecutive assignments in night shifts.
- **S9 - Assignment in non-preferential area**
  This penalty happens when the physician is assigned to a non-preferential area, independently of the day/shift.
- **S10 - Assignment in non-preferential day/shift**
  This penalty occurs when the physician is assigned to a non-preferential day/shift, regardless of the area.

Considering the example presented in Table I, where both days are business days, and the additional information from Table II. The solution is shown in Table II will be penalized as described below:

TABLE III
EXAMPLE OF PHYSICIAN SCHEDULING (INFORMATION)

iPhys	Monthly Hours	Non-Preferential	
		Day/Shift	Areas
P1	18	-	-
P2	18	-	A1
P3	18	-	-
P4	18	Day 1/Night Day 2/Night	-
P5	12	-	-
P6	12	-	-
P7	12	-	-
P8	12	-	-

- As early and late shifts have 6 hours and night shift has 12 hours, the physicians $P4$ and $P8$ were assigned above their monthly hours - 6 and 12 hours more, respectively. The penalization will be the product of the exceeded number of hours by the weight of the soft constraint $S2$.
- As the physician $P2$ has non-preference for working on area $A1$, both assignments are non-preferential.
  The penalization will be the product of the number of non-preferential assignments (2) by the weight of the soft constraint $S9$.
- As the physician $P4$ has non-preference for night shifts, regardless of the day, both assignments are non-preferential.
  The penalization will be the product of the number of non-preferential assignments (2) by the weight of the soft constraint $S10$.

The previous version [13] did not consider the soft constraints about the assigned hours on non-business days - $S3$, $S4$ and $S5$ - and non-preferential areas - $S9$. In contrast, a soft constraint counting the maximum consecutive working days, which was called $S2$, is not considered anymore. According to the authors, the removed soft constraint was one of the main impacts on the final cost of the solutions.

IV. Proposed VNS

This section presents a Variable Neighborhood Search (VNS) heuristic to approach the physician rostering problem. The heuristic uses an initial solution generated by a truncated Branch & Bound (TB&B) algorithm with an improvement phase focused on the constraints with higher penalization.

A. Initial Solution

The initial solution is generated in two phases. First, a truncated Branch & Bound (TB&B) algorithm builds a feasible solution, considering all the hard constraints of the problem. Afterward, the generated solution is improved by a local search to reduce its cost.

The TB&B creates a solution through a choice tree, where each node represents an assignment or a day off for one day and one physician. Once the assignments and day offs are defined for all days and physicians, the solution is complete. The building process of a solution proceeds as follows. First, a shift and an area must be assigned to each physician on the first day of the month. The same occurs with the second day, and so on, until the last day of the month is reached.

Choosing the daily task - an assignment or a day off - for a physician may negatively influence the performance of the TB&B. When a daily task is defined without taking into account the demand and availability of physicians in each shift/area of the day, the number of infeasible solutions covered grows, causing an increase in the runtime of the algorithm.

A mechanism was implemented to define an order to process the available daily tasks for the physicians, and avoid to cover the majority of the infeasible solutions in advantage. The process of ordering gives priority to the shift/area where there is a shortage of physicians available to attend the minimum demand, using as a tiebreaker the lowest cost associated with the assignment.

After the TB&B has found a first feasible solution, a local search is performed using a set of neighborhood types, which are focused on the soft constraints of the problem. The soft constraints considered in this phase was made by analyzing the associated cost in the generated solutions in the first stage, selecting those with the higher penalties.

The improvement phase takes the TB&B solution as a pivot and updates it every time that a neighbor with a lower cost is found. If no neighbor is found, the local search uses the next neighborhood until there is no more type available.

The neighborhood types are focused on the following constraints and exchange a daily task of a physician considering different rules to select the old and the new one.

- **S1 - Minimum working hours in a month**: the selection of the current daily task takes into account:
  - Physicians whose time balance is below the workload in their contract and;
  - Days when the physician has a day off.

  The neighbor exchanges the day off for an assignment with an area and a shift, increasing the number of working hours.
- **S2 - Maximum working hours in a month**: the choice of the current daily task takes into account:
  - Physicians whose time balance is above the workload in their contract and;
  - Days when the physician has an assignment to a shift and an area.

  The neighbor exchanges the assignment for a day off, decreasing the number of working hours.
- **S3 - Minimum working hours on non-business days**: the selection of the current daily task takes into account:
  - Physicians whose number of hours assigned to non-business days is below the ideal in their contract and;
  - Non-business days when the physician has a day off.

The neighbor exchanges the day off for an assignment with an area and a shift, increasing the number of working hours on the non-business days.

- **S4 - Maximum working hours on non-business days**: the choice of the current daily task takes into account:
  - Physicians whose number of hours assigned to non-business days is above the ideal in their contract and;
  - Non-business days when the physician has an assignment to a shift and an area.

  The neighbor exchanges the assignment for a day off, decreasing the number of hours assigned on the non-business days.
- **S5 - Balancing day and night working hours on non-business days**: the selection of the current daily task takes into account:
  - Physicians whose number of hours assigned as day shift (early and late) on non-business days is different from the number of hours assigned as night shift on non-business days and;
  - Non-business days when the physician was assigned in the shift with the highest number hours.

  The neighbor switches the shift to the opposite shift with fewer hours, reducing the difference.

Computational experiments demonstrated that an initial solution that respects all the hard constraints and has a lower cost improves the quality of the VNS final cost. However, this part must be fast so that it does not negatively affect the performance of the heuristic. Table IV shows the average runtime, in seconds, for all instances between the two phases implemented for the initial solution.

TABLE IV
INITIAL SOLUTION - RUNTIME

Instance	Time (sec)	
	TB&B	**Improv**
I_LOW_050	0.01	0.07
I_LOW_100	0.01	0.08
I_LOW_150	0.01	0.07
I_HIGH_050	0.00	0.05
I_HIGH_100	0.02	0.32
I_HIGH_150	0.02	0.70

B. VNS Heuristic

A VNS heuristic was implemented to solve this version of the physician rostering problem applied to the HCPA. The VNS uses three neighborhood types, as follows:

- *CHANGE*: the daily task of one physician is changed, considering all physicians and days of the month. The neighbor exchanges the current daily task of the pivot solution for another valid assignment or a day off.
- *SWAP*: the daily tasks of two physicians are swapped, considering that the inversion is valid for both.
- *CHAIN*: the daily tasks of three physicians are reversed, considering that the inversion is valid for all physicians. The first physician receives the daily task from the second, while the second physician takes from the third, and the third physician receives from the first.

Algorithm 1: VNS

```
Data:
- timeExec: current execution time;
- timeLimit: execution timeout;
- pivoSol: best solution found;
- candSol: candidate solution;
- iterNImp: iterations without cost update;
- maxIterNImp: maximum number of iterations without
cost update;
- kN: current neighborhood index;
1  pivoSol = initialSolF1_TB&B();
2  pivoSol = initialSolF2_BL(pivoSol);
3  while (timeExec < timeLimit and iterNImp <
     maxIterNImp) do
4      int kN = 1;
5      while (timeExec < timeLimit and kN <= 3) do
6          candSol = searchCand(pivoSol,kN);
7          if candSol.cost < pivoSol.cost then
8              pivoSol = candSol;
9              kN = 1;
10             iterNImp = 0;
11         else
12             kN++;
13             iterNImp++;
14         end
15     end
16     pivoSol = shake(pivoSol);
17 end
```

Algorithm 1 shows the pseudo-code of the implementation of the VNS heuristic. The algorithm begins with the construction of the initial solution in lines 1 and 2. Then, a candidate solution is selected from the neighborhood kN of the pivot, as shown in line 6. If the candidate has a better cost then the pivot (lines 7-11), the pivot is updated, and the index of the current neighborhood is reset. Otherwise, the current neighborhood and the number of iterations without improvement are increased (lines 11-14). When all neighborhoods are used, the pivot solution passes by a shake-movement in line 16.

The candidate solution is selected from the current neighborhood and passes for the best improvement local search. All change possibilities are evaluated, considering all physicians and days of the month, selecting the one with the lower cost. If more than one solution with the same lower cost exists, a random choice among all of them is applied.

The best improvement local search uses auxiliary structures that allow evaluating each soft constraint without being necessary to change the solution. During the comparison cost between the candidate and the pivot, is not taken into account the soft constraint $S8$, due to the high time-consuming. However, when the pivot solution is changed, the cost of this soft constraint is calculated.

The shake-movement changes the daily task of each physician of a random day of the month, not taking into account the hard constraints $H1$, $H2$, or $H8$. This movement may turn the solution infeasible, however, helps to change the search scope. When a hard constraint is disobeyed, the associated penalty is 1000 times superior to the soft constraints weights. Therefore, even starting the new loop with an invalid solution, as the search is focused on the lower cost, in a few iterations is found a feasible solution.

V. Instance Generator

We implemented a random instance generator using Java language, considering the current real characteristics of the HCPA physician problem The generator provides a group of instances based on the relative percentage of physicians required per day/shift/area and the probability of a day/shift to be a fixed assignment or an absence for a random physician.

For the analysis, we focused on the impact of the relative percentage of demand for physicians per day/shift/area (DSA), which represents the approximation of the number required to the maximum supported so that the solution is still feasible.

$$maxSupDSA = (\#Phys/(\#Area * \#Shift)) * \alpha \quad (1)$$

Equation (1) represents the maximum number of physicians ($maxSupDSA$) that a DSA could require so that the solution is still feasible. A minimum demand above this number will turn the instance infeasible. The terms $\#Phys$ and $\#Area$ represent the number of physicians and areas of the instance, respectively. $\#Shift$ represents the number of shifts on a day (this number is two for non-business days and three for business days). α represents the percentage of approximation of the number of physicians required to the maximum, which varied to the different categories of instances generated.

TABLE V
Group of instances

ID	Demand (α)	#Phys	#Area	%Fix	%Abs
I_LOW_050	Low (10%)	050	3	60	30
I_LOW_100	Low (10%)	100	3	60	30
I_LOW_150	Low (10%)	150	3	60	30
I_HIGH_050	High (80%)	050	3	00	10
I_HIGH_100	High (80%)	100	3	00	10
I_HIGH_150	High (80%)	150	3	00	10

Table V demonstrates the characteristics of the generated instances, where $\#Phys$ and $\#Area$ represent the number of physicians and areas, respectively, and $\%Fix$ and $\%Abs$ indicates the probability of a physician has a fixed assignment and an absence on a day/shift, respectively.

The instances are divided into two groups: low demand with $\alpha = 10\%$ and high demand with $\alpha = 80\%$. For each group, three similar instances were generated with the same number of physicians. Other information concerning the input data is described below.

- *Contractual hours:*
  - 80% of the physicians are full-time, with 200 hours of workload and the ideal of 48 hours on non-business days;
  - 20% of the physicians are part-time, with 150 hours of workload and the ideal of 24 hours on non-business days;
- *Non-preferential areas:* 30% probability that a physician will not have a preference in an allowed area.
- *Non-preferential day/shifts:* 30% probability that a physician will not have a preference in a day/shift.

The random instance generator and the instances used for this work are available on the site https://github.com/taticm13/Physician-Rostering-Problem-HCPA.git.

VI. Computational Results

This section presents the results and the analysis performed. The tests were executed on an Intel Core i7 computer with 16 GB of RAM and 2.20 GHz, with a time limit of 5 minutes to VNS and 60 minutes to CBC. The heuristic ran each instance 5 times, and the results presented are the average.

A. Initial Solution Analysis

Table VI shows a comparison between the first phase of building the initial solution (TB&B) against the second phase (improvement) for all the instances. For each stage is demonstrated the final cost found and the GAP (%) of the solution compared with the lower bound provided by CBC solver.

On average, the second phase improved 8.3% the GAP of the low demand group, while it enhanced 19.4% the GAP of the high demand. Other experiments were conducted without the second phase, resulting in worse quality solutions. Due to the low time consumption, both phases were maintained for the rest of the tests.

B. Comparison between VNS and CBC

Table VII shows a quality comparison between the VNS and CBC for all instances. Each line of the table contains the name of the instance, the final cost found, and the GAP. The GAP of a solution compares the lower bound (LB) of CBC solver with the solution cost found.

The group with low demand had smaller GAPs when the CBC solver was used. However, the difference in the GAP between CBC and VNS is below than 1.4%, while the time variation is about 55 minutes more to CBC. As the exact method did not prove the optimality of the solution and takes considerably more execution time, the heuristic is a good option for conditions with less time available.

Differently from the low demand group, the high demand instances presented lower GAPs for VNS than for CBC. The instances with 100 and 150 physicians, for example, had a GAP in the VNS of almost half of the CBC solver, indicating better solutions for the heuristic.

TABLE VI
INITIAL SOLUTION - QUALITY COMPARISON

Instance	TB&B		Improv	
	Cost	GAP	Cost	GAP
I_LOW_050_1	90821	21.7%	81637	12.9%
I_LOW_050_2	92819	23.4%	80586	11.7%
I_LOW_050_3	89542	20.5%	82679	13.9%
I_LOW_100_1	297983	17.8%	259868	5.7%
I_LOW_100_2	293932	16.2%	261097	5.6%
I_LOW_100_3	295343	14.3%	269694	6.1%
I_LOW_150_1	465716	9.4%	442154	4.5%
I_LOW_150_2	460533	9.8%	434325	4.4%
I_LOW_150_3	482872	10.0%	451374	3.8%
I_HIGH_050_1	101189	96.8%	25076	87.1%
I_HIGH_050_2	104068	96.9%	23682	86.3%
I_HIGH_050_3	100066	96.1%	27439	85.8%
I_HIGH_100_1	214750	91.6%	52578	65.8%
I_HIGH_100_2	219598	93.7%	47358	70.6%
I_HIGH_100_3	222458	91.3%	55389	65.1%
I_HIGH_150_1	360188	91.7%	81492	63.1%
I_HIGH_150_2	356504	94.6%	75661	74.7%
I_HIGH_150_3	339944	93.1%	84779	72.4%

TABLE VII
VNS X CBC - QUALITY COMPARISON

Instance	LB	TB&B		Improv	
		Cost	GAP	Cost	GAP
I_LOW_050_1	71130	72908	2.4%	71489	0.5%
I_LOW_050_2	71133	72746	2.2%	72284	1.6%
I_LOW_050_3	71153	73098	2.7%	71602	0.6%
I_LOW_100_1	245054	250115	2.0%	245626	0.2%
I_LOW_100_2	246380	251589	2.1%	246922	0.2%
I_LOW_100_3	253183	257942	1.8%	257897	1.8%
I_LOW_150_1	422120	428526	1.5%	422721	0.1%
I_LOW_150_2	415322	421436	1.5%	415983	0.2%
I_LOW_150_3	434361	440543	1.4%	435205	0.2%
I_HIGH_050_1	3233	4962	34.8%	14717	78.0%
I_HIGH_050_2	3233	4910	34.2%	4554	29.0%
I_HIGH_050_3	3908	5490	28.8%	4297	9.1%
I_HIGH_100_1	17995	23086	22.1%	53017	66.1%
I_HIGH_100_2	13937	19231	27.5%	39348	64.6%
I_HIGH_100_3	19337	24881	22.3%	45191	57.2%
I_HIGH_150_1	30072	40870	26.4%	64711	53.5%
I_HIGH_150_2	19174	31028	38.2%	58240	67.1%
I_HIGH_150_3	23368	33917	31.1%	71228	67.2%

When an instance has a large number of physicians, like 100 or more, the number of possible combinations among the daily tasks of each physician grows exponentially. If the instance also has a high demand characteristic per DSA, the majority of the search scope will be infeasible.

VII. CONCLUSIONS

This study approached a physician rostering problem, based on real data provided by the Hospital de Clínicas de Porto Alegre (HCPA), complementing two previous works [11] and [13]. The contributions are the implementation and distribution of an instance generator, the study of new constraints added to the problem, and the development and analysis of a VNS heuristic against the Coin-OR solver (CBC).

The instance generator considers real data information from HCPA, allowing the user to change a set of parameters that is common to be variable month-by-month, like the fixed assignments, the absences, and the number of physicians. All the instances were generated with the same file pattern, considering similar rules of the input file of the INRC-II [14].

The VNS heuristic uses three different neighborhoods, and the results were evaluated with different initial solutions, showing better performance with a better initial solution.

The results show a better performance for the VNS heuristic when the instances are closer to the infeasibility - the high demand group. Approximate results are found in the work [13], where the LAHC heuristic implemented had better performance than the CBC and CPLEX solvers. The instances considered in that problem had a percentage of minimum demand about 90%, a similar value of the high demand group generated in the current paper.

VIII. ACKNWLEDGMENTS

We would like to thank to the reviewers for their valuable considerations. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

REFERENCES

[1] Rais, Abdur, and Ana Viana. “Operations research in healthcare: a survey.” International transactions in operational research 18.1 (2011): 1-31.

[2] Brunner, Jens O., Jonathan F. Bard, and Rainer Kolisch. “Flexible shift scheduling of physicians.” Health care management science 12.3 (2009): 285-305.

[3] Brunner, Jens O., and Günther M. Edenharter. “Long term staff scheduling of physicians with different experience levels in hospitals using column generation.” Health care management science 14.2 (2011): 189-202.

[4] Burke, Edmund K., et al. “The state of the art of nurse rostering.” Journal of scheduling 7.6 (2004): 441-499.

[5] Erhard, Melanie, et al. “State of the art in physician scheduling.” European Journal of Operational Research 265.1 (2018): 1-18.

[6] Mansini, Renata, and Roberto Zanotti. “Optimizing the physician scheduling problem in a large hospital ward.” Journal of Scheduling (2019): 1-25.

[7] Bruni, Renato, and Paolo Detti. “A flexible discrete optimization approach to the physician scheduling problem.” Operations Research for Health Care 3.4 (2014): 191-199.

[8] Gunawan, Aldy, and Hoong Chuin Lau. “Master physician scheduling problem.” Journal of the Operational Research Society 64.3 (2013): 410-425.

[9] Adams, Thomas, Michael O’Sullivan, and Cameron Walker. “Physician rostering for workload balance.” Operations Research for Health Care 20 (2019): 1-10.

[10] Gendreau, Michel, et al. “Physician scheduling in emergency rooms.” International Conference on the Practice and Theory of Automated Timetabling. Springer, Berlin, Heidelberg, 2006.

[11] Wickert, Toni I and Neto, Alberto F Kummer and Buriol, Luciana S, “An integer programming approach for the physician rostering problem”, PATAT, 2018.

[12] Schoenfelder, Jan, and Christian Pfefferlen. “Decision support for the physician scheduling process at a German hospital.” Service Science 10.3 (2018): 215-229.

[13] Sanchotene, Thor and Buriol, Luciana S, “Abordagem heurística para solução do problema de alocação de médicos do HCPA”, 2018.

[14] CESCHIA, Sara et al. Second International Nurse Rostering Competition (INRC-II)—Problem Description and Rules—. arXiv preprint arXiv:1501.04177, 2015.