Balancing Nursing Workload by Constraint Programming

Gilles Pesant

École Polytechnique de Montréal, Montreal, Canada
CIRRELT, Université de Montréal, Montreal, Canada
gilles.pesant@polymtl.ca

Abstract. Nursing workload in hospitals has an impact on the quality of care and on job satisfaction. Understandably there has been much recent research on improving the staffing and nurse-patient assignment decisions in increasingly realistic settings. On a version of the nurse-patient assignment problem given a fixed staffing of neonatal intensive care units, constraint programming (CP) was shown to perform better than competing optimization methods. In this paper we take advantage of recent improvements to the CP approach to solve the integrated problem of staffing and nurse-patient assignment. We then consider a more difficult but also more realistic version of the problem in which patients are categorized into a small number of types and the workload associated with each type is nurse-dependent.

1 Introduction

Because of its impact on the quality of care, job satisfaction, and staff retention, nursing workload is a constant preoccupation in hospitals and it has received some recent attention in the scientific literature (e.g. [11][7][1]). Arguably the most important factor influencing nursing workload is patient acuity but others have been identified such as job interruption, patient turnover rate, and administrative paperwork [4]. If we define the workload of a nurse as the sum of the acuities of the patients he cares for, then we try to keep that value low but we also try to balance the individual workloads in order to avoid an overworked nurse and to show fairness between staff members.

Given a set of patients distributed in a number of units and an available nursing staff (previously determined as a result of *nurse rostering*), the *nurse staffing problem* consists of assigning an appropriate number of nurses to each unit. The *nurse-patient assignment problem* (NPA) then assigns patients to nurses. The number of patients per nurse may be as low as two or three in an intensive care unit [3] or around six in oncology and surgery units [10]. These two levels of assignments must be made so as to balance the resulting workloads. The typical time frame for the decision maker is one to two hours to perform staffing and 30 minutes to perform NPA [3].

This short paper focuses on solving the integrated nurse staffing and NPA problem in a neonatal intensive care setting (Section 2) and a version of the NPA with nurse-dependent patient acuities (Section 3).

Table 1: CP models for the NPA (left) and staffing (right) problems

minimize σ s.t.				
$\texttt{spread}(\{w_j\}, \sum a_i/\lvert N\rvert, \sigma)$			$x_k \geq \lceil \lvert P_k\rvert / p_{\max} \rceil$	$k \in Z$
$\texttt{gcc}(\{n_i\}, \langle [p_{\min}, p_{\max}], \ldots, [p_{\min}, p_{\max}] \rangle)$			$x_k \leq \lceil \lvert P_k\rvert / p_{\max} \rceil + f$	$k \in Z$
$\texttt{binpacking}(\langle n_i \rangle, \langle a_i \rangle, \langle w_j \rangle)$			$\sum_{k \in Z} x_k = \lvert N\rvert$	
$w_j \geq w_{j+1}$	$j \in \{1, 2, \ldots, \lvert N\rvert - 1\}$		$\sum_{k \in Z} LB_{k,x_k} < ub$	
$n_i \in N$		$i \in P$	$x_k \in \{1, \ldots, \lvert N\rvert\}$	$k \in Z$
$w_j \in \mathbb{N}$		$j \in N$		

2 Integrated Staffing and Nurse-Patient Assignment

The problem originally proposed by Mullinax and Lawley [3] asks for a balanced workload for nurses being assigned patients requiring various amounts of care (acuity) in a neonatal intensive care unit. Patients each belong to a zone, a nurse can only work in one zone, and there are upper limits both on the number of patients assigned to a nurse and on the corresponding workload.

Mullinax and Lawley solve that problem as a mixed integer linear program with a linear objective function minimizing the sum of differences between minimum and maximum workloads in each zone, which may lead to imbalance between zones. Schaus et al. [8] [9] describe a constraint programming model minimizing the standard deviation of workloads globally using the `spread` constraint [6]. They significantly improve the quality of solutions and the computational efficiency, solving two-zone instances optimally. For larger instances they first compute a staffing decision heuristically by solving a continuous relaxation of that problem and then solve each zone separately, often finding provably optimal solutions. Ku et al. [2] applied mixed integer quadratic programming and constraint integer programming (CIP). The latter, coupled with a variable ordering heuristic prioritizing the staffing and workload variables, solves two-zone instances significantly faster than the previous CP approach. A stronger filtering algorithm (achieving domain consistency) was recently proposed by Pesant [5] for the `spread` constraint and evaluated empirically on the NPA (i.e. on individual zones). It was found to solve instances one to two orders of magnitude faster than the CIP approach.[1] Building on that performance we investigate solving the integrated staffing and nurse-patient assignment problem.

2.1 Nurse-Patient Assignment

Our CP model shown in Table 1 on the left is standard: given the set of nurses N, the set of patients P, the list of patient acuities $\langle a_i \rangle$, and the minimum and maximum number of patients per nurse $p_{\min}$ and $p_{\max}$ respectively, we use one variable n_i per patient i indicating which nurse it is assigned to and one variable

[1] Personal communication from the authors of [2].

w_j per nurse j indicating his workload. To the usual constraints we add static symmetry breaking among nurses, enforce domain consistency on the `spread` constraint, and use a simple static branching heuristic that first selects the w_j variables in lexicographic order and then the n_i variables in decreasing order of acuity (values are selected in lexicographic order) [5].

2.2 Staffing

In order to solve the integrated problem exactly in principle we need to consider every staffing configuration. Fortunately most configurations are of poor quality and we can eliminate them implicitly by computing a lower bound on the standard deviation from partial configurations. We first use the heuristic staffing from Schaus et al. [8] to solve the NPA in each zone to optimality in order to provide a good upper bound. We then express the staffing problem as a constraint satisfaction problem coupled with the computation of a lower bound. For a given zone $k \in Z$ with its total patient acuity A_k and number of nurses x_k, Schaus et al. describe a lower bound on its contribution to the standard deviation, which we adapt here:

$$\alpha(\lceil A_k/x_k \rceil - \mu)^2 + \beta(\lfloor A_k/x_k \rfloor - \mu)^2$$

where $\mu = \sum a_i/|N|$ is the mean workload, $\alpha = A_k + x_k(1 - \lceil A_k/x_k \rceil)$, and $\beta = x_k - \alpha$. Summing them over all zones, dividing the result by the total number of nurses, and then taking its square root provides a bound on the standard deviation. We pre-compute these lower bounds in each zone for every possible value of x_k and put them in a matrix LB. Let P_k represent the set of patients in zone k, ub an upper bound on the deviation (provided by the best solution so far), and $f = |N| - \sum_{i \in Z} \lceil |P_i|/p_{\max} \rceil$ the number of nurses that are free to be assigned to any zone. Table 1 (right) gives our CP model for the staffing problem: every solution of this model is a staffing from which we solve a CP model for the NPA in each zone.

2.3 Results

The benchmark instances used in the literature are inspired from a neonatal intensive care unit, with an upper limit of 3 newborns per nurse and of 105 for the total workload of a nurse. They were randomly generated by Schaus et al. [8] using a realistic statistical model proposed in [3]. They range from 2 to 20 zones and up to 102 nurses and 258 patients. All experiments were run on Dual core AMD 2.1 GHz processors with 8 GB of RAM, using IBM ILOG Solver 6.7 as the CP solver.

We solve all ten 2-zone instances in an average of 0.12 seconds and 319 fails compared to 2.07 seconds (on a similar processor) and 9254 fails for Schaus et al. [8] using their two-step approach in which they fix the staffing decision

Table 2: Results on the three larger instances

					Schaus and Régin [9]		this paper		
zones	nurses	patients	mean	SD	fails	time(s)	fails	time(s)	staffings
6	31	78	84.58	4.20	12019	0.57	1387	0.37	1
15	71	198	81.95	5.33	38651	2.27	784	0.46	1
20	102	258	82.71	5.54	1176852	25.17	291286	27.04	5

heuristically.[2] Our approach did not require to evaluate more staffing configurations: all others were discarded during enumeration based on the lower bound calculation. The 2-zone instances used by Ku et al. [2] are not the same but were generated with the same parameters. Their best approach solves the instances to optimality in an average of 74.05 seconds on a faster processor.

For the 3-zone instances we require an average of 0.40 seconds and 1323 fails compared to 0.48 seconds and 16528 fails for Schaus and Régin [9] but for the latter the problem is decomposed by zone and optimized separately. Schaus and Régin could verify the optimality of all their solutions except in the case of Instance 7 — this is indeed the only instance for which we needed to explore a second staffing configuration and we can confirm that their solution is optimal. Ku et al. [2] do not report results beyond two zones.

Table 2 reports results on the three larger instances: we give the size of each instance, its mean workload, the optimal standard deviation on the workloads, the performance of the zone-decomposition approach of Schaus and Régin [9], and the performance of our approach for the integrated staffing and NPA problem including the number of staffing configurations we had to explore. We notice that our approach scales very well: indeed the increase in the number of zones does not increase the size of the problem within a zone and only the 20-zone instance took significantly more time because we needed to explore several staffing configurations (and solve more zones). Still, it is remarkable that the good quality of the lower bounds and of the initial configuration considered keep the number of eligible configurations so low. The optimality of the solution for the 20-zone instance was unknown until now: we confirm that it is. We thus close all current benchmark instances for this problem.

We created a new set of ten instances that are harder to solve to optimality in the sense that the best staffing may be different from the one obtained by solving the continuous relaxation. They were generated on 6 zones using the same parameters as Schaus et al. [8] except for the probability of success in the binomial distribution used to generate the acuity of patients which we increased from 0.23 to 0.33, yielding a wider span of acuities. Table 3 reports the performance of our approach on these instances. We see that for some of them a few

[2] For these 2-zone instances they can show that their solutions are optimal for the integrated problem. Their initial model combining staffing and nurse-patient assignment took about two orders of magnitude more time.

Table 3: Results on ten harder 6-zone instances

nurses	patients	mean	first SD	optimal SD	fails	time(s)	staffings
34	80	94.88	**6.09**	**6.04**	3183	1.05	2
38	88	94.16	5.82	5.82	1133	0.40	1
40	89	92.38	**6.23**	**5.16**	13655	14.25	5
40	88	96.48	**5.84**	**5.79**	212304	26.38	4
37	88	93.00	4.30	4.30	1748	0.62	1
39	93	94.92	4.07	4.07	26740	2.71	1
36	83	93.94	5.57	5.57	3885	0.48	1
39	87	93.49	5.41	5.41	2875	0.81	1
37	83	92.70	**5.45**	**5.08**	2200	1.85	2
35	83	89.46	3.99	3.99	1295	0.56	1

staffing configurations had to be explored and, more importantly, that the optimal standard deviation is sometimes noticeably lower than the first one obtained with the heuristic staffing (shown in bold).

3 Nurse-Dependent Patient Acuity

Patient classification systems (PCS) are commonly used in hospitals to estimate the amount of care needed by each patient. For example AcuityPlus® classifies patients into six types according to a weighted sum of 26 acuity indicators. Sir et al. [10] argue that, because of differences in experience, training, or preferences, nurses may not equally perceive the acuity associated with a given patient type. Through a survey of nurses in oncology and surgery units, they found that there could indeed be quite a bit of variation in perceived acuity between nurses and advocate that nurse-dependent patient acuity should be used when balancing nursing workloads. In this section we consider a variant of the NPA where patients are grouped according to their type and the acuity associated with each patient type is nurse-dependent.

Because the acuity of care provided for a given patient type is not perceived uniformly across nurses, we cannot know in advance what the total workload nor the mean workload will be. Hence there are really two dimensions to the quality of a solution: we wish to keep the total workload of the nursing staff low in order to offer better care and to admit new patients more seamlessly, but we also wish to balance the workload between nurses so as to be fair. In such a situation a useful decision support tool will provide the Pareto optimal front so that the decision maker has a small set of attractive solutions to work with.

Working with a variable mean is problematic for the `spread` constraint: its filtering algorithms either assume a fixed mean or sustain a significant increase in their time complexity. We propose to solve a succession of fixed mean-workload problems where we gradually increase that mean. Finding a good starting mean

proved important for the efficiency of our approach: we initially minimize the mean workload by recasting our problem as a Generalized Assignment Problem.

3.1 Generalized Assignment Problem as Lower Bound on Total Workload

If we relax the balancing aspect of our problem and simply minimize the total workload, we can express it as a generalized assignment problem in which the patients are the tasks and the nurses are the agents, with a capacity to perform multiple tasks corresponding to the minimum and maximum number of patients per nurse. We solve it using the Hungarian algorithm by framing it as the following simpler assignment problem: we make as many copies of each nurse as the maximum number of patients he can care for but add a penalty to the assignment costs for the copies in excess of the minimum number of patients required (to ensure that the minimum is reached). The resulting assignment, with its total workload and its standard deviation from the mean workload, gives us an initial point from which to proceed.

3.2 Solving Fixed Mean-Workload Instances

Each COP we solve imposes a total workload fixed to one unit less than that of the previously found solution (except in the case of our first solution from the Hungarian algorithm, which we try to improve) and a standard deviation upper bounded by that of the previous solution. We branch using the default smallest-domain-first variable selection heuristic and lexicographic value selection heuristic. We stop this iterative process when either we reach a standard deviation of zero or an upper bound on the total workload.

3.3 Empirical Evaluation

The data used by Sir et al. [10] is proprietary but we generated instances using their reported findings. Specifically we consider five patient types (the sixth type was not sufficiently represented in their data) and use the mean acuity associated with each type for the oncology unit as reported in Fig. 6 of their paper to draw nurse-dependent acuities from a normal distribution with standard deviation equal to 2.5 (loosely extrapolated from Table 6). The number of patients of each type is generated using a Poisson distribution with an expected value chosen so that the average number of patients per nurse is close to six, which is consistent with oncology and surgery units. The typical size of such a unit is reported as about 30 patients and 5 nurses. We generated ten instances of that size and ten smaller ones with 3 nurses (and about 18 patients).

All experiments were run on Dual core AMD 2.1 GHz processors with 8 GB of RAM, using IBM ILOG Solver 6.7 as the CP solver. Each COP was given up to 5 minutes to run to completion. The plot on the left at Figure 1 presents the individual Pareto optimal fronts for the ten smaller instances.

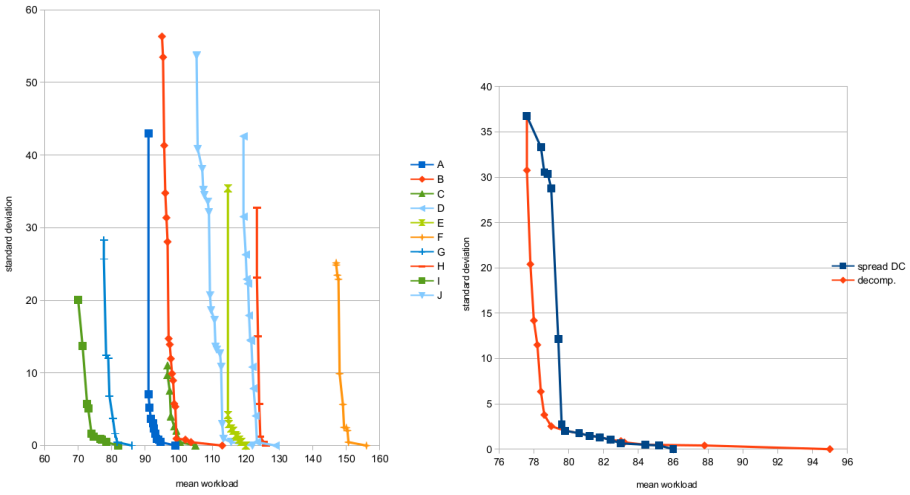

Fig. 1: Pareto solutions to the 3-nurse instances (left) and a comparison of solutions to one 5-nurse instance

Table 4: Solving the 3-nurse instances

	instance	A	B	C	D	E	F	G	H	I	J
DC	time(s)	1.56	112.79	1.03	33.47	1.44	28.82	2.59	6.96	1.50	57.74
	fails	4007	43719	2327	71521	6894	54211	3165	36120	1699	24442
decomp.	time(s)	4.30	9.75	4.84	7.28	5.95	8.08	0.97	3.61	1.23	11.64
	fails	36305	93023	47939	118838	96734	85546	12922	81567	11200	95309

Observe that each instance admits a solution with perfect balance (standard deviation equal to zero). The highest point of each front corresponds to the first solution provided by the Hungarian algorithm; note that sometimes we can find a solution of same mean but with much better balance (e.g for instance A a standard deviation equal to 7.07 instead of 43.01). Table 4 compares the total computation time and number of fails to solve each 3-nurse instance between our model using the `spread` constraint and achieving domain consistency (DC) and a simpler model using linear and quadratic constraints instead (decomp.). Not surprisingly the stronger filtering of DC always exhibits fewer fails. However the latter is sometimes much slower on these instances. Upon closer inspection, there appears to be a strong correlation between the computation time and how long it takes to find solutions with a low standard deviation: instances B, J, D and F are the slowest to solve and also have several solution points in the top part of the plots at Figure 1 (left); in contrast, instance A starts high but immediately finds solutions with a much lower standard deviation. This is not surprising because the time complexity of the domain filtering algorithm is influenced by

Table 5: Solving the 5-nurse instances

	instance	A	B	C	D	E	F	G	H	I	J
DC	time(s)	3207	9638	20496	17078	6819	6444	4811	547	15505	5733
	last mean	86	141	108	130	126	113	102	104	115	91
	last SD	0	0	0	0	0	0	0	0	0	0
decomp.	time(s)	26181	21304	50298	31705	33027	48783	42723	15219	36663	54237
	last mean	95	142	127.6	138	141.8	138.4	125.6	108	128	116.6
	last SD	0	0	0.49	0	0.49	0.4	0.4	0	0.4	0.4

the magnitude of the standard deviation. Regardless of this all these instances are solved well within the practical time frame of the hospital planner.

Moving on to the more challenging 5-nurse instances, Table 5 presents a similar comparison. The computation times jump up by a few orders of magnitude but DC is always faster here. (The difference in the number of fails, even more striking, is not shown in the table.) Note that the 5-minute time limit is often reached on these instances so the solutions found are not necessarily optimal for a given mean workload. Hence we do not provide a Pareto optimal front for them but show the typical behaviour of the two models on instance A in the right plot at Figure 1: initially for higher bounds on the standard deviation the decomposition finds solutions more quickly but as the bound on the standard deviation gets tighter the trend reverses and DC performs better, which is consistent with the previous explanation of the variation in computation times on the 3-nurse instances. We also give in the table the mean and standard deviation of the last solution found by each: DC always finds a solution with perfect balance in the end but this is not always the case with the other model (it eventually terminates because of the upper bound on the total workload), and even when it does find a solution with perfect balance it is always strictly dominated by that of DC.

4 Conclusion

In this paper we considered the problem of balancing the workload of nurses. We closed the benchmark instances for the integrated staffing and nurse-patient assignment problem in the neonatal context and proposed a new set of instances that show better the advantage of our approach. The computation times are well within the usual time frame for this problem. We also considered a challenging variant of the nurse-patient assignment problem in which patient acuities are nurse-dependent. To be useful in practice for this problem, our approach should solve faster the instances considered, which are of realistic size. We could investigate better branching heuristics and it would also be interesting to evaluate the performance of a bound-consistent filtering algorithm for the `spread` constraint here, given what was observed with the domain-consistent filtering algorithm at larger standard deviations.

Acknowledgements

Financial support for this research was provided by Discovery Grant 218028/2012 from the Natural Sciences and Engineering Research Council of Canada.

References

1. A. Hertz and N. Lahrichi. A patient assignment algorithm for home care services. *JORS*, 60(4):481–495, 2009.
2. W.-Y. Ku, T. Pinheiro, and J. C. Beck. CIP and MIQP models for the load balancing nurse-to-patient assignment problem. In Barry O'Sullivan, editor, *Principles and Practice of Constraint Programming - 20th International Conference, CP 2014, Lyon, France, September 8-12, 2014. Proceedings*, volume 8656 of *Lecture Notes in Computer Science*, pages 424–439. Springer, 2014.
3. C. Mullinax and M. Lawley. Assigning patients to nurses in neonatal intensive care. *J Oper Res Soc*, 53:25–35, 2002.
4. D. Myny, A. Van Hecke, D. De Bacquer, S. Verhaeghe, M. Gobert, T. Defloor, and D. Van Goubergen. Determining a set of measurable and relevant factors affecting nursing workload in the acute care hospital setting: A cross-sectional study. *International Journal of Nursing Studies*, 49:427–436, 2012.
5. G. Pesant. Achieving Domain Consistency and Counting Solutions for Dispersion Constraints. *INFORMS Journal on Computing*, 27(4):690–703, 2015.
6. G. Pesant and J.-C. Régin. sPREAD: A Balancing Constraint Based on Statistics. In P. van Beek, editor, *CP*, volume 3709 of *Lecture Notes in Computer Science*, pages 460–474. Springer, 2005.
7. P. Punnakitikashem, J. M. Rosenberber, and D. F. Buckley-Behan. A stochastic programming approach for integrated nurse staffing and assignment. *IIE Transactions*, 45(10):1059–1076, 2013.
8. P. Schaus, P. Van Hentenryck, and J.-C. Régin. Scalable Load Balancing in Nurse to Patient Assignment Problems. In W. J. van Hoeve and J. N. Hooker, editors, *CPAIOR*, volume 5547 of *Lecture Notes in Computer Science*, pages 248–262. Springer, 2009.
9. P. Schaus and J.-C. Régin. Bound-Consistent Spread Constraint. *EURO Journal on Computational Optimization*, 2:123–146, 2014.
10. M. Y. Sir, B. Dundar, L. M. Barker Steege, and K. S. Pasupathy. Nurse-patient assignment models considering patient acuity metrics and nurses' perceived workload. *Journal of Biomedical Informatics*, 55:237–248, 2015.
11. D. Sundaramoorthi, V. C. P. Chen, J. M. Rosenberger, S. Kim, and D. F. Buckley-Behan. A data-integrated simulation-based optimization for assigning nurses to patient admissions. *Health Care Management Science*, 13(3):210–221, 2010.