Scalable Load Balancing in Nurse to Patient Assignment Problems

Pierre Schaus[1], Pascal Van Hentenryck[2], Jean-Charles Régin[3]

[1] Dynadec, One Richmond Square, Providence, RI 02906, USA
pschaus@dynadec.com
[2] Brown University, Box 1910, Providence, RI 02912, USA
pvh@cs.brown.edu
[3] Université de Nice-Sophia Antipolis
930 Route des Colles - BP 145, 06903 Sophia Antipolis Cedex, France
regin@polytech.unice.fr

Abstract. This paper considers the daily assignment of newborn infant patients to nurses in a hospital. The objective is to balance the workload of the nurses, while satisfying a variety of side constraints. Prior work proposed a MIP model for this problem, which unfortunately did not scale to large instances and only approximated the objective function, since minimizing the variance cannot be expressed in a linear model. This paper presents constraint programming (CP) models of increasing complexity to solve large instances with hundreds of patients and nurses in a few seconds using the Comet optimization system. The CP models use the recent spread global constraint to minimize the variance, as well as an exact decomposition technique.

1 Introduction

This paper considers the daily assignment of newborn infant patients to nurses in a hospital described in [5]. In this problem, some infants require little attention, while others need significant care. The amount of work required by the infant during one shift is called the *acuity*. A nurse is in charge of a group of infants and the total amount of acuity is the workload of the nurse during that shift. For ensuring an optimal care quality and perceived fairness for the nurses, it is essential to balance the workload. In addition, the problem features various side constraints:

- A nurse can work in only one zone, but the patients are located in p different zones.
- A nurse cannot be responsible of more than $children^{\max}$ infants.
- The total amount of acuity of a nurse cannot exceed $acuity^{\max}$.

The balance objective and the various constraints make it very difficult to find a good solution in a reasonable time. Since nurses only work in one zone, the number of nurses assigned to each zone has already a huge impact on the quality of the balancing. In [5], the problem was tackled using a MIP model, but the

results were not satisfactory. In this paper, we present a series of increasingly sophisticated constraint programming models in order to reach the required solution quality and scalability.

The rest of the paper is organized as follows. Section 2 presents the instances proposed in [5] and Section 3 describes the MIP model and its limitations. Section 4 reviews the *Spread* constraint for load balancing and characterizes its pruning (as implemented in COMET). Section 5 presents a first constraint programming (CP) model that can solve two-zones instances. Section 6 presents a two-step approach that first assigns the nurses in each zone and then assigns the infants to nurses to balance the load optimally. Finally, Section 7 shows that the second step can be decomposed by zones without losing the optimality guarantees. This final model is instrumental in solving large instances with dozens of zones and hundreds of patients.

2 Problem Instances

Reference [5] specifies a statistical model to generate instances very similar to their real instances. This statistical model was also used to measure the robustness of their solution technique with respect to the number of nurses, the number of infants, and the number of zones. The model contains a single parameter: the number of zones. The maximum acuity per nurse is fixed to $acuity^{\max} = 105$ and the maximum number of infants per nurse is fixed to $children^{\max} = 3$. The instance generator fixes the number of nurses, the number of infants, the acuity, and the zone of each infant. The different steps to generate an instance are as follows:

- The number of patients in a zone is specified by a Poisson random variable with mean 3.8 and offset by 10.
- The acuity Y of a patient is obtained by first generating a number $X \sim Binomial(n = 8, p = 0.23)$ and then choosing the number $Y \sim Unif(10 \cdot (X + 1), 10 \cdot (X + 1) + 9)$.
- The total number of nurses is obtained by solving a First Fit Decreasing (FFD) procedure in each zone. More precisely, the total number is the number of nurses found in each zone by the FFD procedure. The FFD procedure starts by ranking the patients in decreasing acuity. Then, the patient with the highest acuity is assigned to the first nurse. The next patients are assigned successively to the first nurse that can accommodate them without violating the maximum acuity and the number of patient constraints.

3 The MIP Model

We now review the main variables of the MIP model from [5]. We also describe the limitations of the MIP model and suggest why a CP approach may address them. Due to space reasons, we do not reproduce the entire MIP model but readers can consult [5] for more details. The technical details presented here are sufficient for our purposes. The MIP model contains four families of variables:

Fig. 1. Comparison of Two Solutions on a 6 Nurses, 14 Infants, and 2 zones Problem. Solution on the left is obtained by minimizing the range-sum criterion. Solution on the right is obtained by minimizing the variance.

1. $X_{ij} = 1$ if infant i is assigned to nurse j and 0 otherwise;
2. $Z_{jk} = 1$ if nurse j is assigned zone k and 0 otherwise;
3. $Y_{k,\max}$ is the maximum acuity of a nurse in zone k;
4. $Y_{k,\min}$ is the minimum acuity of a nurse in zone k.

All these variables are linked with linear constraints to enforce the constraints of the problem. The objective function implements what we call the *range-sum* criterion and consists of minimizing the sum of the acuity ranges of the p zones, i.e.,

$$\sum_{k=1}^{p}(Y_{k,\max} - Y_{k,\min}).$$

The MIP model has a fundamental limitation: The objective function may produce poorly balanced workloads. It tends to equalize the workload inside the zones but may produce huge differences among the workload of different zones. This is illustrated in Figure 1. The workloads are depicted in the top-right corner of each COMET visualization. The left solution is obtained by minimizing the range-sum criterion and the right solution by minimizing the variance (L_2 norm in the next section). The range-sum objective is minimal on the left because the workloads inside each of the two zones are identical. Unfortunately, nurses in the first zone work twice as much as those in the second zone. The right solution is obtained by minimizing the variance and is significantly more appealing.

This illustrates clearly that "the high level objective that all nurses should be assigned an equal amount of patient acuity" [5] is not properly captured with the range-sum criterion.

It is not immediately obvious how to remedy these problems. The variance is non-linear and is not easily modelled in a MIP approach. In addition, a CP approach may exploit the combinatorial structure in the bin-packing and the side-constraints, while the MIP relaxation is generally bad for bin-packing like problems. Finally, there are important symmetries that are not removed in their model: For a given solution, the nurses are completely interchangeable. We now review load balancing constraints before turning to the CP models.

4 Load Balancing Constraints

Balancing constraints arise in many real-world applications, most often to express the need of a fair distribution of items or work. For instance, Simonis [15] suggested a global constraint to balance the shift distribution among nurses and Pesant [7] proposed the use of balancing constraints for a fair allocation of individual schedules.

Two global constraints and their propagators are available in constraint programming for optimizing load balancing: `spread` [6, 11], which constrains the variance and the mean of a set of variables, and `deviation` [12, 13], which constrains the mean absolute deviation and the mean of a set of variables. We also say that `spread` and `deviation` respectively constrain the L_2 and L_1 norms of a set of variables $X_1..X_n$ with respect to their mean ($s = \sum_{i\in[1..n]} X_i$), i.e.,

- L_1: $\sum_{i\in[1..n]} |X_i - s/n|$;
- L_2: $\sum_{i\in[1..n]} (X_i - s/n)^2$.

These criteria are not equivalent: Minimizing L_1 or L_2 does not lead to the same solutions and it is not always obvious which one to choose. In fact, this is an old and recurrent debate (see for instance [3]). For this application, we use `spread` because the L_2 criteria is more sensitive to outliers, which we consider significant in this application.

We use the following definitions and notations to describe the semantics of the `spread` constraints and propagators.

Definition 1. *Let X be a finite-domain (discrete) variable. The domain of X is a set of ordered values that can be assigned to X and is denoted by $Dom(X)$. The minimum (resp. maximum) value of the domain is denoted by $X^{\min} = \min(Dom(X))$ (resp. $X^{\max} = \max(Dom(X))$. An integer interval with integer bounds a and b is denoted $[a..b] \subseteq \mathbb{Z}$, while a rational interval is denoted $[a, b] \subseteq \mathbb{Q}$. An assignment on the variables $\boldsymbol{X} = [X_1, X_2, ..., X_n]$ is denoted by the tuple $\mathbf{x}$ and the i-th entry of this tuple by $\mathbf{x}[i]$. The extended rational interval domain of X_i is $I_D^{\mathbb{Q}}(X_i) = [X_i^{\min}, X_i^{\max}]$ and its integer interval domain is $I_D^{\mathbb{Z}}(X_i) = [X_i^{\min} \,..\, X_i^{\max}]$.*

We now define the `spread` constraint with a fixed mean.

Definition 2. *Given finite domain variables* $\boldsymbol{X} = (X_1, X_2, ..., X_n)$*, an integer value* s *and a finite domain variable* Δ*,* $\texttt{spread}(\boldsymbol{X}, s, \Delta)$ *holds if and only if*

$$\sum_{i \in [1..n]} X_i = s \quad \text{and} \quad \Delta \geq n \times \sum_{i \in [1..n]} |X_i - s/n|^2.$$

Observe also

$$n \cdot \sum_{i \in [1..n]} |X_i - s/n|^2 = n \times \sum_{i \in [1..n]} X_i^2 - s^2. \tag{1}$$

Since s is an integer, this quantity is integer, which is why it is more convenient to work with $n \times \sum_{i \in [1..n]} X_i^2 - s^2$ than with $\sum_{i \in [1..n]} |X_i - s/n|^2$.

Example 1. Tuple $\mathbf{x} = (4, 6, 2, 5) \in \texttt{spread}([X_1, X_2, X_3, X_4], s = 17, \Delta = 40)$ but $\mathbf{x} = (3, 6, 2, 6) \notin \texttt{spread}([X_1, X_2, X_3, X_4], s = 17, \Delta = 40)$ because $4 \cdot (3^2 + 6^2 + 2^2 + 6^2) - 17^2 = 51 > 50$.

The filtering algorithm for `spread` achieves $\mathbb{Z}$-bound-consistency.

Definition 3 ($\mathbb{Q}$-bound-consistency and $\mathbb{Z}$-bound-consistency). *A constraint* $C(X_1, \ldots, X_n)$ *(*$n > 1$*) is* $\mathbb{Q}$*-bound-consistent (resp.* $\mathbb{Z}$*-bound-consistent) with respect to domains* $Dom(X_i)$ *if for all* $i \in \{1, \ldots, n\}$ *and each value* $v_i \in \{X_i^{\min}, X_i^{\max}\}$*, there exist values* $v_j \in I_D^{\mathbb{Q}}(X_j)$ *(resp.* $v_j \in I_D^{\mathbb{Z}}(X_j)$*) for all* $j \in \{1, \ldots, n\} - \{i\}$ *such that* $(v_1, \ldots, v_n) \in C$*.*

The propagators described in [6, 11] achieve $\mathbb{Q}$-bound-consistency, which means that they assume that the variables can be assigned rational numbers. The propagators implemented in Comet implement the stronger $\mathbb{Z}$-bound-consistency by adapting the algorithms from [6, 11]. In particular, to achieve $\mathbb{Z}$-bound-consistency, the propagators for `spread` compute $\underline{\Delta}^{\mathbb{Z}}$ to filter $\Delta^{\min}$, and $\overline{X}_i^{\mathbb{Z}}$ and $\underline{X}_i^{\mathbb{Z}}$ to filter $X_i^{\max}$ and $X_i^{\min}$:

$$\underline{\Delta}^{\mathbb{Z}} = \min_{\mathbf{x}} \{ n \cdot \sum_{i \in [1..n]} (\mathbf{x}[i] - s/n)^2 \ \text{ s.t. } \sum_{i \in [1..n]} \mathbf{x}[i] = s \quad \text{and} \quad \forall i \in [1..n] : \mathbf{x}[i] \in I_D^{\mathbb{Z}}(X_i) \} \tag{2}$$

$$\overline{X}_i^{\mathbb{Z}} = \max_{\mathbf{x}} \{ \mathbf{x}[i] \text{ s.t. } n \cdot \sum_{j \in [1..n]} (x[j] - s/n)^2 \leq \Delta^{\max} \text{ and } \sum_{j \in [1..n]} \mathbf{x}[j] = s \text{ and } \forall j : x[j] \in I_D^{\mathbb{Z}}(X_j) \}. \tag{3}$$

The filtering of Δ is implemented in $O(n \cdot log(n))$ requiring to sort the bounds of the domains, and that of $\mathbf{X}$ in $O(n^2)$ in the Comet System [2, 10].

5 A Basic CP model.

We now present a CP based resolution which addresses the issues raised for the MIP model.

The CP Model. Let m be the number of nurses, n the number of patients, and a_i be the acuity of patient i. The set of patients in zone k is denoted by $\mathcal{P}_k$ and $[\mathcal{P}_1, ..., \mathcal{P}_p]$ forms a partition of $\{1, ..., n\}$. For each patient i, we use a decision variable $N_i \in [1..n]$ representing her/his nurse. The workload of nurse j is represented by variable $W_j \in [0..acuity^{\max}]$. The objective and constraints are modelled as follows.

- The objective, i.e., minimizing the L_2 norm, is expressed by a `spread` constraint over the workload variables $[W_1, ..., W_m]$, the total acuity, and the acuity spread: `spread(`$[W_1, ..., W_m]$`,totalAcuity,spreadAcuity)`. Note that `spreadAcuity` is the variable to minimize.
- To express that nurses have a total acuity of atmost $acuity^{\max}$, we link the variables N_i, W_j, and the acuities with a global packing/multiknapsack constraint [14]: `multiknapsack`$([N_1, ..., N_n], [a_1, ..., a_n], [W_1, ..., W_m])$.
- To model that a nurse takes care of at most $children^{\max}$ infants and at least one, we use a global cardinality constraint [8]: `cardinality`$(1, [N_1, ..., N_n], children^{\max})$.
- The constraint that a nurse can work in at most one zone is modelled by a pairwise-disjoint constraint `pairwiseDisjoint(`$[Z_1, ..., Z_p]$`)`, where Z_k is an array of variables containing the variables N_i associated with zone k.

The Comet *Program* The model in Comet is shown in Listing 1.1. Lines 1–3 declare the decision variables. Line 4 declares the arrays for the zones, which are filled in lines 5–7. The objective function is in lines 8–9 and 11. Lines 12–14 depict the constraints. The `pairwiseDisjoint` constraint introduces set-variables representing the set of nurses working in each zone $NS_k = \bigcup_{i \in \mathcal{P}_k} N_i$. The set NS_k is maintained with a global constraint `unionOf`. Then, the pairwise empty intersections between the set variables are enforced with a global disjoint constraint. Comet uses a reformulation with channeling constraints and a global cardinality constraint as explained in [9, 1].

The search is implemented in the `using` block in lines 16–24. The search dynamically breaks the value symmetries originating from the nurse interchangeability. The patient having the largest acuity is selected first in line 17. Then the search tries to assign a nurse to this patient, starting first with those with the smaller load (lines 19–22). The symmetry breaking is implemented by considering the already assigned nurses and at most one additional nurse without any assigned patient (a similar technique was used for the steel mill slab problem in [4]). Value `mn` is the maximal index of a nurse already assigned to a patient. The `tryall` statement considers all the nurse indexes until `mn+1` (nurse `mn+1` having currently no patient).

Experimental Results As a first experiment, we generated 10 instances with 2 zones, as was the case for the real instances studied in [5]. These instances have about 10–15 nurses, 20–30 infants, and cannot be solved by the MIP model. All the instances were solved optimally with our Comet model in less than 30 minutes (the time constraint specified in [5] by the hospital to find the assignment).

Listing 1.1. Patient-Nurse Assignment Model

```
1  var<CP>{int} N[patients](cp,nurses);
2  var<CP>{int} W[nurses](cp,1..MaxAcuity);
3  var<CP>{int} spreadAcuity(cp,0..System.getMAXINT());
4  var<CP>{int}[] Z[zones];
5  int k = 1;
6  forall(i in zones,j in 1..nbPatientsInZone[i])
7      Z[i][j] = N[k++];
8  minimize<cp>
9     spreadAcuity
10 subject to {
11     cp.post(spread(W,sum(p in patients) acuity[p],spreadAcuity));
12     cp.post(multiknapsack(N,acuity,W));
13     cp.post(cardinality(minNbPatients,N,maxNbPatients));
14     cp.post(pairwiseDisjoint(Z));
15 }
16 using {
17   forall(p in patients: !N[p].bound()) by (−acuity[p],N[p].getSize()) {
18     int mn = max(0,maxBound(N));
19     tryall<cp>(n in nurses: n <= mn + 1) by (W[n].getMin())
20         cp.label(N[p],n);
21     onFailure
22         cp.diff(N[p],n);
23   }
24 }
```

Table 1. Patients to Nurses Assignment Problem with 2 zones and minimization of L_2 with `spread`.

m	n	#fails	time(s)	avg workload	sd. workload
11	28	511095	170.2	86.09	2.64
11	29	1126480	302.0	80.27	1.76
10	26	104931	24.7	76.50	2.29
12	30	259147	136.5	83.42	1.93
10	28	2990450	1138.5	91.80	6.84
10	26	779969	206.9	88.40	2.29
12	29	555243	198.2	80.08	2.72
10	27	931858	343.9	90.60	5.33
10	25	1616689	434.5	82.70	7.32
8	22	4160	1.2	87.50	3.12

Table 1 depicts the experimental results. All results are using COMET 1.1 [2] on 2.4 GHz Intel Core Duo with 4GB running MacOS 10.5.6.

6 A Two-Step CP Model

The basic CP model can solve 2-zone instances but has great difficulty for 3 zones or more. We now show how to simplify the resolution by a two-step approach which first pre-computes the number of nurses assigned to each zone and then assigns the patients to nurses. This simplifies the resolution by

1. removing one degree of flexibility which is the number of nurses in each zone.
2. removing the disjointness constraint since the set of nurses that can be assigned to each patient can be pre-computed.

A Relaxation This first step is important because the decomposition may be significantly sub-optimal if these numbers are not properly chosen. Indeed, the number of nurses assigned to each zone has a crucial impact on the quality of the balancing. However, after visualizing some optimal solutions, we observed that the workloads of the nurses are extremely well balanced (almost the same) inside the zones. This suggested solving a relaxation of the problem to discover a good distribution of the nurses to the zones. The relaxation allows the acuity of a child in a zone to be distributed among the nurses of that zone (continuous relaxation of the acuity). Since the acuity of a child can be split, the relaxed problem will have an optimal solution where the nurses of a zone have exactly the same workload $\frac{A_k}{x_k}$, i.e., the total acuity $A_k = \sum_{i \in \mathcal{P}_k} a_i$ of zone k divided by the number of nurses x_k in zone k. This is schematically illustrated on Figure 2 for a two-zone relaxation problem and stated in Theorem 1.

Theorem 1. *An optimal solution of the relaxed problem must have the same workload for all the nurses in a given zone.*

Proof. Otherwise, given m variables $[W_1, \ldots, W_m]$ with sum $s = \sum_{i=1}^{m} W_i$, the L_2 criterion can be improved on these variables if two of them can be made closer (2 nurses of the same zone with a different workload). Let W_i and W_j be the variables that can be made closer and assume without loss of generality that $W_i > W_j$. The variables after modification are respectively W_i' and W_j'. If W_i and W_j are made closer this means that $W_i' - W_j' < W_i - W_j$. Since the sum is fixed then $W_i' + W_j' = W_i + W_j$. Thus $W_i - W_i' = W_j' - W_j$ and so there exists δ with $\frac{(W_i - W_j)}{2} \geq \delta > 0$ such that $W_i - W_i' = \delta = W_j' - W_j$. That is $W_i' = W_i - \delta$ and $W_j' = W_j + \delta$. The starting sum of square deviations with formula (1) is $\Delta = m \cdot \sum_{i=1}^{m} (W_i)^2 - s^2$. With W_i' and W_j' it becomes $\Delta' = m \cdot (\sum_{k \neq i,j} (W_k)^2 + (W_i - \delta)^2 + (W_j + \delta)^2) - s^2 = \Delta - 2m\delta \cdot (W_i - W_j - \delta)$. Since $(W_i - W_j - \delta > 0)$, we have $\Delta' < \Delta$. □

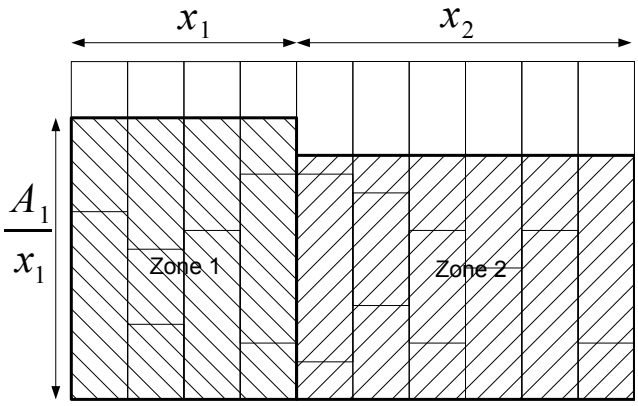

Fig. 2. Illustration of a solution of the relaxation solved to find the number of nurses in each zone.

Given Theorem 1, the mathematical formulation of the relaxed problem is

$$\min \sum_{k=1}^{p} x_k \cdot \left(\frac{A_k}{x_k} - \sum_{j=1}^{p} \frac{A_j}{m} \right)^2 \tag{4}$$

$$s.t. \sum_{k=1}^{p} x_k = m \tag{5}$$

$$x_k \in \mathbb{Z}_0^+ \tag{6}$$

The workload of all the nurses of zone k is $\frac{A_k}{x_k}$ and the average workload is $\sum_{j=1}^{p} \frac{A_j}{m}$. Hence the contribution to the L_2 criterion for the x_k nurses of zone k is $x_k \cdot \left(\frac{A_k}{x_k} - \sum_{j=1}^{p} \frac{A_j}{m} \right)^2$.

Solving the Relaxation In our CP model, we approximate this relaxation in $O(p \cdot log(p))$ time. First, we solve the continuous relaxation of the problem, i.e., we drop the integrality constraint (6). The solution to this continuous optimization problem is $x_k = m \cdot \frac{A_k}{\sum_{j=1}^{p} A_j}$, which corresponds to assigning the average workload $\sum_{j=1}^{p} \frac{A_j}{m}$ to every nurse. The continuous solution $x_k = m \cdot \frac{A_k}{\sum_{j=1}^{p} A_j}$ can be transformed greedily into an integer solution using the following steps:

- By developing the objective (4), it appears that it is equivalent to minimize $\sum_{k=1}^{p} \frac{(A_k)^2}{x_k}$.
- The transformation into an integer solution starts by first rounding up the number of nurses in every zone $x_k = \lceil m \cdot \frac{A_k}{\sum_{j=1}^{p} A_j} \rceil$. The effect is that the constraint (5) may be violated and the objective might decrease.

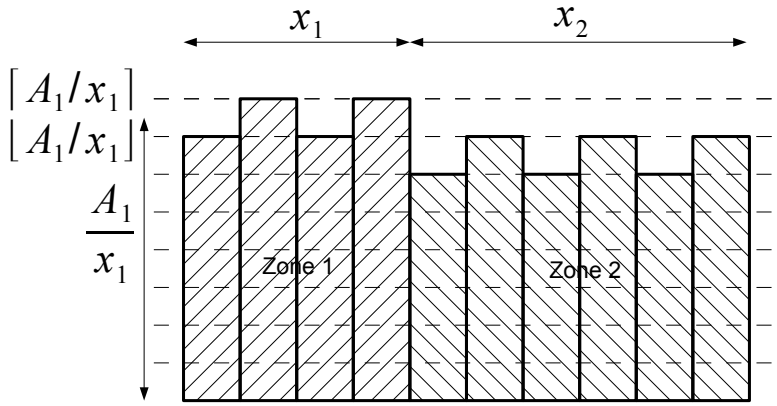

Fig. 3. Illustration of the Lower Bound on L_2 using the Pre-Computation of the Number of Nurses in Each Xone.

- Then, the $x_k > 1$ are considered to be decreased by one unit until the constraint (5) is satisfied again. The index k of the next x_k to be decreased is $\text{argmin}_k\{\frac{A_k^2}{x_k-1} - \frac{A_k^2}{x_k}\}$, i.e., the variable that will increase the least its corresponding term in the equivalent objective $\sum_{k=1}^{p} \frac{A_k^2}{x_k}$.

Our experimental results show that this approximation is optimal on all the instances the first CP model solved.

Lower Bound on the Spread The pre-computation of the number of nurses assigned to each zone is also instrumental in computing a lower bound on the L_2 criterion. Inside a zone, the average load is $\mu_k = A_k/x_k$. Since the acuity of patients are integers, we can strengthen the lower bound of the objective (4) by enforcing the workloads of nurses of zone k to be either $\lfloor \mu_k \rfloor$ or $\lceil \mu_k \rceil$. This is illustrated on Figure 3. Since the total workload of zone k must remain A_k, the distribution of the workload among $\lfloor \mu_k \rfloor$ and $\lceil \mu_k \rceil$ are given respectively by $\alpha_k = A_k + x_k \cdot (1 - \lceil \mu_k \rceil)$ and $\beta_k = x_k - \alpha_k$. The lower bound on the spread variable $\underline{\Delta}^{\mathbb{Z}}$ computed with formula (1) is thus

$$m \cdot \sum_{k=1}^{p} (\alpha_k \cdot \lceil \mu_k \rceil^2 + \beta_k \cdot \lfloor \mu_k \rfloor^2) - (\sum_{k=1}^{p} A_k)^2. \tag{7}$$

The Comet *Model* The two-step CP model in Comet is given in Listing 1.2 and assumes that the x_k are already computed. The model does not create the N variables in line 2: These will be created at the same time as the zone arrays, since their domains are now restricted to a subset of the nurses. Lines 6–12 create the zone arrays, line 10 constructing the array for zone i. Note that the

domains of these variables are defined in lines 9 and 11, using the number of nurses assigned in the zones. Lines 13–15 assign the zone variables to the nurse variables (the opposite of the first model, since the zone variables now have restricted domains). The constraints are similar but there is no longer a need for the `pairwiseDisjoint` constraint. The search in lines 23–34 is a little bit more complicated as the patients are assigned one zone at a time. The dynamic symmetry breaking scheme is the same but adapted to this by zone assignment.

Table 2 reports the results obtained on the same 2-zones instances as for Table 1 using the pre-computation of the number of nurses assigned to each zone. The last column is the lower bound obtained with equation (7). A first observation is that the computation times are greatly reduced. They do not exceed 10 seconds with the new model, while they were over 1000 seconds for the most difficult instances with the old one. The CP model finds the correct number of nurses in the first step, since the standard deviation with previous model are exactly the same (hence optimum) as the optimal values in Table 1. It is also interesting to see that the lower bound is reasonably close to the optimum values which also validates the approach.

Table 2. Patients to Nurses Assignment Problem with 2 zones with precomputation of the number of nurses in each zone

m	n	#fails	time(s)	avg workload	sd. workload	lb. sd.
11	28	25385	4.5	86.09	2.64	2.23
11	29	4916	1.4	80.27	1.76	0.62
10	26	458	0.1	76.50	2.29	2.29
12	30	17558	6.7	83.42	1.93	1.19
10	28	29865	4.8	91.80	6.84	6.81
10	26	3705	1.0	88.40	2.29	1.43
12	29	6115	1.2	80.08	2.72	0.64
10	27	1109	0.4	90.60	5.33	5.22
10	25	3299	0.6	82.70	7.32	6.71
8	22	127	0.0	87.50	3.12	3.04

Since the instances with 2 zones can now be solved easily, we tried to solve instances with 3 zones. The results are presented on Table 3. Only 6 instances (out of 10) could be solved optimally within 30 minutes with this two-step approach.

7 A Two-Step CP Model with Decomposition

The previous approach can solve easily two-zone problems but has difficulties to solve 3 zones problems and instances with more that 3 zones are intractable. It thus seems natural to decompose the problem by zone and to balance the workload of nurses inside each zone independently rather than balancing the workload of all the nurses globally. Interestingly, this decomposition preserves

Listing 1.2. Two steps Patient-Nurse Assignment Model

```
1  Solver<CP> cp();
2  var<CP>{int} N[patients];
3  var<CP>{int} W[nurses](cp,1..MaxAcuity);
4  var<CP>{int} spreadAcuity(cp,0..System.getMAXINT());
5  var<CP>{int}[] Z[zones];
6  range nursesOfZone[zones];
7  int j=1;
8  forall(i in zones) {
9     nursesOfZone[i] = j..j+x[i]-1;
10    Z[i] = new var<CP>{int}[1..nbPatientsInZone[i]](cp,nursesOfZone[i]);
11    j += x[i];
12 }
13 int k = 1;
14 forall(i in zones,j in 1..x[i])
15          N[k++] = Z[i][j];
16 minimize<cp>
17    spreadAcuity
18 subject to {
19    cp.post(spread(W,sum(p in patients) acuity[p],spreadAcuity));
20    cp.post(multiknapsack(N,acuity,W));
21    cp.post(cardinality(minNbPatients,N,maxNbPatients));
22 }
23 using {
24   forall(i in zones){
25     forall(p in Z[i].rng(): !Z[i][p].bound()) by(-acuityByZone[i][p],Z[i][p].getSize()){
26          int shift = i==1? 0 : nursesOfZone[i-1].getUp();
27          int mn = max(0,maxBound(Z[i]))+shift;
28          tryall<cp>(n in nursesOfZone[i]: n <= mn + 1) by (W[n].getMin())
29             cp.label(Z[i][p],n);
30          onFailure
31             cp.diff(Z[i][p],n);
32     }
33   }
34 }
```

Table 3. Patients to Nurses Assignment Problem with 3 zones with precomputation of the number of nurses in each zone

sol	m	n	#fails	time(s)	avg workload	sd. workload	lb. sd.
1	15	42	19488	5.3	84.20	3.04	2.93
1	18	43	3619310	919.2	79.78	5.84	5.49
0	17	43	9023072	1800.0	81.41	4.75	3.45
1	17	42	483032	106.9	83.82	5.65	5.59
0	18	43	7124370	1800.0	81.00	7.11	4.94
1	14	38	590971	145.2	85.36	3.08	2.16
0	19	48	3786580	1800.0	87.42	3.18	2.30
1	16	44	3888210	839.8	84.88	6.70	6.39
0	19	49	5697272	1800.0	86.00	2.70	1.95
1	17	41	61250	17.3	82.18	3.40	3.07

optimality, i.e., it reaches the same solution for the L_2 criterion as the two-step approach of Section 6 for a given pre-computation of the number of nurses assigned in each zone. In other words, given the pre-computed number of nurses in each zone, it is equivalent to minimize L_2 among all the nurses at once or to minimize L_2 separately inside each zone. We now prove this result formally.

Lemma 1. *Minimizing* $n \cdot \sum_{i=1}^{x_k}(y_i - A_k/x_k)^2$ *such that* $\sum_{i=1}^{x_k} y_i = A_k$ *is equivalent to minimizing* $n \cdot \sum_{i=1}^{x_k}(y_i - (A_k/x_k + c))^2$ *such that* $\sum_{i=1}^{x_k} y_i = A_k$.

Proof. The first objective can be reformulated from formula 1 as $x_k \cdot \sum_{i=1}^{x_k} y_i^2 - A_k^2$. The second one can be reformulated after some algebraic manipulations as $c^2 \cdot x_k^2 + x_k \cdot \sum_{i=1}^{x_k} y_i^2 - A_k^2$. Since they differ only by a constant term, they produce the same set of optimal solutions. □

Theorem 2. *It is equivalent to minimize* L_2 *among all the nurses at once or to minimize* L_2 *separately inside each zone.*

Proof. This follows directly from Lemma 1. If the minimization of L_2 is performed globally for all the nurses, the least square L_2 criterion is computed with respect to the global average load of all the nurses that is wrt $\sum_{k=1}^{p} A_k/m$. This corresponds to choosing c in Lemma 1 equal to the difference between the average load in zone k and the global average load: $c = \sum_{k=1}^{p} A_k/m - A_k/x_k$. □

We solved again the 3-zone instances with the decomposition method. The results are given on Table 4. One can observe that, as expected, the objectives are the same for the instances that could be solved optimally in Table 3. For the remaining ones, the algorithm produces strictly better solutions. The time is also significantly smaller. Figure 4 shows a COMET visualization of a solution for a 15-zones instance with 81 nurses and 209 patients. This instance could be solved in only 7 seconds and 10.938 fails.

Table 4. Patients to Nurses Assignment Problem with 3 zones with precomputation of the number of nurses in each zone and decomposition by zone

m	n	#fails	time(s)	avg workload	sd. workload	lb. sd.
15	42	203	0.1	84.20	3.04	2.93
18	43	608	0.1	79.78	5.84	5.49
17	43	8134	1.1	81.41	4.46	3.45
17	42	345	0.1	83.82	5.65	5.59
18	43	24994	3.2	81.00	5.77	4.94
14	38	151	0.0	85.36	3.08	2.16
19	48	3695	0.8	87.42	3.07	2.30
16	44	384	0.1	84.88	6.70	6.39
19	49	2056	0.4	86.00	2.49	1.95
17	41	776	0.2	82.18	3.40	3.07

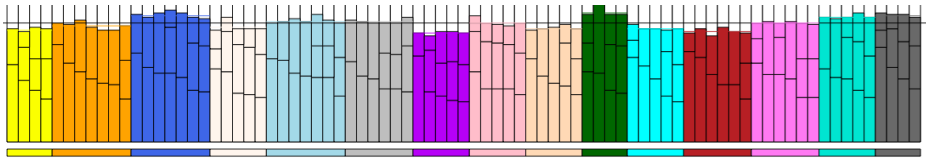

Fig. 4. Solution of a 15-Zone Instance.

8 Conclusion

This paper considered the daily assignment of newborn infant patients to nurses in a hospital. The objective is to balance the workload of the nurses, while satisfying a variety of side constraints. Prior work proposed a MIP model for this problem which exhibits two limitations. It did not scale to large instances and its objective function did not balance the workload properly. The paper presented a direct CP model which balances the load appropriately and easily solve 2-zone instances. To scale the CP approach, the paper showed how to decompose the problem in two steps: an assignment of nurses to zones followed by the assignment of nurses to patients. The first step is obtained from a relaxation of the problem which could be solved quickly. The second step is solved by a simplification of the direct model. This 2-step approach dramatically improved the results on the 2-zone instances and could solve some 3-zone instances. The paper then showed that the zone problems can be solved independently without quality loss. This resulting CP model solves 3-zone problems almost instantly and is highly scalable. For instance, a 15-zone problem with 81 nurses and 209 patients was solved in 7 seconds.

There are a number of interesting issues left to investigate. It would be interesting to study the quality of the approximation performed in the first step. Our experimental results indicate that it is optimal on all our tested intances but a performance guarantee would be desirable. Alternatively, we could con-

sider solving this first step exactly, an algorithmic issue we need to investigate. In addition, it would be interesting to study problems in which nurses have qualifications which restrict their possible zone assignments.

References

1. Christian Bessiere, Emmanuel Hebrard, Brahim Hnich, and Toby Walsh. Disjoint, partition and intersection constraints for set and multiset variables. In *Principles and Practice of Constraint Programming CP 2004*, pages 138–152, 2004.
2. DYNADEC. Comet 1.1 release. *www.dynadec.com*, 2009.
3. Stephen Gorard. Revisiting a 90-year-old debate: The advantages of the mean deviation. *British Journal of Educational Studies*, pages 417–439, 2005.
4. P. Van Hentenryck and L. Michel. The steel mill slab design problem revisited. *CP'AI'OR-08, Paris, France*, 5015:377–381, May 2008.
5. C Mullinax and M Lawley. Assigning patients to nurses in neonatal intensive care. *Journal of the Operational Research Society*, 53:25–35, 2002.
6. G. Pesant and J.C. Régin. Spread: A balancing constraint based on statistics. *Lecture Notes in Computer Science*, 3709:460–474, 2005.
7. Gilles Pesant. Constraint-based rostering. *The 7th International Conference on the Practice and Theory of Automated Timetabling PATAT 2008*, 2008.
8. J-C. Régin. Generalized arc consistency for global cardinality constraint. *AAAI-96*, pages 209–215, 1996.
9. J.C. Régin. Habilitation à diriger des recherches (hdr) : modelization and global constraints in constraint programming. *Université Nice*, 2004.
10. P. Schaus. Balancing and bin-packing constraints in constraint programming. *PhD thesis, Université catholique de Louvain, INGI,*, 2009.
11. P. Schaus, Y. Deville, P. Dupont, and J.C. Régin. Simplification and extension of spread. *3th Workshop on Constraint Propagation And Implementation*, 2006.
12. P. Schaus, Y. Deville, P. Dupont, and J.C. Régin. The deviation constraint. *Proceedings of CP-AI-OR*, 4510:269–284, 2007.
13. Pierre Schaus, Yves Deville, and Pierre Dupont. Bound-consistent deviation constraint. *13th International Conference on Principles and Practice of Constraint Programming (CP 2007)*, 4741, 23/09/2007 2007.
14. Paul Shaw. A constraint for bin packing. In *Principles and Practice of Constraint Programming CP 2004*, pages 648–662, 2004.
15. Helmut Simonis. Models for global constraint applications. *Constraints*, 12:63–92, March 2007.