A Finite Automata Reduction Procedure to Improve MIP `regular` Formulations of Personnel Scheduling Problems

Guillaume Ghienne[1], Odile Bellenguez[1], Guillaume Massonnet[1], and María I. Restrepo[1]

IMT Atlantique, L2SN, UMR CNRS 6004, 4 rue Alfred Kastler, 44300 Nantes, France
{guillaume.ghienne,odile.bellenguez,guillaume.massonnet,
maria-isabel.restrepo-ruiz}@imt-atlantique.fr

Keywords: Unambiguous finite automaton, Biclique decomposition, Regret heuristic, Mathematical programming, Personnel scheduling.

1 Introduction

Inspired by the Constraint Programming (CP) `regular` constraint [8], [3] proposed a similar approach in a Mixed Integer Programming (MIP) context: a `MIP regular` formulation. Their method proved its efficiency to solve several Personnel Scheduling Problems (PSP) [3,7]. The main idea is to consider employee's schedules as words of formal languages to encode a subset of working regulations. Flow formulations in the network structures of these languages can be derived and possibly combined with a classical MIP formulation to entirely model a PSP. In such a formal language, words have a fixed length n equal to the number of time periods in the planning horizon and can be represented by Finite Automata with $n+1$ levels (hereinafter denoted as n-FA) in which each accepting path represents a valid schedule between the first and the last level. The number of decision variables in the resulting network flow formulation corresponds to the number of transitions in the n-FAs.

When each word in the language has exactly one accepting path, we say that an n-FA is unambiguous. Note that deterministic automata (i.e., there is at most one transition per symbol for a given state) are unambiguous by definition. This characteristic allows to break possible symmetries in the mathematical programming formulation of the problem. Therefore, to efficiently obtain small and asymmetric `MIP regular` formulations for PSPs, we are interested in computing an unambiguous n-FA representing a given set of working regulations and with as few transitions as possible. This task faces three main challenges. First, n-FA minimization is NP-hard [1], and our aim is to develop a procedure performed in a negligible amount of time when compared to MIP solving time. Second, unambiguity is not guaranteed in a FA by usual minimization procedures. Third, existing works in the literature mainly focus on reducing the number of states in the automata, while we are more interested in reducing the number of transitions which in turn limits the number of variables in the final MIP formulations.

2 Reduction Approach

[6] describes a procedure to obtain deterministic (hence unambiguous) n-FAs representing a set of given PSP working regulations. We now seek to reduce the size of such automata. The classical minimization of deterministic n-FAs [5] is performed in linear time and is optimal in terms of both state and transition reduction. Although this obvious approach answers the three challenging points previously discussed in an efficient manner, nondeterministic n-FAs allow for an even more compact representation. In particular, [1] generalize Nerode's equivalence to derive an n-FA state reduction heuristic as a sequence of level minimization. This procedure allows to maintain unambiguity during the reduction and we observe short enough computation times when we test it on practical PSPs n-FAs. We propose to adapt this procedure for transition reduction.

Extending the results presented in [1], we show that minimizing the number of transitions entering and exiting a single level of an unambiguous n-FA is equivalent to solving an NP-hard problem that we call Weighted Vertices Biclique Decomposition Problem (WVBDP). Given a bipartite graph B, a biclique decomposition [2] of B is a set of bicliques (i.e., complete bipartite sub-graphs) whose respective edges partition B's edges. Given a positive weight function over B's vertices, we define the WVBDP as the computation of a biclique decomposition of B with minimum total weight (i.e., the sum of the weights of the vertices of each biclique). Our reduction procedure is therefore a succession of such level reductions (see [1] for more details) where each step is equivalent to solve a WVBDP.

The direct MIP formulation of the WVBDP that we propose fails to solve the intermediate steps of our procedure on practical `MIP regular` n-FAs in a reasonable amount of time. Therefore, we develop a heuristic to quickly obtain good approximate solutions for large problems. We build bicliques by starting from single uncovered edges and successively add vertices under some conditions. We define $\texttt{regret}(v, c)$ as the regret of adding a vertex v to the open (i.e. currently being built) biclique c rather than to the trivial biclique composed of v and its neighborhood. A biclique c is considered closed when no vertex v have $\texttt{regret}(v, c) \leq 0$ and open bicliques are extended with the vertex with smallest regret value. We establish some properties of our algorithm guaranteeing the convergence of the n-FA transition reduction procedure. The heuristic allows to obtain solutions close to the optimal on random WVBDP instances with small bipartite graphs. Section 3 shows an insight of the performance of the heuristic when used as an intermediate step in the n-FA reduction, which transfers in larger WVBDPs.

3 Numerical Experiments

We test our approach on instances for the Nurse Rostering Problem (NRP) presented in [4]. These instances represent a planning period of 4 weeks in 9 different hospital services. We model five types of working regulations (forward rotation, minimum and maximum consecutive working shifts, minimum consecutive days-off, maximum number of weekends and fixed days-off) as `MIP regular` constraints, while additional constraints are integrated as classical linear inequalities, as presented in [4]. We compute automata in Python3 and solve optimization problems with a time limit of one hour with the version 22.1.1 of CPLEX on one thread of a CPU Intel Xeon e5-2630-v4 from the CCIPL cluster (https://ccipl.univ-nantes.fr/).

Table 1 shows, for each service, the total number of states and transitions in all unambiguous n-FAs (one by employee) obtained with three different reduction procedures: a classical deterministic minimization [5] (`DeterMin`), the Nerode's equivalence generalization for state reduction [1] (`StateRed`), and our adaptation of [1] for transition reduction using the proposed regret heuristic as intermediate step (`TransRed`). Non-deterministic n-FAs allow a more compact representation of the language and our procedure results in a significant reduction in the number of transitions at the expense of a (mild) increase in the number of states when compared to the state reduction procedure.

Service		`DeterMin`			`StateRed`			`TransRed`		
#w	#s	#states	#trans	time	#states	#trans	time	#states	#trans	time
10	2	3155	6322	**0.48**	**2972**	6175	1.01	3027	**5601**	2.34
16	2	4726	9069	**0.58**	**4450**	8856	1.26	4553	**8085**	2.87
18	3	6691	15287	**1.03**	**6324**	14994	1.97	6404	**13144**	6.29
20	3	7679	18290	**1.30**	**7250**	18009	1.73	7465	**15905**	5.89
30	4	12897	32978	**1.91**	**12155**	32424	3.39	12390	**27814**	11.0
36	4	15798	44473	**2.98**	**15290**	44087	4.33	15461	**36957**	14.7
40	5	18754	53676	**3.37**	**17461**	52728	5.14	17924	**44500**	16.6
50	6	22196	74041	**4.20**	**20653**	72973	6.65	21369	**61935**	15.7
60	10	32693	149246	**8.37**	**30383**	147820	13.8	31676	**122263**	31.1

Table 1: Total number of states (#states) and transitions (#trans) for each instance: #w and #s are the number of nurses and shift types in the hospital service, computation time is indicated in seconds.

For each instance, we create 30 additional ones with the same working regulations and minor variations in the demand, vacation, or preferences to represent new planning periods in the same hospital service. Table 2 compares the solving performances for three MIP formulations: the classical model presented in [4], a `MIP regular` formulation with minimal deterministic n-FAs (`DeterMin MIP regular`) and the same `MIP regular` formulation using n-FAs reduced with `TransRed` (`TransRed MIP regular`). We observe that `MIP regular` formulations are generally more efficient when compared to a classical MIP formulation. Also, by comparing `DeterMin MIP regular` and `TransRed MIP regular`, the results show how the proposed unambiguous n-FA transition reduction leads to better MIP solving performances.

4 Conclusion

We propose a procedure for reducing the number of transitions in unambiguous n-FAs. This procedure is adapted from an existing n-FA state reduction approach and requires to solve a WVBDP as an intermediate step. The WVBDP is an NP-hard problem for which we present a regret heuristic in order to quickly find good solutions and therefore

Service		Compact Assignment		`DeterMin MIP regular`		`TransRed MIP regular`	
#w	#s	#sol(gap)	#opt(time)	#sol(gap)	#opt(time)	#sol(gap)	#opt(time)
10	2	30(<1)	29(93.44)	30(<1)	30(12.49)	30(<1)	**30(12.08)**
16	2	30(1.2)	25(615.1)	30(<1)	30(91.60)	30(<1)	**30(67.41)**
18	3	30(<1)	24(945.6)	30(<1)	**30(137.8)**	30(<1)	30(169.4)
20	3	30(5.6)	10(987.6)	30(<1)	30(289.3)	30(<1)	**30(207.0)**
30	4	30(16)	0(—–)	30(4.2)	0(—–)	**30(3.2)**	**1(1791)**
36	4	**28(9.9)**	**11(262.9)**	19(28)	6(1459)	27(23)	8(745.1)
40	5	30(2.7)	19(960.5)	30(<1)	30(231.5)	30(<1)	**30(133.1)**
50	6	30(<1)	29(309.8)	30(<1)	30(757.0)	30(<1)	**30(600.3)**
60	10	**16(18)**	3(1503)	3(<1)	3(1840)	6(<1)	**5(1508)**

Table 2: Solving performances: #sol(gap) = $n(x)$ indicates n integer solutions with a mean optimality gap of x %, #opt(time) = $n(x)$ indicates n optimal solutions with a mean solving time of x seconds (including `MIP regular` n-FAs reduction).

be able to use our n-FA transition reduction to improve `MIP regular` formulations for practical PSPs.

Results on an NRP show that our approach efficiently reduces the number of transitions in the `MIP regular` n-FAs, which leads to mathematical models with less variables. This globally translates into better MIP solving performances when using this type of reduction rather than a classical deterministic minimization approach.

References

1. Amilhastre, J., Janssen, P., Vilarem, M.C.: Fa minimisation heuristics for a class of finite languages. In: International Workshop on Implementing Automata. pp. 1–12. Springer (1999)
2. Amilhastre, J., Vilarem, M.C., Janssen, P.: Complexity of minimum biclique cover and minimum biclique decomposition for bipartite domino-free graphs. Discrete applied mathematics **86**(2-3), 125–144 (1998)
3. Côté, M.C., Gendron, B., Quimper, C.G., Rousseau, L.M.: Formal languages for integer programming modeling of shift scheduling problems. Constraints **16**, 54–76 (2011)
4. Curtois, T., Qu, R.: Computational results on new staff scheduling benchmark instances. Tech. rep., ASAP Research Group, School of Computer Science, University of Nottingham (2014), www.schedulingbenchmarks.org/papers/computational_results_on_new_staff_scheduling_benchmark_instances.pdf
5. Katsirelos, G., Narodytska, N., Walsh, T.: Reformulating global grammar constraints. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research. pp. 132–147. Springer (2009)
6. Menana, J., Demassey, S.: Sequencing and counting with the multicost-regular constraint. In: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems: 6th International Conference, CPAIOR 2009 Pittsburgh, PA, USA, May 27-31, 2009 Proceedings 6. pp. 178–192. Springer (2009)
7. Pandey, P., Gajjar, H., Shah, B.J.: Determining optimal workforce size and schedule at the retail store considering overstaffing and understaffing costs. Computers & Industrial Engineering **161**, 107656 (2021)

8. Pesant, G.: A regular language membership constraint for finite sequences of variables. In: International conference on principles and practice of constraint programming. pp. 482–495. Springer (2004)