Solution-Based Phase Saving and MaxSAT for Employee Scheduling: A Computational Study

Emir Demirović · Felix Winter · Nysret Musliu · Peter J. Stuckey

1 Introduction

Employee scheduling is a well-known problem that appears in a wide-range of areas including healthcare, airlines, transportation services, and other workforce-related institutions. This problem arises whenever there is a need for effective management and distribution of workforce over periods of time. The task is to schedule the working times of employees while respecting the requirements defined by the given institution and labour laws. Finding the ideal workforce roster, however, is not an easy task: even a basic variant of employee scheduling was proved to be **NP**-hard [8]. In this paper, we study the applicability of solution-based phase saving, an existing but not widely-used technique [1], within the linear maxSAT algorithm [17] for employee scheduling. Following the line of work by [14], we model employee scheduling as a propositional Boolean formula, experimenting with four different cardinality constraint encodings, and use a maximum Boolean satisfiability (maxSAT) solver to provide solutions. Our results demonstrate that our approach outperforms previous MaxSAT approaches. Furthermore, we show that MaxSAT solvers can be used to achieve competitive results for many instances when compared with existing metaheuristic techniques.

We concentrate on the employee scheduling instances introduced by Curtois and Qu [9]. According to the authors, those instances were designed to describe realistic and challenging staff scheduling problems while still being

Emir Demirović and Peter J. Stuckey
School of Computing and Information Systems, University of Melbourne, Australia
E-mail: (edemirovic,pstuckey)@unimelb.edu.au

Nysret Musliu and Felix Winter
Christian Doppler Laboratory for Artificial Intelligence and Optimization for Planning and Scheduling
DBAI, TU Wien
E-mail: (musliu,winter)@dbai.tuwien.ac.at

Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling (PATAT-2018), Vienna, Austria, August 28–31, 2018

straightforward to use. The included scheduling periods range from one week up to one year, requiring up to 180 employees and 32 shift types to be assigned.

Different approaches have been studied in the literature. An integer programming formulation was provided by Curtois and Qu [10]. The maxSAT approach described in [14] was able to solve smaller benchmarks, but did not scale well. With the use of a branch and price algorithm [7] and a metaheuristic based on ejection chains [6], optimal solutions could be found for most of the smaller instances and new lower/upper bounds could be determined for many instances. Furthermore, in [21] the authors used constraint programming in an iterated local search algorithm. Another publication [5] proposes a hybridization of ant colony optimization with integer programming methods to solve the instances. There is still room for improvement, as optimal results for a number of instances are still unknown and the best solutions cannot be consistently reached quickly.

In this paper, the focus is laid on using maxSAT for employee scheduling. The Satisfiability problem (SAT) is a fundamental problem in computer science. Given a Boolean formula, the question is whether there exists an assignment of truth values to Boolean variables such that the formula evaluates to *true*. The optimisation variant, maxSAT, rather than returning a binary answer instead aims to find an assignment which maximizes satisfiability. After a problem, in our case employee scheduling, is converted into a propositional formula, a maxSAT solver can be invoked to provide a solution. Given tremendous improvements in solving technology, maxSAT has found a wide-range of applications in the field of combinatorial optimisation, such as timetabling [13, 12, 2], planning, and scheduling, to name a few. See [3, 19] for more details.

The SAT community developed a number of maxSAT solvers [18, 23, 11, 4, 16]. The Linear maxSAT algorithm [17] is the focus of this paper. It computes an optimal solution by repeatedly solving a series of decision (SAT) problems, each time adding constraints that force the new solution to be better than the previously computed one. The intermediate SAT problems are solved in an exhaustive branch-and-bound fashion: iteratively, a variable is selected, followed by a truth value assignment, and then the algorithm either backtracks if a conflict is found or recursively repeats the procedure. Moreover, the algorithm incorporates sophisticated techniques such as conflict-driven clause-learning [25], phase saving [22], VSIDS [20], and other techniques.

In a recent computational study by Demirović and Stuckey [15], the authors found that the linear maxSAT algorithm, when equipped with solution-phase phase saving as opposed to standard phase saving, provides significantly better results for certain classes of problems including timetabling and scheduling, especially in a low amount of time. In SAT solving, standard phase saving is a value-heuristic that instructs the solver, once an unassigned variable is selected, to assign the truth value used most recently for the variable. This is particularly useful for pure SAT problems, as it directs the solver to a *similar* search region after backtracking, where recently learned clauses are effective. However, as shown by Abio [1], *solution-based phase saving* has proven to be more efficient for optimisation, where the assignment is based on the best

	Optiriss	**WPM3**	**SBPS**	**ILS**	**B & P**	**Gurobi**
Instance 1	**607**	**607**	**607**	**607**	**607**	**607**
Instance 2	853	**828**	**828**	**828**	**828**	**828**
Instance 3	3475	2759	1002	1003	**1001**	**1001**
Instance 4	3608	3189	1718	1717	**1716**	**1716**
Instance 5	3645	4037	1237	1237	1160	**1143**
Instance 6	6941	6150	2057	2155	1952	**1950**
Instance 7	5421	4596	1084	1185	1058	**1056**
Instance 8	7617	11018	1635	1552	1308	**1306**
Instance 9	6394	10949	440	461	**439**	**439**
Instance 10	15350	16435	5137	4774	**4631**	**4631**
Instance 11	15420	12183	3568	3467	**3443**	**3443**
Instance 12	28598	18770	5964	4564	4046	**4040**
Instance 13	69203	6110163	53154	3340	o.m.	**2663**
Instance 14	9776	16303	1831	1549	o.m.	**1278**
Instance 15	25975	30833	11300	5289	o.m.	**4843**
Instance 16	13026	10292	3656	3862	3323	**3225**
Instance 17	22073	22002	8267	6597	o.m.	**5749**
Instance 18	20710	18498	5691	5396	o.m.	**5078**
Instance 19	52672	1698538	32470	4918	o.m.	**3591**
Instance 20	161497	5519316	115224	**7504**	o.m.	132445
Instance 21	o.m.	14715064	o.m.	**25017**	o.m.	265504
Instance 22	o.m.	18391055	o.m.	**239762**	o.m.	o.m.
Instance 23	o.m.	o.m.	o.m.	**53127**	o.m.	o.m.
Instance 24	o.m.	o.m.	o.m.	**247977**	o.m.	o.m.

Table 1 Columns 1 to 4 show the final results obtained for Instance 1-24 using MaxSAT solvers WPM3[4], Optiriss[16], Open-WBO with solution-based phase saving (SBPS) and a hybrid approach based on iterated local search and constraint programming[21] (ILS). These experiments were performed within a 60-minute time limit on an Intel Xeon E5345 2.33 GHz machine with a total of 48GB RAM. For some of the larger instances, the solvers went out of memory (o.m.) before a solution could be found. Furthermore, the table includes results from the literature [10, 5] that have been produced within one hour on comparable machine environments with other exact techniques (Branch & Price (B & P) and Gurobi). Recently, improved results for some instances achieved with an ant colony optimization integer programming hybrid have been reported in [5] and additional improved results using integer programming have been announced on the web [9]. However, these results are not included in this comparison as they were conducted with longer or unrestricted running times.

solution found so far. If the previous search was in a space where no better solution exists, time is effectively wasted with standard phase saving. Solution-based phase saving avoids this by searching around the best solution found. We note that standard phase saving is still used until the first solution is found. The behaviour is reminiscent of local search, as the algorithm is directed *near* the best solution, and is particularly relevant when high-quality solutions are expected within tight time budgets. It can also be seen as a kind of Large Neighbourhood Search [24].

We implemented solution-based phase saving for the linear maxSAT algorithm in Open-WBO [18] (labelled as SBPS) and compared its performance against the best maxSAT solvers previously studied in [14], Optiriss [16] and WPM3 [4], and iterated local search (ILS) [21]. Table 1 shows the summarised results of our experiments and compares them with the best results produced by exact techniques in the literature [10,5]. Although solution-based phase saving cannot compete with state-of-the-art integer programming methods in all but one of the instances, we conclude that our proposed approach provides a clear improvement when compared to the previously used maxSAT solvers. As maxSAT solvers have been significantly advancing in the recent years and SBPS is already able to produce comparable results for many instances, we believe that maxSAT based approaches have the potential to become a reasonable alternative to integer programming based methods in the future. Moreover, SBPS is competitive with ILS on smaller benchmarks. On larger benchmarks, however, ILS outperforms SBPS, proving that its domain-specific techniques are increasingly more effective as the size of the benchmarks grows. For future work, incorporating ILS within SBPS would be a possible research direction.

Acknowledgements The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development is gratefully acknowledged.

References

1. Abío Roig, I.: Solving hard industrial combinatorial problems with SAT. Ph.D. thesis, Technical University of Catalonia (UPC) (2013)
2. Achá, R.J.A., Nieuwenhuis, R.: Curriculum-based course timetabling with SAT and maxSAT. Annals of Operations Research **218**(1), 71–91 (2014). DOI 10.1007/s10479-012-1081-x
3. Ansótegui, C., Bonet, M.L., Levy, J.: Solving (weighted) partial maxsat through satisfiability testing. In: Proceedings of SAT'09, pp. 427–440
4. Ansótegui, C., Gabàs, J.: WPM3: an (in)complete algorithm for weighted partial maxsat. Artificial Intelligence journal **250**, 37–57 (2017)
5. Bunton, J., Ernst, A.T., Krishnamoorthy, M.: An integer programming based ant colony optimisation method for nurse rostering. In: Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, pp. 407–414 (2017)
6. Burke, E.K., Curtois, T.: New approaches to nurse rostering benchmark instances. European Journal of Operational Research **237**(1), 71–81 (2014)
7. Burke, E.K., Curtois, T., Post, G.F., Qu, R., Veltman, B.: A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. European Journal of Operational Research (2), 330–341 (2008)

8. Chuin Lau, H.: On the complexity of manpower shift scheduling. Computers & Operations Research **23**(1), 93–102 (1996)
9. Curtois, T.: Staff scheduling benchmark instances (2014). URL http://www.schedulingbenchmarks.org/. Accessed: 2018-03-11
10. Curtois, T., Qu, R.: Computational results on new staff scheduling benchmark instances. Tech. rep., ASAP Research Group, School of Computer Science, University of Nottingham, NG8 1BB, Nottingham, UK (2014)
11. Davies, J., Bacchus, F.: Exploiting the power of mip solvers in maxsat. In: Proceedings of SAT'13, pp. 166–181
12. Demirović, E., Musliu, N.: MaxSAT based large neighborhood search for high school timetabling. Computers & Operations Research **78**, 172–180 (2017)
13. Demirović, E., Musliu, N.: Modeling high school timetabling as partial weighted maxSAT. technical draft - extended LaSh paper, (2017). URL https://cloudstor.aarnet.edu.au/plus/s/fqYugNB7KheF7w2
14. Demirović, E., Musliu, N., Winter, F.: Modeling and solving staff scheduling with partial weighted maxsat. Annals of Operations Research (2017)
15. Demirović, E., Stuckey, P.J.: Local-style search in the linear maxSAT algorithm: A computational study of solution-based phase saving. technical draft URL https://cloudstor.aarnet.edu.au/plus/s/tOL2Nb6E2q0HzFz
16. Kahlert, L., Krüger, F., Manthey, N., Stephan, A.: Riss solver framework v5. 05. SAT-Race (2015)
17. Le Berre, D., Parrain, A.: The Sat4j library, release 2.2 system description. Journal on Satisfiability, Boolean Modeling and Computation **7**, 59–64 (2010)
18. Martins, R., Manquinho, V., Lynce, I.: Open-WBO: a modular maxSAT solver. In: Proceedings of SAT-14, pp. 438–445
19. Morgado, A., Heras, F., Liffiton, M.H., Planes, J., Marques-Silva, J.: Iterative and core-guided maxsat solving: A survey and assessment. Constraints **18**(4), 478–534
20. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engineering an efficient SAT solver. In: Proceedings of DAC'01, pp. 530–535
21. Musliu, N., Winter, F.: A hybrid approach for the sudoku problem: Using constraint programming in iterated local search. IEEE Intelligent Systems **32**(2), 52–62 (2017)
22. Pipatsrisawat, K., Darwiche, A.: A lightweight component caching scheme for satisfiability solvers. In: Proceedings of SAT'07, *Lecture Notes in Computer Science*, vol. 4501, pp. 294–299. Springer
23. Saikko, P., Berg, J., Järvisalo, M.: LMHS: A SAT-IP hybrid maxsat solver. In: Proceedings of SAT'16, pp. 539–546
24. Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: M. Maher, J.F. Puget (eds.) Proceedings of CP'98, pp. 417–431. Springer
25. Silva, J.P.M., Lynce, I., Malik, S.: Conflict-driven clause learning SAT solvers. In: Handbook of Satisfiability, pp. 131–153 (2009). DOI 10.3233/978-1-58603-929-5-131