Applications of a flow-based algorithmic framework for real-world resource scheduling

Georgia Lazaridou * Alexander Souza (Speaker) * Denys Trieskunov *
Hsuan-Pin Wu *

1 Introduction

Resource scheduling is one of the central operational and planning challenges for many industries, e.g., logistics, transportation, health-care, entertainment, and others. We have designed and implemented a flow-based algorithmic framework and put it into practice for several customers. It is especially well suited whenever non-movable tasks, e.g., shifts, have to be allocated to resources, e.g., staff, as to minimize some (linear) cost function, subject to respecting a mix of local and global constraints over the tasks. Specific applications are CREW DIAGRAMMING and SHIFT SCHEDULING problems. The algorithms we found yield very favorable running times and consistently produce high-quality solutions in practice. We are confident that the approach will prove useful for further applications, beyond the two we just mentioned.

2 Edge Constrained Flow

We call the overall approach EDGE CONSTRAINED FLOW: We are given a directed graph G with edge-capacities and edge-costs. Furthermore, there is a collection $\mathcal{S}$ of edge-subsets given. Two numbers are defined for each subset $S \in \mathcal{S}$: a lower bound $\alpha(S)$ and an upper bound $\beta(S)$. We seek a MINIMUM COST FLOW f, respecting flow-conservation at the vertices, capacity-restrictions at the edges, and flow-bounds $\alpha(S) \leq f(S) \leq \beta(S)$ on the edge-subsets (where $f(S) = \sum_{e \in S} f(e)$), as to minimize the flow-edge-costs. Notice that SET COVER is a direct special case. Thus EDGE CONSTRAINED FLOW is NP-complete (in its decision version).

2.1 Crew Diagramming

This research originates from our work at the Swiss Federal Railways (SBB). Our optimizer is used for annual driver- and conductor-planning and for strategic simulations.

The CREW DIAGRAMMING problem asks to partition a set of crew tasks into a set of crew diagrams, while respecting temporal consistency, geographic consistency, break requirements, and other labor rules. The objective is to minimize a weighted sum over the crew diagrams. The problem has attracted substantial research, see, e.g., [2, 5, 7, 3,

---

* { `georgia | alexander | denys | hsuan-pin }@algomia.com` Algomia GmbH, Schachenallee 29, CH-5000 Aarau, Switzerland.

4, 6]. The dominant approach is column-generation [8], with SET COVER as the master problem [2, 3, 6] and the NP-complete RESOURCE CONSTRAINED SHORTEST PATH as the subproblem [10, 3, 9, 6].

We applied the EDGE CONSTRAINED FLOW approach as follows: The graph captures the possible transitions between the tasks. States stored on the vertices represent the relevant history of a crew diagram that contains some specific task. A source vertex marks the start of a planning day at a depot, while a target vertex marks its end there. By construction, each path from the source to the target is a valid crew diagram. Capacity constraints ensure that at most one unit of flow passes through each vertex. Hence any integral flow decomposes into disjoint paths. For each crew task, there is a set S defined of the edges that lead to vertices containing the task. The set bound constraints $\alpha(S) = 1$ and $\beta(S) = 1$ ensure that each crew task occurs in exactly one vertex with positive flow.

2.2 Shift Scheduling

This application of the EDGE CONSTRAINED FLOW framework culminated into the creation of our software-as-a-service product SHIFTADVISER (`https://shiftadviser.com`). There, we serve small- and medium sized companies in their monthly duty plannig. Examples include a Swiss casino and a local traffic operator.

The SHIFT SCHEDULING problem asks to assign a collection of shift tasks to staff while respecting preferences, skills, availabilities, and minimum rest durations, as to minimize the number of unassigned shifts.

We model the problem as follows: Each staff gets its own connected component in the graph, with a source and target vertex. Each ordinary vertex represents a shift. Then there are layers of vertices for the admissible shifts. The edges represent the possible transitions. A path from the source to the target captures a sequence of the shifts assigned to the staff. For each shift, a set S collects all edges that lead to it. The set bound constraints $\alpha(S) = 0$ and $\beta(S) = 1$ ensure that each shift is assigned at most once. A negative cost (a reward) is defined for each shift.

2.3 Algorithmics Aspects

We have designed and implemented several algorithms that solve EDGE CONSTRAINED FLOW. A direct INTEGER LINEAR PROGRAMMING implementation is capable to solve small to medium sized instances, i.e., with about 500'000 edges, in several minutes.

To increase the admissible instance size further, we resort to approximations: Our most successful approach is to solve the LINEAR PROGRAMMING relaxation, to use the fractional flow for thinning the graph, and then to solve the problem exactly. This simplification procedure yields that we can solve instances with up to 6-10 million edges in several minutes up to a few hours.

To push the limits even further, we resort to column generation. The master problem is given by the set bound constraints on the edges of the graph. The graph itself can be decomposed in a suitable manner into connected components. Each component represents a subproblem requiring a solution of the MINIMUM COST FLOW problem. A treatment is given in the master thesis of Wu [11]. Thus, a major advantage of our approach is that we only have polynomial time solvable subproblems, instead of NP-complete ones. (For example, this is in sharp contrast to the dominant approach for Crew Diagramming, see above.) The subproblems can be solved independently. Hence we have implemented a parallelization mechanism. This allows either computation on local hardware or to use cloud-based compute-nodes that are started on the fly. In order to gain an integer solution, we have implemented several rounding procedures that meet our speed-requirements. (Branch-and-Bound yields prohibitive running times.) To accelerate the column generation further, we used a machine learning heuristic which decides upon the set of subproblems to be solved in the respective next iteration. The appraoch yields a significant reduction of the number of subproblems that are actually executed and solved. See the master thesis of Trieskunov [12] for details.

References

[1] R. Kwan (2011). Case studies of successful train crew scheduling optimisation. J. Scheduling. 14(5). 423-434. pp. 49-54, 2011.

[2] B.M. Smith and A. Wren (1988). *A bus crew scheduling system using a set covering formulation.* In: Transportation Research, vol. 22A, pp. 97-108, 1988.

[3] A.S.K. Kwan (1999). *Train Driver Scheduling*, PhD Thesis, University of Leeds

[4] G. P. Silva, N. D. F. Gualda and R. Kwan (2000). *Bus Scheduling Based on an Arc Generation – Network Flow Approach*, 8th International Conference on Computer-Aided Scheduling of Public Transport - CASPT 2000

[5] J.E. Beasley and B. Cao (1996). *A tree search algorithm for the crew scheduling problem*, J. Operational Research 94, pp. 517 – 526, 1996.

[6] S. Jütte (2012). *Large-Scale Crew Scheduling: Models, Methods, and Applications in the Railway Industry*, Springer Verlag, 2012.

[7] R. Borndörfer, M. Grötschel, and A. Löbel (1998). *Optimization of Transportation Systems.* Technical Report ZIB-Report 98-09, Zuse-Institut Berlin, 1998

[8] G. B. Dantzig and P. Wolfe (1960). *Decomposition Principle for Linear Programs.* Operations Research, 8(1):101–111, 1960.

[9] S. Irnich and G. Desaulniers (2005). *Shortest Path Problems with Resource Constraints.* In: Desaulniers G., Desrosiers J., Solomon M.M. (eds) Column Generation. Springer, Boston, MA, 2005.

[10] M. Desrochers (1986). *La Fabrication d'horaires de travail pour les conducteurs d'autobus par une méthode de génération de colonnes.* Ph.D Thesis, Centre de recherche sur les Transports, Publication 470, Université de Montréal, Canada, 1986.

[11] H.-P. Wu (2022). *Alternative Algorithmic Approaches for Crew Diagramming.* MSc Thesis, University of Zurich, 2022.

[12] D. Trieskunov (2023). *Machine Learning Approaches to Accelerate Column Generation.* MSc Thesis, University of Zurich, 2023.