A Decision Support System Prototype for Automated Bus Driver Scheduling

Nikolaus Frohner[1], Esther Mugdan[1], Lucas Kletzander[1], and Nysret Musliu[1]

Christian Doppler Laboratory for Artificial Intelligence and Optimization for Planning and Scheduling, TU Wien, Vienna, Austria
{firstname.lastname}@tuwien.ac.at

Keywords: Bus driver scheduling, Multi objective optimization, Multi criteria decision making, Decision support systems

The bus driver scheduling problem (BDSP, [11]) deals in its essence with assigning bus tours' legs to drivers. Such an assignment induces a shift for each driver, consisting of categorized pieces of work like active driving time, manipulation time, passive ride time to change between tours, breaks and shift splits (see Figure 1). Feasible assignments must adhere to labor law and regulations imposed by collective agreements, such as a maximum drive time and a sufficient number of breaks depending on the duration and layout of a shift. For assignments to be implemented in practice, they must be economically viable for the bus operator. At the same time, employees need to be happy with their shifts to avoid absenteeism and unnecessary staff fluctuation.

In an ongoing project, we develop and study different features of a decision support system (DSS) prototype for automated BDS together with the personnel planning consultancy company XIMES. The concrete problem variant at hand was introduced by Kletzander et al. [4] featuring complex break constraints and seven different objectives to capture the main qualities of an assignment: the paid time t^{paid} (corresponding to the actual costs), the wasteful time paid up to the minimum shift length of six hours if they are shorter t^{mpaid}, the overall shift span including unpaid breaks t^{shift}, the number of employees n^{emp}, the number of shift splits n^{splits}, where we have a long unpaid break, the number of tour changes employees have to perform n^{change}, and the passive ride time t^{ride}.

So far, this BDSP variant has been studied with domain experts and a well-defined single weighted-sum objective with manually tuned weights in an expensive trial-and-error phase. The current state-of-the-art approaches are based on Branch and Price by Kletzander et al. [6] and a large neighbourhood search (LNS) by Mazzoli et al. [8]. Instead, we focus on finding assignments using a DSS without having to explicitly state any prior preferences by weights but by setting goals for different objectives. Furthermore, the DSS should help visualize a diverse set of solutions, facilitate learning about dependencies between objectives, and suggest concrete hints on how to adjust overly optimistic goals. To this end, we adopt and extend three different decision support methods:

- *Automated Weight Tuning (AWT)*: A recent automated approach introduced by Böðvarsdóttir et al. [2] and extended by Kletzander et al. [5] which performs intertwined

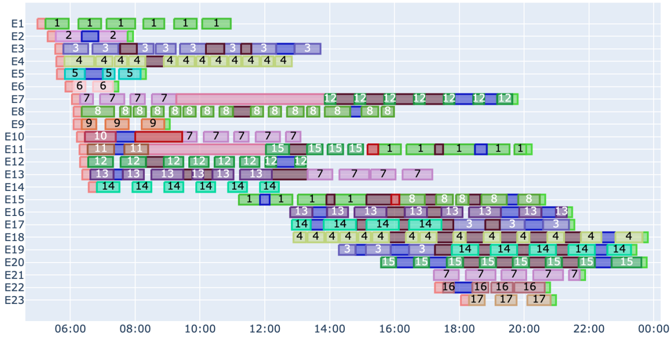

Fig. 1: Example solution to BDS instance with 23 employees and 17 tours. Active driving times are tour-numbered blocks, brown/blue blocks are unpaid/paid breaks, red blocks are passive ride time for a tour change, and long pink blocks are unpaid shift splits.

violation-dependent weight updates and optimization runs until an acceptable solution is found or too-conflicting goals are identified and reported back to the decision maker (DM). Acceptability is defined by a feasible solution that meets current thresholds on objectives, which are updated interactively until the DM is satisfied.
- *Reference eXplainable Interactive Multiobjective Optimization (R-XIMO)*: A reference point based method by Misitano et al. [10]. First, the Pareto front is approximated to quickly retrieve solutions by a scalarization function, taking a reference point as input. SHAP values [7] are then used to quantify the contribution of different input dimensions to the output and are converted into a hint on how to update the reference point to improve a selected objective.
- *R-XIMO with Shapley regression values*: Mischek and Musliu [9] extend the R-XIMO approach by directly using Shapley regression values instead of SHAP values using a black box predictor which permits missing inputs. Their experiments use weights as a preference structure to retrieve a solution from the Pareto front. Then, Shapley values identify a rival of a desired target that the DM wishes to improve. This leads to improvements more frequently for a test laboratory scheduling problem than updating only the target's weight.

First results. AWT consists of an exploratory phase taking time t^{exp} of intertwined shorter optimization runs, in our case using simulated annealing (SA) and weight updates depending on the current violations, either hard or soft. A final longer SA run of duration t is performed using the weights of the best-found solution during exploration. Table 1 shows the result of eight AWT runs over 50 BDS instances from [4]. BDS-1 is the problem variant without thresholds on the objectives starting from hard and soft constraint weights all-equal-1 $\mathbf{1}$ and $k = 3$ exploration threads. The acceptance rate, how often an acceptable solution was found, is denoted by r^{acp}.

The values of the objectives are normalized instance-wise using gross leg times L (including bus idle times), either by the total sum or in units of 8 h blocks. In BDS-1, we

Table 1: Interactive AWT for BDS instance-average results with acceptance rate r^{acp} using k threads and uniform ($\mathbf{1}$) or learned ($\boldsymbol{\mu}_{h,s}$) initial weights $\boldsymbol{w}_0$ within n^{it} iterations and (exploration) runtimes (t^{exp}) t in minutes. Further solution metrics minimum paid, work time, span, number of employees, shift split frequency, tour changes, and ride time.

pid	*thresholds*	k	$\boldsymbol{w}_0$	r^{acp} [%]	$\overline{n^{\mathrm{it}}}$	$\overline{t^{\mathrm{exp}}}$ [′]	$\bar{t}$ [′]	$\overline{t_{\mathrm{L}}^{\mathrm{mpaid}}}$	$\overline{t_{\mathrm{L}}^{\mathrm{work}}}$	$\overline{t_{\mathrm{L}}^{\mathrm{paid}}}$	$\overline{t_{\mathrm{L}}^{\mathrm{shift}}}$	$\overline{n_{\mathrm{L8h}}^{\mathrm{emp}}}$	$\overline{T_{\mathrm{L8h}}^{\mathrm{splits}}}$	$\overline{n_{\mathrm{L8h}}^{\mathrm{change}}}$	$\overline{t_{\mathrm{L8h}}^{\mathrm{ride}}}$
BDS-1		3	$\mathbf{1}$	100	3.1	3.4	13.5	0.44	0.97	1.41	**1.02**	1.68	**940.3**	**0.29**	**0.56**
BDS-2	$t_{h^{\mathrm{emp}}}$	3	$\mathbf{1}$	98	8.4	9.2	19.2	0.16	0.96	1.13	1.13	1.34	10.4	2.69	12.07
BDS-3	$+t_{s^{\mathrm{change}}}$	3	$\mathbf{1}$	86	9.2	10.0	20.0	0.23	0.98	1.20	1.11	1.38	18.2	1.03	11.07
			$\boldsymbol{\mu}_h$	100	3.8	4.2	14.2	0.21	0.96	1.17	1.08	1.38	35.9	0.90	7.13
BDS-4	$+t_{s^{\mathrm{mpaid}}}$	3	$\boldsymbol{\mu}_h$	84	9.5	10.2	20.3	**0.02**	1.11	1.13	1.47	**1.27**	2.9	1.32	52.77
BDS-5	$+t_{s^{\mathrm{span}}}, t_{s^{\mathrm{splits}}}$	1	$\boldsymbol{\mu}_h$	90	7.3	7.3	17.4	0.17	**0.96**	**1.13**	1.09	1.32	23.3	1.05	9.88
		3	$\boldsymbol{\mu}_h$	98	4.4	4.7	14.8	0.17	0.97	1.14	1.09	1.33	27.9	0.99	10.85
			$\boldsymbol{\mu}_{h,s}$	98	**2.4**	**2.6**	**12.7**	0.16	0.97	1.13	1.09	1.33	30.2	1.05	10.76

see that 1.68 employees are used on average to serve a gross 8 h block, with 0.29 tour changes per 8 h block and very rare shifts splits (every $T^{\mathrm{splits}} = 940$th block on average). There are way too many employees with too short shifts. Therefore, for BDS-2, the DM sets a threshold of 1.35 on the number of employees per 8 h block. This leads to a desired massive reduction in costs per gross leg time (from 1.41 down to 1.13) at the inconvenience of quite frequent shift splits (every 10th block) and tour changes (2.7 per block). Hence, the DM introduces another threshold in BDS-3 to reduce the tour changes, which mildly increases costs and number of employees. Acceptable solutions are found slightly less frequently, in 86% of cases. This process is continued by adding two more thresholds until the DM is satisfied. The initial hard and soft weights are either all-equal-1 or taking (hard or hard/soft) centroid weights $\boldsymbol{\mu}$ derived from AWT runs on separate training data set to speed up the search of newly occurring online instances. The impact of using parallel weight updates ($k = 1$ vs $k = 3$) and such learned centroid weights is best seen for BDS-5, where the mean number of iterations goes down from

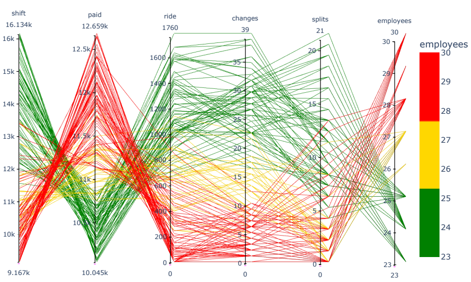

Fig. 2: Parallel coordinate plot of non-dominated solutions for a BDS instance.

7.3 to 2.4, also increasing the acceptance rate to almost 100%, to achieve the goals set by the DM.

The other DSS features require an initial approximation of the Pareto front for a given instance. A parallel coordinates plot [1] of non-dominated solutions created by a first mutation-only evolutionary algorithm with NSGA-II selection rule shows first promising results in Figure 2. The tradeoff between paid time/number of employees and the shift ergonomy aspects are visible. Current work deals with a comparison with Pareto Simulated Annealing (PSA) [3] and the implementation of other DSS approaches [10,9] with Shapley value based guidance to update preference structures like reference points and weights.

Acknowledgments The financial support by the Austrian Federal Ministry of Labour and Economy, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

References

1. Bagajewicz, M., Cabrera, E.: Pareto optimal solutions visualization techniques for multiobjective design and upgrade of instrumentation networks. Industrial & engineering chemistry research **42**(21), 5195–5203 (2003)
2. Böðvarsdóttir, E.B., Smet, P., Berghe, G.V.: Behind-the-scenes weight tuning for applied nurse rostering. Operations Research for Health Care **26**, 100265 (2020)
3. Czyzżak, P., Jaszkiewicz, A.: Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. Journal of multi-criteria decision analysis **7**(1), 34–47 (1998)
4. Kletzander, L., Musliu, N.: Solving large real-life bus driver scheduling problems with complex break constraints. In: Proceedings of the International Conference on Automated Planning and Scheduling. vol. 30, pp. 421–429 (2020)
5. Kletzander, L., Musliu, N.: Dynamic weight setting for personnel scheduling with many objectives. Proceedings of the International Conference on Automated Planning and Scheduling **33**(1), 509–517 (2023)
6. Kletzander, L., Musliu, N., Hentenryck, P.V.: Branch and price for bus driver scheduling with complex break constraints. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Event, February 2-9, 2021. pp. 11853–11861. AAAI Press (2021)
7. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. Advances in neural information processing systems **30** (2017)
8. Mazzoli, T.M., Kletzander, L., Van Hentenryck, P., Musliu, N.: Investigating large neighbourhood search for bus driver scheduling. In: Proceedings of the Internat. Conference on Automated Planning and Scheduling. vol. 34, pp. 360–368 (2024)
9. Mischek, F., Musliu, N.: Preference explanation and decision support for multi-objective real-world test laboratory scheduling. In: Proceedings of the International Conference on Automated Planning and Scheduling. vol. 34, pp. 378–386 (2024)
10. Misitano, G., Afsar, B., Lárraga, G., Miettinen, K.: Towards explainable interactive multiobjective optimization: R-XIMO. Autonomous Agents and Multi-Agent Systems **36**(2), 43 (2022)
11. Wren, A., Rousseau, J.M.: Bus driver scheduling—an overview. In: Computer-Aided Transit Scheduling: Proceedings of the Sixth International Workshop on Computer-Aided Scheduling of Public Transport. pp. 173–187. Springer (1995)