Towards Multi-Objective Optimization for Rotating Workforce Scheduling

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

[a]Christian Doppler Laboratory for Artificial Intelligence and Optimization for Planning and Scheduling
TU Wien, Vienna, Austria

Abstract. While the satisfaction version of the Rotating Workforce Scheduling (RWS) problem has been extensively studied, real-life applications of the problem often require the study of an optimization variant, considering stakeholder preferences. This requires the use of multi-objective methods to solve the problem. In particular, the introduction of soft constraints often leads to conflicting objectives and, therefore, to a large pool of solutions of varying quality in different dimensions. Hence, it is important to be able to find suitable solutions that best meet the requirements for conflicting objectives. We propose the use of Pareto Simulated Annealing (PSA) to solve the optimization variant of RWS. This allows us to study large sets of solutions that approximate the Pareto front. Specifically, different solutions and the relations between soft constraints can be analyzed. In addition, this approach allows stakeholders to choose solutions that best meet the desired properties and comprehend which trade-offs must be made when trying to optimize conflicting objectives.

1 Introduction

Workforce scheduling is a topic of high practical importance as the resulting schedules provide the structure for a significant part of the employee's lives. Many different versions of such problems have been addressed over the years [9, 24, 7]. While traditionally the focus has been on reducing cost, a growing body of work on employee well-being [6] shows the importance of incorporating these aspects. It is known that effects like fatigue due to bad workplace conditions impact not only economic costs of companies [23] but also lead to severe consequences in both psychological and physiological health of individuals [17, 12], including a higher risk for certain diseases and disorders [19], and reduced social contacts with family and friends [1]. Therefore, practical guidelines [13] and metrics have been proposed to evaluate risk-related characteristics [10, 11].

Petrovic et al. [22] presented case studies on personnel planning focusing on the impact of including well-being aspects in the objective function for heuristic optimization approaches. Shifts are created by minimizing a weighted sum of violations of undesired characteristics, like working weekends, long night shift stints, and others.

However, the increasing number of objectives leads to a difficult process of tuning the weights in such an approach, and weighted sum objectives can not be used to show the trade-offs and dependencies between different objectives. This leads to the use of multi-objective methods, which allow the generation of a whole set of solutions, approximating the Pareto front. These solutions allow to show the different relations between objectives to the decision makers. There are only few lines of work considering these approaches in the area of employee scheduling, among them are Burke et al. [4], Li et al. [16].

The Rotating Workforce Scheduling (RWS) problem can be classified as a single-activity tour scheduling problem with non-overlapping shifts and rotation constraints [2]. Over the years, several approaches have been used to solve the problem, including use in commercial software for almost 20 years [20]. At this point, only the satisfaction version of the problem was studied, where only hard constraints need to be fulfilled. A complete method based on constraint programming (CP) was introduced by Musliu et al. [21] and further extended by Kletzander et al. [14], in particular by introducing several optimization goals used in practice, turning the satisfaction problem into an optimization problem. The CP method was used in an instance space analysis by Kletzander et al. [15], leading to the creation of new instances to better cover the transition between feasible and infeasible instances. The current state-of-the-art approach is a branch-and-cut (b&c) framework by Becker et al. [3], which reduces the runtime substantially compared to previous methods for the satisfaction version and a limited set of optimization goals.

While the satisfaction version of RWS is rapidly solved with b&c, in practice, a range of different objectives are required to judge solutions for the problem. These include non-local properties like the distance between free weekends, which are difficult to incorporate into the exact approaches. Therefore, we propose to use Pareto Simulated Annealing (PSA) to solve the multi-objective version of RWS.

We apply PSA to an extended version of the RWS problem, considering six different soft constraints. We mainly focus on a combination of three conflicting constraints and aim to improve and adapt the general PSA algorithm introduced by Czyżak and Jaszkiewicz [5] We explore how different configurations enhance the results of PSA. To do so we evaluate the configurations on three problem instances of various sizes and compare the hypervolumes and other metrics of the resulting solution sets.

2 Problem Domain

The Rotating Workforce Scheduling domain deals with problems where shifts are fixed. These have to be assigned to employees according to several constraints such as allowed sequences of shifts and limitations to consecutive shift assignments. In many applications, a rotating schedule, where each employee rotates through the same sequence of shifts with different offsets, is a preferred way of scheduling. We provide a formal definition of the problem and introduce new objectives that focus on employee well-being.

* Corresponding Author. Email: esther.mugdan@tuwien.ac.at

Formal definition. The satisfaction version of RWS is based on definitions and notation by Musliu et al. [20, 21]:

- n: Number of employees.
- d: Length of the schedule. The total length of the planning period is $n \cdot d$, as each employee rotates through all n rows. We set $d = 7$, corresponding to the number of days in a week.
- $\mathbf{A}$: Set of work shifts (activities), enumerated from 1 to m. A day off is denoted by a special activity O with numerical value 0, $\mathbf{A}^+ = \mathbf{A} \cup \{O\}$.
- T: Temporal requirements matrix, an $m \times d$-matrix where each element $T_{i,j}$ corresponds to the number of employees that need to be assigned shift $i \in \mathbf{A}$ at day j.
- ℓ_w and u_w: Minimal and maximal length of blocks of consecutive work shifts.
- ℓ_a and u_a: Minimal and maximal lengths of blocks of consecutive assignments of shift a for each $a \in \mathbf{A}^+$.
- $\mathbf{F}_2$ and $\mathbf{F}_3$: Sequences of shifts of length 2 and 3 that are forbidden in the schedule (e.g. N D, a night shift followed by a day shift). This is typically required due to legal or safety concerns.

The task is to construct a cyclic schedule S, represented as an $n \times d$-matrix, where each $S_{i,j} \in \mathbf{A}^+$ denotes the shift or day off that employee i is assigned during day j in the first period of the cycle. An exemplary schedule can be found in Table 1.

A feasible solution must cover all requirements of T, such that no employee is assigned to multiple shifts per day, and meet the hard constraints induced by $\ell_\mathrm{w}, u_\mathrm{w}, \ell_a, u_a$ and $\mathbf{F}_2, \mathbf{F}_3$.

Table 1. Example of a rotating workforce schedule

Empl.	Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	D	D	D	D	N	N	-
2	-	-	A	A	A	A	N
3	N	N	-	-	D	D	D
4	A	A	N	N	-	-	-

Soft constraints. We focus on the optimization variants and extend the problem introduced by Kletzander et al. [14] with soft constraints to be minimized. These objectives aim to increase well-being of employees and meet applicable best practices of Britain's national regulator for workplace health and safety (HSE) shift work guidelines [13]. We consider the following soft constraints:

- $s^{\mathrm{N}>3}$: Minimize the number of consecutive night shifts exceeding 3.
- s^{ℓ_dev}: Minimize the squared deviation of working sequence lengths from length 5.
- s^{ww}: Minimize the number of working weekends. A weekend is free if Saturday and Sunday are off. Otherwise, it is a working weekend.
- $s^{\mathrm{d_{max}}}$: Minimize the maximum distance $d_{\max}$ between consecutive free weekends ($d_{\max} = n + 1$ if no weekend is free).
- $s^{\mathrm{d_{rms}}}$: Minimize the root weeks mean of the sum of squared distances between weekends $\sqrt{1/n \sum_{i=1}^{n} \hat{d}_i^2}$, where $\hat{d}_i$ is the distance minus 1 to the next free weekend if weekend i is free, and n otherwise.
- s^{Nww}: Minimize the number of working weekends, including Friday night.

Classically, a solution has a weight vector λ and a violation vector v. An optimal solution minimizes the objective function given in Equation 1. In order for the solution to be feasible, the first sum must be equal to zero.

$$obj = \sum_{h \in \substack{\text{Hard} \\ \text{constraints}}} \lambda_h \cdot v_h + \sum_{s \in \substack{\text{Soft} \\ \text{constraints}}} \lambda_s \cdot v_s \tag{1}$$

In our case, we are interested in feasible Pareto-optimal solutions. To determine the non-dominance of solutions, we do a pairwise comparison of the unweighted violations v_i for each constraint i. We enforce that feasible solutions always dominate infeasible ones.

3 Method

Previously, the Rotating Workforce Scheduling (RWS) problem was mainly solved with methods that generate a single solution based on pre-specified weight vectors. Our interest lies in generating multiple non-dominated solutions from which the decision maker can select a suitable solution or learn about trade-offs between different objectives. To achieve this, we adapt the Pareto Simulated Annealing (PSA) framework [5] for RWS to generate sets of solutions that approximate the Pareto fronts in different multidimensional objective spaces.

PSA performs Simulated Annealing (SA) runs in parallel with a set of generating solutions G while keeping a non-dominated solutions archive S. In each iteration, a move is applied to every solution $x \in G$, generating neighbouring solutions. A new solution y that is generated is compared to the current set of non-dominated solutions S. The set S is updated according to the hard constraint violations and soft objectives of y. If y is not dominated by any solution, it is added to S, and any solution dominated by y is removed.

The applied move is accepted, resulting in the replacement of x by y as generating solution in G if one of the following applies:

- y is added to the set of non-dominated solutions S. (This criterion was added and is not proposed in the original implementation)
- y dominates x.
- with probability sl (given in Equation 2), which depends on the current temperature t.

$$sl(x, y, t) = \min\left(1, e^{\sum_{i=0}^{k} \frac{\lambda_i(x) \cdot (v_i(x) - v_i(y))}{t}}\right) \tag{2}$$

Every solution has its own weight vector λ, which is updated in each iteration. This should lead to solutions diverging into different directions within the solution space and allow for better coverage of the space. For each solution, x, the closest neighbouring solution $x' \in S$, which is not dominated by x, is determined. This is classically done by comparing the absolute distance in terms of violations of two solutions, i.e. $\|v(x) - v(x')\|_1$. The weights of x are updated according to the violations of solution x' using the update factor α (see Equation 3).

$$\lambda_i(x) = \begin{cases} \lambda_i(x) \cdot \alpha & \text{if } v_i(x) \leq v_i(x') \\ \lambda_i(x)/\alpha & \text{otherwise} \end{cases} \tag{3}$$

We additionally propose an alternative approach to updating the weight vectors, where the closest neighbour is determined by comparison of the weight vectors, i.e. x' st. $\|\lambda(x) - \lambda(x')\|_1$ is minimal. This can lead to a more diverse distribution of solutions in the search space, especially when considering a small number of dimensions.

Lastly, the weight vectors are normalized such that the sum of weights equals 1. We enforce a minimum weight of 0.001, an addition recently introduced by Mischek and Musliu [18]. This ensures that every constraint is still relevant during the search.

A further modification, initially proposed by Drexl and Nikulin [8], consists of restarting generating solutions $x \in G$. For this variant of PSA solutions x, which have not contributed new solutions to the set of non-dominated solutions S for a certain number of consecutive iterations, are replaced by a random solution from S. The aim of this addition is to replace 'bad' generating solutions with new generating solutions that are more promising.

4 Computational Study

We now present the application of PSA to the RWS problem. Our computational testbed was a cluster running Ubuntu 22.04.2 LTS with 2× Intel Xeon CPU E5-2650 v4 (2.2 GHz, 12 physical cores, no hyperthreading). The algorithm was implemented in Python 3.9 and run with the fast PyPy[1] interpreter.

We evaluate our approach on three different real-life RWS instances[2] of different size and difficulty. The first instance (instance 10) only considers $n = 27$ employees, while the second (instance 15) and third (instance 20) deal with 64 and 163 employees. As initial construction solutions we use solutions generated using the branch-and-cut [3] approach.[3] This ensures that all our construction solutions are feasible, allowing us to concentrate on the optimization towards soft-constraints. Additionally, this ensures that all solutions added to the set of non-dominated solutions are feasible, as no infeasible solution can dominate a feasible solution. For our experiments, we use 8 generating solutions, which should allow for a good tradeoff between runtime and quality of the solution set [5]. Additionally, all experiments are run with a starting temperature $t = 1$, cooling factor $\beta = 0.999$ and 1 million iterations. When the temperature falls below 10^{-5}, t is reheated to the starting temperature of 1. We use a fixed weight of 5 for hard constraints and only update weights for soft constraints using the factor $\alpha = 1.05$. The initial soft constraint weights are chosen at random for each generating solution.

For the application of moves, we consider *PeriodIntervalSwaps*. An application swaps two intervals of length 1 to 7 (number of days) that differ by a period of 7, meaning two sequences of consecutive shifts are swapped between two employees. Thereby, the number and type of shifts assigned to each day remain the same, as only shifts between the same days of the week are swapped.

We have conducted brief experiments, including all of the proposed soft constraints. Based on the most conflicting constraints, length deviation (ℓ_{dev}), working weekends (ww) and weekend distance ($d_{\max}$) we have decided to focus on the quality of the solutions for the 3-dimensional case.

The quality of the generated solution is evaluated by using the hypervolume metric. The hypervolume indicates the percentage of the multidimensional objective space, which is dominated by the approximated Pareto front. The size of the whole objective space is determined by the ideal(min) and anti-ideal (max) points. The dominated space spans from the anti-ideal point to the approximated front. The (anti-)ideal point in every dimension is given minimal (maximal) violation value of the corresponding constraint that a solution can have. The ideal points for s^{ww} and $s^{\ell_{\text{dev}}}$ were calculated using the branch-and-cut framework by Becker et al. [3], to which we added the squared length deviation minimization. Also, the anti-ideals for s^{ww} and $s^{\ell_{\text{dev}}}$ could be determined by using the branch-and-cut approach but by aiming to maximize the violation. Lastly, the ideal for $s^{d_{\max}}$ is given by the ratio between working and free weekends $\lceil \frac{min(s^{\text{ww}})}{n-min(s^{\text{ww}})} \rceil$, while the anti-ideal points correspond to the anti ideal of s^{ww} (except for instance 20 where no weekend is free

[1] https://www.pypy.org/
[2] https://www.dbai.tuwien.ac.at/staff/musliu/benchmarks/
[3] https://github.com/tribec/bac-rwsp

Table 2. Ranges from ideal to anti-ideal points for soft constraints on different instances.

	$s^{\ell_{\text{dev}}}$		s^{ww}		$s^{d_{\max}}$	
instance	$[min,$	$max]$	$[min,$	$max]$	$[min,$	$max]$
10	[1,	48]	[12,	18]	[1,	18]
15	[20,	154]	[45,	54]	[3,	54]
20	[2,	962]	[120,	163]	[3,	164]

in the worst-case). The ideals (max) and anti-ideals (min) for the three dimensions and instances can be found in Table 2. We have used these points for the calculation of the normalized hypervolume (further just referred to as hypervolume).

6-Dimensional Experiments. We use parallel coordinate plots to visualize the approximated fronts. This allows stakeholders to investigate individual solutions while providing insights into correlations between different constraints. The plot can be studied interactively by filtering regions for each dimension. Thereby only solutions which satisfy the filters in all dimensions are shown. This shows clearly which tradeoffs have to be made in which dimensions when a stakeholder wants to improve a specific objective. Such a plot can be found in Figure 1. Each line corresponds to a non-dominated solution, indicating the violation values by the intersection with the corresponding axis. The solutions are color-coded with respect to the length deviation dimension.

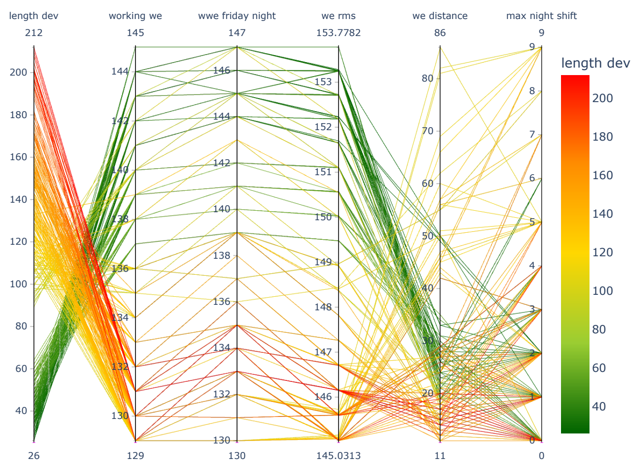

Figure 1. Approximated Pareto front after 1M iteration for instance 20 with all dimensions, making weight updates according to closest violation neighbours

We can observe that while the number of working weekends, working weekends including Friday night and equal distance between free weekends show a strong correlation, the length deviation constraint seems to be in conflict with optimizing free weekends.

We validate our observations using the sample Pearson correlation coefficient $r(dim1, dim2)$. We compared the dimensions length deviation, working weekends and weekend distance for all three instances. For this we took the violation vectors for each non-dominated solution generated by runs considering all dimensions and making weight updates using closest-violation neighbours. Table 3 shows, that while the correlations of the objectives differ for each instance, a general tendency of negative correlation can be observed. In addition each instance provides a strong negative correlation for a different combination of the three dimensions.

Table 3. Sample Pearson correlation coefficients for three objectives.

instance	$r(s^{\ell\text{dev}}, s^{\text{ww}})$	$r(s^{\text{ww}}, s^{\text{dmax}})$	$r(s^{\text{dmax}}, s^{\ell\text{dev}})$
10	-0.368	0.109	**-0.736**
15	-0.013	**-0.414**	-0.201
20	**-0.873**	-0.134	-0.179

2-Dimensional Experiments. Our motivation for introducing weight-neighbour weight updates stems from the observation, that for the 2-dimensional case the weight vectors of the generating solution converged toward 1 in one of the dimensions. This lead to solutions almost only optimizing one of the two dimensions. As a result, only few non-dominating solutions could be found that optimize both constraints in equal manner. When using weight-neighbour updates this gap could be closed. however, fewer solutions that mainly optimize one of the constraints could be obtained. A visualization of the observation for instance 20 and dimensions length deviation and working weekends can be found in Figure 2. Comparing the hypervolumes for all three instances over 10 runs considering length deviation and working weekends yielded similar average hypervolumes for both weight updating procedures. Nevertheless, we investigated both approaches also for multi-dimensional cases.

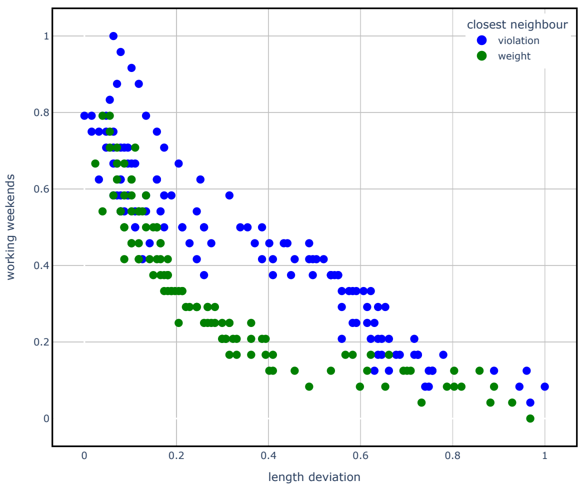

Figure 2. Combined solution sets over 10 runs using closest weight and violation neighbour for weight updates for instance 20. The solutions are given in the min/max normalized space over both dimensions.

3-Dimensional Experiments. We additionally calculated exact 2D Pareto fronts for s^{ww} and $s^{\ell\text{dev}}$ with the ε-constraint method using the branch-and-cut (b&c) approach. There, the optimization of $s^{\ell\text{dev}}$ is converted into a constraint with an upper bound starting from its ideal, iteratively optimizating s^{ww} while increasing the constraint bound stepwise, until the ideal of s^{ww} is hit. This could be performed within a couple of minutes runtime. A visualization of this front compared to a Pareto front approximation using PSA with weight-vectors to determine the closest neighbouring solution can be found in Figure 3. The generating solutions are given as arrows, directed in the direction within the 2-dimensional space toward which they are optimizing. It can be observed that while the solution set generated with b&c provides better solutions for length deviation and working weekend constraints, weekend distance violations are comparably high.

We evaluated the hypervolume for the different configurations of PSA over all three instances considering the length deviation, working weekends and weekend distance constraints. For each configuration, 10 runs were performed using the same random generator seeds for the different configurations. The hypervolume is calculated in regard to the ideal (min) and anti-ideal (max) for each of the three dimensions. The development of the average hypervolume when restarting solutions after a certain number of non-improvement iterations (consecutive iterations where a solution does not contribute to the set of non-dominated solutions) can be found in Figure 4. As a reference, the average hypervolume for random weight updates, where weights are assigned at random in each iteration, as well as the average hypervolume for classical weight updates using the weight or violation vectors to determine closest neighbours, are indicated. It can be observed that frequent restarts (after 10 to 1k non-improvement iterations) lead to worse hypervolumes. This can

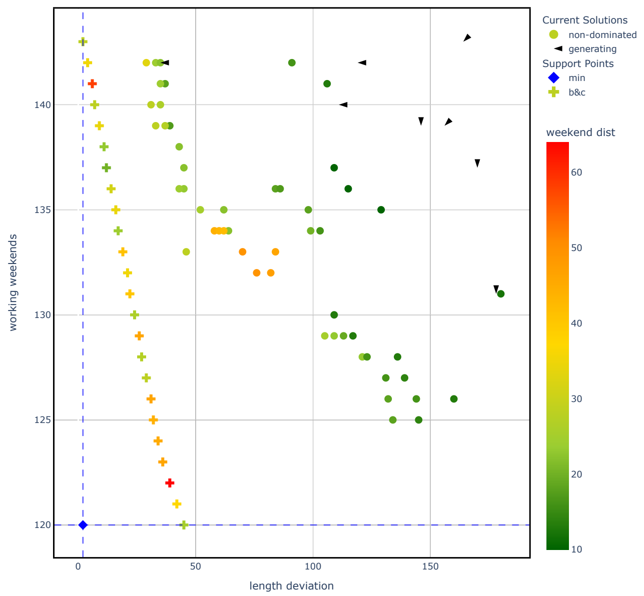

Figure 3. Approximated Pareto front over 3 dimensions after 1M iteration for instance 20. The 2-dimensional ideal and non-dominated solutions using branch-and-cut are given as reference.

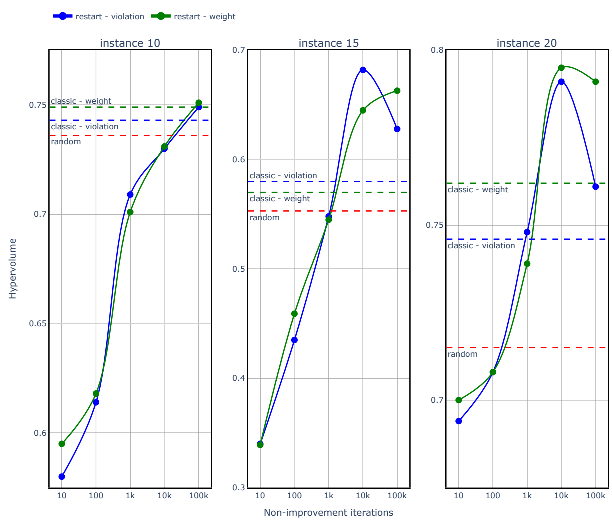

Figure 4. Comparison of hypervolume for random and classic weight updates with spline hypervolume for restarting approach.

be explained by the fact that multiple generating solutions can be replaced by the same non-dominated solutions, leading to a lack of variety within the generating solutions. However, by increasing the threshold on non-improvement iterations the hypervolume can be increased compared to the classical PSA approach. For instances 15 and 20 restarting after 10k non-improvement iterations leads to better results than for a threshold of 100k iterations, where the hypervolume tends to decline. In contrast, the hypervolume for instance 10 can hardly be increased by using the restarting method. This implies that a suiting threshold depends on the instance but should not be set too low, as this reduces the diversity among the generating solutions.

In Table 4 the average, maximal and standard deviation for the hypervolume over the 10 runs for different PSA configurations are given. For restarting runs the configurations with the best and worst average hypervolume are indicated. Again, it is evident that choosing the threshold is crucial, as low thresholds lead to worse hypervolumes than the approach using no restarting and random weight updates. In addition to the hypervolume, the average runtime over 10 runs for the PSA procedure is given. It can be observed, that the configuration using random weight updates takes the least runtime. This is evident, as no closest neighbour must be determined to update weights, hence decreasing the number of comparisons that have to be made. However, random weight updates lead to worse hypervolumes than the classical approach. It can also be observed that frequent restarts (after 10 iterations) increase the runtime while high thresholds for restarting yield only slightly increased runtimes compared to the classic approach but can improve the hypervolume of the obtained solution sets. This indicates that seldom restarting is a feasible addition to the classical implementation of PSA.

Comparing the performance of the presented approaches to the branch-and-cut reference points calculated with the ε-constraint method, we observe that while PSA can provide solution sets with more hypervolume for instance 10 and for some runs on instance 15, it is outperformed by branch-and-cut on instance 20. As the exact approach can not explicitly optimize weekend distance, we also analyze the mean violation for the best PSA configuration and b&c over the non-dominated solutions. The results can be obtained in Table 5.

As expected, the branch-and-cut approach has difficulties in optimizing $s^{\mathrm{d}_{\max}}$, especially in comparison to the PSA configurations. While the results for s^{ww} are very similar for both algorithms, b&c yields better results for $s^{\ell_{\mathrm{dev}}}$, which is to be expected as this objective can be optimized using branch-and-cut.

These results show, that while the hypervolume provides a good general indication on the quality of the solution sets, it does not allow for insights on the individual dimensions. We have shown that branch-and-cut provides very good solutions considering local objectives such as length deviation or working weekends. However, PSA can outperform branch-and-cut on non-local properties. In general, PSA with the right configuration manages to produce results comparable with branch-and-cut, while it also benefits from the initial feasible solutions provided by the exact solver. Therefore, interesting future work would be a further hybridization of both by (re-)starting PSA from partially optimized branch-and-cut solutions.

Table 4. Average, max and standard deviation for hypervolume and average runtime over 10 runs for different PSA configurations. Weight update configuration (v/w) and restart thresholds (10, 10k, 100k) are indicated. Hypervolumes of b&c are given as reference.

instance	configuration	hv - avg	hv - max	hv - stdv	t[h]
10	random	0.736	0.748	0.009	1.160
	classic-v	0.743	**0.757**	0.008	1.307
	classic-w	0.749	0.755	0.004	1.343
	restart-v10	0.580	0.678	0.074	1.760
	restart-w100k	**0.751**	**0.757**	0.005	1.406
	b&c	0.561			
15	random	0.553	0.644	0.050	1.575
	classic-v	0.580	0.648	0.037	1.726
	classic-w	0.570	0.663	0.054	1.758
	restart-v10	0.340	0.497	0.067	2.370
	restart-v10k	**0.682**	**0.838**	0.082	1.659
	b&c	0.782			
20	random	0.715	0.750	0.014	2.261
	classic-v	0.746	0.796	0.024	2.425
	classic-w	0.769	0.794	0.017	2.481
	restart-v10	0.694	0.741	0.019	3.360
	restart-w10k	**0.795**	**0.841**	0.024	2.597
	b&c	0.872			

Table 5. Mean values for three objectives comparing PSA configurations with highest avg hypervolume to branch-and-cut (b&c).

instance	configuration	$s^{\ell_{\mathrm{dev}}}$	s^{ww}	$s^{\mathrm{d}_{\max}}$
10	restart-w100k	7.02	12.89	8.18
	b&c	2.25	13.25	12.75
15	restart-v10k	33.84	47.79	11.34
	b&c	23.55	47.22	17.33
20	restart-w10k	51.04	130.51	16.52
	b&c	22.98	130.84	41.70

5 Conclusions

We have implemented the Pareto Simulated Annealing (PSA) algorithm for the Rotating Workforce Scheduling (RWS) problem. We proposed different modifications and adaptions of the algorithm and evaluated them in three instances of varying difficulty. Three different combinations of employee-centric objectives have been studied. First, we applied PSA using all six soft constraints, allowing us to determine the most conflicting constraints among them. After that, we evaluated the influence of using closest weight neighbour updates in the 2-dimensional space (length deviation and working weekends). While the obtained hypervolumes were very similar, we observed that using the closest weight-neighbours generates better compromising solutions, while violation-based updates enhance solutions favouring one of the dimensions. In the 3-dimensional space, additionally considering the weekend distance, again, no significant difference in hypervolume between weight or violation-based updates could be observed. Nevertheless, it would be interesting to use different metrics to evaluate the generated solutions, as also for the 3-dimensional space, a lack of diversity among the weight vectors of generating solutions and slightly worse results for compromising solutions could be observed. We have shown that sporadically resetting generating solutions that do not contribute to the set of non-dominating solutions can improve the hypervolume compared to the classical PSA approach. Additionally, these sparse resets do not increase runtime significantly. Lastly, we compared our approach to the current state-of-the-art branch-and-cut. While PSA could not consistently outperform branch-and-cut, it yielded substantially better results for the non-local weekend distance objective. It would be interesting to evaluate how this observation applies to other non-local objectives such as root mean squared weekend distance $d^{\mathrm{d}_{\mathrm{rms}}}$. In the future, we want to further improve our approach by making use of the strengths that b&c provides. It would be interesting to use parameter tuning to determine optimal configurations, to investigate how instance characteristics influence suiting configurations, and what other metrics should be considered to evaluate solution quality. Finally, we also want to validate our results on a larger set of benchmark instances.

Acknowledgements

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

References

[1] A. Arlinghaus, P. Bohle, I. Iskra-Golec, N. Jansen, S. Jay, and L. Rotenberg. Working time society consensus statements: Evidence-based effects of shift work and non-standard working hours on workers, family and community. *Industrial Health*, 57(2):184–200, 2019.

[2] K. R. Baker. Workforce allocation in cyclical scheduling problems: A survey. *Journal of the Operational Research Society*, 27(1):155–167, 1976.

[3] T. Becker, M. Schiffer, and G. Walther. A general branch-and-cut framework for rotating workforce scheduling. *INFORMS Journal on Computing*, 34(3):1548–1564, 2022.

[4] E. K. Burke, J. Li, and R. Qu. A pareto-based search methodology for multi-objective nurse scheduling. *Annals of Operations Research*, 196: 91–109, 2012.

[5] P. Czyżak and A. Jaszkiewicz. Pareto simulated annealing. In *Multiple Criteria Decision Making: Proceedings of the Twelfth International Conference Hagen (Germany)*, pages 297–307. Springer, 1997.

[6] C. Dall'Ora, J. Ball, A. Recio-Saucedo, and P. Griffiths. Characteristics of shift work and their impact on employee performance and wellbeing: A literature review. *International journal of nursing studies*, 57:12–27, 2016.

[7] P. De Bruecker, J. Van den Bergh, J. Beliën, and E. Demeulemeester. Workforce planning incorporating skills: State of the art. *European Journal of Operational Research*, 243(1):1–16, May 2015. ISSN 03772217. doi: 10.1016/j.ejor.2014.10.038.

[8] A. Drexl and Y. Nikulin. Multicriteria airport gate assignment and pareto simulated annealing. *IIE Transactions*, 40(4):385–397, 2008.

[9] A. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, Feb. 2004.

[10] S. Folkard and D. A. Lombardi. Modeling the impact of the components of long work hours on injuries and "accidents". *American journal of industrial medicine*, 49(11):953–963, 2006.

[11] S. Folkard, K. A. Robertson, and M. B. Spencer. A fatigue/risk index to assess work schedules. *Somnologie*, 11(3), 2007.

[12] J. Gärtner, P. Bohle, A. Arlinghaus, W. Schafhauser, T. Krennwallner, and M. Widl. Scheduling matters-some potential requirements for future rostering competitions from a practitioner's view. In *PATAT 2018-Proceedings of the 12th international conference on the practice and theory of automated timetabling*, pages 33–42, 2018.

[13] Health and Safety Executive. Managing shiftwork-health and safety guidance, 2006.

[14] L. Kletzander, N. Musliu, J. Gärtner, T. Krennwallner, and W. Schafhauser. Exact methods for extended rotating workforce scheduling problems. *Proceedings of the International Conference on Automated Planning and Scheduling*, 29:519–527, 2019.

[15] L. Kletzander, N. Musliu, and K. Smith-Miles. Instance space analysis for a personnel scheduling problem. *Annals of Mathematics and Artificial Intelligence*, 89:617–637, 2021.

[16] J. Li, E. K. Burke, T. Curtois, S. Petrovic, and R. Qu. The falling tide algorithm: a new multi-objective approach for complex workforce scheduling. *Omega*, 40(3):283–293, 2012.

[17] A. Lock, D. Bonetti, and A. Campbell. The psychological and physiological health effects of fatigue. *Occupational Medicine*, 68(8):502–511, Nov. 2018. ISSN 0962-7480. doi: 10.1093/occmed/kqy109. URL https://doi.org/10.1093/occmed/kqy109.

[18] F. Mischek and N. Musliu. Preference explanation and decision support for multi-objective real-world test laboratory scheduling. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 2024. to appear.

[19] C. R. Moreno, E. C. Marqueze, C. Sargent, K. P. Wright Jr, S. A. Ferguson, and P. Tucker. Working time society consensus statements: Evidence-based effects of shift work on physical and mental health. *Industrial Health*, 57(2):139–157, 2019.

[20] N. Musliu, J. Gärtner, and W. Slany. Efficient generation of rotating workforce schedules. *Discrete Applied Mathematics*, 118(1-2):85–98, 2002.

[21] N. Musliu, A. Schutt, and P. J. Stuckey. Solver independent rotating workforce scheduling. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 15th International Conference, CPAIOR 2018, Delft, The Netherlands, June 26–29, 2018, Proceedings 15*, pages 429–445. Springer, 2018.

[22] S. Petrovic, J. Parkin, and D. Wrigley. Personnel scheduling considering employee well-being: insights from case studies. In *Proceedings of the 2020 International Conference on the Practice and Theory of Automated Timetabling-PATAT*, volume 1, page 14, 2021.

[23] M. Rosekind, K. Gregory, M. Mallis, S. Brandt, B. Seal, and D. Lerner. The cost of poor sleep: Workplace productivity loss and associated costs. *Journal of occupational and environmental medicine / American College of Occupational and Environmental Medicine*, 52:91–8, Jan. 2010. doi: 10.1097/JOM.0b013e3181c78c30.

[24] J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, and L. De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, May 2013.