COLLINS, J., ZĂVOIANU, A.-C. and MCCALL, J.A.W. 2023. Comparison of simulated annealing and evolution strategies for optimising cyclical rosters with uneven demand and flexible trainee placement. In *Bramer, M. and Stahl, F. (eds.) Artificial intelligence XL: proceedings of the 43rd SGAI (Specialist Group on Artificial Intelligence) Artificial intelligence international conference 2023 (AI-2023), 12-14 December 2023, Cambridge, UK*. Lecture notes in computer science, 14381. Cham: Springer [online], pages 451-464. Available from: https://doi.org/10.1007/978-3-031-47994-6_39

Comparison of simulated annealing and evolution strategies for optimising cyclical rosters with uneven demand and flexible trainee placement.

COLLINS, J., ZĂVOIANU, A.-C. and MCCALL, J.A.W.

2023

Comparison of Simulated Annealing and Evolution Strategies for Optimising Cyclical Rosters with Uneven Demand and Flexible Trainee Placement

Joseph Collins[1,2,3][0000−0003−0969−024X], Alexandru-Ciprian Zăvoianu[1,2][0000−0003−1003−7504], and John A.W. McCall[1,2][0000−0003−1738−7056]

[1] School of Computing, Robert Gordon University, Aberdeen, Scotland, UK
[2] National Subsea Centre, Aberdeen, Scotland, UK
{j.collins1, c.zavoianu, j.mccall}@rgu.ac.uk
[3] Port of Aberdeen, Scotland, UK

Abstract. Rosters are often used for real-world staff scheduling requirements. Multiple design factors such as demand variability, shift type placement, annual leave requirements, staff well-being and the placement of trainees need to be considered when constructing good rosters. In the present work we propose a metaheuristic-based strategy for designing optimal cyclical rosters that can accommodate uneven demand patterns. A key part of our approach relies on integrating an efficient optimal trainee placement module within the metaheuristic-driven search. Results obtained on a real-life problem proposed by the Port of Aberdeen indicate that by incorporating a demand-informed random rota initialisation procedure, our strategy can generally achieve high-quality end-of-run solutions when using relatively simple base solvers like simulated annealing (SA) and evolution strategies (ES). While ES converge faster, SA outperforms quality-wise, with both approaches being able to improve the man-made baseline.

Keywords: simulated annealing · evolution strategies · staff rostering · staff training · combinatorial optimisation · uncertainty

1 Introduction and Motivation

Staff rosters are an essential tool in scheduling personnel, as the usage of well-established rota patterns allows personnel to plan their activities, both in and out of work. In general, a successful roster considers the needs of the personnel, while aiming to satisfy work commitments. Highly advanced staff rosters involve hundreds or thousands of employees and can incorporate multiple rota patterns as well as dedicated training time for staff members [1].

In recent decades, specialised metaheuristic solvers have become an increasingly popular option for automatically generating timetables [2], complex rosters and rota patterns [3]. While having strong interactions with timetabling problems from sectors like education [4][5], transport [6] and sports [7], efficient personnel scheduling is important in scenarios where a limited but skilled workforce must ensure adequate service availability even when confronted with dynamic demand patterns. This is often the case with emergency response staff [8][9], airline crew [10], healthcare workers [11] and even call centre staff [12].

Popular solvers used include tabu search (TS) [13], hybrid scatter search [14], noising methods combined with simulated annealing (SA) [15], ant colony optimisation (ACO) [16] and evolutionary algorithms (EAs) [17].

The present work is motivated by a pilot roster modelling scenario proposed by the Port of Aberdeen (PoA). As pilots provide a critical service for vessels, full daily coverage must be ensured. However, PoA receives most pilotage requests during early and late shifts, and least on night shifts. Furthermore, pilotage needs also vary per weekday. For example, Tuesdays have the highest demand and Saturdays the lowest. These demand trends are consistent over multiple years. The historical PoA pilot roster, shown in Figure 1, is cyclical and based on a weekly rota pattern. Designed for twelve staff members, the rota pattern is twelve weeks long with a single shift type – Early (d), Late (l), Night (n), Flexible (a) or TimeOff (o) – assigned to each day in the rota. The ratios of each shift type within the rota are based on consultation with staff. There is a mandated three week block of time off at the end of the rota. The PoA currently used the man-

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Flexi	Early	Early	Early	Night	Night	Night
Night	TimeOff	TimeOff	TimeOff	Late	Late	Late
Late	Night	Night	Night	TimeOff	TimeOff	TimeOff
Early	Early	Early	Early	Early	Early	Early
TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff
Flexi	Late	Late	Late	Night	Night	Night
Night	TimeOff	TimeOff	TimeOff	Early	Early	Early
Late	Night	Night	Night	TimeOff	TimeOff	TimeOff
Early	Late	Late	Late	Late	Late	Late
TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff
TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff
TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff

Fig. 1: Historical / baseline pilot rota pattern (x_h) at PoA

made pilot rota shown in Figure 1. The inauguration of a new harbour within the PoA is expected to increase pilot demand and challenge the existing roster. A critical feature when designing a new rota is a bespoke requirement to consider placement of a variable number of trainee pilots. A rota should factor in the need to train pilots without impacting high levels of service or overall schedule quality in terms of work-life balance. Automation of rota generation will allow the PoA the flexibility of easily testing and adapting to staffing scenarios that can rapidly adjust to unknown demand dynamics introduced by the Port's extension.

The remainder of this paper is organised as follows: Section 2 describes our formalisation of the PoA pilot rostering problem. Section 3 describes our approach to optimising the pilot rota and the placement of trainees. In Section 4 we present the setup and results of our numerical experiments alongside their interpretation. Finally, Section 5 contains conclusions and gives an outlook on future work.

2 Problem Formalisation

Using previous notations for shift types, a rota pattern to the PoA rostering problem can be encoded by an n-tuple (i.e., array x of size N) where N is the total number of days in the roster – i.e., $x \in \{d, l, n, a, o\}^N$. As the rotas we aim to generate are cyclical, N is equal to the number of staff multiplied by seven. However, given that the number of time off weeks at the end of the rota is predefined based on user input, this part of the roster can be decoupled to reduce problem size (e.g., $N = 63$ instead of $N = 84$ for the 12 person rota in Figure 1) and complexity (i.e., enforcing extended time off periods via constraints).

Discussions with the PoA have revealed the best way to model the quality of a given rota x is by penalising undesirable shift sub-patterns and understaffing.

Penalties were allocated on a scale from zero to one hundred, based on relative severity of constraint violations elicited from end users. These penalties were observed empirically to generate an appropriate response from the tested solvers.

Isolated shift penalty: $\boldsymbol{p_I(x)}$ A shift $x_i \neq f, 1 \leq i \leq N$ is considered isolated if $x_i \neq x_{i-1}$ and $x_i \neq x_{i+1}$ [4]. When marking with ω_I the number of isolated shift occurrences: $p_I(x) = 50 \cdot \omega_I$.

Late shift before early shift penalty: $\boldsymbol{p_{LE}(x)}$ If $x_i = l$ and $x_{i+1} = d$ a medium base penalty is incurred. If ω_{LE} marks the number of late→early infringements: $p_{LE}(x) = 50 \cdot \omega_{LE}$.

Insufficient rest penalty: $\boldsymbol{p_R(x)}$ If the set R contains all the disjoint rota sub-patterns $r_i = x_{s+1} \ldots x_{s+k}$ with the property that $|r_i| = k \geq 7$ and $x_{s+i} \neq o, \forall 0 \leq i \leq k$ then $p_R(x) = 25 \cdot \sum_{r_i \in R}(2^{|r_i|-6} - 1)$ as the aim is to generally discourage working more than 6 days in a row without rest. This penalty is designed to increase exponentially based on the seriousness of its violation(s).

Too many successive night shifts penalty: $\boldsymbol{p_{SN}(x)}$ If the set SN contains all the disjoint rota sub-patterns $n_i = x_{s+1} \ldots x_{s+k}$ with the property that $|n_i| = k \geq 4$ and $x_{s+i} = n, \forall 0 \leq i \leq k$ then $p_{SN}(x) = 10 \cdot \sum_{n_i \in SN}(2^{|n_i|-3} - 1)$ as the aim is to discourage working more than 3 night shifts in a row.

[4] As we are operating on cyclical rotas, $x_0 = x_N$ and $x_{N+1} = x_1$

Insufficient time off after night shift penalty: $p_O(x)$ If ω_O denotes total number of occurrences when $x_i = n$ and $x_{i+1} \neq n$, and for the j^{th} such occurrence o_j denotes number of consecutive time off shifts after the night shift (with the count starting at x_{i+1}), the partial time off penalty score is defined as:

$$p(o_j) = \begin{cases} 7 & \text{if } o_j = 0 \\ 3 & \text{if } o_j = 1 \\ 1 & \text{if } o_j = 2 \\ 0 & \text{if } o_j \geq 3 \end{cases}$$

and the total penalty is $p_O(x) = 100 \cdot \sum_{1 \leq j \leq \omega_O} p(o_j)$. The idea behind this penalty is to encourage adequate rest periods after night shifts.

Unmatched shift demand penalty: $p_U(x)$ Let $dd_i, dl_i, dn_i \in \mathbb{N}$ with $1 \leq i \leq 7$ denote the historical/expected demand for early, late and night shift pilots on the i^{th} day of the week. The supply of pilots for each (day, shift) pair is marked by sd_i, sl_i, sn_i and computed via a column-wise summation of the relevant shifts in the rota pattern (e.g., for the rota in Figure 1, $sd_i = sl_i = sn_i = 2, \forall i$). The unmatched shift demand penalty is computed as:
$p_D(x) = 100 \cdot \sum_{1 \leq i \leq 7} [max(dd_i - sd_i, 0) + max(dl_i - sl_i, 0) + max(dn_i - sn_i, 0)]$

Insufficient trainee supervision penalty: $p_T(x)$ Given that both trainee and experienced pilots can be placed on the rota, the previously defined supply of pilots can be broken down into $sd_i = sd_i^T + sd_i^E$, $sl_i = sl_i^T + sl_i^E$, $sn_i = sn_i^T + sn_i^E$ with sd_i^E, sl_i^E and sn_i^E denoting the number of experienced pilots for each (day, shift) pair. Given that it is highly preferable for trainees to always be supervised by at least one experienced member of staff, rota occurrences when this is not the case are penalised using:
$p_T(x) = 80 \cdot \sum_{1 \leq i \leq 7} [max(1 - sd_i^e, 0) + max(1 - sl_i^e, 0) + max(1 - sn_i^e, 0)]$.
It is important to note that the particular placement of trainees on the rota heavily impacts the value of p_T. For example: when placing two trainees on start weeks 1 and 2 (as trainee placement combination $C1$) on the rota in Figure 1 we obtain $p_T(x, C1) = 0$, but when the trainees are assigned start weeks 2 and 7 (as placement combination $C2$) we obtain $p_T(x, C2) = 80$ as $sn_1^e = 0$. This aspect is discussed at length in Section 3.2. The total penalty associated with a rota x is obtained by summing the seven penalty types and the search for a high-quality rota can be formalised as:

$$\min\ f(x) = p_I(x) + p_{LE}(x) + p_R(x) + p_{SN}(x) + p_O(x) + p_U(x, D) + p_T(x, C_b), \quad (1)$$

where D is a set that aggregates predefined pilot demand values for all (day, shift) pairs and C_b denotes the best possible start week placement for a predefined number of trainees that must be placed on the rota.

The global optimum for Equation 1 is $f(x^*) = 0$ and indicates a rota pattern x^* that does not incur any penalties. When considering the pilotage demand vectors obtained from historical PoA data:

$$\begin{cases} dd & = [1, 2, 2, 2, 2, 2, 1] \\ dl & = [2, 3, 2, 2, 2, 2, 2] \\ dn & = [2, 1, 2, 2, 1, 1, 1] \end{cases} \tag{2}$$

and no trainee placement, the total penalty associated with the rota x_h shown in Figure 1 is $f(x_h) = 470$ as: $p_I(x_h) = 100$, $p_{LE}(x_h) = 0$,$p_R(x_h) = 250$, $p_N(x_h) = 20$, $p_O(x_h) = 0$, $p_U(x_h, D) = 100$ and $p_T(x_h, C_b) = 0$. One trainee can also be placed on x_h without penalty (irrelevant of start week). If two trainees are to be placed on x_h, starting them in weeks 1 and 4 would not impact $p_T(x_h)$.

3 Proposed Approach

3.1 Rota Initialisation

After fixing N – the size of the rota – and subtracting the total number of flexible (a) and time off (o) days, the main focus of the initialisation stage is to compute how many Early (d), Late (l) and Night (n) shifts are to be allocated.

This allocation is based on historical or expected total relative demand across the three types of work shifts. Assuming that Equation (2) reflects expected demand trends for each weekday, the total relative demand for late shifts is: $\mathbb{F}_l = \frac{\sum dl_i}{\sum dd_i + \sum dl_i + \sum dn_i}$, $1 \leq i \leq 7 \Rightarrow \mathbb{F}_l = \frac{15}{12+15+10} = 0.4054$. Similarly $\mathbb{F}_d = 0.3243$ and $\mathbb{F}_n = 0.2702$. For example, given that after removing 40 time off and 2 flexible shifts from the rota in Figure 1, 42 days remain to be allocated among the three types of work shifts, we would obtain an allocation of $\mathbb{F}_d \cdot 42 = 13.62$ early shifts, $\mathbb{F}_l \cdot 42 = 17.02$ late shifts and $\mathbb{F}_n \cdot 42 = 11.34$ night shifts for x_h based on the previously computed relative work shift demand. Settling on the initialisation of (i) 17 Late shifts, 14 Early shifts and 11 Night shifts or (ii) 17 Late shifts, 13 Early shifts and 12 Night shifts is a somewhat subjective modelling decision. In the experiments we report on in Section 4, we opted for (ii) in light of the heavy penalties related to Night shift placements (i.e., p_N and p_O).

Once all the individual shift type counts have been determined based on historical/expected demand for the analysed use case, our rota patterns are initialised randomly to reflect the desired distribution of shift types.

3.2 Trainee Placement

A key part of our automated rostering strategy revolves around the optimal placement of a variable number of trainees. This feature allows the Port of Aberdeen flexibility to balance conflicting objectives under uncertain pilotage demand trends. Rosters with an overconcentration of trainees will be unsatisfactory should there be a surge in demand for highly skilled pilotage. Rosters with insufficient trainee slots will fail to provide sufficient pilotage experience.

For determining trainee placement, we first compute all combinations of locations where a predefined number of trainees nt can be assigned a start week on a rota pattern that covers nw work weeks. The total number of distinct trainee placement locations is $\binom{nw}{nt}$. As the roster is cyclical, initial trainee placements will iterate in a round robin fashion that induces an equivalence relation between different placements. For example, in the case of a four week roster, with two trainees (T) and two experienced staff members (E), as we have $nt = 2$ and $nw = 4$, there are $\binom{4}{2} = 6$ individual trainee placements and they are grouped into two equivalence classes: $\{TETE, ETET\}$ and $\{TTEE, ETTE, EETT, TEET\}$.

Our strategy for efficiently evaluating trainee placements applies a min-max approach on top of the resulting trainee placement equivalence classes and is described in Algorithm 1. As our goal is to discover a placement that results in a minimal insufficient trainee supervision penalty for a given rota x, our approach:

1 computes the class penalty score associated with each equivalence class of placements as the maximum $p_T(x)$ among the distinct trainee placements (i.e., members) of that class;
2 selects the minimum class penalty score among all equivalence classes as $p_T(x, C_b))$ and a class representative (e.g., the first member) as C_b – the best trainee placement option.
3 preemptively stops evaluating an equivalence class once a member displays a $p_T(x)$ value that is higher than a previously computed class penalty score.

Algorithm 1 Trainee Placement Approach

Require: Rota x, number of trainees nt
Ensure: Best trainee placement C_b, insufficient trainee supervision penalty $p_t(x, C_b)$
Extract the number of weeks in x: nw
Compute the trainee placement combinations: $C_1, C_2, \ldots, C_{\binom{nw}{nt}}$
Divide $C_1, C_2, \ldots, C_{\binom{nw}{nt}}$ into equivalence classes: $E_1, E_2, \ldots E_k$
Initialise: $C_b = C_1$, $p_T(x, C_b) = \infty$
for $i = 1$ to k **do**
  Initialise: $cps = 0$
  for $C \in E_k$ **do**
    Compute penalty score for trainee placement C: $p_T(x, C)$
    if $p_T(x, C) \geq cps$ **then**
      $cps = p_T(x, C)$
    end if
    if $p_T(x, C) \geq p_T(x, C_b)$ **then**
      Break the loop
    end if
  end for
  if $cps < p_T(x, C_b)$ **then**
    Extract class representative from E_k: C_r
    $C_b = C_r$
    $p_T(x, C_b) = cps$
  end if
end for
return C_b, $p_T(x, C_b)$

3.3 Metaheuristic Solvers

Given that the initialisation method described in Section 3.1 ensures that the distribution of work shifts required for obtaining a reasonable solution is present in any randomly generated rota, our optimisation strategy is centred on the deployment of a simple *shift swap (variation) operator* within several metaheuristic approaches that (re)-position shifts whilst aiming to solve Equation 1. When applied on a given rota x, the shift swap operator randomly selects two (day) indices i and j, with $1 \leq i, j \leq N$, and switches their shift types:

$$\begin{cases} aux & = x_i \\ x_i & = x_j \\ x_j & = aux \end{cases}$$

The first strategy experimented with was local search (LS) [18], but a majority of LS runs fell into local minima. Therefore, we continued with a slightly more advanced trajectory-based solver: Simulated Annealing (SA) [19]. Similarly to LS, at each iteration of SA a new candidate solution $x^{'}$ is generated by applying the swap operator on the current solution of the algorithm: x^c. Unlike in LS, $x^{'}$ can be accepted, with a certain probability, as the new current solution in SA even when $f(x^c) < f(x^{'})$, thus enabling the avoidance of local minima. The acceptance probability of a non-improving candidate solution is inversely proportional to the difference in quality with respect to $f(x^c)$ and directly proportional to a temperature parameter that is gradually reduced to 0 during the search (i.e., annealed). Preliminary parameter tuning tests with SA indicated that the solver is able to produce high-quality solutions for our use cases.

In order to contextualise SA performance in terms of convergence speed and final solution quality, we integrated the swap operator in a population-based solver, namely a $(1+\lambda)$ Evolution Strategy (ES) [20], as a mutation operator. At each iteration (i.e., generation) of the ES, λ offspring (i.e., candidate solutions) are created by applying the mutation operator to a single parent x^p (i.e., current solution). $x^{'}$ – the best among the λ offspring – becomes the parent of the next generation provided that $f(x^{'}) < f(x^p)$. Otherwise, x^p remains the parent.

We used standard versions for both solvers as they discovered high-quality solutions for the tested use case with zero trainees.

As the ability to optimally place trainees on the rosters is a PoA requirement, an aim of our numerical optimisation runs with SA and ES is studying differences in final rota qualities when opting between two trainee integration strategies:

- *SwapCheck (SC)*: computes the trainee placement penalty using the strategy outlined in Algorithm 1 in order to accurately evaluate $f(x^{'})$ whenever the solver generates a new candidate solution $x^{'}$;
- *FinalCheck (FC)*: disregards the $p_T(x^{'}, C_b)$ component from the computation of $f(x^{'})$ during the run and applies Algorithm 1 only on the best rota to estimate its final quality and associated best possible trainee placement.

The *SwapCheck* strategy provides the solvers with an accurate view of the fitness landscape at all stages of the optimisation run whilst the *FinalCheck* strategy has the advantage of proposing a simplified (but hopefully similar) fitness formulation during the search. The main disadvantage of *SwapCheck* is that it is computationally expensive. The main disadvantage of the *FinalCheck* strategy is that the best possible trainee placement on a high-quality solution for a problem formulation lacking trainees might still yield very large penalties.

4 Numerical Experiments and Results

4.1 Experimental Setup

While a limited test series indicates that our proposed approach can scale well across problem instances of up to 26 weeks (i.e., $N = 175$) and 7 trainees (especially with the *FinalCheck* strategy), the single use case focused on in this work is optimising the 12 week pilot rota used by the PoA when considering the historical pilotage demand of previous years (see Equation 2) and a wish to place one, two or no trainees on the rota. As previously mentioned, the historic rota x_h in Figure 1 has a baseline quality of 470 regardless of the number of trainees.

Across all SA and ES optimisation runs, we used a computational budget of 20,000 shift swap operations (each generating a new candidate solution). We performed 1000 independent repeats for each solver run on the no trainees scenario and 100 independent repeats for each solver run on the one and two trainees scenarios. The reduced runs on the latter scenarios were due to significantly increased computational cost of checking trainee constraints. Sufficient experiments were conducted to support statistical testing. When testing the statistical significance of differences between central tendency estimators, we applied a one-sided Mann-Whitney U Test [21] with a preset significance level of 0.025.

Given the simplistic nature of our two solvers, the options for parameterising them are fairly limited and are mainly intended to discover their best search space exploration-intensification trade-off for the analysed use case.

In the case of SA, we experimented with different parameterisations of the annealing schedule in order to force LS runs of various lengths at the end of the optimisation. Thus, we tested variants that dedicated the last 10000, 7500, 5000, 2500 shift swaps to LS and also tested an SA variant with 0 dedicated LS swaps.

For the $(1 + \lambda)$ ES, we used population sizes of λ =100, λ =50, λ =40 and λ =25 which resulted in optimisation runs of 200, 400, 500 and 800 generations.

4.2 Results and Interpretation

No Trainees Scenario. In Figure 2 we plot the average convergence behavior of SA and ES across the tested parameterisation options when there is no need to place trainees on the optimised rotas.

Results indicate that ES has a faster convergence speed as all 4 variants are, on average, able to discover rotas with a lower penalty (i.e., higher quality) than

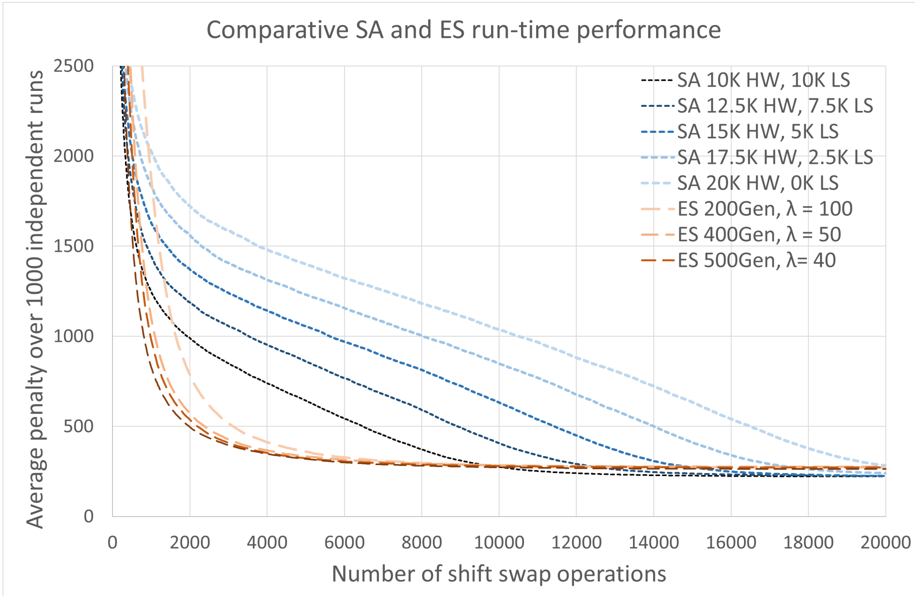
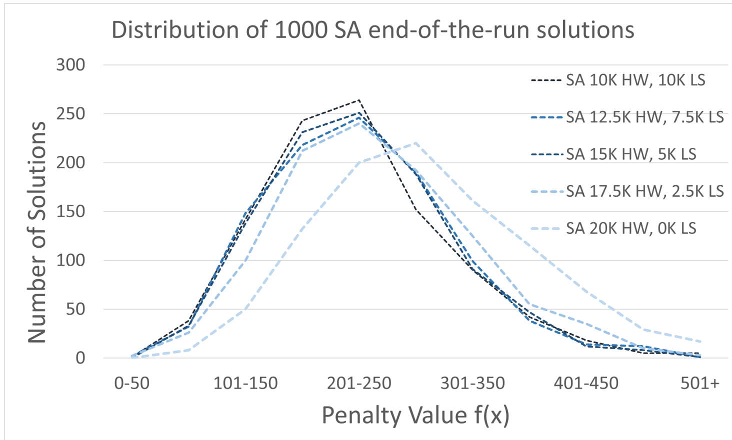


Fig. 2: Comparative convergence behavior of 5 SA and 4 ES variants.

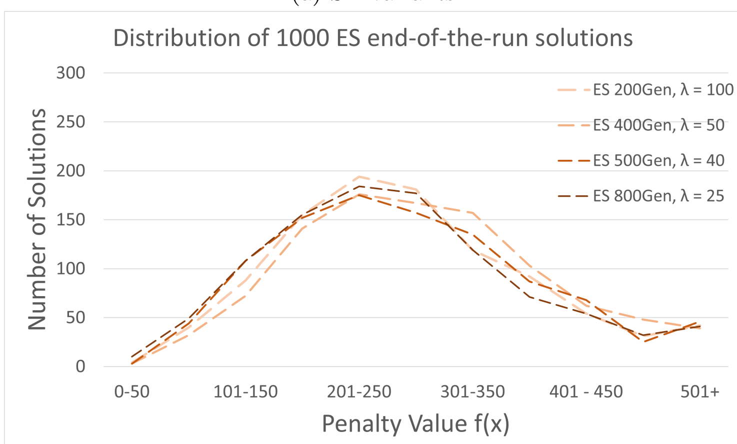
the baseline after only 4000 shift swap operations. Conversely, all 4 SA variants that have dedicated LS shift swaps after their hot working (HW) phase discover end-of-run solutions that are better than ES results. Across both solvers, it is noteworthy that the two variants that prioritise exploitation of the search space by integrating a long LS component (i.e., SA 10HW, 10K LS) or by evolving a smaller population over a longer period (i.e., ES 800gen $\lambda = 25$) outperform their peers both in terms of convergence speed and end-of-run solution quality.

Penalty-wise distributions of end-of-run solutions discovered by SA and ES are plotted in Figure 3. Details regarding the central tendency indicators of these distributions alongside information regarding the quality of the best solutions can be found in Table 1. Statistical significance testing confirms three observations:

1. Each of the SA variants that includes a meaningful LS phase (i.e., 10K, 7.5K, 5K, 2.5K) delivers better results than any ES variant.
2. The SA variant without a dedicated LS phase at the end of the run (i.e., 0K LS) underperforms the other 4 SA variants.
3. There is no meaningful difference between the end-of-run solution qualities obtained by the 4 ES variants.



(a) SA variants



(b) ES variants

Fig. 3: Histograms of end-of-run solution penalty.

Table 1: End-of-run solution quality for the no trainees scenario

Solver: variant	**Best**	**Average (μ)**	**Median**	**Std. deviation (σ)**
SA: 10K HW, 10K LS$^+$	30	222	215	81.1
SA: 12.5K HW, 7.5K LS$^+$	40	224.7	215	79.6
SA: 15K HW, 5K LS$^+$	55	224.8	215	78.9
SA: 17.5K HW, 2.5K LS$^+$	40	239.8	230	84
SA: 20K HW, 0K LS$^-$	60	283.2	280	92.8
ES: 200gen, $\lambda = 100$	105	271.9	257.5	115.8
ES: 400gen, $\lambda = 50$	25	256.6	265	115.9
ES: 500gen, $\lambda = 40$	25	275.2	260	115.9
ES: 800gen, $\lambda = 25$	10	263.4	250	117.3

Despite their average underperformance when compared with SA, three ES variants were able to find near-perfect solutions (i.e., $f(x) \leq 25$).

1 Trainee Scenario. Table 2 contains information regarding the differences between end-of-run solution penalties when wishing to place one trainee on the 12 week rota compared with complementary end-of-run results for the no trainee scenario. All solver variants used the computationally expensive *SwapCheck (SC)* strategy for determining the optimal placement of the trainee on the rota.

As expected, the lowest (i.e., best), average and median penalties achieved when placing one trainee on generated rotas are higher than equivalents for the no trainee scenario. Across all variants[5] differences are slightly higher than 80 – the minimal non-zero value of the insufficient trainee supervision penalty $p_T(x)$. As standard deviations are similar between scenarios, magnitude and consistency of the best, average and median penalty increases indicate that any form of trainee placement (i.e., including $P_T(x, C_b)$ in Equation 1) over-constrains the PoA rostering.

Table 2: End-of-run differences in solution quality for the 1 trainee scenario when compared with results from Table 1. Positive values indicate the 1 trainee result is worse.

Solver: variant	**ΔBest**	$\Delta\mu$	**ΔMedian**	$\Delta\sigma$
SA: 10K HW, 10K LS	125	127	125	10.4
SA: 12.5K HW, 7.5K LS	100	100.4	110	4.7
SA: 15K HW, 5K LS	135	92.3	97.5	0.5
SA: 17.5K HW, 2.5K LS	115	98.4	95	3
SA: 20K HW, 0K LS	105	96.9	95	0.4
ES: 200gen, $\lambda = 100$	5	100.5	107.5	0.5
ES: 400gen, $\lambda = 50$	150	128.5	117.5	-3.5
ES: 500gen, $\lambda = 40$	120	87.6	90	-1.2
ES: 800gen, $\lambda = 25$	175	123.0	137.5	-8.6

2 Trainees Scenario. Results in Table 3 indicate that when compared with the 1 trainee scenario – solved using *SwapCheck (SC)* –, the best end-of-run rotas for the 2 trainees scenario have a generally increased average and median penalty only when using the faster *FinalCheck (FC)* trainee placement strategy. When applying solvers on the 2 trainees scenarios using the *SC* trainee placement strategy, the impact of the extra trainee placement on $P_T(x, C_b)$ is minimal for SA variants and reduced in comparison with *FC* in the case of ES.

[5] Apart from the best penalties for ES 200gen, $\lambda = 100$.

Table 3: End-of-run differences in quality for 2 trainee scenario compared with 1 trainee *SwapCheck (SC)* results. Positive values mean 2 trainees result is worse.

Solver: variant	Δ**Best**		$\Delta\mu$		Δ**Median**		$\Delta\sigma$	
	SC	*FC*	*SC*	*FC*	*SC*	*FC*	*SC*	*FC*
SA: 10K HW, 10K LS	20	30	-12.3	39.3	-20	40	-1.6	-6.5
SA: 12.5K HW, 7.5K LS	-20	60	3.1	50	0	45	4.2	-4.7
SA: 15K HW, 5K LS	-25	-25	30	62.3	37.5	55	21.8	8.7
SA: 17.5K HW, 2.5K LS	10	-15	-3	54.8	15	57.5	3.4	17.4
SA: 20K HW, 0K LS	0	-25	0.5	42	5	55	18	15.7
ES: 200gen, $\lambda = 100$	15	-30	26	31.25	40	37	11	-13.1
ES: 400gen, $\lambda = 50$	-10	-35	-9.7	22	-15	22	-8.6	10.2
ES: 500gen, $\lambda = 40$	30	10	29.4	79.65	32.5	87.5	6.2	16
ES: 800gen, $\lambda = 25$	-50	5	13.7	36.7	0	30	31.5	22.8

Figure 4 shows the penalty-wise distributions of the end-of-run solutions discovered by the best performing SA and ES variants when using *SwapCheck* and *FinalCheck*. Statistical significance testing confirms the observation that the best performing SA variant obtains better results than the best performing ES variant regardless of which trainee placement strategy is used.

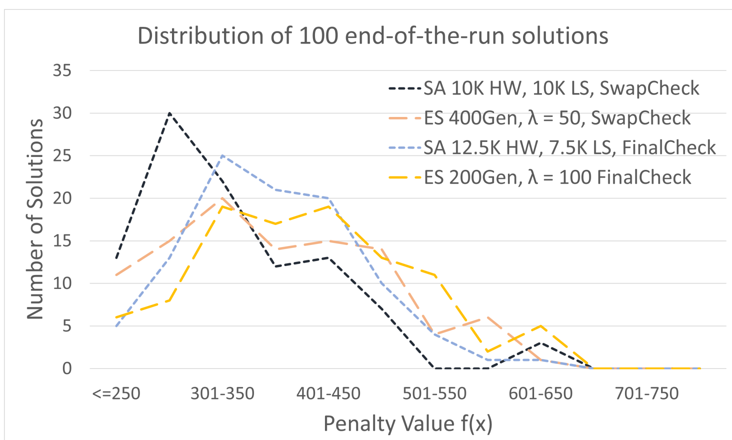


Fig. 4: Histograms of end-of-run solution penalty distribution for best performing SA and ES variants using *SwapCheck* and *FinalCheck* placement strategies.

The best solution obtained by the ES: 200gen, $\lambda = 100$ variant on the 2 trainee scenario achieved a total penalty of $f(x) = 80$ that was entirely due to a trainee supervision penalty (i.e., $p_T(x) = 80$). This means that the discovered rota (shown in Figure 5) is a perfect solution (i.e., global optimum) for the no trainee scenario (see Table 1).

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Late	Late	Night	Night	Night	TimeOff	TimeOff
TimeOff	Late	Late	Late	Flexi	Early	Early
TimeOff	TimeOff	Early	Early	Early	Night	Night
Night	TimeOff	TimeOff	TimeOff	Late	Late	Flexi
Early	Early	TimeOff	TimeOff	Late	Late	Night
Night	Night	TimeOff	TimeOff	TimeOff	Late	Late
Late	Late	Late	Late	TimeOff	TimeOff	Late
Late	Night	Night	Night	TimeOff	TimeOff	TimeOff
TimeOff	Early	Early	Early	Early	Early	Early
TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff
TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff
TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff	TimeOff

Fig. 5: Perfect solution for the no trainee scenario discovered using ES.

5 Conclusions and Future Work

The presented work shows how a bespoke trainee placement method based on an efficient min-max search that speculates equivalent placements in cyclical rosters can be combined with basic metaheuristic solvers like simulated annealing (SA) and evolution strategies (ES) to produce high-quality rotas that can successfully accommodate uneven shift demand patterns while also satisfying multiple staff preferences related to their work-life balance.

Our numerical experiments indicate that SA variants that allow for a significant LS phase at the end of the optimisation run outperform their faster-converging ES counterparts with regard to final solution quality. Furthermore, whilst solver performance is improved by evaluating trainee placement suitability during all stages of the optimisation, a much faster approach of simply placing the required number of trainees on the best solution for the simplified no trainee scenario also produced high quality rotas (for a limited number of trainees).

Future work will address larger problem instances (increased length and number of trainees) likely to pose difficulties to both our optimal trainee placement approach and the two base solvers we considered in our experiments so far.

Acknowledgements The authors would like to acknowledge the support of staff members at the Port of Aberdeen that have kindly contributed to this research by providing the historical pilot roster pattern, demand data and feedback that informed the problem formalisation.

This work was supported by the Port of Aberdeen and InnovateUK through a Knowledge Transfer Partnership project (KTP reference number 12046).

References

1. Fyvie, M., McCall, J.A.W., Christie, L.A., Brown-lee, A.E.: Explaining a staff rostering genetic algorithm using sensitivity analysis and trajectory analysis. In: Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal (2023)

2. Burke, E.K., Petrovic, S.: Recent research directions in automated timetabling. European journal of operational research 140(2), 266–280 (2002)
3. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: A review of applications, methods and models. European journal of operational research 153(1), 3–27 (2004)
4. Petrovic, S., Burke, E.K.: University timetabling. (2004)
5. Qu, R., Burke, E.K., McCollum, B., Merlot, L.T., Lee, S.Y.: A survey of search methodologies and automated system development for examination timetabling. Journal of scheduling 12, 55–89 (2009)
6. Kwan, R.S., Wren, A., Kwan, A.S.: Hybrid genetic algorithms for scheduling bus and train drivers. In: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512). vol. 1, pp. 285–292. IEEE (2000)
7. Ribeiro, C.C.: Sports scheduling: Problems and applications. International Transactions in Operational Research 19(1-2), 201–226 (2012)
8. Li, Y., Kozan, E.: Rostering ambulance services. Industrial engineering and management society pp. 795–801 (2009)
9. Taylor, P.E., Huxley, S.J.: A break from tradition for the san francisco police: Patrol officer scheduling using an optimization-based decision support system. Interfaces 19(1), 4–24 (1989)
10. Arabeyre, J., Fearnley, J., Steiger, F., Teather, W.: The airline crew scheduling problem: A survey. Transportation Science 3(2), 140–163 (1969)
11. Maenhout, B., Vanhoucke, M.: The impact of incorporating nurse-specific characteristics in a cyclical scheduling approach. Journal of the Operational Research Society 60, 1683–1698 (2009)
12. Gans, N., Koole, G., Mandelbaum, A.: Telephone call centers: Tutorial, review, and research prospects. Manufacturing & Service Operations Management 5(2), 79–141 (2003)
13. Burke, E., De Causmaecker, P., Vanden Berghe, G.: A hybrid tabu search algorithm for the nurse rostering problem. In: Simulated Evolution and Learning: Second Asia-Pacific Conference on Simulated Evolution and Learning, SEAL'98 Canberra, Australia, November 24–27, 1998 Selected Papers 2. pp. 187–194. Springer (1999)
14. Maenhout, B., Vanhoucke, M.: A hybrid scatter search heuristic for personalized crew rostering in the airline industry. European Journal of Operational Research 206(1), 155–167 (2010)
15. Parr, D., Thompson, J.M.: Solving the multi-objective nurse scheduling problem with a weighted cost function. Annals of Operations Research 155(1), 279 (2007)
16. Gutjahr, W.J., Rauner, M.S.: An aco algorithm for a dynamic regional nurse-scheduling problem in austria. Computers & Operations Research 34(3), 642–666 (2007)
17. Ainslie, R., McCall, J., Shakya, S., Owusu, G.: Tactical plan optimisation for large multi-skilled workforces using a bi-level model. In: 2018 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8. IEEE (2018)
18. Russell, S., Norvig, P.: Artificial intelligence: A modern approach (2003)
19. Kirkpatrick, S., Gelatt Jr, C.D., Vecchi, M.P.: Optimization by simulated annealing. science 220(4598), 671–680 (1983)
20. Beyer, H.G., Schwefel, H.P.: Evolution strategies–a comprehensive introduction. Natural computing 1, 3–52 (2002)
21. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. The annals of mathematical statistics pp. 50–60 (1947)