# Quantifying the effects of increasing user choice in MAP-Elites applied to a Workforce Scheduling and Routing Problem

**Abstract.** Quality-diversity algorithms such as MAP-Elites provide a means of supporting the users when finding and choosing solutions to a problem by returning a set of solutions which are diverse according to set of user-defined features. The number of solutions that can potentially be returned by MAP-Elites is controlled by a parameter that discretises the user-defined features into 'bins'. For a fixed evaluation budget, increasing the number of bins increases user-choice, but at the same time, can lead to a reduction in overall quality of solutions while vice-versa, decreasing the number of bins can lead to higher-quality solutions at the expense of reducing choice. The goal of this paper it to explicitly quantify this trade-off, through a study of the application of Map-Elites to a Workforce Scheduling and Routing problem, using a large realistic instances based in London and Edinburgh. We note that for the problems under consideration 30 bins or above maximises coverage (and therefore choice to the end user), whilst fewer bins maximises performance.

**Keywords:** First keyword · Second keyword · Another keyword.

## 1 Introduction and Motivation

The Workforce Scheduling and Routing Problem (WSRP) is a commonly encountered real-world problem in which there is a requirement to allocate tasks to individuals within a workforce, and to provide an optimised routing-plan between allocated tasks. Typical domains in which these problems arise include health-care (e.g scheduling home-visits from care-workers) and maintenance scheduling (e.g. scheduling service-engineers to jobs). As with many real-world problems, the ability to provide an end-user with a set of potential solutions is of great importance, enabling them to select between alternatives with respect to specific company priorities or objectives.

One approach to generating a set of potential solutions is to use a quality-diversity algorithm [10] which returns a set of diverse but high fitness solutions. In previous work, we applied one such algorithm, MAP-Elites[1], to the WSRP in order to generate multiple solutions that were diverse with respect to a set of four user-defined features ($CO_2$ produced, car use, travel cost and staff cost). This enabled a planner to see how the WSRP solution could be tailored to specific requirements, for example examining the effects of implementing a policy that

[1] Multi-dimensional Archive of Phenotypic Elites

increased public transport usage, and noting the effects on other objectives such as financial cost or emissions.

The MAP-Elites algorithm requires the user to define $n$ features (in this case $CO_2$ produced, car use, travel cost and staff cost) of interest which are mapped to an $n$-dimensional grid. The grid is discretised according to a user-defined parameter $d$ which controls the number of *bins*, i.e. discrete cells, on each axis of the grid. The maximum number of cells and therefore solutions that can be obtained is therefore $d^n$. Given a fixed evaluation budget, then increasing $d$ is very likely to lead to a reduction in solution quality given that the algorithm is forced to maintain solutions across a larger space, with a consequent reduction in selection pressure. For the end-user, this introduces a trade-off between selecting a grid at one end of the spectrum that provides a *large* number of *lower-quality* solutions and at the other, a grid providing a *small* number of *high-quality* solutions. *The goal of this paper is to quantify this trade-off, by conducting a thorough empirical investigation of the influence of the number of bins (d) on the quality of solutions produced and the coverage obtained when solving instances of the WSRP given a fixed optimisation budget.*

## 2 Background and Previous Work

The WSRP was defined in [2] as a scenario that involves the scheduling and routing of personnel in order to perform activities at different locations. Although similar to vehicle routing problems, the focus of the WSRP is on individuals rather than vehicles. For a comprehensive introduction to the WSRP and an overview of the latest developments, the reader is directed towards [3], [1] and [6]. A number of previous researchers have examined the scheduling and routing of workforces, including home care scheduling[11], security personnel scheduling [8] and technician scheduling [4]. A attempt to model the WSRP as a bi/multi-objective problem can be found in [1], the authors use cost and patient convenience as the twin objectives. The solution cost is the travel cost and staff overtime costs, patient convenience is defined as to whether the member of staff allocated is preferred, moderately preferred or not preferred, with penalties allocated as appropriate. The results presented show a strong relationship between convenience and cost; the more convenient a solution the higher the cost is likely to be.

Quality-Diversity algorithms (QD) [10] produce a large array of high-quality solutions with respect to a set of user-defined features. Although a number of QD algorithms now exist, including Novelty-Search with Local Competition [7], and MAP-ELites [9], here we focus only on the latter given its prevalence in the literature. The Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) was first introduced by Mouret *et al* [9] and provides a mechanism for illuminating search spaces by creating an archive of high-performing solutions mapped onto solution characteristics defined by the user. The majority of applications of illumination algorithms have been to *design* problems [9,15]. The basic approach has been extended by surrogate-assistance to reduce the computation time associated with real-world evaluations [?] and more recently by including a user in an evolutionary loop to focus search on particular regions of the feature-space of interest [5]. There are very few applications of QD algorithms apparent in the combinatorial optimisation domain; to the best of our knowledge, we were the first to show that MAP-Elites could be successfully applied to the combinatorial optimisation using the WSRP problem as an example[13] this paper investigated its use in producing multiple solutions to instances modelled using real data from Edinburgh and London, with four dimensions of user interest. The discretisation parameter $d$ was arbitrarily chosen to give 160,000 cells in this work however, motivating the need for further exploration.

## 3 Methodology

### 3.1 WSRP Problem Description

The WSRP considered is defined as follows. An organisation has to service a set of clients, who each require a single visit. Each visit $v$ must be allocated to an employee, such that all client visits are made by an employee. Each visit $v$ is located at $g_v$, where $g$ represents an actual UK post-code, and has a visit length $d_v$ and a time-window in which it must commence $\{e_v, l_v\}$. Visits are grouped into *journeys*, where each journey is allocated to an employee and contains a subset $V_j$ of the $V$ visits, starting and ending at a central office. In this formulation an unlimited number of employees are available.

Two modes of travel are available to employees, private transport (car) or public-transport, encouraging more sustainable travel, each journey is carried out using one of these modes for the entire journey. The overall goal of our WSRP is to minimise the total distance travelled across all journeys completed , discussions with end-users [14] highlights four characteristics of solutions that are of interest:

- **Emissions** incurred by all employees on their journeys
- **Employee Cost** the cost (based on £/hour) of paying the workforce for the duration of the journeys and visits
- **Travel Cost** the cost of all of the travel activities undertaken by the workforce
- The % of Employees using **car travel** for their journeys

We use a problem-representation described in [14]. The genotype defines a permutation of all $v$ required visits, this is divided into individual feasible journeys using a *decoder*. For each visit, the genotype also includes an additional gene that denotes the preferred mode of transport to be used for the visit (public or private).

The decoder converts the genome into a set of employee journeys by examining each visit in order. Initially, the first visit in the genotype is allocated to the first journey. The travel mode(car or public transport) associated with this visit in the genome is then allocated to that journey. The travel mode associated with the first visit is then adopted for the entire journey (regardless of the information associated with a visit in the genome). The decoder then examines the next visit in the permutation this is added to the current journey if it is *feasible*. Feasibility requires that the employee arrives from the previous visit using the mode of transport allocated to the journey within the time window associated with the visit. Subsequent visits are added using the journey mode until a hard constraint is violated, at which point the current journey is completed and a new journey initiated.

### 3.2 Problem Instances

We use a set of problem instances based upon the city of London, divided into two problem sets, termed Lon (60 visits) and BLon (110 visits). These instances were first introduced in [14] and also used later in [13]. For each of the problem sets, 5 instances are produced in which the duration of each visit is fixed to 30 minutes. Visits are randomly allocated to one of $t$ time-windows, where $t \in \{1, 2, 4, 8\}$. For $= 1$, the time-window has a duration of 8 hours, for $t = 2$, the time-windows are "9am-1pm" and "1pm-5pm" etc. These instances are labelled using the scheme *<set>- numTimeWindows*, i.e. *Lon-1* refers to an instance in the London with one time-window and *Blon-2* refers to an instance of the BigLondon problem with 2 time windows. The 'rnd' instance (e.g. BLon-rnd) represents a randomly chosen mixture of time windows based on 1,2,4 and 8 hours.

When a journey is undertaken by car,the distance and time is calculated according to the real road-network using the GraphHopper library[2]. This relies on Open StreetMap data[3]. Car emissions are calculated as 140 g/km based upon values presented in [12]. For journeys by public-transport, data is read from the Transport for London (TfL) API[4] which provides information including times, modes and routes of travel by bus and train. Public transport emissions factors are based upon those published by TfL [12].

### 3.3 MAP-Elites

The implementation of MAP-Elites used in this paper was used previously by [13] and was taken directly from [9]. The algorithm commences with an empty, N-dimensional map in which {solutions $\mathcal{X}$ and their performances $\mathcal{P}$ } can be placed. An initialisation method generates $G$ random-solutions which are placed in the archive (map); subsequent solutions are generated from elites stored in the archive. Following the random selection of a solution (or solutions) from the archive, the *RandomVariation()* method applies either crossover followed by mutation, or just mutation, depending on the experiment. All operators utilised are borrowed from the authors' previous work on these problems[**?**]. The *mutation* operator moves a randomly selected entry in the chromosome to a randomly

[2] https://graphhopper.com/
[3] https://openstreetmap.org/
[4] https://api.tfl.gov.uk/

selected point in the tour. The *crossover* operator selects a random section of the tour from parent-1 and copies it to the new solution. The missing elements in the child are copied from parent-2 in the order that they appear in parent-2. For each solution $x'$, a *feature-descriptor* $b$ is obtained by discretising the four features of interest associated with the solution. The upper and lower bounds required for discretisation are taken as the maximum and minimum values observed within these data sets in [14]. A new solution is placed in the cell in the archive corresponding to $b$ if its fitness $p$ (calculated as total distance travelled) is better than the current solution stored, or the cell is currently empty.

**Algorithm 1** MAP-Elites Algorithm, taken directly from [9]

```
procedure MAP-ELITES ALGORITHM
    (P ← ∅, X ← ∅)
    for iter = 1 → I do
        if iter < G then
            x' ← randomSolution()
        else
            x' ← randomSelection(X)
            x' ← randomVariation(X)
        end if
        b' ← featureDescriptor(x')
        p' ← performance(x')
        if P(b') = ∅ or P(b') < p' then
            P(b') ← p'
            X(b') ← x'
        end if
    end for
    return feature-performance map(P and X )
end procedure
```

### 3.4 Experimental Set up

We vary the number of bins ($d$) from 5 to 50 in steps of 5 (see table 1). The number of cells is therefore $cells = d^n$ where $n$ is the number of dimensions within the problem. Note that the *range* for each feature axis remains constant regardless of the number of bins that it is discretised into.

MAP-Elites was executed on each problem instance 10 times for each bin configuration. The function evaluation budget is fixed at 5,000,000 evaluations all experiments. We use the coverage metric to measure the area of the feature-space covered by a single run of the algorithm, i.e. the number of cells filled. For a single run $x$ of algorithm $y$, $coverage = noOfCellsFilled/C_{Max}$ where $C_{Max}$ is the total number of cells filled by combining all of the solutions produced to the problem under consideration.

| Bins ($d$) | Cells | Bins ($d$) | Cells |
|---|---|---|---|
| 5 | 625 | 30 | 810000 |
| 10 | 10000 | 35 | 1500625 |
| 15 | 50625 | 40 | 2560000 |
| 20 | 160000 | 45 | 4100625 |
| 25 | 390625 | 50 | 6250000 |

Table 1: Numbers of cells in each map. The number of cells may be calculated as $cells = d^n$

## 4 Results

### 4.1 The Effects of Bin Quantity on Fitness

The average, maximum and minimum fitness values found for each problem instance are given in table 3: entries in the table are of the format <avg>(min/max) over the 10 runs undertaken on each problem instance. It is clear that as the number of bins increases the average and best fitness decreases. The drop off in performance is relatively consistent as evidenced by table 4: the increase in average fitness is in the range of 23 - 34 % and increases as the problems become more constrained (e.g. with smaller time windows). The relationship between performance (as evidenced by the lowest fitness found over 10 runs) and the number of bins is shown in figure 5. Correlation coefficients of $R = 0.8943$ for the Lon data and $R = 0.9619$ for the BLon data indicate a strong positive relationship between the best solution found and the number of bins.

### 4.2 The Effects of Bin Quantity on Coverage

Figures 3 and 4 show the average levels of coverage achieved. The reader is reminded that coverage measures the proportion of the cells in the archive that contain a solution. We notice in both figures that the highest coverage is obtained with 25-30 bins, with the average coverage dropping before that. When looking for a relationship between coverage and the number of bins ($d$) we find that a correlation coefficient of $R = 0.5314$ and $R = 0.291$ for BLon and Lon. This suggests that the relationship between Bins and Coverage is not strong. We do note that for BLon less coverage is achieved with smaller values of $d$. The differing problem instances show different levels of coverage, but follow the same overall trends in the case of BLon. It is worth noting that when a small number of bins are used the coverage can drop as low as 50% (Blon): this translates to user-choice being limited to very few solution. At a discretisation level of 5 bins, this results in 312 solutions; this is in stark contrast to using e.g. 25 bins which gives a potential of 390625 solutions of which 80% are covered (albeit at lower quality). The best/worst coverage scores (averaged over 10 runs) for each value of $d$ may be seen in figures 1 and 2. The larger datasets (Blon) show a more consistent performance in relation to $d$ with the values of 30 or over giving the best coverage. The smaller dataset (Lon) gives a less consistent performance, generally smaller values of $d$ produce less coverage, but there is not a value of $d$ which differentiates consistently between best and worst performances. In order to confirm that there is a significant difference between the best and worst performance we apply a t-test to the coverage values obtained from the individual runs (table 2), we note that a significant difference is obtained in every case, and in 6 cases the difference is classed as extremely significant.

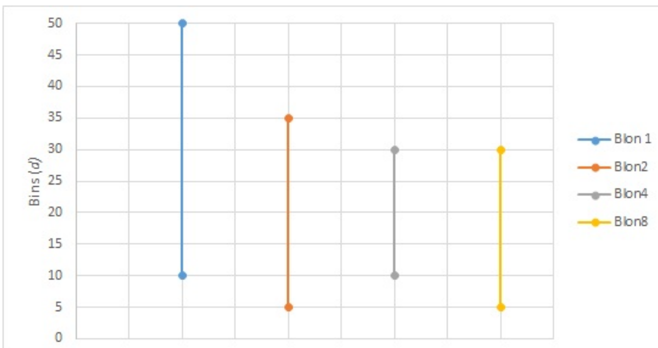


Fig. 1: The best/worst average coverage obtained for the BLon datasets. The worst is consistently obtained with 5-10 bins, the best coverage is obtained with $\geq$ 30 bins.

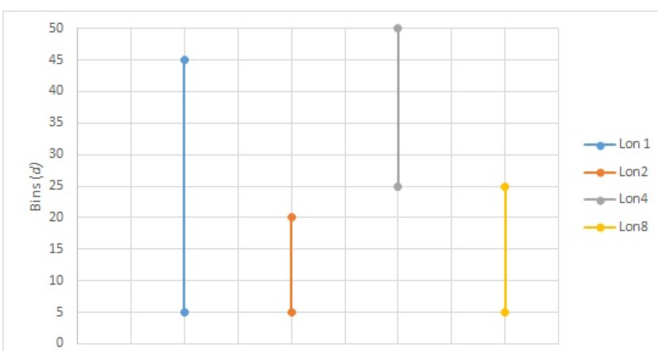


Fig. 2: The best/worst average coverage obtained for the Lon datasets. The least coverage is obtained with 5bins, except for Lon4 the best coverage is obtained with bins ranging from 20 to 50.

| Dataset | $P$ value | Classification | Dataset | $P$ value | Classification |
|---|---|---|---|---|---|
| **Lon1** | 0.0393 | S | **BLon1** | 0.0001 | E |
| **Lon2** | 0.0003 | E | **Blon2** | <0.0001 | E |
| **Lon4** | <0.0001 | E | **Blon4** | <0.0001 | E |
| **Lon8** | <0.0001 | E | **Blon8** | 0.0022 | V |

Table 2: The t-test results for the best/worst coverage (see figures 1 and 2) based on the coverage obtained on each individual run. The t-tests were carried out using (https://www.graphpad.com/), the classifications based on https://www.graphpad.com are 'S' - significant, 'E' - extremely significant and 'V' very significant.

### 4.3 Effects of Bin Quantity on Other Solution Characteristics

A visual indication of the solutions found is shown in figures 6 and 7 which examine results achieved from the BLon-rnd dataset. The figures chart the results obtained with differing numbers of bin: note that each row represents the final result of a specific run. We note that although the number of cells increases, the shape of the filled in area, largely remains the same. The resolution of the heat map increases, but the image remains largely unaltered. Those areas which are not filled in on the 5 bin heat maps are largely the same as those not filled in on the 50 bin heat maps. The lightest cells (green) represent those solutions with the least distance cost and we see largely the same distribution of colours.

## 5 Conclusions

In this paper we set out to quantify the trade-off between user choice (coverage) and solution quality, by varying the number of bins ($d$). In this case user choice is based upon the size of the map which is determined by the number of bins.

Table 4 shows a decrease in the performance of the algorithm (measured as best and average fitness) as the number of bins ($d$) increases. When the average fitness value across all cells is considered, increases of between 23% and 35% are noted. The average increase in the lowest (best) fitness obtained, rises to between 32% and 38% when the best solution is considered. These statistics represent a decline in performance as the number of cells increases, the correlations shown in figure 5 confirm that there exists a relationship between the fitness of the solutions found by the algorithm and the number of bins ($d$). Table 3 confirms that the lowest fitness and lowest average fitness's are found with low values of $d$, across all of the problem instances examined in this study/ The user is advised to choose a low value of $d$ if their priority is to find low fitness solutions.

| **Problem** | **Number of Bins** | | | | |
|---|---|---|---|---|---|
| | **5** | **10** | **15** | **20** | **25** |
| **lon-1** | **343.17** (**169.72**/582.58) | 368.32 (177.72/604.06) | 392.55 (181.45/640.38) | 404.09 (194.59/651.94) | 419.19 (221.44/670.35) |
| **lon-2** | **316.14** (189.68/500.40) | 335.26 (**185.45**/617.62) | 361.00 (191.99/639.24) | 382.90 (219.27/662.77) | 398.83 (243.32/668.99) |
| **lon-4** | **289.52** (201.27/489.72) | 302.10 (**187.22**/551.87) | 328.39 (214.04/585.18) | 350.54 (231.61/615.31) | 369.23 (247.72/631.39) |
| **lon-8** | **267.03** (209.85/415.40) | 277.14 (**202.67**/504.63) | 301.58 (218.08/552.50) | 323.88 (233.18/573.96) | 342.29 (250.64/598.11) |
| **lon-rnd** | **302.87** (202.48/516.74) | 323.81 (**196.94**/607.91) | 347.86 (219.05/647.06) | 371.41 (245.39/654.72) | 391.31 (263.34/662.31) |
| | **30** | **35** | **40** | **45** | **50** |
| **lon-1** | 426.57 (225.94/678.49) | 438.23 (231.51/705.43) | 440.74 (240.23/693.97) | 449.25 (245.00/690.69) | 450.32 (264.40/689.91) |
| **lon-2** | 412.29 (253.64/664.07) | 424.12 (264.65/672.79) | 433.92 (277.33/692.45) | 442.50 (271.10/685.20) | 450.24 (292.77/677.21) |
| **lon-4** | 386.14 (256.57/626.87) | 401.86 (281.49/647.94) | 414.90 (284.74/642.31) | 425.24 (291.31/644.13) | 433.86 (307.19/642.20) |
| **lon-8** | 359.06 (266.18/602.82) | 374.84 (275.39/601.51) | 388.46 (287.01/618.53) | 399.58 (286.76/609.64) | 409.79 (307.17/607.84) |
| **lon-rnd** | 409.99 (278.47/677.60) | 424.29 (289.09/679.12) | 437.99 (284.55/658.81) | 448.22 (308.62/665.29) | 456.55 (313.57/669.29) |
| | **5** | **10** | **15** | **20** | **25** |
| **blon-1** | **908.88** (576.88/1353.63) | 984.16 (**575.24**/1485.79) | 1049.33 (678.25/1544.02) | 1077.76 (722.07/1629.96) | 1100.84 (777.11/1660.29) |
| **blon-2** | **843.40** (**595.33**/1258.17) | 921.98 (644.37/1426.47) | 979.64 (698.01/1536.59) | 1027.11 (707.28/1577.54) | 1061.74 (797.55/1603.90) |
| **blon-4** | **812.18** (**593.46**/1218.16) | 850.28 (616.68/1384.28) | 914.15 (688.46/1447.85) | 962.87 (719.31/1556.15) | 1005.62 (781.23/1547.89) |
| **blon-8** | **739.43** (575.02/1123.34) | 762.15 (**558.48**/1264.02) | 814.39 (621.05/1388.22) | 876.25 (688.50/1425.02) | 921.19 (745.42/1441.38) |
| **blon-rnd** | **772.14** (**560.96**/1318.77) | 841.77 (614.57/1355.84) | 906.13 (646.29/1465.39) | 963.29 (721.45/1599.49) | 1010.42 (770.88/1580.51) |
| | **30** | **35** | **40** | **45** | **50** |
| **blon-1** | 1127.61 (794.66/1639.93) | 1144.45 (787.04/1707.63) | 1155.95 (828.50/1712.46) | 1171.13 (830.16/1714.22) | 1179.35 (874.86/1709.89) |
| **blon-2** | 1089.34 (814.67/1604.73) | 1115.92 (856.11/1634.92) | 1135.05 (862.98/1664.27) | 1148.45 (889.40/1660.21) | 1168.40 (919.96/1653.63) |
| **blon-4** | 1039.31 (802.53/1528.20) | 1071.97 (847.85/1562.03) | 1094.58 (861.59/1581.44) | 1114.82 (867.15/1585.64) | 1132.02 (903.20/1578.84) |
| **blon-8** | 961.76 (745.99/1505.71) | 996.44 (808.32/1529.49) | 1024.60 (832.17/1503.73) | 1050.35 (858.94/1515.39) | 1069.19 (893.62/1495.86) |
| **blon-rnd** | 1044.26 (828.00/1565.58) | 1077.62 (859.40/1593.16) | 1099.46 (847.55/1598.61) | 1121.43 (900.96/1639.58) | 1141.09 (906.73/1604.65) |

Table 3: The fitness (distance) values achieved on each of the problem instances, the results are in the format of <avg>(<lowest>/<highest>) based on 10 runs. The results in **bold** show the lowest overall and lowest average fitness values found.

| | Avg | | | Best | | |
|---|---|---|---|---|---|---|
| | **5** | **50** | **% increase** | **5** | **50** | **% Increase** |
| **lon-1** | 343.17 | 450.32 | 23.79% | 169.72 | 264.4 | 35.81% |
| **lon-2** | 316.14 | 450.24 | 29.78% | 189.68 | 292.77 | 35.21% |
| **lon-4** | 289.52 | 433.86 | 33.27% | 201.27 | 307.19 | 34.48% |
| **lon-8** | 267.03 | 409.79 | 34.84% | 209.85 | 307.17 | 31.68% |
| **lon-rnd** | 302.87 | 456.55 | 33.66% | 202.48 | 313.57 | 35.43% |
| **blon-1** | 908.88 | 1179.35 | 22.93% | 576.88 | 874.86 | 34.06% |
| **blon-2** | 843.4 | 1168.4 | 27.82% | 595.33 | 919.96 | 35.29% |
| **blon-4** | 812.18 | 1132.02 | 28.25% | 593.46 | 903.2 | 34.29% |
| **blon-8** | 739.43 | 1069.19 | 30.84% | 575.02 | 893.62 | 35.65% |
| **blon-rnd** | 772.14 | 1141.09 | 32.33% | 560.96 | 906.73 | 38.13% |

Table 4: The increase in average and best fitness between the lowest and highest number of bins ($d$) used.

Figures 8 and 9 provide a visual representation of typical sets solutions presented to the user. We can see that when $d = 5$ there are far fewer solutions and far fewer opportunities for trading off between objectives. When $d = 50$ (figure 9) we note the dramatic increase in solutions and the potential to to find trade-offs.

The increase in the quantity of solutions found (Table 5) represents an increase in the choice available to users selecting a solution. Figures 3 and 4 show that the the numbers in table 1 represent less than 80% of the solutions that could be found.

In summary we demonstrate the implications of altering the value of $d$ within a MAP-Elites algorithm. Reducing $d$ is likely to result in the production of MAPS containing solutions with lower fitness values, but limiting the choice available to the user. Increasing $d$ will result in increased choice, but with lower overall and average fitness.

| | **5** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **50** |
|---|---|---|---|---|---|---|---|---|---|---|
| **lon-1** | 69 | 432 | 1450 | 3344 | 6360 | 10328 | 15524 | 21129 | 28759 | 38252 |
| **lon-2** | 84 | 1080 | 4640 | 12374 | 25013 | 42039 | 65781 | 89835 | 118815 | 153787 |
| **lon-4** | 122 | 1501 | 6926 | 19325 | 38961 | 65684 | 101561 | 140768 | 185533 | 236537 |
| **lon-8** | 106 | 1490 | 6979 | 19144 | 39533 | 67495 | 103840 | 145798 | 194424 | 248200 |
| **lon-rnd** | 120 | 1552 | 6878 | 18749 | 37590 | 62441 | 95737 | 131973 | 175174 | 222401 |
| | | | | | | | | | | |
| **blon-1** | 56 | 362 | 1271 | 3263 | 6258 | 11038 | 17539 | 24545 | 33898 | 44642 |
| **blon-2** | 54 | 565 | 2549 | 7180 | 15470 | 28173 | 44827 | 64970 | 88276 | 112549 |
| **blon-4** | 68 | 752 | 3305 | 9376 | 20473 | 35916 | 56615 | 81887 | 109628 | 141184 |
| **blon-8** | 72 | 859 | 3821 | 10993 | 22940 | 41354 | 63907 | 92374 | 123681 | 157905 |
| **blon-rnd** | 84 | 888 | 4113 | 11478 | 24330 | 42021 | 65049 | 92556 | 122091 | 156211 |

Table 5: Average number of solutions produced for each problem instance.

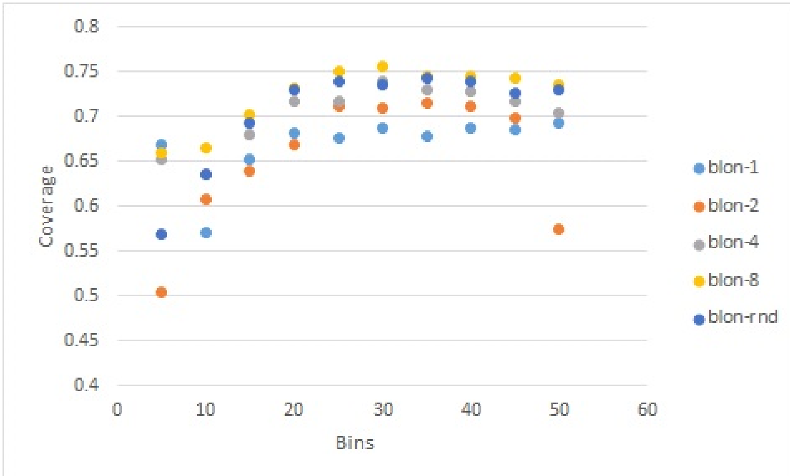


Fig. 3: The coverage achieved using with BLon datasets, each data point represents the average coverage over 10 runs.

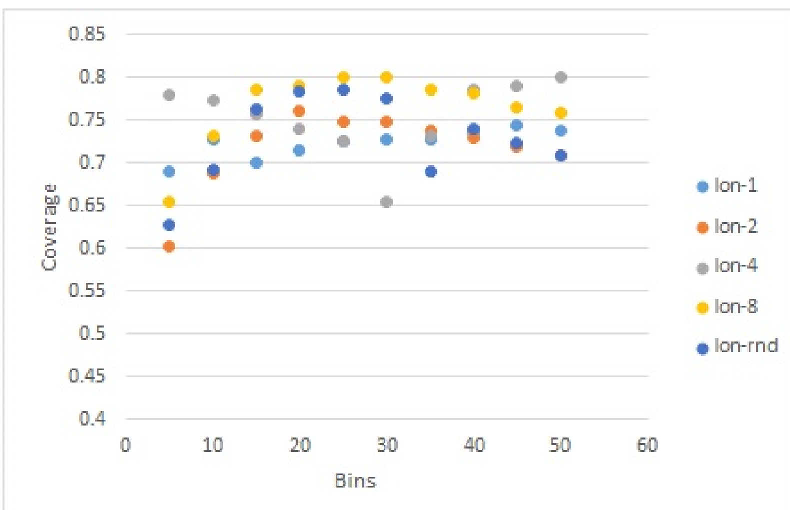


Fig. 4: The coverage achieved using with Lon datasets, each data point represents the average coverage over 10 runs.

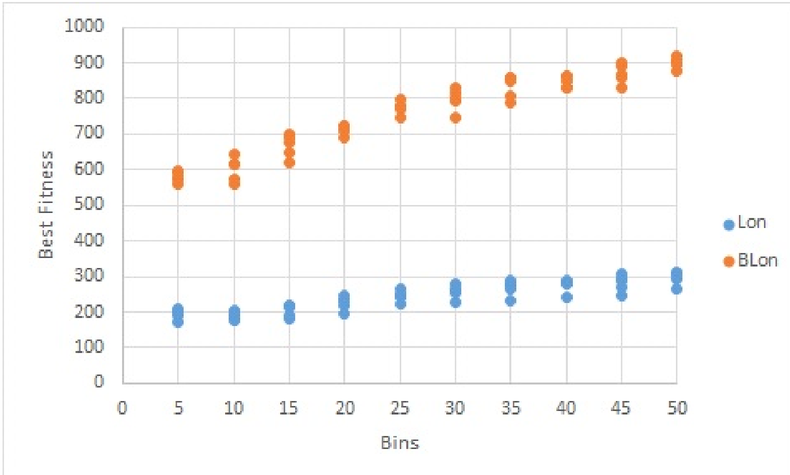


Fig. 5: Best fitness achieved (over 10 runs) plotted against the number of bins. The correlation coefficients obtained were $R = 0.8943$ and $R = 0.9619$ for the Lon and BLon data.

## References


1. Braekers, K., Hartl, R.F., Parragh, S.N., Tricoire, F.: A bi-objective home care scheduling problem: Analyzing the trade-off between costs and client inconvenience. European Journal of Operational Research **248**(2), 428 – 443 (2016)
2. Castillo-Salazar, J.A., Landa-Silva, D., Qu, R.: A survey on workforce scheduling and routing problems. In: Proceedings of the 9th international conference on the practice and theory of automated timetabling. pp. 283–302 (2012)
3. Castillo-Salazar, J.A., Landa-Silva, D., Qu, R.: Workforce scheduling and routing problems: literature survey and computational study. Annals of Operations Research **239**(1), 39–67 (2016). https://doi.org/10.1007/s10479-014-1687-2, http://dx.doi.org/10.1007/s10479-014-1687-2
4. G¨unther M, N.V.: Application of particle swarm optimization to the british telecom workforce scheduling problem. Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012), Son, Norway (2012)
5. Hagg, A., Asteroth, A., Bäck, T.: Prototype discovery using quality-diversity. In: Auger, A., Fonseca, C.M., Lourenço, N., Machado, P., Paquete, L., Whitley, D. (eds.) Parallel Problem Solving from Nature – PPSN XV. pp. 500–511. Springer International Publishing, Cham (2018)
6. Hiermann, G., Prandtstetter, M., Rendl, A., Puchinger, J., Raidl, G.R.: Metaheuristics for solving a multimodal home-healthcare scheduling problem. Central European Journal of Operations Research **23**(1), 89–113 (2015). https://doi.org/10.1007/s10100-013-0305-8, http://dx.doi.org/10.1007/s10100-013-0305-8
7. Lehman, J., Stanley, K.O.: Evolving a diversity of virtual creatures through novelty search and local competition. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation. pp. 211–218. ACM (2011)

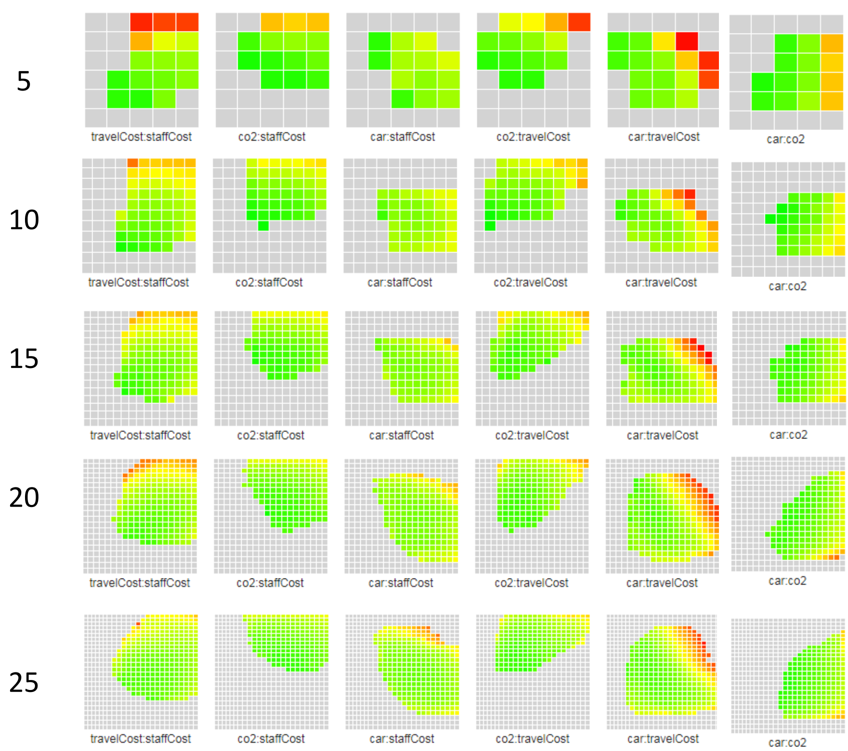


Fig. 6: Heat maps showing the solutions contained within the map. Each row contains a heat map for each pair of dimensions.

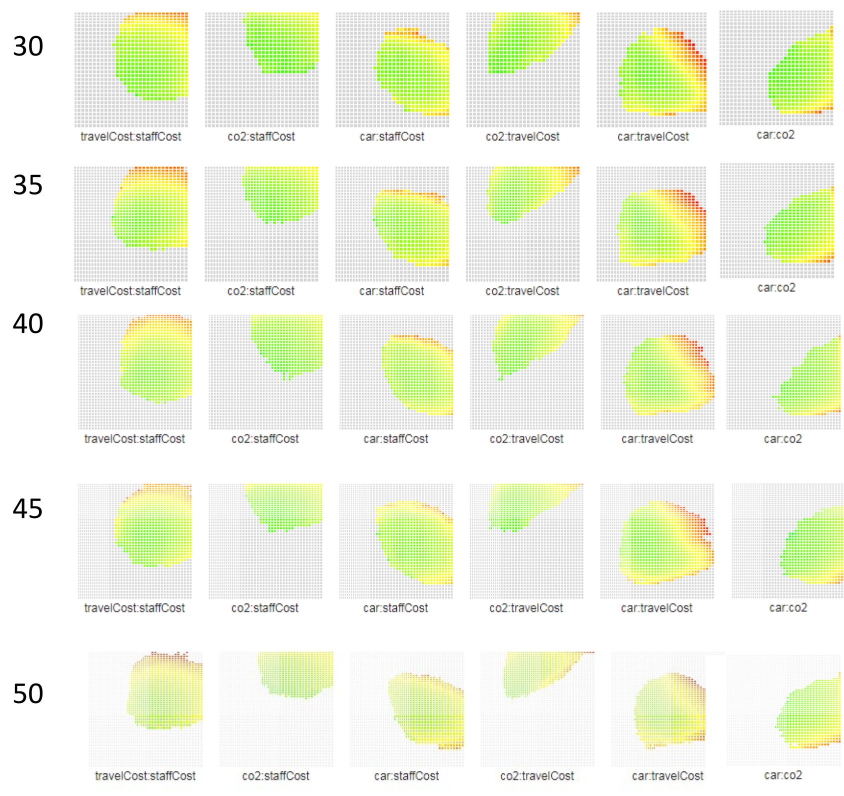


Fig. 7: Heat maps showing the solutions contained within the map. Each row contains a heat map for each pair of dimensions.

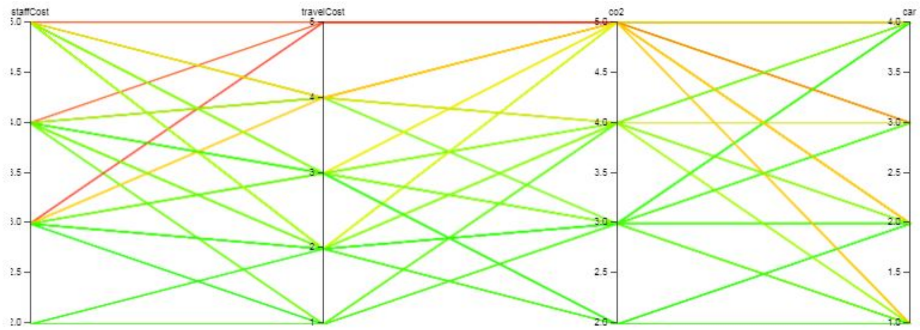


Fig. 8: A parallel coordinates plot showing the solutions found for a run with the BLon-rnd data set with $d = 5$.

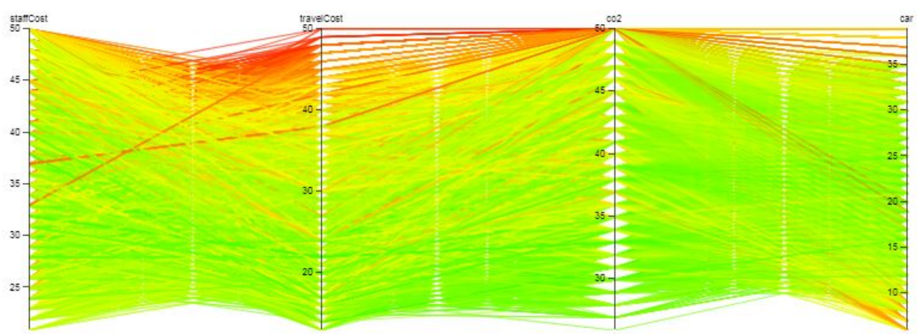


Fig. 9: A parallel coordinates plot showing the solutions found for a run with the BLon-rnd data set with $d = 50$.

8. Misir, M., Smet, P., Verbeeck, K., Berghe, G.V.: Security personnel routing and rostering: a hyper- heuristic approach. Proceedings of the 3rd International Conference on Applied Operational Research, ICAOR11 (2011)
9. Mouret, J., Clune, J.: Illuminating search spaces by mapping elites. CoRR (2015)
10. Pugh, J.K., Soros, L.B., Stanley, K.O.: Quality diversity: A new frontier for evolutionary computation. Frontiers in Robotics and AI **3**, 40 (2016)
11. Rasmussen, M., Justesen, T., Dohn, A., Larsen, J.: The Home Care Crew Scheduling Problem:: Preference-Based Visit Clustering and Temporal Dependencies. DTU Management (2010)
12. TFL: Travel in london: Key trends and developments. Tech. rep., Transport for London (2009)
13. Urquhart, N., Hart, E.: Optimisation and illumination of a real-world workforce scheduling and routing application (wsrp) via map-elites. In: Auger, A., Fonseca, C.M., Lourenço, N., Machado, P., Paquete, L., Whitley, D. (eds.) Parallel Problem Solving from Nature – PPSN XV. pp. 488–499. Springer International Publishing, Cham (2018)
14. Urquhart, N.B., Hart, E., Judson, A.: Multi-modal employee routing with time windows in an urban environment. In: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation. pp. 1503–1504. ACM (2015)
15. Vassiliades, V., Chatzilygeroudis, K., Mouret, J.B.: Using centroidal voronoi tessellations to scale up the multi-dimensional archive of phenotypic elites algorithm pp. 1–1 (08 2017)