10th International Conference of the Practice and Theory of Automated Timetabling
PATAT 2014, 26-29 August 2014, York, United Kingdom

Diversity-Oriented Bi-Objective Hyper-heuristics for Patrol Scheduling

Mustafa Mısır · Hoong Chuin Lau

Abstract The patrol scheduling problem is concerned with assigning security teams to different stations for distinct time intervals while respecting a limited number of contractual constraints. The objective is to minimise the total *distance* travelled while maximising the *coverage* of the stations with respect to their security requirement levels. This paper introduces a hyper-heuristic strategy focusing on generating diverse solutions for a bi-objective patrol scheduling problem. While a variety of hyper-heuristics have been applied to a large suite of problem domains usually in the form of single-objective optimisation, we suggest an alternative approach for solving the patrol scheduling problem with two objectives. An adaptive weighted-sum method with a variety of weight schedules is used instead of a traditional static weighted-sum technique. The idea is to reach more diverse solutions for different objectives. The empirical analysis performed on the Singapore train network dataset demonstrate the effectiveness of our approach.

Keywords Hyper-heuristics · Bi-objective Optimisation · Patrol Scheduling

1 Introduction

In this paper, we consider the Patrol Scheduling Problem (PSP) on a train network as studied in [10]. Hyper-heuristics have been previously used to solve similar type of problems as the security personnel routing and rostering problem [14]. The objective of those problems is to assign a number of security personnel to the sites where security is needed while respecting the contractual constraints of the personnel. Besides this rostering aspect of the problem,

M. Mısır, H. C. Lau
Living Analytics Research Centre, School of Information Systems, Singapore Management University
Tel.: +65-6808-5227
E-mail: mustafamisir@smu.edu.sg, hclau@smu.edu.sg

routing is also critical due to travelling between different sites. Related to this problem, the home care scheduling problem [15] which was solved by hyper-heuristics, also involves routing and rostering characteristics. The goal is to assign a group of carers to assist the people who need help at their homes. The PSP presented in this paper is a bi-objective optimisation problem for assigning a number of security teams to the train stations during a day. The first objective is to minimise the total travelling time spent by the security teams. The latter objective represents the coverage of the stations with respect to the stations' security requirements related to the station size and the passenger density. A hyper-heuristic framework is proposed to solve this problem.

Hyper-heuristics [2] are high-level search and optimisation techniques for managing a given set of heuristics or automatically generating heuristics. The primary reason behind studying hyper-heuristics is their problem-independent nature which separates the problem domain and the algorithm design components. This characteristic is achieved by performing search at a higher level, i.e. heuristic level, instead of a problem level. As a consequence, hyper-heuristics promise a high level of generality for solving different kinds of problems under varying search-related challenges. Hence, in principal, a hyper-heuristic can be applied to any target problem with no additional effort. This brings a unique advantage to hyper-heuristics in comparison to most of the existing search and optimisation methods.

Hyper-heuristic designs are generally divided into two types: *selection hyper-heuristics* and *generation hyper-heuristics.* The first type operates on a suite of existing low-level heuristics that are implemented to solve a given problem. The latter type aims at automatically building problem-specific heuristics, particularly via genetic programming [3,1] and hybridisation. This paper is concerned with selection hyper-heuristics. A traditional selection hyper-heuristic is composed of a heuristic selection method and a move acceptance criterion. The selection method tries to choose the best heuristics at each decision step. The acceptance part is required to evaluate the performance of these chosen heuristics for deciding whether to accept or reject the solutions generated by the heuristics.

Majority of existing hyper-heuristics aim at solving single-objective optimisation problems even though these problems could be multi-objective in nature. The usual methodology to solve these multi-objective problems is to consider them as single-objective optimisation problems by defining a weighted sum of the objectives. Although this idea is reasonable to quickly deliver solutions, it is likely to suffer from missing good solutions. Besides that, it is hard to assign appropriate weights to the objectives since they are usually different in metrics and their precise importance is unobvious. In this respect, a plausible solution strategy would be to deliver a number of solutions forming a set call *pareto front [7]*. A pareto front refers to a group of solutions that are non-dominated considering all the objectives. Non-dominance of solutions mean that none of these solutions are better in terms of all the objectives. Besides the quality of the solutions with respect to each objective separately, it is critical to maintain some level of diversity to have a good pareto front. Providing high diversity gives a high variety of solutions that can meet different needs. In order to furnish such diverse solutions in a weighted-sum setting, a hyper-heuristic framework with a number of weight adaptation schemes is proposed. The experimental results on the PSP show the diversity performance of the proposed approach.

The paper continues as follows. Section 2 defines the problem. Section 3 explains the hyper-heuristic approaches used to solve the problem. A detailed experimental analysis is provided in Section 4. Section 5 finalises the paper with a discussion and possible future research.

2 The Patrol Scheduling Problem

The Patrol Scheduling Problem (PSP) studied here is about addressing the security personnel requirements of a train network composed a group of stations. Each train station needs a number of security teams that should be present during different time periods. A PSP solution provides assignments of the available security teams to the stations. The goal is to generate solutions requiring short travels between a set of stations while providing better security by taking the stations' risks into account. The PSP with the first objective aiming to minimise the travelling distance was studied in [10]. A real-world dataset on the Singapore MRT network was used for the experiments. It was shown in CPLEX failed to find a solution for this particular instance within a reasonable amount of computation time. The problem was then solved by considering each line as a separate problem. The PSP with the both objectives was approached in [11], where an exact model was introduced.

Besides optimising the aforementioned objectives, a feasible PSP solution should satisfy the following constraints:

– Each team can visit only one station during each time period
– Each station should be visited at least for a number of minimum visits requested
– Each station should not be visited more than a number of maximum visits requested
– Each station cannot be visited more than a single team during each time period
– Stations visited by each team should be reachable from one station to the next
– Break periods of each team should be respected

The PSP objectives are considered in the basic weighted-sum form as generalised in Equation 1. In the equation, w refers to the weight for the objective o . However, it is well known that the weighted sum approach suffers the challenge to determine reasonable weights for the objectives. From the search landscape perspective, changing the weights can result in a fitness function converting a hard landscape to an easy one, or vice versa. It is hence a challenging task to determine what the weights should be in order to have a easy-to-search landscape for a particular algorithm. Moreover, when different parts of a landscape are separately analysed, it is even likely to see that different weights can be useful for different parts. Our proposed strategy aims at easily accessing distinct search regions which result in high solution diversity with better solution quality.

$$\sum_{i}^{n} w_i \times o_i \text{ where } \sum_{i}^{n} w_i = 1 \quad (1)$$

3 A Diversity-Oriented Hyper-heuristic Framework

We apply multi-objective selection hyper-heuristics for solving the bi-objective PSP. In the literature, a limited number of hyper-heuristics were introduced for the multi-objective optimisation, where each objective is separately considered. TSRoulWheel [4] was introduced as a selection hyper-heuristic that learns the right heuristics for optimising each objective. A genetic-programming based hyper-heuristic was proposed for automatically generating heuristics to solve the bounded-diameter minimum spanning tree problem in [9]. A population-based Markov chain hyper-heuristic incorporating reinforcement learning was proposed in [13]. Another population-based multi-objective hyper-heuristic was studied to solve the 2D guillotine strip packing and 2D cutting stock problems in [6]. A multi-objective version of a hyper-heuristic with choice function was proposed in [12].

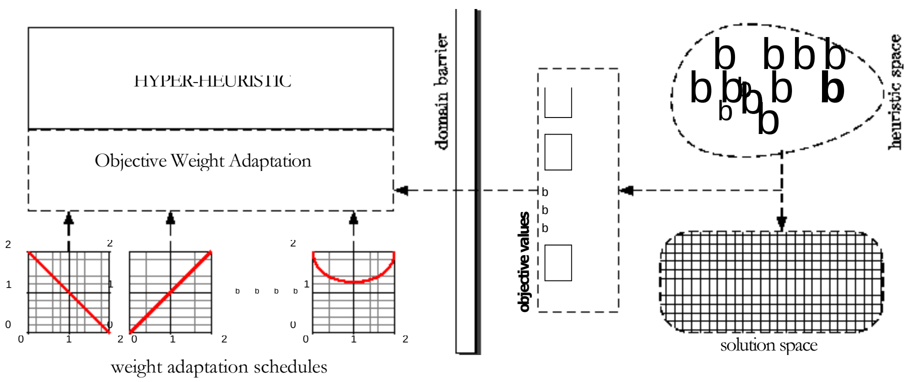

Fig. 1 A multi-objective single-point search hyper-heuristic framework

Unlike the existing hyper-heuristic methods, we introduce a hyper-heuristic framework targeting at solution diversity for multi-objective optimisation. Figure 1 illustrates our proposed framework. The framework is based on single-point search selection hyper-heuristics which manages a set of given low-level heuristics to deliver quick and high quality solutions while manipulating a

Algorithm 1: A multi-objective hyper-heuristic framework

Solution initialisation: $S \leftarrow S_{init}$

Check solution quality: $Q = eval(S)$ where $Q = \{v, o_1, o_2, ..., o_n\}$ and $f(S) = \sum_j^n w_j \times o_j + v$

v: violation, o_j: solution quality wrt. objective j, w_j: weight for the objective j

while *!stoppingCriteria()* **do**

  Choose a LLH_i

  Generate a new solution: $S' \leftarrow LLH_i(S)$

  Evaluate S': $Q = eval(S')$ and $f(S')$

  if *accept(f(S), f(S'))* **then**

    $S \leftarrow S'$

    updateParetoFront(S)

  end

  updateWeights(W)

end

single solution. We incorporate the idea of adaptive weights when a multi-objective optimisation problem is to be solved. For the weight adaptation, it is required to have one or more functions that provide update schedules. Updating weights actually refer to changing the focus of a search. In other words, if the weight of a particular objective is higher than other objectives, the solutions found by an algorithm are likely to be better for this objective. Thus, changing weights means changing the search direction of an algorithm. For this process, three basic functions are used. In the bi-objective case, the first function is *Linear* that sets the weight of the first objective to 1 and the weight of the second objective is set to 0. The weights linearly changes in the other direction over time. In the final phase of search, the first weight becomes 0 while the second objective is set to 1. The second function, i.e. *sin180,* simply updates the weights between 1-0-1 considering the spent time for the 180 degrees of the Sine function. Hence, the objective focus starts from the first objective, gradually moves to the second objective and comes back to the first objective. The inverse case, i.e. *cos180,* applies the Cosine function for updating the weights. Of course, updating weights at each iteration may result in moving around a very small search region. That way, each update is performed at each $1/50$ time of the whole search process. Algorithm 1 explains the steps of the complete framework in details.

For applying this framework, the simple random heuristic selection mechanism [5] is combined with the great deluge move acceptance criterion [8] using exponential diversification scheme. The selection method randomly chooses a heuristic at each iteration. The acceptance criterion accepts better or equal quality solutions and accepts worsening solutions w.r.t. the initial solution and time. Algorithm 2 explains the acceptance procedure.

For initialisation, solutions are randomly constructed while taking some of the constraints into account in order to deliver (near-)feasible solutions. In particular, station consecutiveness, break times and team availability information were considered.

Algorithm 2: Great deluge move acceptance

```
if f(S') ≤ f(S) then
    S ← S
else if f(S') ≤ f(S_initial) X (t_remaining / t_total)^2 then
    S ← S'
end
```

7 low-level heuristics are implemented to solve the PSP. These heuristics are detailed as follows:

– ***LLH*** 1: Change a randomly selected visit with another station
– ***LLH*** 2: Shift left visits from a randomly selected team and add a randomly selected visit instead of last shifted visit while removing the first visit
– ***LLH*** 3: Shift right visits from a randomly selected team and add a randomly selected visit instead of first shifted visit while removing the last visit
– ***LLH*** 4: Change a randomly selected visit causing per station minimum visit violation
– ***LLH*** 5: Change a randomly selected visit causing per station maximum visit violation
– ***LLH*** 6: Swap two visits between two randomly selected team
– ***LLH*** 7: Swap two visits for a randomly selected team

4 Computational Results

The experiments are performed on an Intel i5 1.7 GHz PC with 4 GB of memory. Each test is repeated for 10 times due to the stochastic nature of the hyper-heuristics. Different execution time limits are used for the PSP instances retrieved from the Singapore MRT network as shown in Figure 2. Table 1 presents these instances. Each of these instances is spread across 20 time periods. The first 4 instances represent separate lines. The EW+NS instance is a combination of two lines and EW+NS+NE is composed of three lines as stated in their instance names. The last instance, i.e. ALL, refers to the complete train network involving all the aforementioned lines. 1 minute is set as the running time for the first two instances. The execution time increases to 10 minutes for the next two instances and 30 minutes for the two subsequent instances. The proposed approach is run for 1 hour on the ALL instance.

Table 2 provides the best objective values found on each objective. The results indicate that there is no single weight adaptation scheme that will deliver the best performance. This is consistent with the underlying idea behind hyper-heuristics where there is no single heuristic that always work well. In this respect, a selection method for the update scheme or a learning method to actually adapt the update scheme might be an effective way to resolve this issue. The other way is to run all the update functions to deliver a pareto front together, which refers to the method called ***Combined***.

Table 1 The patrol scheduling problem instances where the total number of periods is 20 (Per station: minimum number of visits = 1, maximum number of visits=2)

Instance	#Teams	#Stations
NE	3	16
CC	3	16
NS	5	25
EW	6	31
EW+NS	9	53
EW+NS+NE	12	67
ALL	14	79

Fig. 2 The Singapore MRT network

Table 2 Best objective values achieved for the total distance travelled and the coverage of the stations w.r.t. their security team requirements (Distance | Coverage)

Instance	Linear	Cos180	Sin180	Combined
NE	15 | 367	15 | 325	15 | 361	15 | 367
CC	15 | 281	15 | 257	15 | 277	15 | 281
NS	25 | 947	25 | 955	25 | 954	25 | 955
EW	30 | 1009	30 | 1027	30 | 942	30 | 1027
EW+NS	47 | 1309	47 | 1379	47 | 1369	47 | 1379
EW+NS+NE	70 | 1893	71 | 1943	70 | 1958	70 | 1958
ALL	85 | 1693	80 | 1724	74 | 1685	74 | 1724

Figure 3 indicates the number of times when each station is visited in the pareto solutions. The solutions reveal that a few stations are frequently visited on certain time periods. For instance, the Raffles Place station that is used by both the NS and EW lines, is visited 9 times during the time period 13 as the highest frequently visited station on a single time period. It is additionally

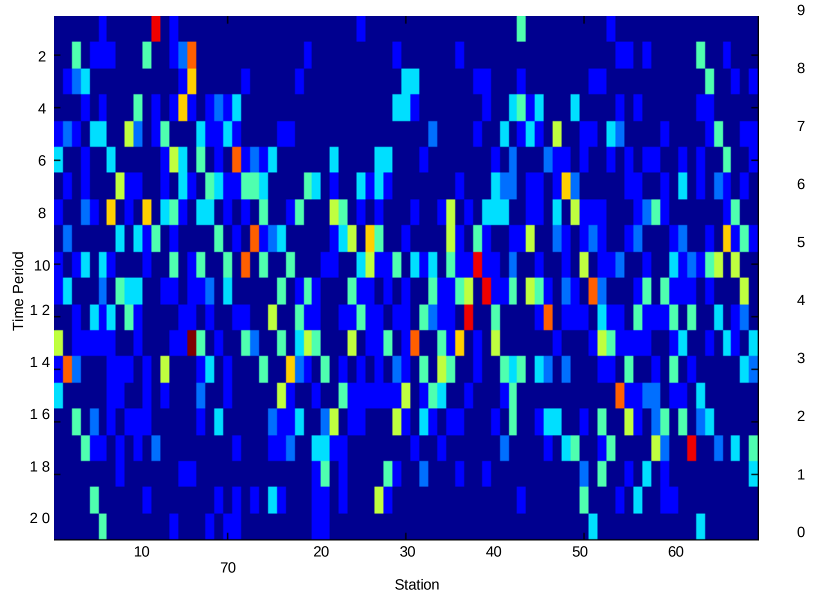

Fig. 3 Visit frequency of the stations on different time periods based on 16 pareto solutions found

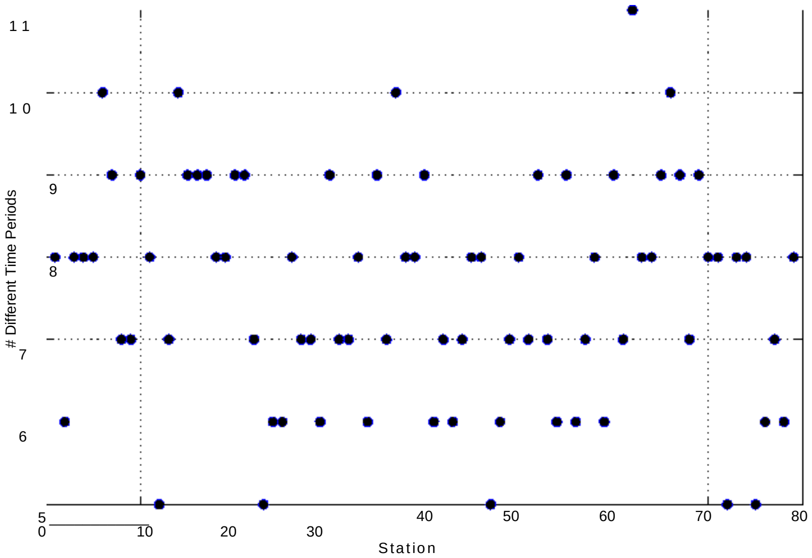

Fig. 4 Number of different time periods for each station visit based on 16 pareto solutions found

visited 7 times on the time period 2 and 6 times on the time period 3. However, it is still possible to visit this particular station during 9 different time periods due to the solution diversity provided by the proposed strategy. Among all the stations, each station is visited at least 4 times considering the highest visit frequency time periods like the City Hall station. The Serangoon station which is a common station between the CC and NE lines, is visited during 11 different time periods, thus it can be considered the most flexible station in terms of visiting time. Figure 4 shows clearer details about the number of different time periods where each station visited. The results indicate that diversity is achieved at the time period level by generating different patrol schedules.

Figure 5 presents the pareto fronts found after using each objective weighing schedules and the one using all methods as black-box, i.e. Combined. The results show that the linear schedule provides high diversity on both objectives while the solutions are relatively low quality compared to both the cos180 and sin180 schedules. Since the cos180 schedule aims to minimise the distance objective more, the solution quality in terms of this objective is better than the rest. However, this approach provides diversity on the other objective. Inversely, sin180 is able to deliver better solutions in terms of the coverage objective and higher diversity for the distance objective. In the Combined version, the pareto is composed of the solutions found both by using cos180 and sin180.

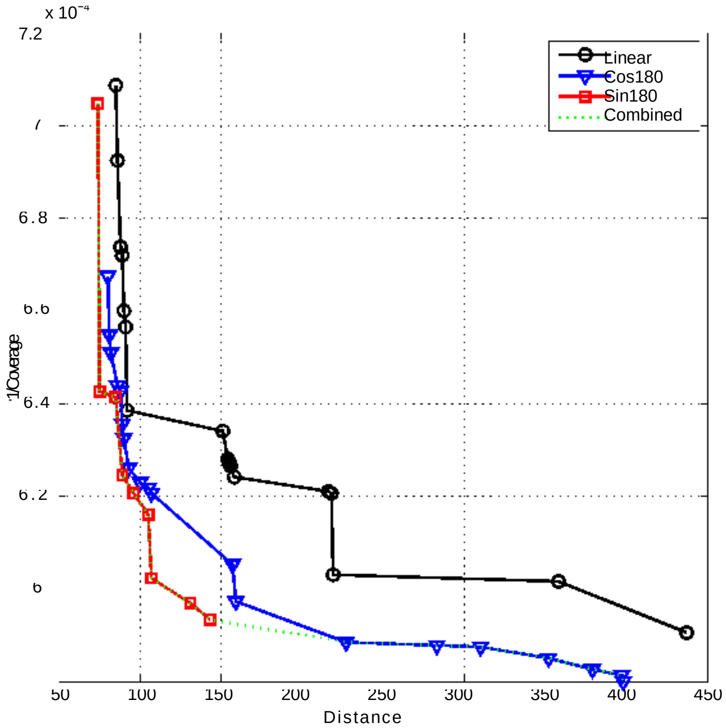

Fig. 5 Pareto fronts determined by different objective weighting schedules on the ALL instance

Figure 6 shows the pareto solutions returned for the remaining instances. The linear strategy delivers the best performance both on the NE and CC instances. Although sin180 has a higher effect on the combined pareto front of the NS instance, the combined pareto front consists of the solutions from all these weight schedule schemes. For the EW instance, the final pareto front blends the linear and cos180 solutions. All these three methods contribute to the pareto front of the EW+NS instance while the linear and sin180 schedules provide pareto solutions on the EW+NS+NE instance. As discussed on the numerical results, these pareto fronts indicate that there is no single weight adaptation strategy for high level solution diversity as well as a better pareto front. Thus, combining different weight adaptation schemes in a black-box form, i.e. Combined, is an effective way to overcome this issue. However, this doesn't necessarily mean that there is no a single mathematical function that can deliver similar or better pareto fronts.

5 Conclusion

This paper studies the problem of generating diverse schedules for the bi-objective patrol scheduling problem. We propose an approach for incorporating objective weight schedules or adaptation schemes that change over time as a single-point search selection hyper-heuristic framework. The idea is to change the objective focus by updating the objectives' weights while solving a given problem instance. The weight updating process is handled by incorporating basic mathematical functions. Besides independently using these functions, a combined approach is additionally proposed to deliver a better overall performance. Experimentally, we evaluated the performance in terms of solution diversity among the resulting pareto solutions. We performed empirical analysis on a real-world dataset for the Singapore rail network, and our results indicated that the weights are extremely critical for diversity. Among the tested weight update schemes, there is no a single scheme which always works well. Thus, choosing the right scheme can be considered another interesting selection problem for hyper-heuristics research. However, we showed that the combined strategy using the strengths of multiple update schemes addressed this issue reasonably well.

Our future research will be about incorporating better weight update schemes using different functions, not just for patrol scheduling, or potentially for any multi-objective optimization problem where solution diversity is the key concern. The test domains will also be extended to those with more than two objectives to evaluate the generality of our approach. Finally, a distributed version of this approach will be devised to take advantage of using multiple machines.

Acknowledgements This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

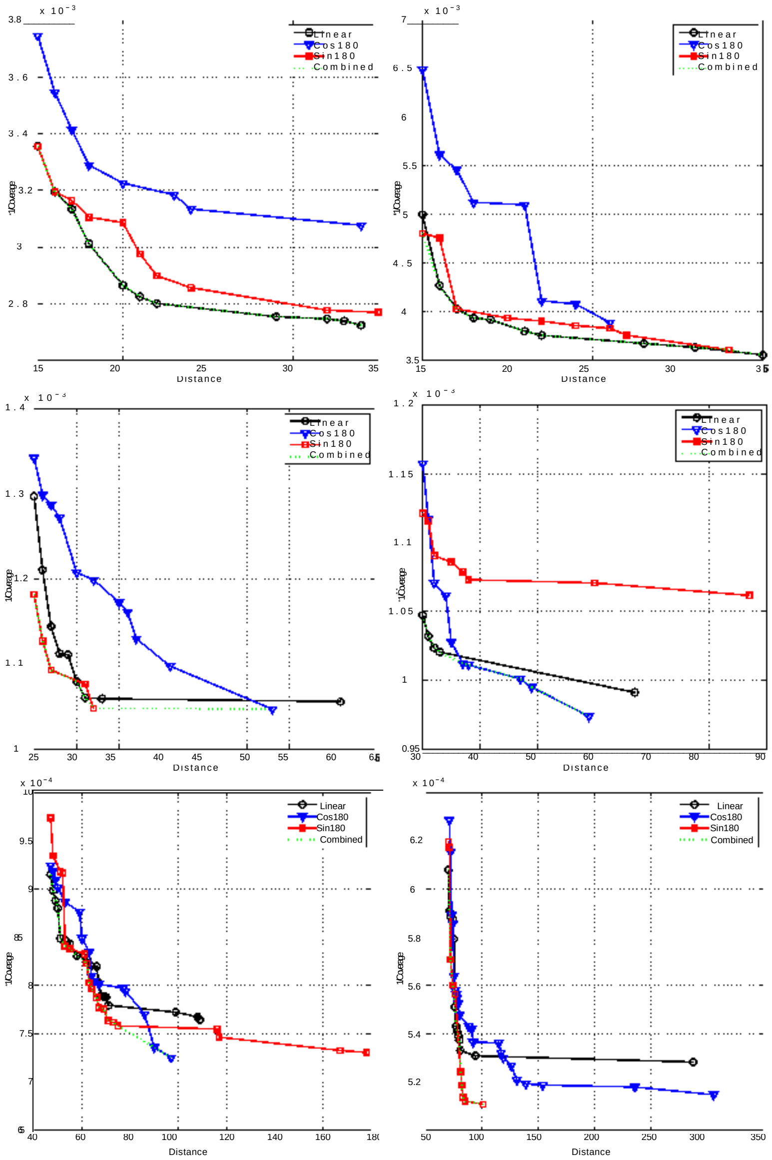

Fig. 6 Pareto fronts determined by different objective weighting schedules on the NE, CC, NS, EW, EW+NS and EW+NS+NE instances (from left to right, top to bottom)

References

1. Bader-El-Den, M., Poli, R., Fatima, S.: Evolving timetabling heuristics using a grammar-based genetic programming hyper-heuristic framework. Memetic Computing **1**(3), 205–219 (2009)
2. Burke, E., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., Qu, R.: Hyper-heuristics: A survey of the state of the art. Journal of the Operational Research Society

695–1724 (2013)
3. Burke, E., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., Woodward, J.: Exploring hyper-heuristic methodologies with genetic programming. Collaborative Computational Intelligence. Springer (2009)
4. Burke, E., Silva, J.L., Soubeiga, E.: chap. Multi-objective Hyper-heuristic Approaches for Space Allocation and Timetabling, pp. 129–158. Springer (2005)
5. Cowling, P., Kendall, G., Soubeiga, E.: A hyperheuristic approach to scheduling a sales summit. In: Selected papers from the 3rd International Conference on Practice and Theory of Automated Timetabling (PATAT'00), pp. 176–190. Springer-Verlag, London, UK (2001)
6. de Armas, J., Miranda, G., Le´on, C.: Hyperheuristic encoding scheme for multi-objective guillotine cutting problems. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation, pp. 1683–1690. ACM (2011)
7. Deb, K., et al.: Multi-objective optimization using evolutionary algorithms, vol. 2012. John Wiley & Sons Chichester (2001)
8. Dueck, G.: New optimization heuristics: The great deluge algorithm and the record-to-record travel. Journal of Computational Physics 104(1), 86–92 (1993)
9. Kumar, R., Bal, B.K., Rockett, P.I.: Multiobjective genetic programming approach to evolving heuristics for the bounded diameter minimum spanning tree problem. In: Proceedings of the 11th Annual conference on Genetic and Evolutionary Computation (GECCO'09), pp. 309–316. ACM (2009)
10. Lau, H.C., Gunawan, A.: The patrol scheduling problem. In: Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT'12). Son, Norway (2012)
11. Lau, H.C., Yuan, Z., Gunawan, A.: Patrol scheduling in urban rail network. Annals of Operations Research (In press)
12. Maashi, M., ¨Ozcan, E., Kendall, G.: A multi-objective hyper-heuristic based on choice function. Expert Systems with Applications (to appear)
13. McClymont, K., Keedwell, E.: Markov chain hyper-heuristic (MCHH): an online selective hyper-heuristic for multi-objective continuous problems. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO'11), pp. 2003–2010 (2011)
14. Mısır, M., Smet, P., Verbeeck, K., Vanden Berghe, G.: Security personnel routing and rostering: a hyper-heuristic approach. In: Proceedings of the 3rd International Conference on Applied Operational Research (ICAOR'11), *LNMS*, vol. 3, pp. 193–205. Istanbul, Turkey (2011)
15. Mısır, M., Verbeeck, K., De Causmaecker, P., Vanden Berghe, G.: Hyper-heuristics with a dynamic heuristic set for the home care scheduling problem. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC'10), pp. 2875–2882. Barcelona, Spain (2010)