

Robustness and Evolutionary Dynamic Optimisation of Airport Security Schedules

Darren M. Chitty, Shengxiang Yang, and Mario Gongora

Centre for Computational Intelligence (CCI),
School of Computer Science and Informatics,
De Montfort University,
The Gateway, Leicester, LE1 9BH, U.K.
darrenchitty@googlemail.com

Abstract. Reducing security lane operations whilst minimising passenger waiting times in unforeseen circumstances is important for airports. Evolutionary methods can design optimised schedules but these tend to *over-fit* passenger arrival forecasts resulting in lengthy waiting times for unforeseen events. Dynamic re-optimisation can mitigate for this issue but security lane schedules are an example of a *constrained* problem due to the *human* element preventing major modifications. This paper postulates that for dynamic re-optimisation to be more effective in *constrained* circumstances consideration of schedule *robustness* is required. To reduce *over-fitting* a simple methodology for evolving more robust schedules is investigated. Random delays are introduced into forecasts of passenger arrivals to better reflect actuality and a range of these randomly perturbed forecasts are used to evaluate schedules. These steps reduced passenger waiting times for actual events for both static and dynamic policies with minimal increases in security operations.

Keywords: airport security lane scheduling, robust dynamic optimisation, evolutionary algorithm

1 Introduction

Airports face pressures to reduce their operational costs whilst improving passenger experiences, a conflicting objective. A key operational cost is the manning of security lanes but reducing security lane operations has the net effect of increasing passenger waiting times and thereby dissatisfaction. Consequently, it is a problem area suited to optimisation to derive security lane schedules that reduce both waiting times and lane operations. Evolutionary methods have been frequently used to solve scheduling problems such as this. However, in uncertain environments such as passenger arrivals, there is a tendency for optimised schedules to *over-fit* predictions of future events with poor performance for unforeseen events. Real-time schedule modification can mitigate for this issue and the authors of this work successfully demonstrated evolutionary dynamic re-optimisation methods to improve security lane schedules at airports [3].

However, within a *constrained* environment such as staff scheduling there are limitations to the modification of schedules restricting the effectiveness of dynamic approaches. Therefore, it is postulated that the *robustness* of original schedules must be of consideration for *constrained* environments to reduce *over-fitting* and improve the performance of both static and dynamic scheduling policies. To raise the robustness of security lane schedules two steps will be considered. Firstly, modifying the supplied forecasts of passenger arrivals by increasing arrival variability to better reflect actual events. The second step then uses this methodology generate a range of varying forecast scenarios which candidate schedules can be tested upon to measure their robustness.

The paper is laid out as follows: Section 2 will present the security lane optimisation problem and briefly describe the evolutionary design of security lane schedules for both static and dynamic policies. Section 3 will present methods to improve the robustness of schedules and contrast the performance of derived schedules with non-robust methods for unforeseen actual events. Finally, Section 4 will profile related works and Section 5 will draw conclusions and present options for future work.

2 Dynamic Optimisation of Security Lane Schedules

2.1 The Security Lane Optimisation Problem

Passengers travelling by air are required to pass through stringent security checks such as hand baggage searches and passing through metal detectors etc. with a number of available security lanes for processing passengers. Security checks are staff intensive and cannot be compromised as maintaining security is paramount. One aspect of security that is open to optimisation is the schedules of opening these security lanes. Clearly, minimising passenger waiting times at security reduces passenger dissatisfaction. Thus, opening all security lanes will achieve this but to the expense of the airport but alternatively, closing lanes will increase passenger waiting times and hence increase dissatisfaction. Therefore, it can be considered that the problem is multi-objective in nature, minimising waiting times and minimising security operations are mutually exclusive objectives. However, passenger demand will ebb and flow and therefore the problem becomes the design of a schedule that ensures low passenger waiting times at peak times and lower security lane opening hours at times of low passenger demand. Figure 1 demonstrates the ebb and flow of passenger demand during a 24h period at an airport for four exemplar problems with regards actual passenger flow data and a supplied generalised forecast of passenger arrivals.

2.2 Evolutionary Optimisation of Security Lane Schedules

The optimisation objective of the security lane problem is to simultaneously reduce passenger waiting times whilst also minimising the degree to which security lanes are open hence reducing costs. The two objectives are mutually exclusive

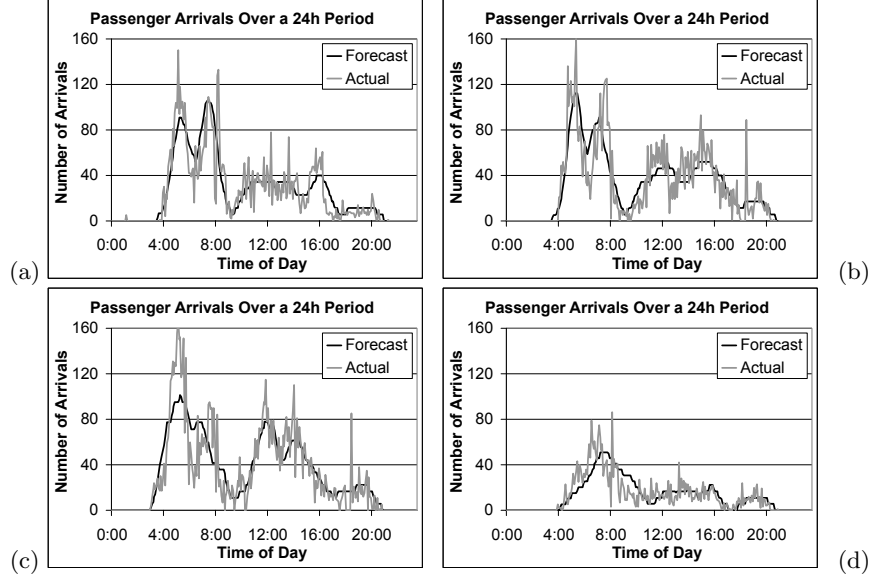


Fig. 1. The forecast arrivals of passengers at security and a set of actual passenger arrivals over a 24 hour period for the exemplar problems labelled (a) F_PAXflow_2425, (b) F_PAXflow_2428, (c) F_PAXflow_2501 and (d) F_PAXflow_21113 respectively.

hence the problem is multi-objective. The primary objective is to minimise passenger waiting times as defined by:

$$\text{minimise} \quad f_1 = \max_{i \in \{1, \dots, m\}} (W_i), \quad (1)$$

where W_i is the waiting time experienced by the i^{th} passenger at the security queue and m is the number of passengers that arrive over the time period.

The secondary objective is to minimise the degree of time to which security lanes are open during over the stated time period, defined as follows:

$$\text{minimise} \quad f_2 = \sum_{i=1}^{i \leq n} S_i, \quad (2)$$

where S_i is the time for which the i^{th} security lane shift lasts and n is the number of shifts within the schedule.

Essentially, the key objective is to minimise the maximum passenger waiting time experienced by a single passenger across the whole time period. Therefore, the multi-objective problem can be simplified to finding the lowest maximum waiting time experienced by a passenger with the fewest hours of security lane operation. To derive the optimal security lane operational schedule an evolutionary approach can be used by deploying a Genetic Algorithm (GA) [7] whereby a candidate schedule is represented as a set of shifts defined by a start and finish

time with shifts restricted to being between two and four hours in length and each gene represents a shift. Since a set of shifts constituting a schedule can be variable in nature, a variable GA approach is used [6]. Two point crossover swaps subsets of shifts between two candidate solutions with these subsets being of differing size. Mutation consists of either swapping a subset of shifts with a random replacement set or a low probability bitwise mutation of starting and finishing times of shifts. In terms of fitness selection, a candidate schedule with a lower maximum passenger waiting time is considered the fitter. If the times are identical then the schedule with the lower degree of lane operation is considered the fitter. A simulation based approach is used to measure passenger waiting times. Passengers are simulated arriving at security defined by the passenger flow forecast and enter a queue operating in a First In First Out (FIFO) manner. Open security lanes take passengers from this queue and process them which is defined as randomly taking between 15 and 21 seconds per passenger. Ten simulations are used to account passenger processing variance.

However, as evidenced in Figure 1, passenger arrivals will often not reflect the predicted forecast with bad weather weather or road traffic accidents causing changes to passenger arrivals. With schedules optimised to the forecast this will likely cause significant queues with security lanes not being open, these schedules essentially *over-fit* the forecast and by minimising lane opening hours there is no spare capacity. To address this issue a dynamic re-optimisation approach can be used to improve these optimised schedules by modifying the shifts. In fact, resource managers often alter schedules to suit demand known as real-time shift updating [9]. However, there are constraints with this policy in that shifts due to their human component may only have their start time brought forward or pushed back by up to an hour and similarly for the finish time with shifts restricted to being between two and four hours in length. To dynamically modify security lane schedules a re-optimisation is performed every hour using the same aforementioned evolutionary approach. Forecast passenger flow is used for simulated future arrivals and actual passenger events are represented purely by the current passengers in the queue which could be much larger than expected. Further details of the approach can be found in [3].

Table 1. GA Parameters used throughout unless otherwise stated.

Population Size	100
Max Generations	2,000
Tournament Size	7
Crossover Probability	0.9
Mutation Probability	0.1
Primary Fitness Measure	Minimisation of max. passenger waiting time
Secondary Fitness Measure	Minimisation of total lane opening time

To establish the effectiveness of an evolutionary approach to the design of security lane schedules for both static and dynamic policies experiments are conducted for the four exemplar problems with results averaged over 25 random runs. Initial schedules are evolved using the forecast passenger flow information. These are then tested against the actual passenger flow information. Moreover,

Table 2. The average maximum waiting times and the scheduled total lane opening time using actual passenger arrivals for a range of available lanes for the optimal static and the dynamically re-optimised schedules. Results averaged over 25 evolved schedules and 10 simulations with varying passenger processing times.

Problem	Max. Lanes	Max. Wait Experienced Over 24h Period (in hours)		Schedule Total Lane Opening (in hours)	
		Static	Dynamic	Static	Dynamic
F_PAXflow_2425	4	2.70 ± 0.26	1.80 ± 0.32	38.14 ± 1.07	42.88 ± 1.55
	5	2.39 ± 0.13	1.90 ± 0.29	42.04 ± 2.06	45.97 ± 2.72
	6	2.39 ± 0.15	1.95 ± 0.26	46.74 ± 2.48	50.67 ± 2.72
	7	2.29 ± 0.45	1.77 ± 0.45	60.06 ± 2.40	65.38 ± 2.46
F_PAXflow_2428	4	2.27 ± 1.17	0.80 ± 0.09	41.02 ± 0.78	44.03 ± 1.61
	5	1.04 ± 0.21	0.69 ± 0.13	44.48 ± 1.68	47.90 ± 2.06
	6	0.45 ± 0.02	0.43 ± 0.05	51.68 ± 2.34	53.80 ± 2.58
	7	0.26 ± 0.00	0.23 ± 0.03	64.10 ± 2.70	66.18 ± 2.89
F_PAXflow_2501	4	3.20 ± 0.86	1.33 ± 0.44	45.12 ± 1.00	48.30 ± 1.70
	5	0.78 ± 0.08	0.73 ± 0.01	48.82 ± 1.81	52.70 ± 1.72
	6	0.43 ± 0.01	0.43 ± 0.01	65.22 ± 1.70	67.70 ± 1.87
	7	0.23 ± 0.00	0.23 ± 0.00	73.34 ± 3.23	76.70 ± 3.57
F_PAXflow_21113	4	0.14 ± 0.00	0.14 ± 0.00	35.20 ± 1.45	36.87 ± 1.88
	5	0.12 ± 0.01	0.12 ± 0.01	44.64 ± 1.78	47.38 ± 2.07
	6	0.12 ± 0.02	0.12 ± 0.02	53.56 ± 2.21	56.21 ± 2.69
	7	0.12 ± 0.02	0.12 ± 0.02	60.08 ± 2.83	63.05 ± 3.35
	8	0.10 ± 0.01	0.10 ± 0.01	66.82 ± 4.66	70.91 ± 5.07

the dynamic re-optimisation of these schedules is also tested against actual passenger flow events. The parameters used by the GA are shown in Table 1.

The results in terms of maximum passenger waiting times and total security lane opening hours from evolving schedules and dynamically re-optimising them throughout the given time period are shown in Table 2. It is clear to see that there are considerable maximum passenger waiting times over several hours in length for the static schedules for actual arrival events. Dynamic re-optimisation of these schedules results in significant reductions in these maximum waiting times. This demonstrates how the static schedules have *over-fit* the forecast in terms of the pattern of lane opening hours matching projected peaks in passenger demand. Deviations from this forecast results in significant passenger delays. Clearly, the dynamic approach is highly effective in mitigating for the *over-fitting* issue but it can be considered that there is a limit to the degree to which it can improve passenger experiences as a result of the constrained nature of modifying schedules.

3 Evolving More Robust Security Lane Schedules

3.1 Considering a More Realistic Forecast of Passenger Arrivals

A problem with the optimisation of the airport security lane schedules can be considered that the supplied projected forecast of passenger arrivals is too smooth when compared to actual events as evidenced in Figure 1. Moreover, from the results in Table 2 it was observed that the static schedules have unacceptably lengthy passenger queueing times most likely a result of the evolved schedules *over-fitting* this smooth forecast. Consequently, it is considered that a more realistic forecast could improve results. A simple methodology to achieve this is to take each passenger's forecast arrival time at security and add a random time penalty causing a passenger to be either later or earlier than predicted. A normal (gaussian) distribution with a mean of zero and a standard deviation of fifteen minutes is used to generate penalties. The new forecasts for each exemplar problem are shown in Figure 2 alongside the actual passenger arrival events data whereby it can be observed that the forecasts are now more variable than previously although not mirroring actual events which is to be expected.

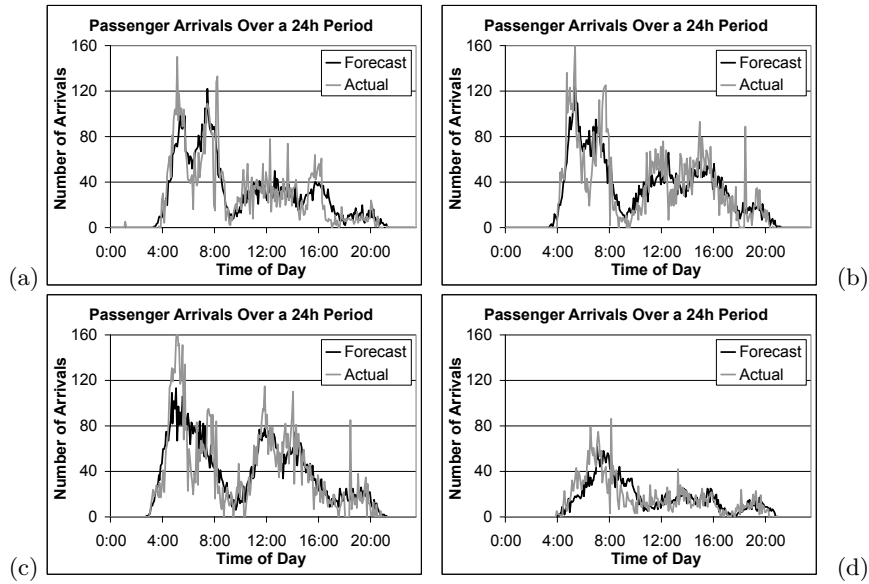


Fig. 2. Randomly perturbed forecast arrivals of passengers at security and a set of actual passenger arrivals during a given 24 hour period for the exemplar problems labelled (a) F_PAXflow_2425, (b) F_PAXflow_2428, (c) F_PAXflow_2501 and (d) F_PAXflow_21113 respectively.

The experiments with evolving security lane schedules are repeated using this new forecast of passenger arrivals with the results from actual events shown in Table 3. Comparing these results to those in Table 2 it can be observed

Table 3. The average maximum waiting times and lane opening hours using the actual passenger volumes for static and dynamically re-optimised schedules evolved using the randomly modified forecast information. Results averaged over 25 evolved schedules and 10 simulations with varying passenger processing times. Bold values indicate improvements in waiting times over those in Table 2

Problem	Max. Lanes	Max. Wait Experienced Over 24h Period (in hours)		Schedule Total Lane Opening (in hours)	
		Static	Dynamic	Static	Dynamic
F_PAXflow_ 2425	4	2.36 ± 0.18	1.44 ± 0.01	38.88 ± 1.71	42.97 ± 1.80
	5	1.90 ± 0.14	1.43 ± 0.01	43.64 ± 1.60	46.18 ± 1.92
	6	1.72 ± 0.52	1.29 ± 0.35	54.28 ± 3.39	57.26 ± 3.39
	7	1.60 ± 0.61	1.23 ± 0.42	56.70 ± 2.54	60.16 ± 2.77
	8	1.84 ± 0.25	1.41 ± 0.10	66.74 ± 3.78	71.11 ± 4.08
F_PAXflow_ 2428	4	1.32 ± 0.28	0.82 ± 0.06	42.11 ± 1.27	45.48 ± 1.41
	5	0.78 ± 0.13	0.60 ± 0.10	45.98 ± 1.48	48.89 ± 1.83
	6	0.45 ± 0.02	0.43 ± 0.04	54.02 ± 2.02	56.65 ± 2.33
	7	0.26 ± 0.01	0.24 ± 0.03	64.78 ± 2.63	66.78 ± 3.13
	8	0.22 ± 0.01	0.21 ± 0.02	70.41 ± 3.34	72.94 ± 3.64
F_PAXflow_ 2501	4	2.50 ± 0.73	1.25 ± 0.04	46.36 ± 0.78	49.73 ± 1.34
	5	0.75 ± 0.02	0.73 ± 0.01	52.02 ± 1.29	55.71 ± 1.41
	6	0.43 ± 0.01	0.43 ± 0.01	65.34 ± 3.02	67.76 ± 3.44
	7	0.23 ± 0.00	0.23 ± 0.00	70.74 ± 2.52	74.04 ± 2.57
	8	0.20 ± 0.02	0.17 ± 0.02	81.74 ± 3.45	86.35 ± 4.02
F_PAXflow_ 21113	4	0.17 ± 0.00	0.17 ± 0.00	36.78 ± 1.61	39.31 ± 1.82
	5	0.13 ± 0.01	0.13 ± 0.01	44.88 ± 2.16	47.12 ± 2.50
	6	0.12 ± 0.01	0.12 ± 0.01	51.56 ± 2.61	53.60 ± 3.19
	7	0.09 ± 0.01	0.09 ± 0.01	62.14 ± 2.93	65.58 ± 3.07
	8	0.09 ± 0.01	0.09 ± 0.01	68.86 ± 2.15	72.77 ± 2.88

that in many cases the static schedules have much lower maximum passenger waiting times when using the updated forecast that is randomly perturbed. Indeed, in some cases the maximum waiting time experienced by a passenger is up to 50% lower. Reductions in the maximum passenger waiting times are also observed for the dynamic re-optimisation policy although not to the extent as those seen for the static schedules demonstrating the effectiveness of the dynamic re-optimisation for sub-optimal schedules. However, an obvious reason that passenger waiting times are improved from using the updated forecast is that there has been an increase in security lane opening hours and inspection of Table 3 and comparing to Table 2 bears this out. However, in most cases from using the updated forecast information, the increase in opening hours is relatively small. The increases though get larger as the number of available lanes increases.

3.2 Using Multiple Randomly Manipulated Predictions

The current methodology has been to use a single forecast of passenger arrivals to evolve initial static security lane schedules that minimise both passenger waiting times and security lane opening hours. To deal with optimised schedules *overfitting* this forecast evolutionary dynamic re-optimisation has been shown to successfully mitigate this issue to some degree. However, dynamic re-optimisation is limited in its success due to the constrained nature of schedules, a schedule unfit

for actual events cannot be greatly improved. Using a more accurate forecast that better reflects actual events can provide improved results for both static and dynamic policies.

Consequently, it is considered that further steps could be taken to reduce this *over-fitting* of a forecast of passenger arrivals by ensuring initial schedules are more robust. Indeed, the approach from the previous section of generating a new forecast by applying random time penalties to projected passenger arrival times could be extended. Instead of doing this once, a range of randomly perturbed forecasts can be generated to prevent the security lane schedules matching a single prediction of passenger events. A set of ten forecasts can be generated using the approach from the previous section. Candidate schedules can then be tested against each forecast using the simulation based approach as described previously. The maximum passenger waiting time experienced is then derived as the average across these ten forecasts. This approach should derive schedules that are more robust to unexpected passenger arrivals as evolved schedules do not fit a single forecast.

Table 4. The average maximum waiting times and lane opening hours using the actual passenger volumes for the static and dynamically re-optimised schedules evolved using a range of randomly modified forecasts. Results averaged over 25 evolved schedules and 10 simulations with varying passenger processing times. Bold values indicate improvements in waiting times over those in Table 3

Problem	Max. Lanes	Max. Wait Experienced Over 24h Period (in hours)		Schedule Total Lane Opening (in hours)	
		Static	Dynamic	Static	Dynamic
F_PAX flow_2425	4	2.40 ± 0.14	1.44 ± 0.01	38.77 ± 0.82	43.28 ± 1.30
	5	1.78 ± 0.31	1.39 ± 0.13	44.98 ± 1.75	47.56 ± 2.03
	6	1.79 ± 0.45	1.34 ± 0.31	59.40 ± 2.08	63.66 ± 2.51
	7	1.74 ± 0.52	1.30 ± 0.37	67.46 ± 2.28	71.93 ± 2.50
	8	1.92 ± 0.10	1.43 ± 0.00	77.76 ± 3.59	82.93 ± 4.18
F_PAX flow_2428	4	1.34 ± 0.28	0.77 ± 0.07	42.56 ± 1.09	46.10 ± 1.66
	5	0.70 ± 0.12	0.60 ± 0.09	47.36 ± 1.55	50.74 ± 1.96
	6	0.22 ± 0.00	0.22 ± 0.01	62.98 ± 1.62	65.19 ± 1.89
	7	0.21 ± 0.02	0.19 ± 0.03	74.48 ± 2.19	77.60 ± 2.67
	8	0.21 ± 0.02	0.19 ± 0.03	82.60 ± 2.19	85.88 ± 2.90
F_PAX flow_2501	4	1.85 ± 0.28	1.25 ± 0.01	47.02 ± 1.22	50.50 ± 1.72
	5	0.76 ± 0.02	0.73 ± 0.01	52.77 ± 1.80	56.54 ± 1.97
	6	0.42 ± 0.00	0.42 ± 0.00	69.32 ± 1.46	72.23 ± 1.82
	7	0.23 ± 0.00	0.23 ± 0.00	79.79 ± 1.97	83.57 ± 2.41
	8	0.16 ± 0.02	0.15 ± 0.02	91.36 ± 3.47	96.00 ± 3.98
F_PAX flow_21113	4	0.12 ± 0.01	0.12 ± 0.01	42.34 ± 1.17	43.80 ± 1.55
	5	0.09 ± 0.00	0.09 ± 0.00	51.36 ± 1.78	53.07 ± 2.02
	6	0.08 ± 0.00	0.08 ± 0.00	59.90 ± 2.05	62.97 ± 2.67
	7	0.07 ± 0.00	0.07 ± 0.00	68.84 ± 2.09	71.43 ± 2.86
	8	0.07 ± 0.00	0.07 ± 0.00	78.60 ± 4.28	82.25 ± 4.86

The experiments from previously are repeated but now using ten differing randomly perturbed forecasts with the results shown in Table 4. By comparing these results to those from Table 3 it can be observed that the use of multiple

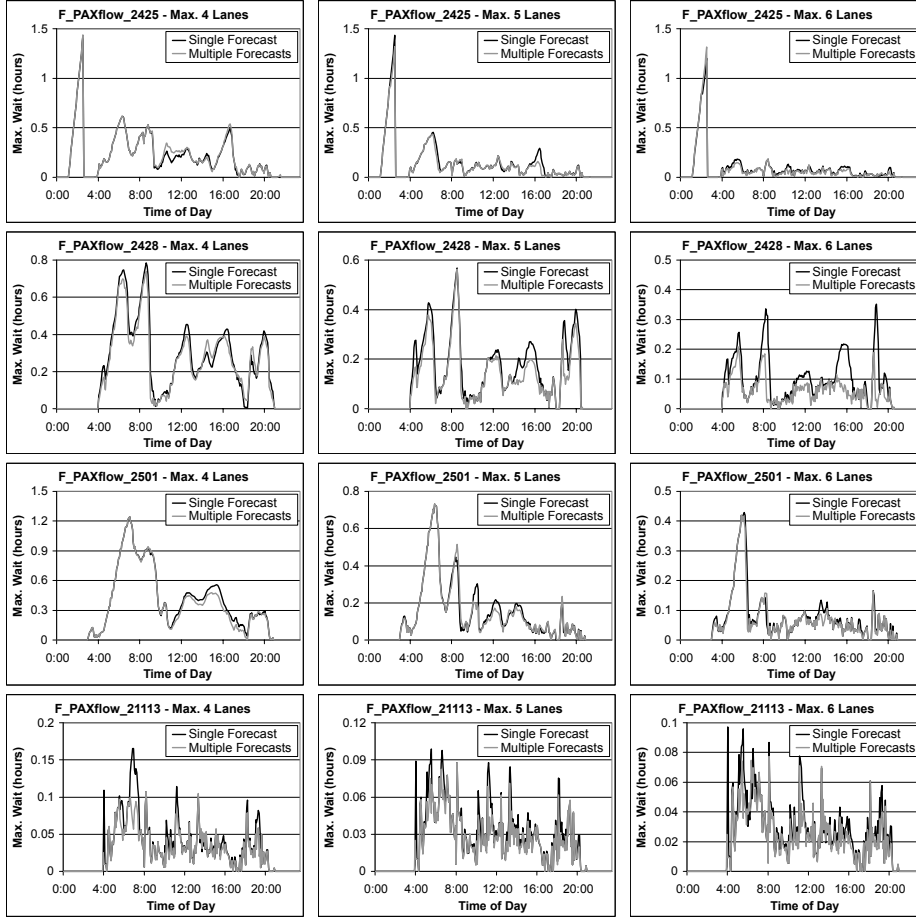


Fig. 3. The average passenger waiting time at security at five minute intervals for the dynamically re-optimised schedules using a single forecast and a range of forecasts.

forecasts to reduce *over-fitting* has provided generally better results in terms of the maximum passenger waiting time for both static and dynamic schedules although some increases occur especially for the F_PAXflow_2425 problem. However, these are caused by the highly unusual nature of passengers arriving very early in the day when no lanes are open which is difficult to mitigate for. A similar situation can be observed with the evolutionary dynamic re-optimisation of the static schedules. Clearly, as a result of the constraints on schedules, if the original schedule is poor then the dynamic re-optimisation approach will only be able to achieve minor improvements in passenger waiting times. Greater detail is shown in Figure 3 for the dynamic policy in terms of average passenger waiting times throughout the time period. Although the waiting times are mixed between using a single and multiple forecasts it can be observed that the use of multiple forecasts does reduce passenger waiting times in the main. Of additional note is that in contrast to Table 3, the degree of security lane openings has increased

for both static and dynamic policies in some cases over 10% although this is with greater lane availability whereby passenger waiting times are lower. Clearly though, more robust schedules will have an increase in security lane opening hours in any case. The original methodology of using a single forecast meant the evolved schedules reduced security lane operations to fit only this forecast.

4 Related Work

There is limited literature associated with the optimisation of airport security lane schedules to reduce passenger waiting times. Soukour *et al.* [16] used a memetic algorithm merged with an evolutionary algorithm to assign security staff concentrating on reducing over and undertime and raising staff satisfaction. However, the security lane problem is similar to optimising airport check-in desks to minimise passenger delays and the degree to which desks are open. Wang and Chun [17] used a GA for optimal counter assignment for check-in desks. Chun and Mak [4] used simulation and search heuristics to determine the optimal check-in desk allocation that reduces the time desks are open and acceptable queue lengths for Hong Kong Airport. Bruno and Genovese [2] proposed a number of optimisation models for the check-in service balancing operational costs with passenger waiting times for Naples airport. Araujo and Repolho [1] present a new methodology to optimise the check-in desk allocation problem of maintaining a service level whilst reducing operational costs. Three phases are used whereby the first optimises the number of desks based upon [2], the second uses simulation to test the service level and the third uses an optimisation model to solve an adjacent desk constraint. Integer programming is used to solve both a common and dedicated desk problem. Mota [11] uses an evolutionary algorithm and a simulation approach to establish the allocation and opening times of check-in desks to reduce passenger waiting times.

The dynamic optimisation of check-in desks has been investigated by Parlar *et al.* [15, 14] with regards the optimal opening of desks to minimise a monetary cost determined as the financial cost of waiting passengers and the cost of open check-in desks and aircraft delays solved using dynamic programming for a single flight scenario. A static policy was recommended as a dynamic policy was found to suffer from the curse of dimensionality [14]. Hsu *et al.* [8] investigated the dynamic allocation of check-in facilities and passengers to desks defined as a Sequential Stochastic Assignment Problem and solved using binary integer programming with positive results. Nandhini *et al.* [12] investigated the dynamic optimisation of check-in desks to minimise the conflicting objectives of resource allocation and passenger waiting times using a GA.

With regards robustness and evolutionary dynamic scheduling there have been some significant works within the related area of job shop scheduling. Jensen [10] considers a robustness measure for evolved schedules based on the performance of other schedules within the neighbourhood since a modified schedule will likely be one of these. The measure was found to outperform alternatives from the literature. Hart *et al.* [5] consider the use of an Artificial Immune System (AIS) to design robust dynamic job shop schedules by generating sets of

schedules such that if a reschedule is required another from the set can be used. More generally for evolutionary methods Paenke *et al.* [13] use a similar approach to achieve robustness as used in this paper by averaging over a range of scenarios. However, the authors use a fitness approximation approach rather than pure simulation to reduce computational cost.

5 Conclusion

This paper has considered the role of robustness in evolutionary dynamic re-optimisation of schedules in *constrained* environments. It was demonstrated that when evolving optimised schedules for airport security lane problem for minimising the conflicting objectives of waiting times and lane operations, schedules would *over-fit* forecasts and perform badly for unforeseen passenger arrival events with lengthy delays. Dynamic re-optimisation of schedules was shown to mitigate for this issue. However, modification of schedules involving human operators is an example of a *constrained* environment in that scheduled shifts cannot be changed to a great degree. Consequently, this paper hypothesised that *robustness* must be considered when optimising schedules to improve schedule performance in unforeseen circumstances. Two simple measures were used to facilitate robustness, the generation of forecasts of passenger arrivals with randomised delays to better reflect actual events and using a range of these scenarios to evaluate candidate schedules upon. Experiments demonstrated significant reductions in waiting times when introducing random delays into the forecasts of passenger arrivals for both static and dynamic policies highlighting the importance of improving forecast accuracies. Further experiments with robustness demonstrated reductions in passenger waiting times for both static and dynamic re-optimisation policies over a non-robust approach in a majority of cases with minimal increases in security lane operations. However, the degree of the reductions is much larger for static schedules rather than an evolutionary dynamic re-optimisation policy. This demonstrates the effectiveness of evolutionary dynamic re-optimisation of security lane schedules but that some consideration of robustness will benefit the approach.

Further work should consider improved methods of generating forecasts of passenger arrivals perhaps using Bayesian methods to apply penalties taking into consideration the time of day whereby delays are more likely to happen and also be lengthier such as at rush hour. Furthermore, schedule flexibility could be incorporated into the initial evolution of schedules to improve the evolutionary dynamic re-optimisation methodology.

6 Acknowledgment

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of U.K. under grant EP/K001310/1.

References

1. Araujo, G.E., Repolho, H.M.: Optimizing the airport check-in counter allocation problem. *Journal of Transport Literature* 9(4), 15–19 (2015)
2. Bruno, G., Genovese, A.: A mathematical model for the optimization of the airport check-in service problem. *Electronic Notes in Discrete Mathematics* 36, 703–710 (2010)
3. Chitty, D.M., Gongora, M., Yang, S.: Evolutionary dynamic optimisation of airport security lane schedules. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–8 (Dec 2016)
4. Chun, H.W., Mak, R.W.T.: Intelligent resource simulation for an airport check-in counter allocation system. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 29(3), 325–335 (1999)
5. Hart, E., Ross, P., Nelson, J.: Producing robust schedules via an artificial immune system. In: *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence*. pp. 464–469. IEEE (1998)
6. Harvey, I.: Species adaptation genetic algorithms: A basis for a continuing saga. In: *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. pp. 346–354 (1992)
7. Holland, J.H.: *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press (1975)
8. Hsu, C.I., Chao, C.C., Shih, K.Y.: Dynamic allocation of check-in facilities and dynamic assignment of passengers at air terminals. *Computers & Industrial Engineering* 63(2), 410–417 (2012)
9. Hur, D., Mabert, V.A., Bretthauer, K.M.: Real-time work schedule adjustment decisions: An investigation and evaluation. *Production and Operations Management* 13(4), 322–339 (2004)
10. Jensen, M.T.: Generating robust and flexible job shop schedules using genetic algorithms. *IEEE Transactions on evolutionary computation* 7(3), 275–288 (2003)
11. Mota, M.A.M., Zuniga, C.: A simulation-evolutionary approach for the allocation of check-in desks in airport terminals. In: *ATOS 2013, 4th International Air Transport and Operations Symposium* (2013)
12. Nandhini, M., Palanivel, K., Oruganti, S.: Optimization of airport check-in service scheduling (2012)
13. Paenke, I., Branke, J., Jin, Y.: Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. *IEEE Transactions on Evolutionary Computation* 10(4), 405–420 (2006)
14. Parlar, M., Rodrigues, B., Sharafali, M.: On the allocation of exclusive-use counters for airport check-in queues: static vs. dynamic policies. *Opsearch* 50(3), 433–453 (2013)
15. Parlar, M., Sharafali, M.: Dynamic allocation of airline check-in counters: a queueing optimization approach. *Management Science* 54(8), 1410–1424 (2008)
16. Soukour, A.A., Devendeville, L., Lucet, C., Moukrim, A.: A memetic algorithm for staff scheduling problem in airport security service. *Expert Systems with Applications* 40(18), 7504–7512 (2013)
17. Wang, B., Chun, H.: Check-in counter allocation using genetic algorithm. In: *Proc. Of the 7th Conf. on Artificial Intelligence and Expert Systems Applications* (1995)