Evolutionary Dynamic Optimisation of Airport Security Lane Schedules

Darren M. Chitty
Centre for Computational
Intelligence (CCI),
School of Computer Science
and Informatics,
De Montfort University,
Leicester LE1 9BH, UK
Email: darrenchitty@googlemail.com

Mario Gongora
Centre for Computational
Intelligence (CCI),
School of Computer Science
and Informatics,
De Montfort University,
Leicester LE1 9BH, UK
Email: mgongora@dmu.ac.uk

Shengxiang Yang
Centre for Computational
Intelligence (CCI),
School of Computer Science
and Informatics,
De Montfort University,
Leicester LE1 9BH, UK
Email: syang@dmu.ac.uk

***Abstract*—Reducing costs whilst maintaining passenger satisfaction is an important problem for airports. One area this can be applied is the security lane checks at the airport. However, reducing costs through reducing lane openings typically increases queue length and hence passenger dissatisfaction. This paper demonstrates that evolutionary methods can be used to optimise airport security lane schedules such that passenger dissatisfaction and staffing costs can be minimised. However, it is shown that these schedules typically over-fit the forecasts of passenger arrivals at security such that in actuality significant passenger delays can occur with deviations from the forecast. Consequently, this paper further demonstrates that dynamic evolutionary re-optimisation of these schedules can significantly mitigate this over-fitting problem with much reduced passenger delays.**

I. Introduction

Airports are under increasing financial pressures and as such achieving financial savings through a reduction in operational costs is becoming increasingly important. One major staff intensive operation within an airport is the manning of security lanes which passengers are required to pass through on transition to the air-side. Security is paramount at an airport hence there can be no operational reduction in the actual screening of passengers as this would increase security risks. However, the number of open security lanes can be optimised in order to reduce staffing costs although this will naturally come at a cost of extra waiting times for passengers queuing to pass through security.

Consequently, the work presented in this paper is concerned with optimising the security lane operations such that the number of open security lanes during a given time period are minimised at all times thereby reducing staffing costs. However, consideration must also be given to ensuring that the waiting times of passengers queuing to pass through security is reduced as much as possible. Additionally, the work presented in this paper considers the robustness of these optimal schedules to variations in patterns of passenger arrivals and seeks to mitigate any significant passenger delays through a dynamic re-optimisation of these optimised schedules.

The work is laid out as follows. Section II will describe the security lane optimisation problem being tackled. Section III will profile the technique used to evolve optimal security lane schedules. Section IV demonstrates how the optimised schedules can *over-fit* expected passenger arrivals leading to significant delays whilst online dynamic evolutionary re-optimisation can mitigate this problem. Finally, Section V provides an overview of related work in the field and Section VI summarises and draws conclusions from the presented work.

II. The Security Lane Optimisation Problem

All passengers wishing to travel by air are required to forego a security check in order to reach the air-side of an airport. This requires passing through a number of security checks such as hand luggage checking and passing through metal detector machines etc. Consequently, an airport will have a number of available security lanes that can process passengers for their transition to the air-side. However, the volume of passengers arriving at security will ebb and flow during the day primarily as a result of the numbers of departing aircraft at a given time. Therefore, in order to reduce the staffing costs of manning open lanes, not all the security lanes need to be open especially if demand is low. Indeed, airports are under rising financial pressures and minimising the staffing costs of manning security lanes is of key consideration.

However, reducing the number of security lanes open at any time will consequently increase the waiting time of passengers wishing to pass through security. Indeed, as demand can vary greatly during the day, if there are not enough security lanes open in times of high demand, waiting times could increase significantly which could impinge on passenger satisfaction. Indeed, Gkritza *et al.* [1] conducted a study into how increasing waiting times impacted passenger satisfaction and found that it had a profound effect on passengers although the authors conceded that other factors can also effect passenger satisfaction. Moreover, it is conceivable that if frequent delays occurred to departing aircraft as a result of problems at security, airlines may consider moving their aircraft to another airport resulting in financial implications for the airport under consideration.

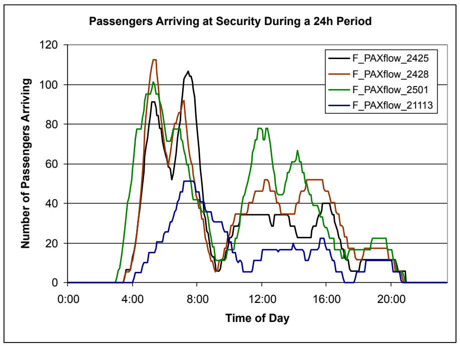


Fig. 1. The forecast arrivals of passengers at security for differing days labelled, F_PAXflow_2425, F_PAXflow_2428, F_PAXflow_2501 and F_PAXflow_21113 respectively.

Therefore, the problem becomes the minimisation of staff costs of manning the security lanes by reducing the amount of time which lanes are open whilst also reducing the waiting times experienced by passengers queuing. These two objectives are mutually exclusive. Reducing the number of open security lanes will likely result in a consequent increase in the waiting times experienced by passengers. However, as previously mentioned, passenger demand ebbs and flows during the day dependant on the number of departing flights. Therefore, it should be possible to reduce the number of open security lanes at times of low demand whilst increasing the number of open security lanes at peak demand times. This should overall provide a minimisation of the staffing costs associated with open security lanes but with minimal impact on increasing passenger waiting times at security.

A. Passenger Flow

In order to be able provide an optimisation of the opening times of security lanes such that in times of low demand the number of open lanes is reduced, information with regards passenger numbers arriving at security throughout the day is required. This information is easily obtainable and enables a forecast to be made throughout a given time period of the numbers of passengers expected to be arriving at the security lane area. This will enable the planning of the opening and closing of security lanes to meet passenger demand.

Four examples of the forecast of passenger arrivals at the security area for a 24h period for differing flight schedules are shown in Figure 1 labelled *F_PAXflow_2425*, *F_PAXflow_2428*, *F_PAXflow_2501* and *F_PAXflow_21113* respectively. From these forecasts it can be seen that there is considerable variation in the number of passengers arriving at security during the 24h period. In general, two peaks are observed, one early in the morning and the other early in the afternoon and likely coincide with peak aircraft departures.

III. Optimising Security Lane Schedules

The goal of the optimisation process is to reduce the waiting times experienced by passengers when queuing to pass through security whilst also reducing the degree to which the security lanes are open thereby reducing staffing costs. The problem is multi-objective in nature as these objectives are mutually exclusive. The primary objective will be considered here as the reduction of the waiting times experienced by passengers measured as the absolute maximum waiting time experienced by a single passenger during the given time period as this will reflect the greatest level of dissatisfaction of a passenger:

$$\text{minimise} \qquad Objective_1 = \max_{i=1}^{i \leq m}(W_i) \tag{1}$$

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 at security within the given time period.

The secondary objective is to minimise the amount of time that the security lanes are open during the given time period:

$$\text{minimise} \qquad Objective_2 = \sum_{i=1}^{i \leq n} S_i \tag{2}$$

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.

As this problem is multi-objective in nature several methods could be used. One method is to find a Pareto front of non-dominated solutions. Alternatively a weighted sum approach can be used whereby a weight is assigned to the importance of each objective resulting in a single optimal schedule. For the purposes of security lane schedule optimisation, a simpler version of the weighted sum approach will be used. The key objective is the minimisation of passenger delay at times of high demand. Reducing the staffing costs is only a secondary consideration. Essentially, when comparing two potential solutions, the fitter of the pair will be considered the solution with the lower maximum waiting time experienced by passengers. However, if the two possible solutions have the same maximum waiting times experienced by passengers, then the secondary objective of minimising the time which security lanes are open will be minimised. Hence, of the two solutions, the one with the reduced number of security lane opening hours will be considered the better of the two.

To solve this optimisation problem a Genetic Algorithm (GA) [2] approach will be used. A GA uses the principles of evolution to optimise a given problem using a population of potential solutions and selection (survival of the fittest) with crossover and mutation of the parent genomes to form new descendant potential solutions. The representation of a security lane opening schedule will consist of a set of shifts defined simply by a start and end time. Moreover, there are constraints associated with these shifts in that it is assumed that a shift will be performed by a group of individuals such that the shift must not be less than two hours in length and no more than four hours in length. A shift will be automatically allocated at its start time to the next available security lane that is not already open. If no security lane can be found that will accommodate the given shift (all available security lanes are currently open), then this shift be considered invalid and ignored.

TABLE I

An example evolutionary algorithm representation of a schedule of shifts for security lane openings.

Shift	Start Time	End Time	Shift	Start Time	End Time
1	03:30	05:30	10	13:30	16:30
2	07:00	09:00	11	18:30	19:30
3	09:30	11:30	12	04:00	08:00
4	11:30	13:30	13	08:00	11:30
5	13:30	16:30	14	15:30	17:30
6	17:30	19:30	15	04:30	08:00
7	04:00	07:00	16	05:00	08:30
8	07:00	09:00	17	05:00	08:30
9	11:30	13:30			

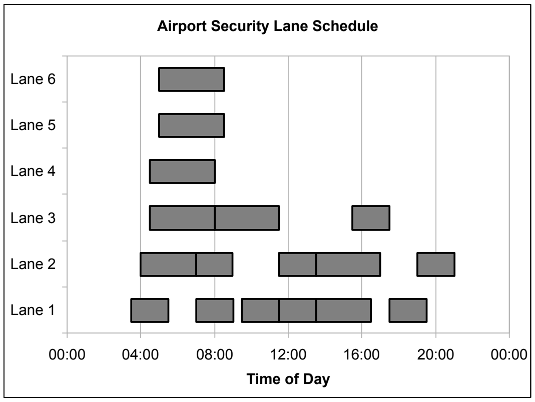


Fig. 2. An example airport security lane shift schedule generated from a candidate solution as described by Table I.

As a set of shifts can differ in number, a variable genome approach will be used [3] and can be considered similar to Linear Genetic Programming (LGP) [4] whereby the genome consists of a variable list of instructions to execute. The genome in this case consists of a variable list of shifts to be performed defined by a start time and end time. Two point crossover is performed by taking a subset of shifts from each parent individual and copying them across replacing the current subset. These subsets do not need to consist of the same number of shifts hence leading to variability in the genome size. Two forms of mutation are used. The first is similar to crossover with a subset of shifts from a solution being replaced by a subset of randomly generated shifts which can be of a differing number. The second mutation operator simply performs a bitwise random mutation of a shift start or end time with a given probability. An example of a candidate solution representing a security lane shift schedule is shown in Table I with a matching gant chart shown in Figure 2.

As previously discussed, the fitness of a candidate solution will consist of two measures. Firstly, the maximum waiting time experienced by a passenger over the given time period. The second fitness measure is the total time of the valid shifts performed during the given time period. Taking two candidate solutions, if one has a lower maximum waiting time than the other then this is considered the fitter. However, if they have the same maximum waiting time then the solution with the lesser degree of shift time considered the fitter.

To measure the primary fitness of candidate solutions a simulation based approach is used. The genome of a candidate solution is translated into a schedule of valid shifts that operate security lanes during the given time period. A simulation is then conducted whereby passengers arrive at the security lane area as specified by the expected forecast for a set of time steps over the given time period. At each time step passengers arrive at the security lanes and are placed at the end of a queue. Passengers are then taken from the front of the queue in a first in first out (FIFO) manner by each open security lane and processed through security until the end of the time step is reached. This process is repeated at each time step throughout the given time period. Security processing is defined as taking between fifteen and twenty one seconds. The time to process a passenger is randomly generated within this range using a uniform distribution. The waiting times of each passenger through the queue are tracked and the maximum time experienced by a single passenger is reported as the primary fitness. A number of simulations are conducted to enable variations in passenger processing times to be accounted for and the maximum waiting time experienced by an individual passenger averaged over these simulations.

A. Evolutionary Optimisation Results

To test the capability of this evolutionary approach to minimising both the maximum passenger waiting time experienced and the time for which security lanes are open, the four exemplar days of passenger numbers will be used as described in Figure 1. For each example, the forecast predicts the number of passengers arriving at security at each time step during the given time period. In this case the time period is a 24 hour period and each time step is defined as a five minute interval. Ten separate simulations are used to account for variations in passenger processing times and the effect on the maximum passenger waiting times. The parameters used for the evolutionary algorithm approach are shown in Table II. The algorithms are implemented in C++ and are executed using an Intel i7 processor using eight parallel threads of execution to fully exploit the processor.

TABLE II

Parameters used throughout unless otherwise stated.

Population Size	100
Max Generations	2,000
Tournament Size	7
Crossover Probability	0.9
Mutation Probability	0.1
Fitness Measure	Minimisation of maximum passenger waiting time and in addition the total shift time

Results for each problem instance and with a varying number of maximum security lanes available are shown in Table III. Twenty five separate experimental runs of the evolutionary algorithm are performed with ten simulations and the results averaged over these experiments. The two fitness measures are shown with a third observation for reference

TABLE III

The average maximum waiting times and average overall waiting times and the total amount of time security lanes are open in hours for each problem forecast and for a range of maximum available lanes. Results are averaged over the 25 evolved schedules and 10 simulations with varying passenger processing times.

Problem	Max. Lanes	Max. Waiting Time Experienced	Av. Waiting Time Experienced	Total Lane Opening
F_PAXflow_ 2425	4	0.67 ± 0.00	0.26 ± 0.02	38.14 ± 1.07
	5	0.24 ± 0.00	0.08 ± 0.00	42.04 ± 2.06
	6	0.09 ± 0.00	0.04 ± 0.00	46.74 ± 2.48
	7	0.08 ± 0.00	0.03 ± 0.00	60.06 ± 2.40
	8	0.07 ± 0.00	0.03 ± 0.00	64.08 ± 3.17
F_PAXflow_ 2428	4	0.74 ± 0.00	0.35 ± 0.03	41.02 ± 0.78
	5	0.26 ± 0.00	0.09 ± 0.01	44.48 ± 1.68
	6	0.12 ± 0.00	0.04 ± 0.00	51.68 ± 2.34
	7	0.08 ± 0.00	0.03 ± 0.00	64.10 ± 2.70
	8	0.07 ± 0.00	0.03 ± 0.00	69.88 ± 2.47
F_PAXflow_ 2501	4	0.71 ± 0.00	0.36 ± 0.03	45.12 ± 1.00
	5	0.21 ± 0.00	0.08 ± 0.01	48.82 ± 1.81
	6	0.09 ± 0.00	0.03 ± 0.00	65.22 ± 1.70
	7	0.08 ± 0.00	0.03 ± 0.00	73.34 ± 3.23
	8	0.07 ± 0.00	0.03 ± 0.00	83.34 ± 3.07
F_PAXflow_ 21113	4	0.07 ± 0.00	0.03 ± 0.00	35.20 ± 1.45
	5	0.06 ± 0.00	0.02 ± 0.00	44.64 ± 1.78
	6	0.05 ± 0.00	0.02 ± 0.00	53.56 ± 2.21
	7	0.04 ± 0.00	0.02 ± 0.00	60.08 ± 2.83
	8	0.04 ± 0.00	0.02 ± 0.00	66.82 ± 4.66

purposes, the average waiting time experienced by all the passengers over the given time period. The first observation that can be made is that as the number of available security lanes increases, the maximum waiting time experienced by a passenger reduces significantly. Indeed, by simply increasing the maximum number of lanes from four to five results in a reduction in the maximum waiting time of up to 71%. However, as the maximum number of available security lanes increases the number of hours that the security lanes are open increases likewise. Thus, the evolutionary approach exploits the availability of extra security lanes to reduce the maximum waiting times experienced even though this increases staffing requirements. Note that the variation in the maximum passenger waiting times is zero indicating that in every case the minimisation of the passenger maximum waiting time has been achieved. However, there is variation in the total amount of time that the security lanes are open.

Also of note is the average waiting times experienced by all the passengers over the twenty four hour period. Aside from when there are only four available security lanes, the average passenger waiting times at security are merely a matter of being five minutes or less.

IV. Dynamic Re-Optimisation of Security Lanes

The optimal security lane opening time schedules generated by an evolutionary approach in the previous section use a forecast of the expected passenger numbers. However, in practice it is unlikely that actual passenger arrivals at the security lanes will mirror this forecast, there will be some differences. Moreover, a number of factors can have a significant impact on how passengers arrive at the security lane area. For instance, predicted bad weather could result in passengers arriving at the airport early to avoid problems or later by being delayed. Alternatively, a traffic accident near the airport could result in passengers arriving late. A further scenario that could impact passenger arrival numbers is that there could be a block booking of passengers such as a school trip whereby many passengers will arrive at security at the same time.

Figure 3 shows the forecast passenger numbers arriving during a 24h period at the security lane area alongside a set of actual passenger arrivals for a single 24h period for each of the four exemplar days. From this it can be broadly seen that the actual passenger volumes follow the forecast but with a large degree of variation. Indeed, it can be observed that the forecast numbers are a much smoother prediction than the actual reality of passengers arriving at security.

The optimal security lane opening schedules evolved in the previous section can be measured as to their performance against these actual passenger arrivals by running them through the airport security lane simulation. These results are shown in Table IV. From these it can be observed that there has been a considerable increase in the maximum waiting time experienced by a passenger for all problem instances and ranges of maximum security lanes available compared to using the forecast passenger numbers. The most significant of these occurs for the F_PAXflow_2425 problem instance with as much as a 32 fold increase although this arises from a very low maximum wait from the forecast passenger flows. However, it is clear that with a greater availability of security lanes there is a greater capacity to deal with variations in passenger flow from expectations although the evolved schedules for greater numbers of lanes did have a greater number of hours of open security lanes which would account for this.

The reasons for this increase become clearer when analysing the maximum wait experienced by passengers in the current queue at five minute intervals across the given time period whereby it was observed that in some cases passengers have arrived early or late such that there are no security lanes open incurring significant waiting times at security. This effect is most obvious with the F_PAXflow_2425 problem whereby passengers arrive earlier than expected at the beginning of the time period before any lanes are open. Looking at the actual passenger flow numbers from Figure 3 it can be observed that these delays are caused by a very small number of passengers arriving earlier than expected. The same can be seen with a couple of passengers arriving at security later than expected hence security lanes have all closed for the time period.

A. Evolutionary Dynamic Re-Optimisation

It has been demonstrated that there has been a considerable increase in passenger waiting times for schedules evolved using forecast data and tested using actual passenger arrival data. A primary reason for this is that the evolved schedules are essentially over-fitting the forecast passenger arrivals data.

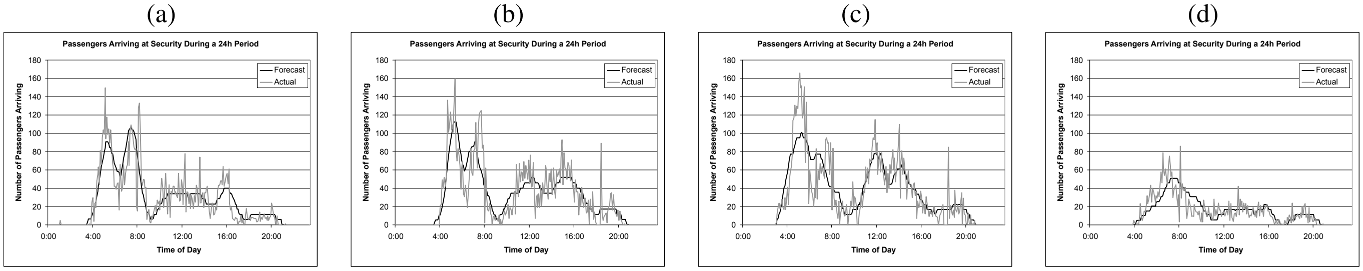


Fig. 3. The forecast arrivals of passengers at security and a set of actual passenger arrivals during a given day for the exemplar days labelled (a) F_PAXflow_2425, (b) F_PAXflow_2428, (c) F_PAXflow_2501 and (d) F_PAXflow_21113 respectively.

TABLE IV

THE AVERAGE MAXIMUM WAITING TIMES IN HOURS USING THE FORECAST PASSENGER VOLUMES AND THE ACTUAL PASSENGER VOLUMES AND THE MULTIPLE OF THE INCREASE IN MAXIMUM WAITING TIMES OVER A RANGE OF MAXIMUM AVAILABLE LANES. RESULTS ARE AVERAGED OVER THE 25 EVOLVED SCHEDULES AND 10 SIMULATIONS WITH VARYING PASSENGER PROCESSING TIMES.

Problem	Max. Lanes	Max. Forecast Waiting Time Experienced	Max. Actual Waiting Time Experienced	Multiple Increase
F_PAXflow_ 2425	4	0.67 ± 0.00	2.70 ± 0.26	4.04x
	5	0.24 ± 0.00	2.39 ± 0.13	9.94x
	6	0.09 ± 0.00	2.39 ± 0.15	25.29x
	7	0.08 ± 0.00	2.29 ± 0.45	28.03x
	8	0.07 ± 0.00	2.33 ± 0.32	32.12x
F_PAXflow_ 2428	4	0.74 ± 0.00	2.27 ± 1.17	3.07x
	5	0.26 ± 0.00	1.04 ± 0.21	4.03x
	6	0.12 ± 0.00	0.45 ± 0.02	3.86x
	7	0.08 ± 0.00	0.26 ± 0.00	3.03x
	8	0.07 ± 0.00	0.25 ± 0.01	3.39x
F_PAXflow_ 2501	4	0.71 ± 0.00	3.20 ± 0.86	4.48x
	5	0.21 ± 0.00	0.78 ± 0.08	3.80x
	6	0.09 ± 0.00	0.43 ± 0.01	4.88x
	7	0.08 ± 0.00	0.23 ± 0.00	2.91x
	8	0.07 ± 0.00	0.20 ± 0.02	2.98x
F_PAXflow_ 21113	4	0.07 ± 0.00	0.14 ± 0.00	1.97x
	5	0.06 ± 0.00	0.12 ± 0.01	2.18x
	6	0.05 ± 0.00	0.12 ± 0.02	2.49x
	7	0.04 ± 0.00	0.12 ± 0.02	2.69x
	8	0.04 ± 0.00	0.10 ± 0.01	2.68x

In both reducing the maximum passenger waiting time and the number of hours the security lanes are open there is no flexibility within the schedules to be able to deal with any deviation from forecast passenger arrivals at security.

A methodology that could be used to deal with passenger arrivals deviating from the expected forecast is to dynamically re-optimise the optimised schedules as time progresses. Indeed, managers frequently alter schedules when demand differs from expectations and this is known as online or real-time shift updating [5]. Thus, when passengers begin to arrive earlier than expected and thereby increasing the size of the queue, opening extra security lanes or ensuring that currently open security lanes remain open for longer will obviously have a positive effect in reducing the size of the queue and thereby passenger waiting times. However, there are limitations to achieving this. Clearly, security lanes cannot simply be opened in an ad-hoc manner, staff need to be on-site to be able to do this. Equally, security lanes cannot be kept open if staff have completed too many hours or finish work for other reasons.

Therefore, it is envisaged that whilst it is possible to modify a schedule dynamically, this can only be performed to a limited degree. In this case it is considered that a schedule can be dynamically modified by bringing forward or pushing back the start time of a security lane opening by up to an hour. Equally, the closing time of a security lane can be brought forward or pushed back by up to an hour. Additional instances of security lanes being opened cannot be added. Also, as with the generation of optimal static schedules, security lane shifts must not exceed four hours or be less than two hours in length.

To solve the dynamic re-optimisation of schedules once again an evolutionary algorithm is considered as previously. However, there are some differences from the previous use of an evolutionary algorithm to generate the optimal schedule. The primary difference being that the number of shifts in the schedule are fixed, no shifts can be added or removed. Therefore, the only changes that can occur are the start and end times of each shift. Moreover, these can only be within plus or minus one hour and shifts must be within a minimum of two hours and a maximum of four hours. Subsequently, the evolutionary algorithm in this case more closely resembles a GA. Each individual is of a fixed length and as such two point crossover can be used such that the sizes of the two offspring are the same as the parents. Mutation also differs from previously in that the start and end times of shifts are merely modified as a result of a mutation probability and only within plus or minus one hour. Fitness remains the same as previously, the minmisation of the maximum waiting time experienced by a passenger during the time period. However, with dynamic re-optimisation, clearly events that are in the past cannot be changed thus the dynamic re-optimisation only considers events in the future. Therefore, the fitness measure of the average maximum passenger waiting time is only considered from the current time T in the simulation whereby the re-optimisation occurs and the end of the given time period.

To implement a dynamic re-optimisation approach, the optimised schedules from the first phase detailed in Section III will be run through the security lane simulation against actual recorded days passenger flow numbers. Moreover, at each hour during the simulation, the schedule will be re-optimised using

TABLE V

THE AVERAGE MAXIMUM WAITING TIMES AND OVERALL AVERAGE WAITING TIMES IN HOURS USING THE ACTUAL PASSENGER VOLUMES OVER A RANGE OF MAXIMUM AVAILABLE LANES FOR THE OPTIMAL STATIC SCHEDULES AND THE DYNAMICALLY RE-OPTIMISED SCHEDULES. RESULTS ARE AVERAGED OVER THE 25 EVOLVED SCHEDULES AND 10 SIMULATIONS WITH VARYING PASSENGER PROCESSING TIMES.

Problem	Max. Lanes	Maximum Wait Experienced Over 24h Period (in hours)		Average Max. Wait Over Five Minute Intervals (in hours)	
		Static Schedule	Dynamic Schedule	Static Schedule	Dynamic Schedule
F_PAXflow_ 2425	4	2.70 ± 0.26	1.80 ± 0.32	0.46 ± 0.05	0.26 ± 0.06
	5	2.39 ± 0.13	1.90 ± 0.29	0.29 ± 0.05	0.20 ± 0.05
	6	2.39 ± 0.15	1.95 ± 0.26	0.26 ± 0.05	0.18 ± 0.04
	7	2.29 ± 0.45	1.77 ± 0.45	0.21 ± 0.06	0.14 ± 0.05
	8	2.33 ± 0.32	1.85 ± 0.35	0.22 ± 0.06	0.14 ± 0.05
F_PAXflow_ 2428	4	2.27 ± 1.17	0.80 ± 0.09	0.55 ± 0.17	0.22 ± 0.04
	5	1.04 ± 0.21	0.69 ± 0.13	0.23 ± 0.06	0.13 ± 0.02
	6	0.45 ± 0.02	0.43 ± 0.05	0.10 ± 0.01	0.08 ± 0.01
	7	0.26 ± 0.00	0.23 ± 0.03	0.05 ± 0.00	0.04 ± 0.00
	8	0.25 ± 0.01	0.22 ± 0.04	0.04 ± 0.00	0.04 ± 0.00
F_PAXflow_ 2501	4	3.20 ± 0.86	1.33 ± 0.44	0.96 ± 0.18	0.34 ± 0.14
	5	0.78 ± 0.08	0.73 ± 0.01	0.23 ± 0.05	0.13 ± 0.01
	6	0.43 ± 0.01	0.43 ± 0.01	0.06 ± 0.00	0.06 ± 0.00
	7	0.23 ± 0.00	0.23 ± 0.00	0.05 ± 0.00	0.04 ± 0.00
	8	0.20 ± 0.02	0.17 ± 0.02	0.04 ± 0.00	0.03 ± 0.00
F_PAXflow_ 21113	4	0.14 ± 0.00	0.14 ± 0.00	0.03 ± 0.00	0.03 ± 0.00
	5	0.12 ± 0.01	0.12 ± 0.01	0.03 ± 0.00	0.02 ± 0.00
	6	0.12 ± 0.02	0.12 ± 0.02	0.02 ± 0.00	0.02 ± 0.00
	7	0.12 ± 0.02	0.12 ± 0.02	0.02 ± 0.00	0.02 ± 0.00
	8	0.10 ± 0.01	0.10 ± 0.01	0.02 ± 0.00	0.02 ± 0.00

the evolutionary algorithm as detailed. It should be stated that at each re-optimisation phase the original optimised schedule will be used as the template to seed the initial population with variations of this original optimal schedule using mutations of the start and end times of shifts within plus or minus one hour. Only fifty generations of evolution are required due to the rigidity of schedules. The re-optimisation process uses the forecast passenger flow as previously from the re-optimisation point in time T. Additionally, the current state of the queue is used from the actual events. Thus is the queue has a large number of passenger within it then the re-optimisation process will likely open more desks than planned to reduce passenger waiting time.

B. Dynamic Re-optimisation Experimental Results

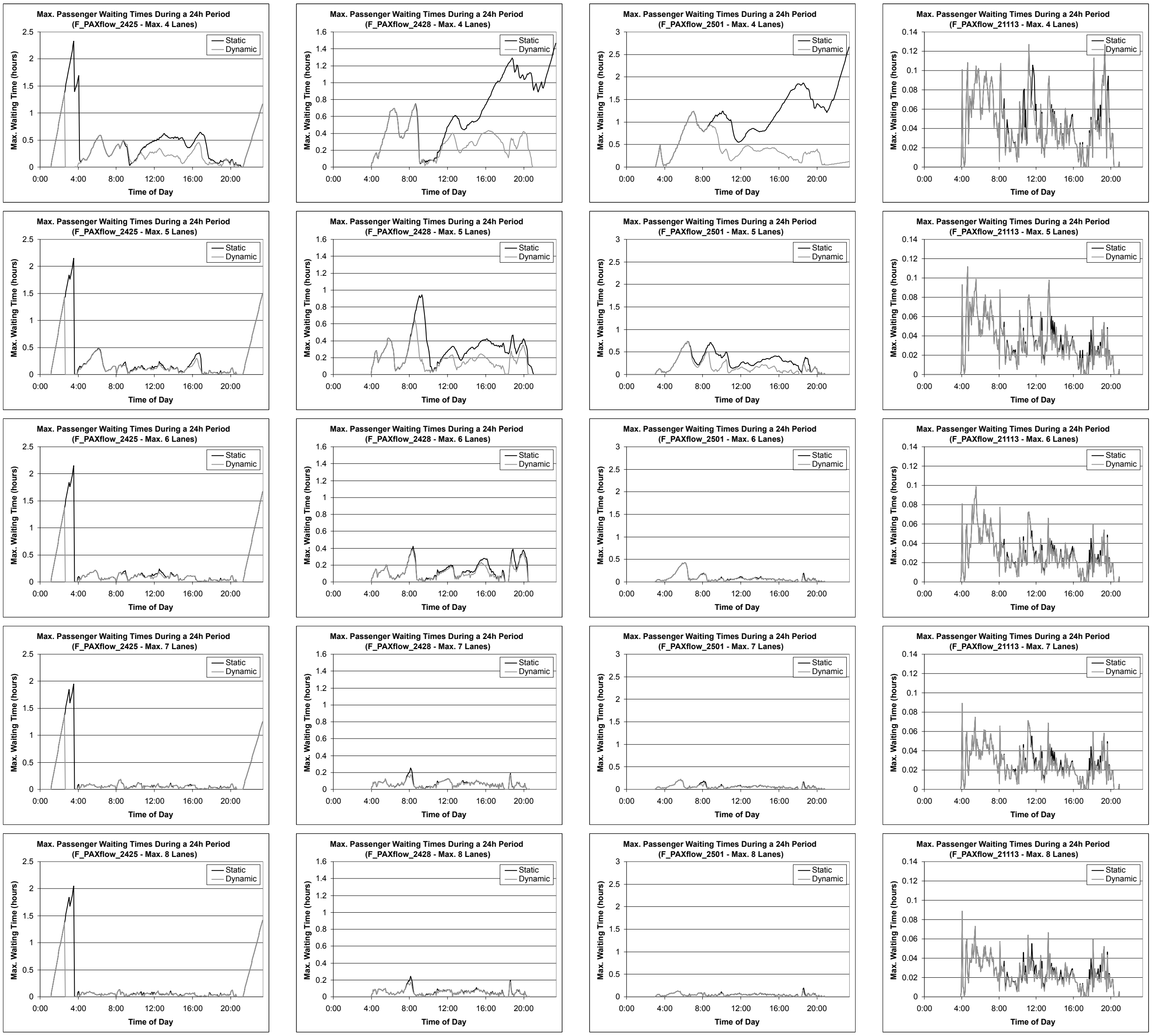
To test the effectiveness of a dynamic re-optimisation process using an evolutionary algorithm, each of the 25 evolved schedules for each problem from Section III will be run though the re-optimisation process using 10 simulations with random passenger processing times. The results will be averaged over the 25 schedules and the 10 simulations. The results in terms of the maximum waiting time experienced by a passenger over the given time period and the average maximum wait experienced at each five minute interval across the time period are shown in Table V.

From these results it can be observed that there is a reduction in the maximum waiting time experienced by a passenger when using a dynamic approach vs. a static approach for most problems and ranges lane availability. Indeed, the reduction in maximum wait is significantly reduced when only a few security lanes are available demonstrating how easily fewer security lanes can be overwhelmed by differences in passenger flows. A reason for this is that the maximum passenger delay cannot likely be reduced even with a dynamic approach hence there is no change. Additionally, the low passenger volumes for this problem instance ensure that passenger waiting times at security are relatively low in any case. However, it should be noted that when considering the average maximum wait across the time period there has still been a small reduction.

Visualisations of the differences between the optimised static schedules and dynamic re-optimised schedules in terms of the maximum passenger wait at five minute intervals throughout the exemplar 24h periods are shown in Figure 4. From these it can be seen that the dynamic approach shows significant improvement when there are only four or five available security lanes. Beyond that performance is very similar. It can also be observed that for problem F_PAXflow_2425 the results are highly skewed by the early and late arrival of passengers at either end of the 24h period. Moreover, even the dynamic approach cannot mitigate this issue to a significant degree although passenger delays are reduced.

V. RELATED WORK

Within the field of airport security lane optimisation there has been limited work concerned with the optimal opening of security lanes to reduce passenger waiting times. However, there has been work associated with staff scheduling for security lanes. Soukour *et al.* [6] use a memetic algorithm merged with an evolutionary algorithm to assign staff using an objective function based on over and undertime and staff satisfaction. Some work has also been performed with regards security screening. McLay *et al.* [7] investigated assigning passengers to varying levels of security screening with the goal of maximising security coverage subject to capacity and



Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2425 - Max. 4 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2428 - Max. 4 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2501 - Max. 4 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_21113 - Max. 4 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2425 - Max. 5 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2428 - Max. 5 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2501 - Max. 5 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_21113 - Max. 5 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2425 - Max. 6 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2428 - Max. 6 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2501 - Max. 6 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_21113 - Max. 6 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2425 - Max. 7 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2428 - Max. 7 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2501 - Max. 7 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_21113 - Max. 7 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2425 - Max. 8 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2428 - Max. 8 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_2501 - Max. 8 Lanes)

Max. Passenger Waiting Times During a 24h Period (F_PAXflow_21113 - Max. 8 Lanes)

Fig. 4. The average maximum waiting time experienced by a passenger using actual passenger flow data for the static optimal schedule and a dynamically re-optimisation schedule for each of the four problem instances with results averaged over the 25 evolved schedules and 10 simulations with varying passenger processing times.

assignment constraints. This optimisation problem was solved using a linear programming approach.

However, the security lane optimisation problem shares many characteristics with the optimisation of airport check-in desks. Indeed, early work by Wang and Chun [8] used a GA for optimal counter assignment for check-in desks. Chun and Mak [9] used simulation and search heuristics to determine the optimal check-in desk allocation that reduces the time desks are open and ensures queue lengths are within an acceptable tolerance for Hong Kong Airport. Bruno and Genovese [10] proposed a number of optimisation models for the check-in service balancing operation costs with passenger waiting times. Simulations based upon Naples airport were used with ten problems considered which used 50 flights. Solutions were found using FICO XPress Optimization Suite 7 although no advantages were reported. Parlar *et al.*[11] investigated the allocation of exclusive use check-in counters. A static policy is recommended as it is contended that a dynamic policy suffers from the curse of dimensionality. The overall objective is to minimise passenger waiting times, counter operational costs and passenger delays. The static policy only consists of a few decision variables and hence is an easy to optimise system. Araujo and Repolho [12] present a new methodology to optimise the check-in desk allocation problem maintaining a service level whilst reducing operational costs. Three phases are used whereby the first optimises the number of desks, the second uses simulation to test the service level and the third uses an optimisation model to solve an adjacent desk constraint. The optimisation for phase one is based upon the Bruno and Genovese paper [10] with modifications for minimising operational costs whilst maintaining service levels at each time step. Phase 2 uses the ARENA software tool to conduct airport simulations. Integer programming is used to solve both a common and dedicated desk problem. Mota [13] uses an evolutionary algorithm and a simulation approach to establish the allocation and opening times of check-in desks to reduce passenger waiting times.

Dynamic optimisation of check-in desks has been investigated by Parlar and Sharafali [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 delay costs. A stochastic dynamic programming model is then used to minimise this cost for a single flight scenario. However, the dynamic aspect is based on optimising for a set of time-windows rather than unforseen events. Hsu *et al.* [15] investigated the dynamic allocation of check-in facilities and the dynamic allocation of passengers to desks. The problem is modelled as a Sequential Stochastic Assignment Problem (SSAP) and solved using binary integer programming. Results from data at Taoyuan International Airport, Taiwan showed that a dynamic allocation approach can increase desk utilisation and reduce waiting times for passengers. Nandhini *et al.* [16] investigated the dynamic optimisation of check-in desks to minimise the conflicting objectives of resource allocation and passenger waiting times. A genetic algorithm was used to create optimised schedules for the check-in desks. However, the dynamic aspect is not so much a re-optimisation but a methodology of combining desks to cover multiple flights.

VI. Conclusion

Reducing airport costs whilst maintaining passenger satisfaction levels is becoming an increasingly important issue with a key area being the airport security checking area consisting of a number of lanes. The work presented in this paper successfully used an evolutionary algorithm to design a security lane schedule that minimised the maximum waiting time a passenger may experience when queuing to pass through security whilst minimising the amount of time that security lanes are open. Using forecast passenger flow data, security lanes opened during periods of predicted high passenger flow and quickly closed when forecast demand dropped.

However, although it was successfully demonstrated that an evolutionary algorithm could effectively reduce the degree of delay a passenger experiences at security, deviations from passenger flow forecasts had considerable effect on these optimised schedules. Since the schedules had the number of security lanes optimised to the forecast demand there is no spare capacity within the schedule to cope with deviations in passenger arrivals, the schedules over-fit the forecast passenger flow. To handle this issue, a dynamic re-optimisation methodology was introduced whereby an optimal schedule can be modified throughout a time period at given intervals to satisfy unexpected passenger flows. Again, an evolutionary algorithm was used to perform these dynamic modifications to an optimised schedule with significant reductions in passenger delays observed with minimal schedule changes.

Further work needs to consider an alternative robust scheduling technique whereby a range of actual passenger flows are considered rather than a forecast. A further aspect to consider is that some evolved security lane schedules may be difficult to dynamically re-optimise possibly due to constraints on working hours. Therefore, it may be useful to test schedules from the first optimisation process as to how well they react to dynamic re-optimisation. Finally, a unified model with staff scheduling could be considered.

Acknowledgment

This work was supported by De Montfort University through a Higher Education Innovation Fund (HEIF) grant.

References

[1] K. Gkritza, D. Niemeier, and F. Mannering, “Airport security screening and changing passenger satisfaction: An exploratory assessment,” *Journal of Air Transport Management*, vol. 12, no. 5, pp. 213–219, 2006.

[2] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

[3] I. Harvey, “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*, 1992, pp. 346–354.

[4] M. Brameier and W. Banzhaf, “A comparison of linear genetic programming and neural networks in medical data mining,” *Evolutionary Computation, IEEE Transactions on*, vol. 5, no. 1, pp. 17–26, Feb 2001.

[5] D. Hur, V. A. Mabert, and K. M. Bretthauer, “Real-time work schedule adjustment decisions: An investigation and evaluation,” *Production and Operations Management*, vol. 13, no. 4, pp. 322–339, 2004.

[6] A. A. Soukour, L. Devendeville, C. Lucet, and A. Moukrim, “A memetic algorithm for staff scheduling problem in airport security service,” *Expert Systems with Applications*, vol. 40, no. 18, pp. 7504–7512, 2013.

[7] L. A. McLay, S. H. Jacobson, and J. E. Kobza, “Integer programming models and analysis for a multilevel passenger screening problem,” *IIE Transactions*, vol. 39, no. 1, pp. 73–81, 2007.

[8] B. Wang and H. Chun, “Check-in counter allocation using genetic algorithm,” in *Proc. Of the 7th Conf. on Artificial Intelligence and Expert Systems Applications*, 1995.

[9] H. W. Chun and R. W. T. Mak, “Intelligent resource simulation for an airport check-in counter allocation system,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 29, no. 3, pp. 325–335, 1999.

[10] G. Bruno and A. Genovese, “A mathematical model for the optimization of the airport check-in service problem,” *Electronic Notes in Discrete Mathematics*, vol. 36, pp. 703–710, 2010.

[11] M. Parlar, B. Rodrigues, and M. Sharafali, “On the allocation of exclusive-use counters for airport check-in queues: static vs. dynamic policies,” *Opsearch*, vol. 50, no. 3, pp. 433–453, 2013.

[12] G. E. Araujo and H. M. Repolho, “Optimizing the airport check-in counter allocation problem,” *Journal of Transport Literature*, vol. 9, no. 4, pp. 15–19, 2015.

[13] M. A. M. Mota and C. Zuniga, “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.

[14] M. Parlar and M. Sharafali, “Dynamic allocation of airline check-in counters: a queueing optimization approach,” *Management Science*, vol. 54, no. 8, pp. 1410–1424, 2008.

[15] C.-I. Hsu, C.-C. Chao, and K.-Y. Shih, “Dynamic allocation of check-in facilities and dynamic assignment of passengers at air terminals,” *Computers & Industrial Engineering*, vol. 63, no. 2, pp. 410–417, 2012.

[16] M. Nandhini, K. Palanivel, and S. Oruganti, “Optimization of airport check-in service scheduling,” 2012.