Intelligent Resource Simulation for an Airport Check-In Counter Allocation System

Hon Wai Chun, *Member, IEEE*, and Raymond Wai Tak Mak

***Abstract*— The objective of resource allocation systems is to assign valuable corporate resources to meet business demands. However, in applications involving stochastic events, demands cannot be determined explicitly beforehand. For those applications, computer simulation is often used to predict resource demands. This paper describes research in developing a knowledge-based simulation system to predict resource requirements at an international airport. Our intelligent resource simulation system (IRSS) determines how many check-in counters should be allocated to each departure flight while providing passengers with sufficient quality of service. This predicted resource requirement is then used by a constraint-based resource allocation system to allocate the actual check-in counters. Because IRSS considers many more factors than a human can, the resulting allocation schedule is more efficient. These factors include: 1) different service rates for different destinations, airlines, or handling agents; 2) different passenger arrival rates for different times of the day or days of the week; and 3) different requirements for different service levels, etc. Our experiments show that there are substantial resource savings by combining a resource allocation system with an intelligent resource simulation system.**

***Index Terms*—Constraint-based scheduling, discrete event simulation, knowledge-based simulation, resource allocation.**

I. Introduction

In this paper, we present the overall design of our intelligent resource simulation system (IRSS) used by a constraint-based airport check-in counter allocation system (CCAS) [2]–[4]. The allocation system itself consists of three major components as seen in Fig. 1: a database which forms input facility, the IRSS, and a constraint-based scheduling system. Airlines enter their seasonal check-in counter requests through the database forms input facility. These requests represent "desired" allocation and may not necessarily be the actual number of check-in counters allocated by the authorities. The authorities use the IRSS to predict the minimum number of check-in counters needed to satisfy the predefined service levels. The constraint-based scheduling system will then use both the simulated values and the requested values to perform check-in counter allocation. Our experiments show that there are substantial resource savings by combining a resource allocation system with an IRSS. This paper focuses on the design and implementation of the IRSS. Details of the constraint-based scheduling system itself can be found in [2]–[4].

Previously, the allocation of check-in counters was done manually without any simulation tools. The predicted resource demand was derived from experience and represented only an approximation. Because of the complexity involved, subtle variations in resource demand due to different time of day, flight destination, airline, and handling agent were ignored. Hence, the manually constructed check-in counter schedule might not utilize the airport facilities effectively. With the growing amount of traffic at most airports, a more efficient approach was needed to optimize resource utilization. The CCAS was, therefore, designed to incorporate an intelligent knowledge-based resource simulation module into more precisely predict resource requirements.

Our IRSS has been in use at the Hong Kong Kai Tak International Airport since 1995. At that time, this airport was the third busiest international airport in the world, handling approximately 150 000 flights per year and over 75 000 passengers daily. However, due to the physical limitations of the airport terminal building, there were only a limited number of check-in counters available (a total of about 300). Resources at the old airport were limited, since the new Chek Lap Kok International Airport was due to replace the Kai Tak Airport on July 6, 1997. In order to cope with increasing traffic at this aging airport, the authorities had to rely on advanced computer simulation and scheduling to manage their resources more efficiently. With the use of our IRSS, the airport was able to increase traffic to a total of over 165 000 flights and 28.3 passengers in 1997.

The check-in counters at Kai Tak Airport were managed centrally by the Civil Aviation Department (CAD). This Government authority made daily assignments of counters to airlines or their designated handling agents. This problem required an accurate estimation of the actual check-in counter requirement for each departing flight. Our simulation system was designed and developed to estimate this resource requirement.

IRSS is a knowledge-based simulation system and uses rules and heuristics to encode knowledge on how simulation parameters vary with different types of flights. Based on these rules, IRSS automatically constructs a simulation model and associated experimental parameters for each departure flight. Through an iterative simulation and intelligent search algorithm, the IRSS automatically determines the minimum number of counters needed. This minimum check-in counter profile is generated according to the number of passengers

Manuscript received January 6, 1997; revised November 13, 1998. This paper was supported in part by an RGC Competitive Earmarked Research Grant and a Strategic Research Grant from the City University of Hong Kong.
H. W. Chun is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: eehwchun@cityu.edu.hk).
R. W. T. Mak is with Federal Express Corporation, Hong Kong.
Publisher Item Identifier S 1094-6977(99)05326-2.

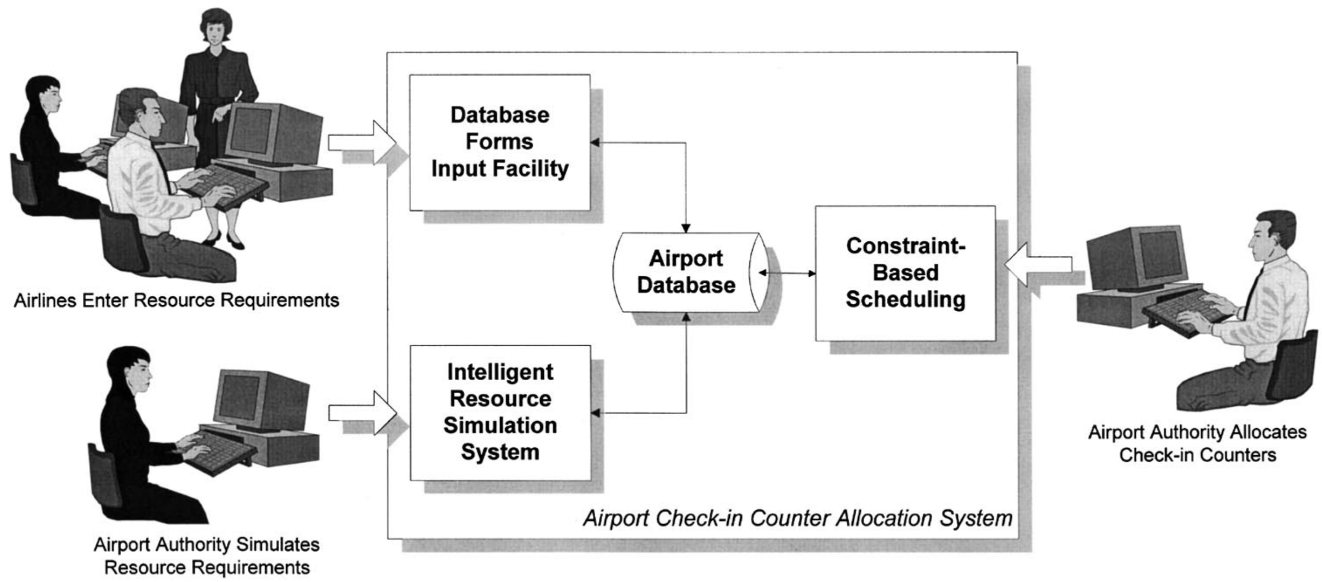

Fig. 1. Overall architecture of the airport CCAS.

boarding a flight, the passenger arrival profile, the average service rate, and the airport service commitment. Given a daily schedule, the IRSS computes the minimum profile for each flight and stores this information in a database. The constraint-based scheduling system then takes this information and produces a check-in counter allocation plan.

In this paper, we will discuss only the design and development of the knowledge-based IRSS. We will also present the results of simulating the check-in process for different categories of flights. The CCAS constraint-based scheduling system is a component that operates independently from the IRSS and is documented separately [2]–[4].

II. The Check-In Counter Problem

The check-in counter allocation problem is quite different from other traditional resource allocation problems, in that the amount of resources required is not known beforehand. The process of checking in passengers is stochastic, and the number of required check-in counters varies with time since the total number of passengers per flight is different. Other factors such as time of day, day of the week, and destination will all influence the amount of resources to be allocated. Due to this complexity, it is practically impossible for a human to predict resource requirements accurately on a daily basis.

Previously, check-in counter allocation was performed manually at the Kai Tak Airport based on prior experience and simple heuristics. The procedure starts by selecting airlines with the most check-in counter requests. Airlines with more flights and check-in counter requests are allocated first. Whenever a resource conflict is encountered, the affected airlines are notified, and conflicts are resolved through verbal negotiation. This procedure continues until all the check-in counter requests from all the airlines have been allocated. The major drawback with this procedure is that it relies on the resource requirements provided by the airline and the experience and skill of the human schedulers. However, airlines tend to request many more counters than actually needed to provide better service to their passengers. Unfortunately, the airport was already overcrowded and could not afford to underutilize their check-in counters. At the same time, it was impossible for a human to judge whether requests made by an airline were reasonable without an intelligent simulation tool.

Our knowledge-based IRSS is different from other conventional airport terminal simulation systems, in that previous systems were designed to simulate the overall situation in an airport terminal building after a schedule has been generated [9], [14], [15]. The simulation models developed by Sokkar and Nelson [14], and Weiss and Lacher [15] were for simulating air traffic flow at a major airport. These models are used to study flight delays and the effect of scheduling patterns on airport congestion. On the other hand, the simulation model developed by Haeme *et al.* [9] helps airlines evaluate their on-time arrival performance. The model described in [5] relates to the simulation of aircraft operations such as the unloading, refueling, and loading of aircraft. SABRE Decision Technologies [7] indicated that simulation was used in the Berlin Interim West Terminal to identify areas of the terminal in which bottlenecks and overcrowding occurred, and that it was used in Miami to ensure the adequacy of the facilities to accommodate the projected passenger flow and air traffic. In the past few years, not much research work has been documented on airport terminal simulation and little, if any, has been documented on simulation of check-in counter resources to support planning and scheduling. Although Drake and Smith [6] described a simulation system to support real-time planning, scheduling, and control, their problem domain is in manufacturing. Hopefully, this paper can provide some insights on how advanced computer simulation can help a scheduling system to optimize the utilization of airport check-in counter resources.

In Hong Kong, airlines submit seasonal and daily check-in counter requests in the form of a flight schedule plus the number of check-in counters requested and their opening and

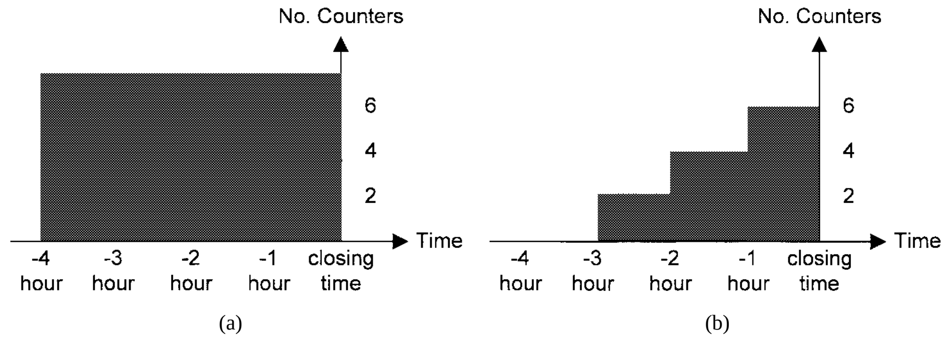

Fig. 2. (a) and (b). Check-in counter profile.

closing times for each flight or flight group. These requests can be represented as rectangular blocks, which we call check-in counter profiles. Fig. 2(a) is an example where seven check-in counters are requested for 4 h. In the ideal case, all that is needed during scheduling is to assign a bank of counters that will fit this profile. Unfortunately, since counters are limited, the authorities very often will not be able to meet all airline requests. The authorities needed to differentiate between what is the "real" need versus what is the "desired" need. The desired need is usually much larger than the real need, since airlines would like to provide better service to their customers by operating more counters for a longer period of time. For illustration purposes, the real need might be represented by a check-in counter profile of the shape displayed in Fig. 2(b). In this example, check-in counters are only needed 3 h before check-in closing time. In order to check in all the passengers within a given level of service quality, two counters should be opened during the first hour of operation, then four counters, and finally six counters in the last hour of operation. The main objective of our knowledge-based simulation system is to compute and determine this real need. Using results from the IRSS, the CCAS allocation system will then allocate counters within the range defined by the desired and real profiles depending upon the availability of check-in counters at that time.

III. Simulation Parameters

The knowledge-based IRSS automatically generates a simulation program from basic flight information such as the airline, destination, departure time, and total number of passengers. The following describes how the simulation parameters are automatically generated.

A. Simulation Duration

The simulation duration is the time range between which events are to be simulated. Two types of simulation duration are used by IRSS: simulation duration for a single flight and simulation duration for a common check-in group. At Kai Tak Airport, each check-in counter is normally assigned to service passengers from a single flight. However, an airline may decide to group some of their flights together to form a common check-in group. The check-in counters assigned to a common check-in group are shared among all the flights within that group.

For single flight simulation, the duration of the simulation is simply the duration of the check-in counter profile, which is determined by the type of flight and destination. This can range from 1.5 to 4 h depending upon whether the flight is short, local, or long. At our airport, the authorities impose a special rule of closing the counters 40 min before the scheduled departure time. Hence, our simulation duration also ends at counter closing time, which is 40 min before flight departure.

For a common check-in group, the duration of the simulation will depend upon the first and last flight within the common check-in group. The simulation duration will start with the check-in counter start time of the first flight and end with the check-in counter closing time of the last flight.

B. Number of Counters

The number of counters to be simulated will vary with time. The example in Fig. 2(b) shows that the number of counters varied from two to four and six, and that the changes take place on an hourly increment. In IRSS, the time of change can be more fine-tuned and can be in increments of 15 min. The number of counters to be simulated will depend on the simulation goals. The basic goal is that there should be enough counters to process all the passengers boarding this flight before counter closing time. Another goal, related to service quality, is the tolerable queue length and waiting time. To provide better service, the queue length and waiting time cannot be too long. On the other hand, since there are only a limited number of counters, we must restrict how many counters can be assigned. It is not easy to find a balance between these conflicting goals. The approach taken is first to compute a minimum check-in counter profile that will maintain a maximum queue length and waiting time during the time counters are open. During allocation, the CCAS will then allocate more counters if additional counters are available. In addition, the simulation goals can also be seasonal dependent. For example, during peak seasons or holidays when the demand on check-in counters is higher, the maximum allowed queue length and waiting time may also be defined to be longer than offpeak seasons. This will allow more flights to be processed at the expense of reduced quality of service.

1) Arrival Profile: The passenger arrival rates will depend on many factors such as the time of departure and the destination of the flight. For example, if the flight is scheduled

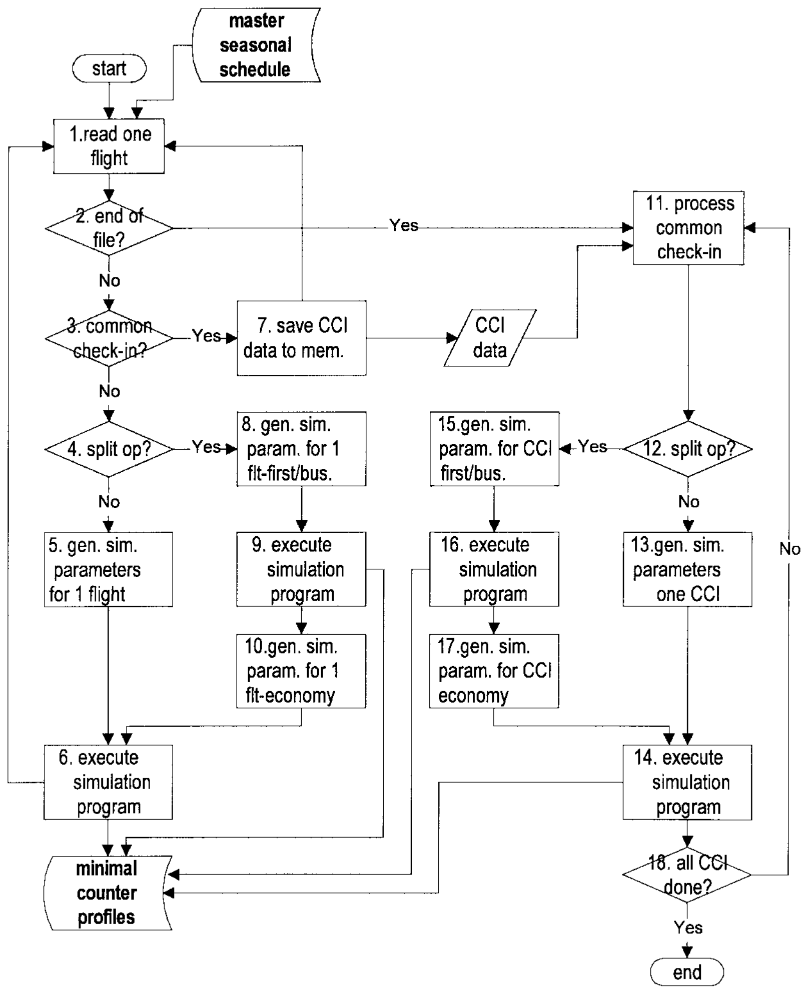

Fig. 3. Simulation methodology flowchart.

to depart early in the morning, passengers will usually arrive a bit later than the statistical average. The approach used by the IRSS is to define a set of statistical average arrival patterns. These arrival patterns are extracted from statistical data collected by the airport. A knowledge base of heuristics is defined in IRSS to adjust the average arrival patterns based on additional features of the current flight. A further complication to the simulation model is that the passenger arrival rate is a nonstationary distribution with more passengers arriving during the middle part of the check-in counter opening period. In our simulation model, we use a passenger turn-up profile to generate the passenger arrival rates using an exponential distribution with time-varying mean. More description of the use of the exponential distribution for the passenger arrival rate is presented in Section IV.

2) Check-In Processing Time: The check-in processing time or the service rate will also depend on many factors related to the flight. For example, flights to certain destinations may require more processing time because travellers may check-in more than the average amount of baggage. Certain countries may require a more stringent passport and visa check. The type of payment used for the departure tax will also affect the processing time. Credit card payments will take longer. These are also encoded as heuristics in our IRSS.

IV. Simulation Methodologies

Many real-world queuing systems such as those found in industries like banking, health care, manufacturing, telecommunications, and transportation have been modeled using discrete-event simulation, e.g., [1], [8], [10], and [11]. Our discrete-event simulation model for the check-in counter problem follows the steps defined in Law [12]. The check-in counter operation is modeled as a multiqueue $M/M/n$ system. The model parameters, such as the probability distribution for passenger arrival and check-in processing time, are defined from collected statistics and a set of heuristics. A simulation model is then developed to simulate either a single flight or a common check-in flight group. Several simulations are performed on the model, and the simulated statistics are then compared with the tolerable service level, i.e., the tolerable waiting time and queue length. The system iteratively generates a new simulation model until it finds the minimal counter profile for that flight.

Fig. 3 is a flowchart of the simulation algorithm used in IRSS. The master seasonal schedule contains all the airlines' flight schedules and check-in counter requests. Each flight in this master schedule is simulated, and a minimum profile is computed before CCAS performs seasonal long-term planning or daily check-in counter allocation.

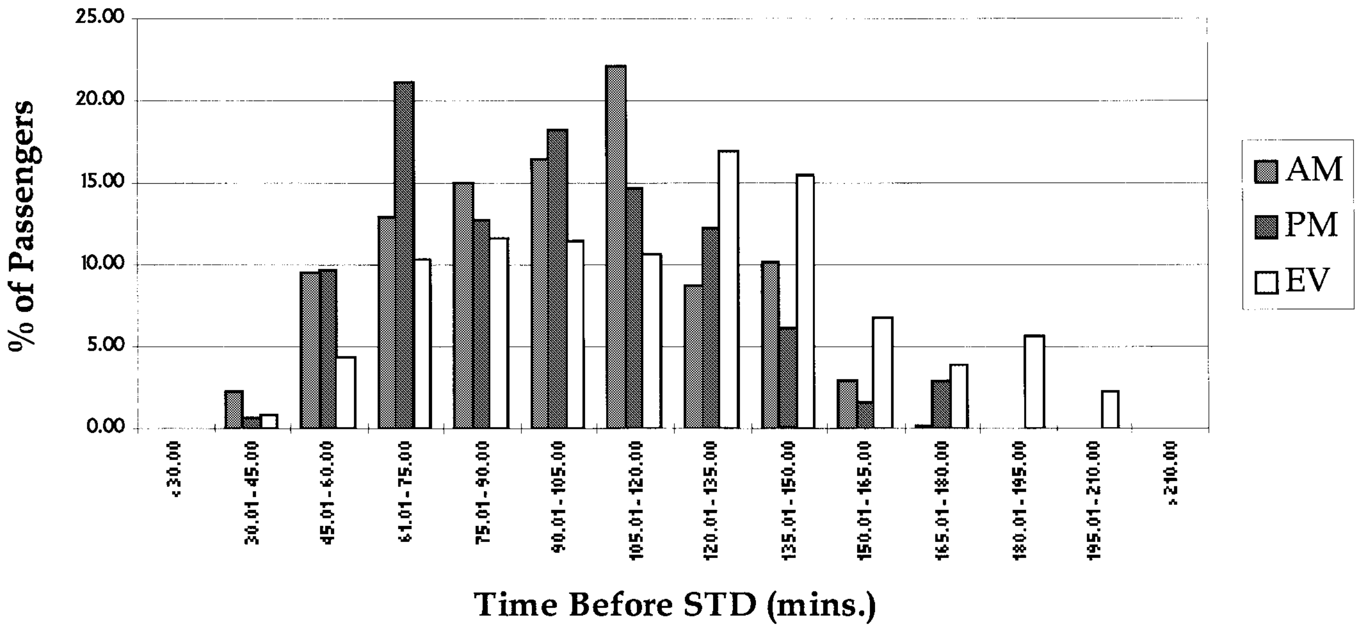

Fig. 4. Typical passenger arrival pattern for first class passengers.

The IRSS will first simulate all single flights before common check-in groups. For a single flight, the airline may opt for split operations, i.e., separate first/business class and economy class check-in counters. For split-operation single flights, IRSS will generate a set of simulation parameters and a model for each fare class. Iterative simulation will be performed twice and there will be two check-in counter profiles, i.e., one for first/business class and one for economy class. For regular single flights, only one series of iterative simulation will be performed. Common check-in groups are simulated last, since they cannot be processed until all the flights in the groups have been read into the IRSS. The system will then sort out the different common check-in flights and merge flights that belong to the same common check-in group. If the common check-in group does not require split operation, a single simulation model will be generated. Otherwise, two models will be generated to correspond with first/business class check-in and economy class check-in.

Both single flight and common check-in simulation parameters are generated from the rules stored in the knowledge base. The knowledge includes how check-in duration, arrival profile, and processing time for different flights may vary with the flight parameters. This information is used to determine what simulation parameter values are to be assigned for a particular flight. These values are obtained from previous statistics collected at the airport.

The search for the minimal profile will then proceed iteratively by executing the simulation model and collecting simulated results. The results are compared with the tolerable average waiting time and average queue length. By using search heuristics defined in the system, the counter profile is modified automatically based on the difference between the simulated results and the target result. A new simulation model is then created and executed. This process is performed iteratively until the minimal check-in counter profile is found.

A. Simulation Model

This section describes the components of the simulation model. There are three major components in the multiqueue model: arrival process, check-in mechanism, and queue discipline.

1) Arrival Process: The arrival process is modeled using passenger arrival statistics collected for different categories of flight. These statistics show that the interarrival times of passengers are IID random variables and follow nonstationary Poisson process, i.e., the arrival rate of passengers is a function of time. The charts in Figs. 4–6 show the typical passenger arrival pattern for morning, afternoon, and evening flights for each class of passengers respectively. To simulate the arrival of passengers for a particular flight, we use a passenger turn-up profile to describe the stochastic arrival pattern of each flight. The passenger turn-up profile records the portion of passengers out of the total average load of a flight that will turn-up in each time interval within the check-in duration. The number of time intervals depends on the check-in duration. For each type of haulage, time of day, and check-in duration, a turn-up profile is constructed based on the passenger arrival statistics.

Using this turn-up profile, the simulation model simulates the arrival of passengers as a nonstationary Poisson process for each flight. Exponential distribution, which is often used for modeling interarrival time [12], is selected to model the passenger arrival. Exponential distribution requires a scale parameter β to generate the random variates. This is determined by estimating the mean interarrival time of passengers in each time interval based on data stored in the passenger turn-up profile for each type of flight.

2) Check-In Mechanism: Similarly, the check-in mechanism is modeled using statistics collected on check-in servicing time. The statistics include check-in servicing time for different destinations and times of the day. Fig. 7 illustrates the

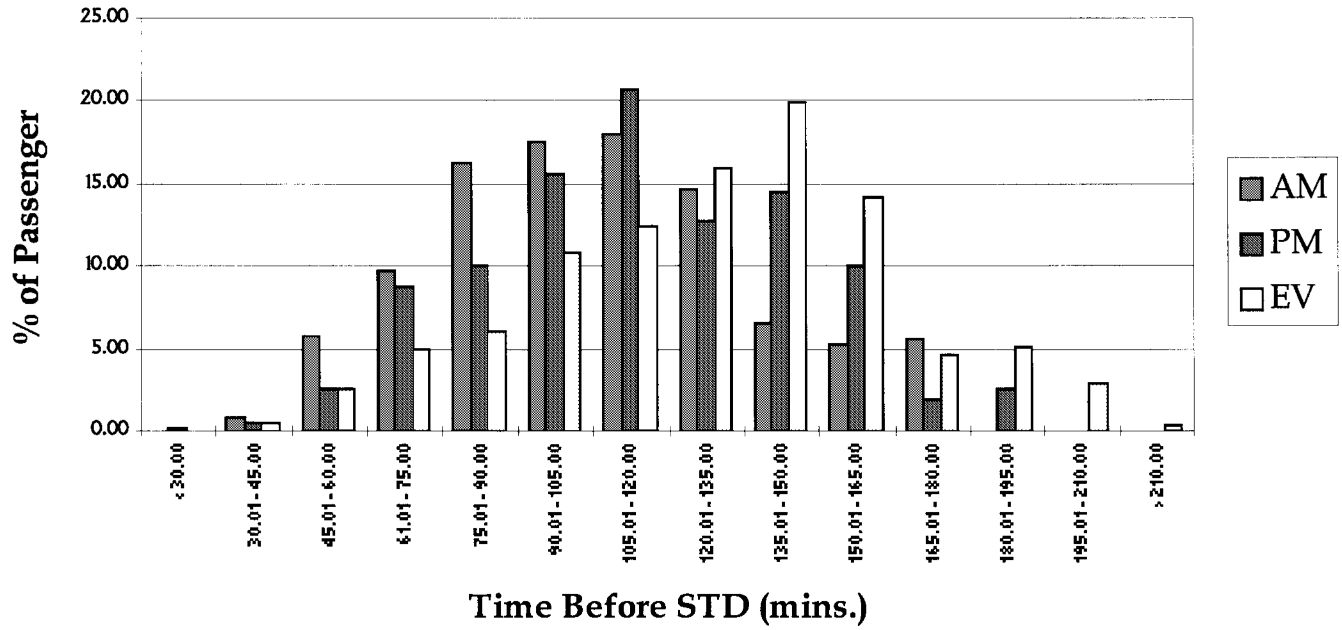

Fig. 5. Typical passenger arrival pattern for economy class passengers.

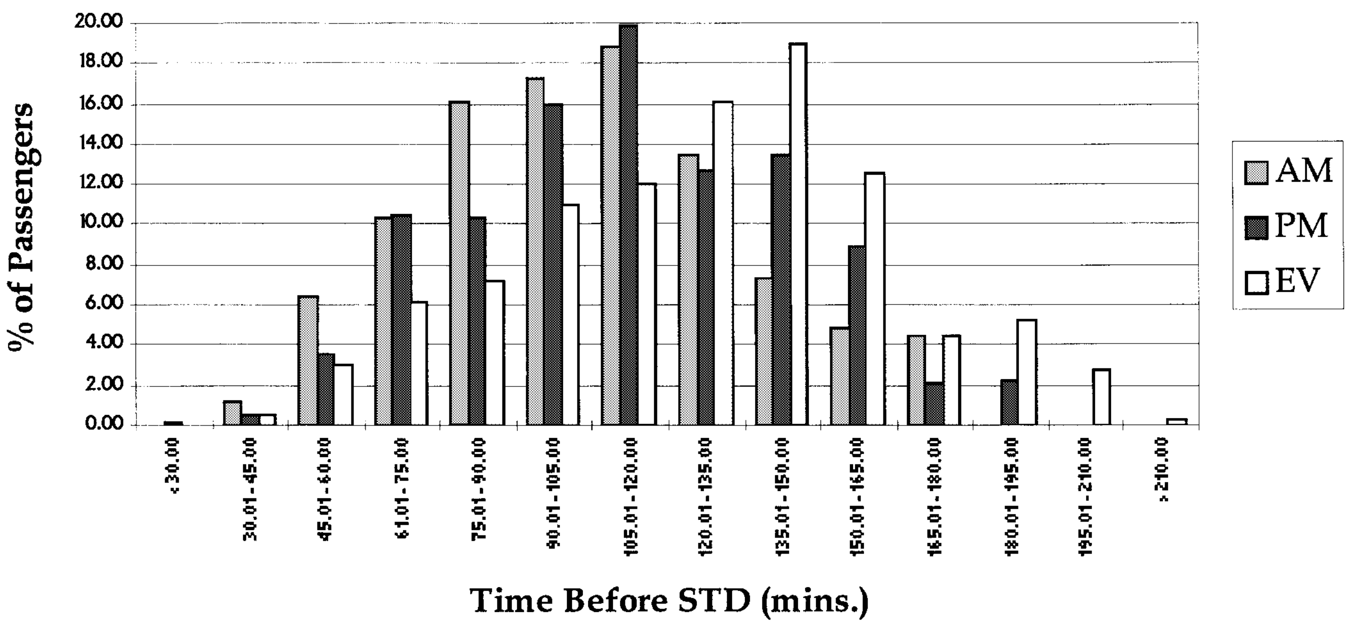

Fig. 6. Typical passenger arrival pattern for combined class passengers.

typical minimum, median, and maximum check-in processing time by different destination. Making use of these estimates, a commonly used distribution for modeling time to complete a task (beta distribution) is constructed [12]. The beta distribution requires the estimation of two shape parameters: α_1 and α_2. The two shape parameters are determined from estimates of the minimum, maximum, and mode of the service time from the statistics collected.

The check-in mechanism used in the simulation model mimics the characteristics of the check-in process and the actual behavior of passengers. When a passenger arrives and people are waiting in line, he/she will pick a counter with the shortest queue and wait for service. In Hong Kong, counters are parallel and each has its own queue. In addition, the check-in processing time depends on the flight destination, which may have different check-in requirements. The processing rate is assumed to be homogeneous for all the counters servicing the same flight, and we assume staff members at counters of the same flight have similar skills and performance levels.

3) Queue Discipline: The queue discipline at the check-in counters is FIFO, which means passengers are served in a first-in, first-out fashion.

B. Implementation

The IRSS is designed to be a knowledge-based simulation system that combines simulation with knowledge engineering.

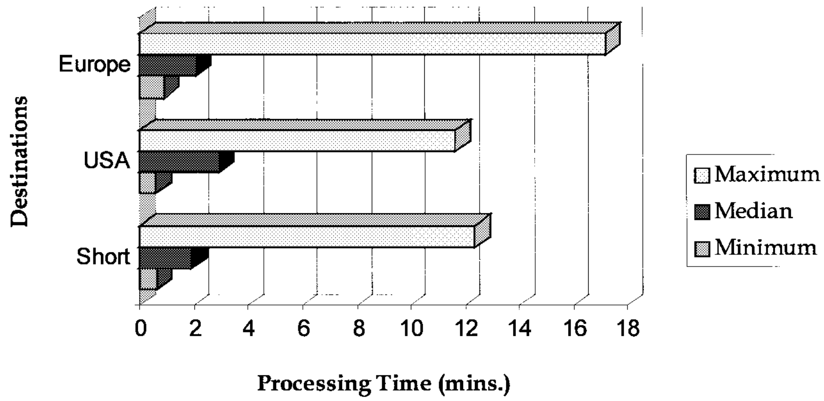

Fig. 7. Typical check-in processing time by different destination.

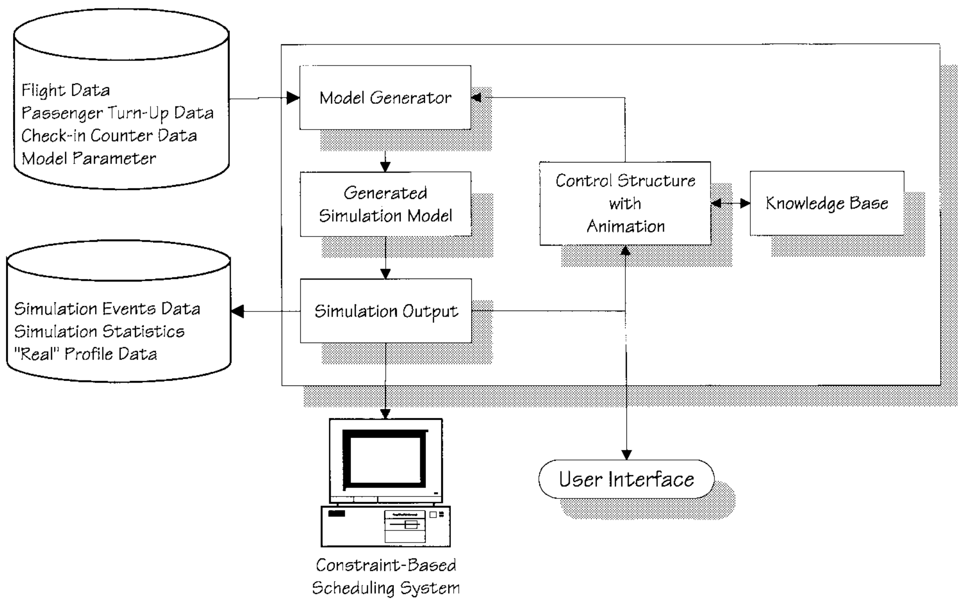

Fig. 8. Architecture of the IRSS.

The IRSS consists of the airport database, a simulation model, a control structure with animation, a knowledge base, and a user interface. The architecture of IRSS is illustrated in Fig. 8, and each element in the system is discussed below.

The database contains the flight schedule, passenger turn-up profile, check-in counter processing times, parameters used in the simulation model, and simulated results and statistics. Whenever a flight is selected for simulation, the model generator will retrieve the passenger turn-up profile, flight information, and check-in counter data from the database to build the corresponding simulation model and experiment data set.

The simulation model defines the static and dynamic characteristics of the system. The experiment data set defines the experimental conditions under which the model is to be executed. This experiment data set can be easily changed without affecting the model. Hence, different "what-if" analysis can be performed by simply changing the parameters in the experiment data set. The simulation model and experiment files are generated using the parameter values stored in the model parameter data file. When the model executes, the simulated passenger arrival time, waiting time, and check-in processing time are collected and stored into the system database.

The control structure is designed to control and monitor the simulation process according to the architecture illustrated in Fig. 3. For each flight or common check-in group in the seasonal schedule, the corresponding simulation model and experimental files will be constructed using the model generator. The generated simulation model is then executed. The simulation results are compared with the tolerable service levels. The search heuristics automatically adjust the simulation model until a minimal check-in counter profile is found for each flight or common check-in group.

A graphic user interface (GUI) provides the means for users to interact with the system. This user interface is called the profile simulator and is used to simulate and animate the check-in counter queuing for one single flight (Fig. 9). The main purpose of the profile simulator is to allow the user to experiment with different combinations of the simulation parameters and to visualize the results. For example, the process of determining the real profile can be inspected visually using the simulator. The animation window graphically simulates passengers queuing in front of a set of check-in counters. The visual animation provides a way for the user to visualize effects of different what-if scenarios. Three types of what-if analysis

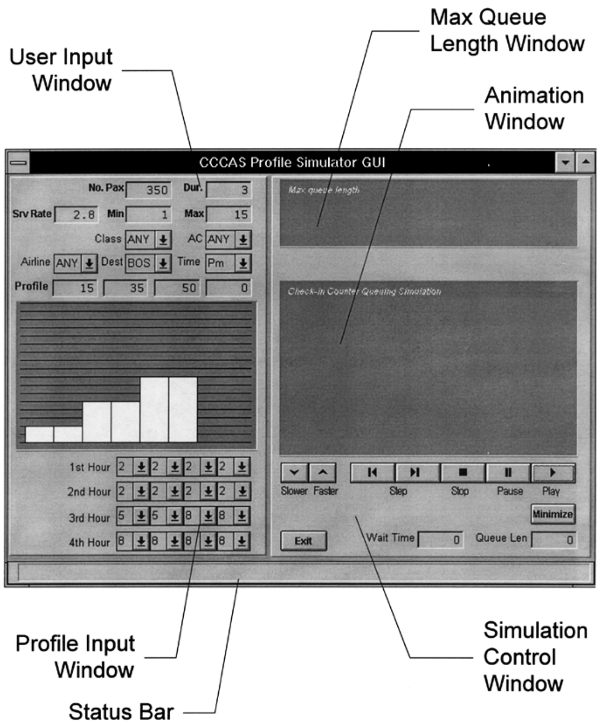

Fig. 9. Profile simulator.

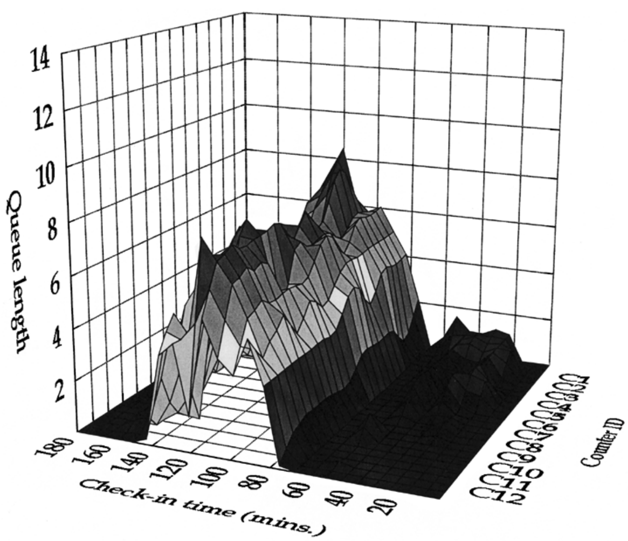

Fig. 10. Simulation result for counter profile 12-12-12.

can be performed. The first is to visualize the results from changing simulation parameters such as the tolerable queue length and waiting time or the service rate. The second is to visualize the results from changing the check-in counter profile itself such as extending the check-in time or adding or reducing the number of check-in counters. The third is to visualize the results from changing the flight parameters such as the number of locally boarding passengers or the time of departure. Through the what-if analysis, the user can fine-tune the IRSS system parameters to better match the reality at the airport. The profile simulator can be used either to test a particular counter profile or to automatically generate a counter profile that minimizes the total number of counter minutes (see Fig. 9,

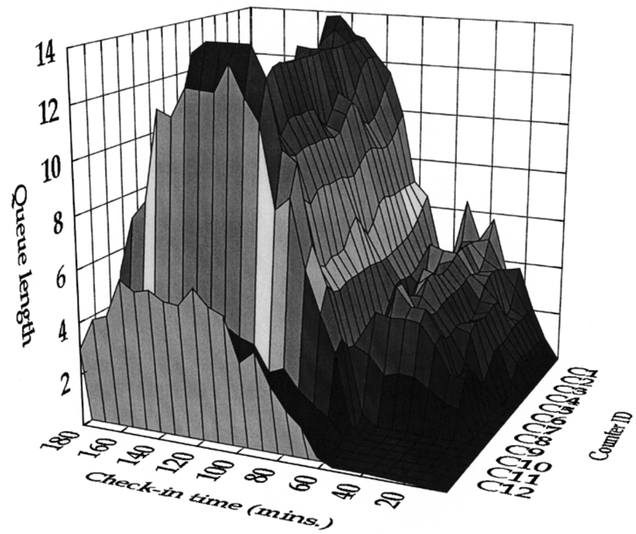

Fig. 11. Simulation result for counter profile 8-10-12.

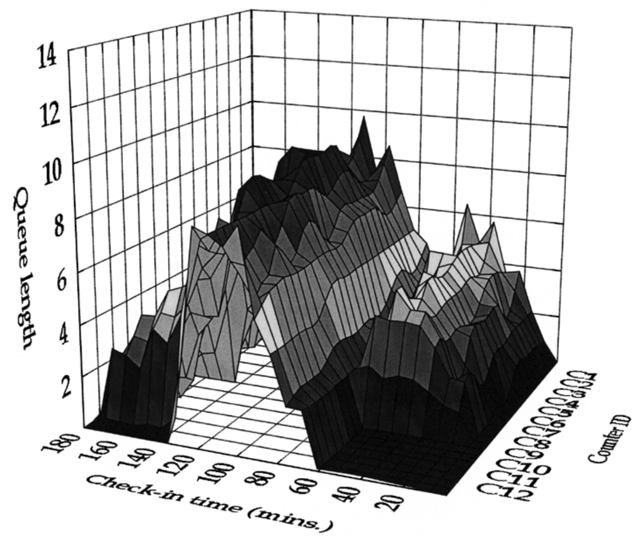

Fig. 12. Simulation result for counter profile 8-12-10.

the "Minimize" button at the lower right-hand corner). Once the user has fine-tuned the desired parameters, the IRSS can then be used to perform simulation over the whole schedule using the new set of parameters.

V. Simulation Results

This section presents a detailed analysis of the results from several simulated flights with the IRSS. Figs. 10–13 show the impact on the queue length when different counter profiles are used for a morning flight to the United States that has a three hour check-in and 418 passengers. Average processing time is assumed to be 3.81 min in all cases. Fig. 10 shows the change in queue length at each counter when twelve counters are opened for the entire check-in period. The service quality provided by this counter profile is within the tolerance level, where average queue length is less than three, and the maximum queue length is less than twelve. A counter profile of "8-10-12" means that eight counters are opened in the first hour of operation, ten counters in the next hour, and 12

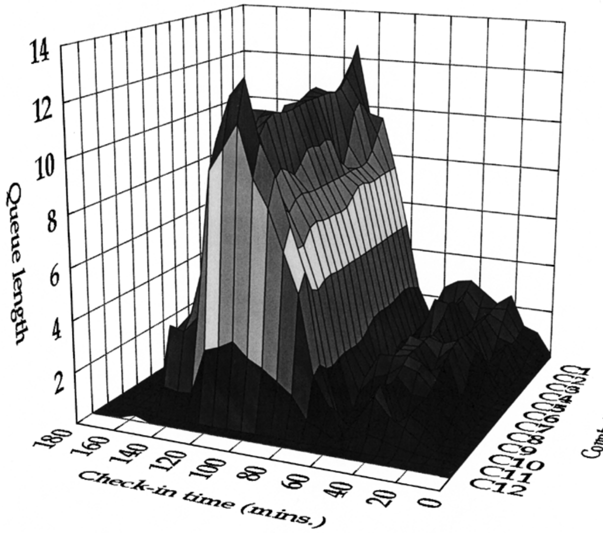

Fig. 13. Simulation result for counter profile 10-10-7.

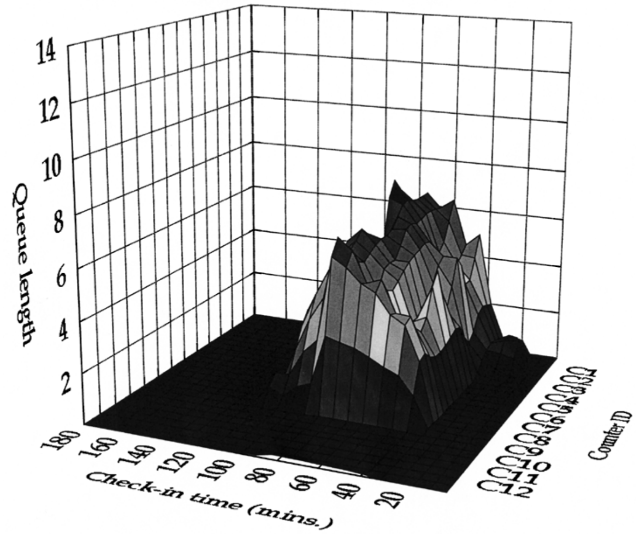

Fig. 14. Simulation result for counter profile 8-8-8.

TABLE I
COMPARISON OF COUNTER PROFILES FOR 3 h DURATION AND 418 PASSENGERS

Profile Type	Total Cntr-Mins.	Avg. Queue Length	Max. Queue Length	Avg. Waiting Time	Max. Waiting Time	% Resource Saving
12-12-12	2160	1.4	8.0	10.0	55.7	-
10-12-10	1920	1.7	8.0	11.5	57.9	11.1%
8-12-12	1920	2.5	9.0	17.6	71.1	11.1%
8-10-12	1800	3.4	14.0	23.6	83.4	16.7%
10-10-8	1680	2.9	12.0	16.9	69.0	22.2%
10-10-7*	**1620**	**2.9 (3)**	**12.0 (12)**	**17.0 (4.68)**	**63.2 (39.58)**	**25.0%**

* Minimum counter profile (Numbers in parenthesis indicate the tolerable service level).

counters in the last hour. Figs. 11–13 show the corresponding surface plot of using alternative profiles: 8-10-12, 8-12-10, and 10-10-7 respectively. Clearly, counter profile 8-10-12 cannot meet the desired service quality level. By using two more counters during the second hour of operation and closing two in the last hour, the effect is reversed because now the desired service quality level is satisfied. The minimum counter profile for this flight was found to be 10-10-7.

Table I compares the different counter profiles considered. The measures for service quality include the average queue length, maximum queue length, average waiting time, and maximum waiting time. The percentage resource saving indicates how many counter minutes are saved by using the alternative profile. The profile with an asterisk is the minimum profile for this flight. The values in parentheses are the tolerance level for the service quality measures. In this case, some of the measures cannot be satisfied even using the current profile. Hence, queue length is used as the primary measure for service quality.

Figs. 14–17 show another morning flight to the United States that has two and a half hours for check-in and 250 passengers. Average processing time is assumed to be 3.81 min. Here, counter profile "8-8-8" means eight counters are opened in the first half hour of operation, eight counters in the second hour, and eight in the last hour. Fig. 14 shows the effect of using

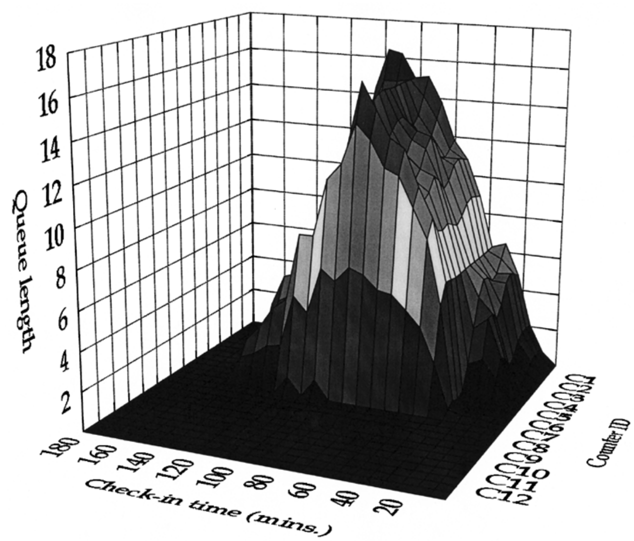

Fig. 15. Simulation result for counter profile 4-6-8.

eight counters during the entire check-in period. Figs. 15–17 show other counter profiles that have been examined. Fig. 16 shows the profile 6-6-6, which is used currently in the manual approach as the alternative profile for resolving conflicts. The minimum profile for this flight was found to be 5-7-3, which saves the most counter minutes but still maintains the desired service quality level. Table II compares the different counter profiles examined.

An afternoon flight to Japan is also studied, and the results are shown in Figs. 18–21. This flight has a two hour check-in, 218 passengers, and an average processing time of 2.28 min. A counter profile of "4-4" means four counters are opened in the first hour of operation and four in the next hour. Fig. 19 shows the profile 4-3, which is used currently in the manual approach as the alternative profile for resolving conflicts. Table III compares the different counter profiles examined. The minimum profile was found to be 5-3, which satisfies all the service quality tolerance levels. Although there is no real percentage resource saving, one counter is freed in the second

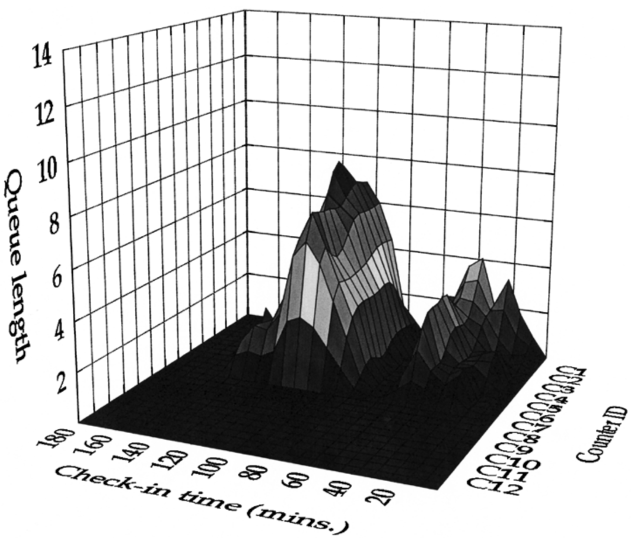

Fig. 16. Simulation result for counter profile 6-6-6.

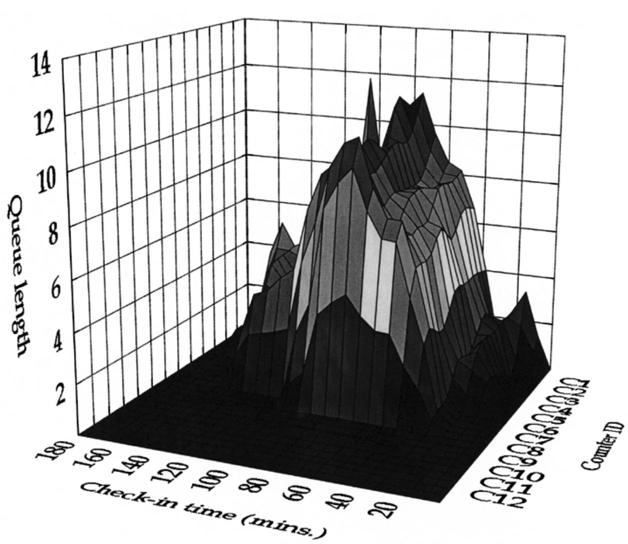

Fig. 17. Simulation result for counter profile 5-7-3.

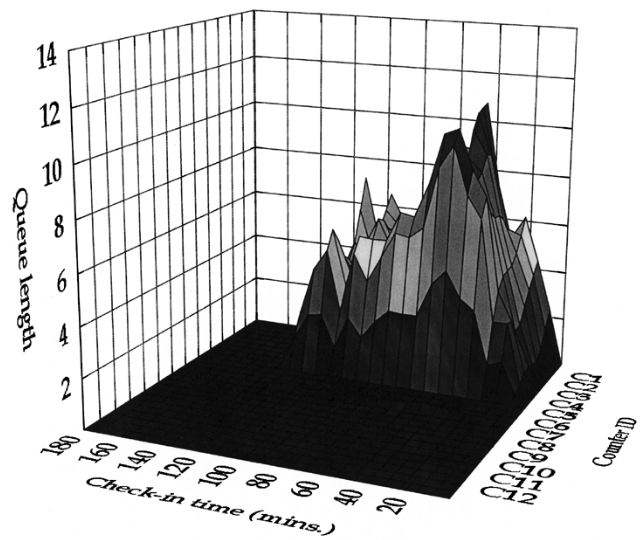

Fig. 18. Simulation result for counter profile 4-4.

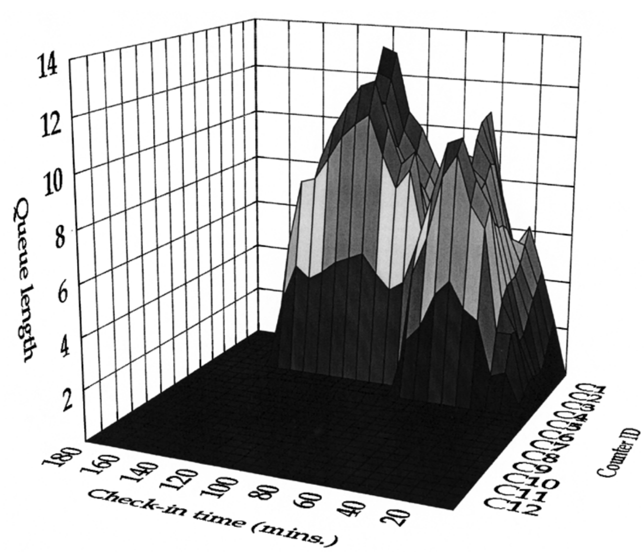

Fig. 19. Simulation result for counter profile 4-3.

TABLE II
COMPARISON OF COUNTER PROFILES FOR 2.5 h DURATION AND 250 PASSENGERS

Profile Type	**Total Cntr-Mins.**	**Avg. Queue Length**	**Max. Queue Length**	**Avg. Waiting Time**	**Max. Waiting Time**	**% Resource Saving**
Human gen. (6-6-6)	900	1.36	8.0	7.85	31.55	25%
8-8-8	1200	1.21	7.0	9.29	37.49	-
6-8-6	1020	1.43	8.0	10.94	50.39	15.0%
4-8-6	960	2.05	9.0	15.74	69.10	20.0%
4-6-8	960	3.19	15.0	24.51	72.14	20.0%
4-7-6	900	3.10	12.0	20.83	66.52	25.0%
5-7-4	810	2.66	11.0	17.85	59.04	32.5%
5-7-3*	**750**	**2.81 (3)**	**11.0 (12)**	**18.89 (4.68)**	**72.76 (39.58)**	**37.5%**

* Minimum counter profile (Numbers in parenthesis indicate the tolerable service level).

hour of operation. This can make a difference if there is a need for an additional counter for another flight during that hour.

VI. CONCLUSION

This paper documents our research and development in creating a knowledge-based simulation system to predict check-in counter resource requirements at one of the world's busiest airports. Results generated by this knowledge-based simulation system are then used by a constraint-based scheduling system that generates a schedule for daily check-in counter allocation. Results from our experiments show that combining intelligent simulation with resource allocation can lead to resource savings of up to 40%. This approach provides a more accurate means of predicting resource requirement than the current manual approach that is based upon the individual experience of each human scheduler. The CCAS and IRSS were in use at the Kai Tak International Airport from 1995 until its last day of operation on July 5, 1997.

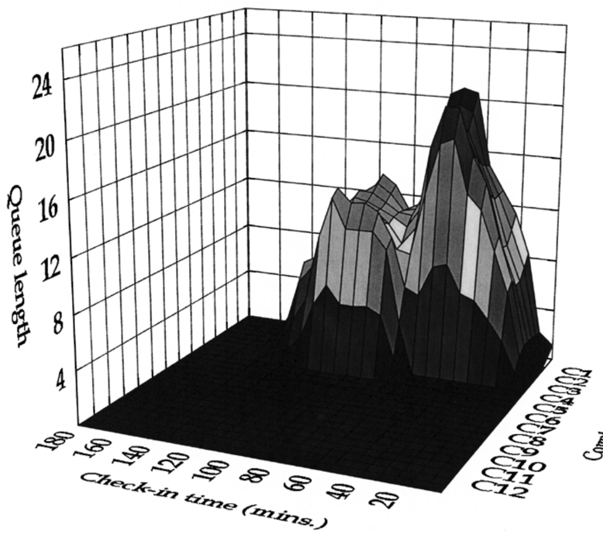

Fig. 20. Simulation result for counter profile 3-4.

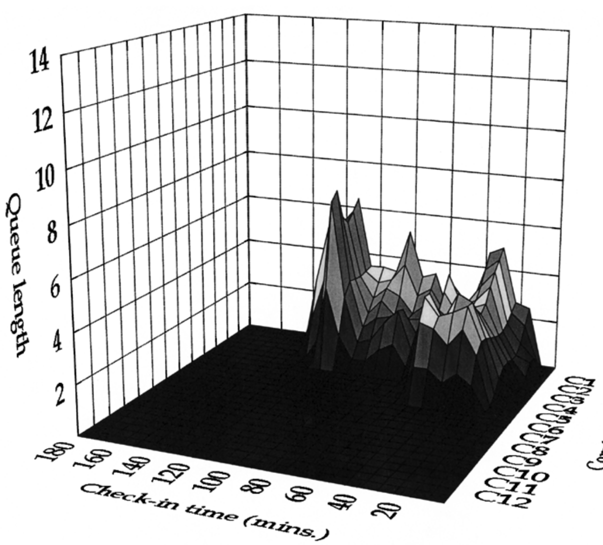

Fig. 21. Simulation result for counter profile 5-3.

TABLE III
COMPARISON OF COUNTER PROFILES FOR 2 h DURATION AND 218 PASSENGERS

Profile Type	Total Cntr-Mins.	Avg. Queue Length	Max. Queue Length	Avg. Waiting Time	Max. Waiting Time	% Resource Saving
Human gen. (4-3)	420	3.52	13.0	15.48	39.17	12.5%
4-4	480	2.66	11.0	11.73	38.54	-
3-4	420	5.20	21.0	22.91	56.03	12.5%
3-5	480	3.30	21.0	18.19	51.89	0.0%
5-3*	**480**	**1.05 (3)**	**7.0 (12)**	**5.77 (10.86)**	**23.09 (25.12)**	**0.0%**

* Minimum counter profile (Numbers in parenthesis indicate the tolerable service level).

ACKNOWLEDGMENT

Commercialization of this research was performed by CityU Consultants Limited and Resource Technologies Limited. Extending the system for stand allocation was performed by Advanced Object Technologies Limited. The authors would like to thank the Hong Kong Civil Aviation Department for making available Hong Kong International Airport statistical data, without which this research would not have been possible.

REFERENCES

[1] K. M. Chandy and C. H. Sauer, *Computer Systems Performance Analysis*. Englewood Cliffs, N.J.: Prentice-Hall, 1981.
[2] H. W. Chun, "Scheduling as a multi-dimensional placement problem," *Eng. Applicat. Artif. Intell.*, vol. 9, pp. 261–274, June 1996.
[3] H. W. Chun, "Solving check-in counter constraints with ILOG SOLVER," in *Proc. 1st ILOG Solver and ILOG Schedule Int. Users Meeting*, Abbaye des Vaux de Cernay, France, 1995.
[4] H. W. Chun, "Representing constraints in airport resource allocation," in *Proc. 8th Int. Symp. Artificial Intelligence*, Monterrey, Mexico, 1995, pp. 314–319.
[5] K. K. Cornett, S. A. Miller, and W. S. Ottman, "An aircraft operations simulation model for the United Parcel Service Louisville park," in *Proc. 1996 Winter Simulation Conf.*, 1996.
[6] G. R. Drake and J. S. Smith, "Simulation system for real-time planning, scheduling, and control," in *Proc. 1996 Winter Simulation Conf.*, 1996.
[7] P. T. Fishburn, J. Oolkar, and K. M. Taaffe, "Simulation of transportation systems," in *Proc. 1995 Winter Simulation Conf.*, 1995.
[8] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*, 2nd ed. New York: Wiley, 1985.
[9] R. A. Haeme, J. L. Huttinger, and R. W. Shore, "Airline performance modeling to support schedule development: An application case study," in *Proc. 1988 Winter Simulation Conf.*, M. Abrams, P. Haigh, and J. Comfort, Eds., San Diego, CA: AIIE, 1988, pp. 800–806.
[10] L. Kleinrock, *Queueing Systems*, vol. 1, *Theory*. New York: Wiley, 1975.
[11] ——, *Queueing Systems*, vol. 2, *Computer Applications*. New York: Wiley, 1976.
[12] A. M. Law and M. G. McComas, "Secrets of successful simulation studies," *Ind. Eng.*, vol. 22, pp. 47–48, 51–53, May 1990.
[13] C. D. Pegden, R. E. Shannon, and R. P. Sadowski, *Introduction to Simulation Using SIMAN*, 2nd ed. New York: McGraw-Hill, 1995.
[14] F. Sokkar and S. V. Nelson, "Examination of air traffic flow at a major airport," in *Proc. 1990 Winter Simulation Conf.*, O. Balci, R. Sadowski, R. Nance Eds., San Diego, CA: AIIE, 1990, pp. 784–792.
[15] W. E. Weiss and E. S. Lacher, "Simulating the national airspace system," in *Proc. 1988 Winter Simulation Conf.*, M. Abrams, P. Haigh, and J. Comfort, Eds., San Diego, CA: AIIE, 1988, pp. 796–799.

Hon Wai Chun (S'79–M'81) received the B.S. degree from the Illinois Institute of Technology, Chicago, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign in 1983 and 1987, respectively.

He was a Senior Scientist at Ascent Technology, Boston, MA, before joining the Information Engineering Department of the Chinese University of Hong Kong as Lecturer in 1989. He later joined the Department of Electronic Engineering, City University of Hong Kong, in March 1994 and is now an Associate Professor there. He is jointly the Managing Director of Advanced Object Technologies Limited and a Principal Consultant for CityU Consultants Limited. His research interests are in simulation, optimization, operations research, constraint programming, object-oriented analysis and design, and distributed object architecture.

Raymond Wai Tak Mak received the B.S. and M.S. degrees in industrial engineering from Northeastern University, Boston, MA, in 1991 and 1993, respectively, where he is pursuing his Ph.D. degree.

He is currently a Technical Business Advisor with Federal Express Corporation, Hong Kong. His research interests are in business process simulation and scheduling.

Dr. Mak is listed in Who's Who Among Students in American Universities and Colleges and is a member of AAAI, IIE, INFORMS, and CASS.