Multilevel Passenger Screening Strategies for Aviation Security Systems

Laura A. McLay and Sheldon H. Jacobson
Department of Mechanical and Industrial Engineering
University of Illinois at Urbana-Champaign

John E. Kobza
Department of Industrial Engineering
Texas Tech University

Abstract

Passenger prescreening is a critical component of aviation security. This paper introduces the Multilevel Allocation Problem (MAP), which models the screening of passengers and baggage in a multilevel aviation security system. A passenger is screened by one of several classes, each of which corresponds to a set of procedures using security screening devices, where passengers are differentiated by their perceived risk levels. Each class is defined in terms of its fixed cost (the overhead costs), its marginal cost (the additional cost to screen a passenger), and its security level. The objective of MAP is to assign each passenger to a class such that the total security is maximized subject to passenger assignments and budget constraints. This paper shows that MAP is NP-hard, introduces two dynamic programming algorithms for solving MAP in pseudo-polynomial time, and introduces a Greedy heuristic that obtains approximate solutions to MAP that use no more than two classes. Examples are constructed using data extracted from the Official Airline Guide (OAG). Analysis of the examples suggests that fewer security classes for passenger screening may be more effective and that using passenger risk information can lead to more effective security screening strategies.

Keywords: aviation security, policy modeling, integer programming, heuristics, dynamic programming.

Introduction

On September 11, 2001, four commercial aircraft were hijacked and used as bombs to destroy the World Trade Center twin towers and inflict severe damage to the Pentagon. These acts of violence have lead to widespread aviation security policy and operational changes throughout the nation's airports. A critical component in aviation security systems is the prescreening of passengers with their checked and carry-on baggage prior to boarding an aircraft. However, developing strategies to effectively and efficiently screen passengers, as well as the allocation and utilization of screening devices, can be quite challenging. Moreover, even after such systems are in place, it can be very difficult to measure their effectiveness.

Aviation security operations in the United States began in 1970 with surveillance equipment in airports in response to hijacking attempts. When hijacking attempts persisted, air carriers were required to physically screen all passengers with metal detectors (beginning in December 1972), with passenger prescreening operations not significantly changing from that time until 1996 (National Research Council 1996). The Commission on Aviation Safety and Security, established on July 25, 1996 and headed by (then) Vice-President Al Gore, recommended that the aviation industry improve security using existing explosive detection technologies, automated passenger prescreening, and positive passenger-baggage matching. Moreover, the Federal Aviation Administration (FAA) had been working with the airlines to annually purchase and deploy explosive detection systems (EDSs) at airports throughout the United States. From 1998 until September 11, 2001, EDSs were only used to screen checked baggage of *selectee* passengers, those who were not cleared by a computer risk assessment system (i.e., the Computer-Aided Passenger Prescreening System—CAPPS) developed in conjunction with the FAA, Northwest Airlines, and the United States Department of Justice. The checked baggage of *nonselectee* passengers, those who were cleared by such a system, received no additional security attention. There were no further differences between selectee and nonselectee passengers.

The terrorist events on September 11, 2001 prompted the Inspector General for the United States Department of Transportation to prescribe additional security procedures to be implemented (Mead 2002; Mead 2003), including screening all checked baggage for explosives. Given such a policy, there would no longer be any distinction between selectee and nonselectee passengers, since all checked baggage would be screened. An FAA official once remarked that the FAA is in "...hot pursuit of equipment and procedures that can spot these [explosive] devices with high degrees of confidence for the nearly one billion pieces of [baggage] and 500 million passengers traveling annually on

United States carriers" (Malotky 1994). Note that in 2000, there were over 600 million passengers, with forecasts of nearly one billion passengers by 2013 (FAA 2002). The primary objective of all these efforts is to improve security operations at all of the nation's airports. To meet this objective, the Transportation and Security Administration (TSA), part of the Department of Homeland Security, must develop new security system paradigms that can optimally use and simultaneously coordinate several security technologies and procedures. Aviation security devices deployed at airport stations (e.g., terminals) provide a level of security for passengers, and determining the type of security devices to deploy can be challenging. Moreover, once such devices are deployed, the practical issue of determining how to optimally use them can be difficult.

Alternatively, experts suggest that greater scrutiny of passengers perceived as greater risks (from a security standpoint) is a more effective approach to aviation security. Butler and Poole (2002) suggest that the TSA's policy of 100% checked baggage screening is not cost-effective and that enhancing the binary screening paradigm to a multilevel screening system would be a more effective approach to process airline passengers. Poole and Passantino (2003) endorse *risk-based airport security*, partitioning passenger and baggage security devices in proportion to perceived risk. They suggest that multiple levels of security for processing passengers may be more effective than treating all passengers the same (from a security standpoint).

The TSA responded to these proposals through the development of CAPPS II, an enhanced computer-based system for systematically prescreening passengers. CAPPS II partitions passengers into three risk classes (as opposed to two classes by CAPPS). A frequently mentioned criticism of any system designed to classify passengers into risk classes, including CAPPS and CAPPS II, is that such systems can be gamed through extensive trial and error sampling by a variety of passengers through the system (Barnett 2001; Chakrabarti and Strauss 2003). Martonosi and Barnett (2003) note that trial and error sampling may not increase the probability of a successful attack and that CAPPS II may not substantially improve aviation security if the screening procedures for each type of passenger are not effective. Barnett (2004) suggests that CAPPS II may only improve aviation security under a particular set of circumstances and recommends that CAPPS II be transitioned from a security centerpiece to one of many components in future aviation security strategies. On July 14, 2004, the TSA announced that CAPPS

II has been dismantled over privacy concerns despite having invested $100M for its development (Hall and DeLollis, 2004). However, it appears that the TSA plans to replace CAPPS II with an automated system akin to CAPPS II (i.e., a system designed to partition passengers into risk classes) for passenger prescreening (Singer 2004).

This paper describes a systematic approach for designing an enhanced passenger screening system using discrete optimization models and algorithms, by formulating a problem that models multilevel passenger prescreening strategies. Multilevel screening considers no fewer than three levels of security to screen passengers, as opposed to the binary system in place prior to September 11, 2001. Therefore, the primary contribution of this effort is to identify models for designing multilevel screening security systems, and show how these models can be used to provide insights into the operation and performance of such systems. Note that this research assumes that a system such as CAPPS has been implemented and is highly effective in identifying passenger risk (TSA 2003a) .

The following definitions are needed to describe these models. An *attack* is any event in which willful human intent disrupts air service. A *threat* is a passenger or baggage directly involved in a planned attack targeted at an aircraft. For example, hijackers and terrorists with bombs in their checked baggage are threats. A *device* is an aviation security technology and/or procedure used to identify a threat. Examples of devices include metal detectors, EDS devices, and detailed hand search by an airport security official. A *class* is defined by a preassigned subset of devices and a procedure through which passengers are processed prior to boarding an aircraft. A risk assessment system, such as CAPPS, assigns each passenger an *assessed threat value*, which quantifies the risk associated with the characteristics of the passenger. The *fixed cost* of a class is the purchase and overhead costs for the devices associated with the class. Note that the fixed cost of a class is assessed against the budget only if there are passengers assigned to the class. The *marginal cost* associated with a class is the direct cost to screen each passenger or bag assigned to the class. These values may be acquired or estimated from statements and press releases given by the TSA and published articles on aviation security in the public domain.

Secondary screening is needed to resolve alarms in each class. In practice, the same secondary screening procedures may be used to resolve alarms in all of the classes, which suggests a degree of overlap between the classes. However, this research focuses on the primary screening procedures associated with each class, and it assumes that there are enough resources for resolving alarms. Hilkevitch (2003) describes the secondary screening procedures used at Heathrow International Airport in London.

Several articles formulate aviation security problems as integer programming and discrete optimization models. Jacobson et al. (2001) provide a framework for measuring the effectiveness of a baggage screening security device deployment at a particular station. Jacobson et al. (2002) introduce three performance measures for baggage screening security systems and use these models to assess the security effect for single or multiple stations. Jacobson et al. (2005) formulate problems that model multiple sets of flights originating from multiple stations subject to a finite amount of resources. These problems consider three performance measures, and examples suggest that one of the performance measures may provide more robust screening device allocations. Virta et al. (2002) consider the impact of originating and transferring passengers on the effectiveness of baggage screening security systems. In particular, they consider classifying selectees into two types; those at their point of origin and those transferring. This is noteworthy since at least two of the hijackers on September 11, 2001 were transferring passengers.

Other research has focused on the experimental and statistical analysis of risk and security procedures on aircraft. For example, Barnett et al. (2001) performed a large-scale two-week experiment at the nation's airports to test which costs and disruptions would arise from using positive passenger baggage matching (PPBM), an aviation security procedure, for all flights. Barnett et al. (1979) and Barnett and Higgins (1989) study mortality rates on passenger aircraft and perform a statistical analysis on this data.

The paper is organized as follows. Section 2 introduces the Multilevel Allocation Problem (MAP), a discrete optimization model that considers budget allocation based on class costs and shows that this problem is NP-hard. Section 3 introduces a heuristic that provides approximate solutions to MAP, and describes two dynamic programming algorithms that solve MAP in pseudo-polynomial time. Section 4 provides a real-world example using data extracted from the Official Airline Guide (OAG). Section 5 provides concluding comments and directions for future research. The Appendix contains the proofs of the theorems and lemmas. A companion paper analyzes theoretical aspects of MAP as it is formulated as a knapsack problem (McLay and Jacobson 2003).

1 Discrete Optimization Model

This section introduces the Multilevel Allocation Problem (MAP), a general framework for multilevel security screening, and formulates it as a discrete optimization problem. MAP is first stated as an optimization problem and then formulated as an integer programming model. The objective is to assign N passengers to M classes such that the total security is maximized subject to budget and assignment constraints. The classes are defined in terms of their fixed and marginal costs, which are determined by the set of devices that define the classes. Although this problem assigns passengers to classes, it ultimately determines how the budget should be allocated to the various classes and which classes should (and should not) be used. MAP is formally stated.

The Multilevel Allocation Problem (MAP)
Given:

A set of N passengers, each of which is characterized by an assessed threat value $AT_1, AT_2, \ldots, AT_N$ with $0 < AT_i \leq 1,\ i = 1, 2, \ldots, N$,

a set of M classes,

a fixed cost associated with each class $FC_1, FC_2, \ldots, FC_M$,

a marginal cost associated with each class $MC_1, MC_2, \ldots, MC_M$,

the total budget B,

the *security level* of each class, L_i, where $0 \leq L_i \leq 1,\ i = 1, 2, \ldots, M$.

Denote passenger assignments for the N passengers to the M classes by $A_1, A_2, \ldots, A_M$, where $A_i \subseteq \{1, 2, \ldots, N\}$ represents the subset of passengers who are assigned to class i, and define the risk level R_i of class $i = 1, 2, \ldots, M$ as the proportion of assessed threat values of the passengers assigned to class i. Hence,

$$R_i = \frac{1}{\sum_{j=1}^{N} AT_j} \sum_{j \in A_i} AT_j,\ \ i = 1, 2, \ldots, M \tag{1}$$

Find passenger assignments $A_1, A_2, \ldots, A_M$ such that $\bigcup_{i=1}^{M} A_i = \{1, 2, \ldots, N\}$ and $A_{i_1} \cap A_{i_2} = \emptyset$ for $i_1,\ i_2 = 1, 2, \ldots, M$, $i_1 \neq i_2$, and such that the budget constraint is satisfied (i.e., $\sum_{i=1}^{M} |A_i|\ MC_i + \sum_{\{i:|A_i|>0\}} FC_i \leq B$) and the total security is maximized (i.e., $\sum_{i=1}^{M} L_i R_i$).

The assessed threat values, the security levels and the risk levels can be set with information and data available from CAPPS and the TSA. The assessed threat values provide risk assessment measures for each passenger (scaled between zero and one). It is assumed that the assessed threat values are accurate representations of the passengers' true level of risk (i.e., passengers with higher assessed threat values are more likely to be a threat than passengers with lower assessed threat values). Another way to interpret passenger risk is to base the assessed threat values on passenger and baggage attributes (e.g., a checked bag having a large amount of metal shielding has a larger assessed threat value than a bag without shielding), and hence, the assessed threat values are assigned as passengers check in for their flights.

The security level of each class (scaled between zero and one) is based on security procedures of each device used to screen passengers in that class. In this case, the security level for class i is defined as the true alarm rate, the probability that a passenger who is a threat is detected given that they are assigned to class i. Likewise, the risk level for class i is defined as the conditional probability that class i contains a passenger who is a threat given that the passenger population contains a passenger who is a threat. In order for the risk levels to be interpreted as this conditional probability, each assessed threat value must be proportional to the probability that the passenger is a threat. The total security is then the overall true alarm rate, the probability that a threat is detected given that there is a passenger who is a threat. To see this, define the following events:

D = a threat is detected in the passenger population,
T = the passenger population contains a threat,
C_i = class i contains a passenger who is a threat, $i = 1, 2, \ldots, M$.

By conditioning on which class contains a threat, the total security can be expressed as

$$P(D|T) = \sum_{i=1}^{M} P(D|C_i, T)\, P(C_i|T) = \sum_{i=1}^{M} L_i\, R_i.$$

Since each of the passengers in class i are screened individually, L_i is the probability of detecting a class i passenger who is a threat. It is a function of the detection probabilities associated with the devices and the procedures used to screen passengers in class i.

MAP is formulated as an integer program (2) with binary decision variables $x_{ij} = 1(0)$ if passenger j is (not) assigned to class i for $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$, and $y_i = 1(0)$ if there is (not) at least one passenger assigned to class $i = 1, 2, \ldots, M$.

$$\max \sum_{i=1}^{M} L_i R_i \quad = \quad \frac{1}{\sum_{j=1}^{N} AT_j} \sum_{i=1}^{M} \sum_{j=1}^{N} L_i AT_j x_{ij} \tag{2}$$

$$\text{subject to} \quad \sum_{i=1}^{M}\sum_{j=1}^{N} MC_i x_{ij} + \sum_{i=1}^{M} FC_i y_i \leq B$$
$$\sum_{i=1}^{M} x_{ij} = 1,\ j = 1,2,\ldots,N$$
$$\frac{1}{N}\sum_{j=1}^{N} x_{ij} - y_i \leq 0,\ i = 1,2,\ldots,M$$
$$y_i \in \{0,1\},\ i = 1,2,\ldots,M$$
$$x_{ij} \in \{0,1\},\ i = 1,2,\ldots,M,\ j = 1,2,\ldots,N.$$

In (2), the objective is to maximize the total security, which is represented by the sum of the products of the security and risk levels. The first constraint is the budget feasibility constraint. The second set of N constraints ensures that each passenger is assigned to exactly one class. The third set of M constraints ensures that the fixed costs are included for all nonempty classes. The last two sets of $M(1+N)$ constraints restrict the y_i and x_{ij} to being 0-1 binary variables.

Theorem 1 shows that MAP is NP-hard.

Theorem 1: MAP is NP-hard.

Note that MAP can be restricted in several ways and still remain NP-hard. First, since the polynomial Turing reduction considered fixed costs of zeros, MAP remains NP-hard when $FC_i = 0,\ i = 1,2,\ldots,M$. Secondly, MAP remains NP-hard when the assessed threat values for all passengers are identical (i.e., when passengers are indistinguishable). Since Theorem 1 shows MAP is NP-hard for a particular risk level form, MAP remains NP-hard for general risk level forms $R_1, R_2, \ldots, R_M$. However, when $M = 2$, MAP is solvable in polynomial time (see Lemma 3 in Section 3.1).

2 Heuristics and Dynamic Programming

This section introduces a Greedy heuristic, the Two Class Greedy Heuristic, that obtains approximate solutions to MAP in polynomial time, and two dynamic programming algorithms that obtain optimal solutions to MAP in pseudo-polynomial time. Without loss of generality, assume that $AT_1 \leq AT_2 \leq \ldots \leq AT_N$ and $L_1 \leq L_2 \leq \ldots \leq L_M$.

The heuristics and algorithms find the number of passengers assigned to each class and then construct the exact passenger assignments. Lemma 1 indicates that for any feasible solution to MAP, the passengers with higher assessed threat values are assigned to classes with higher security levels. Therefore, if n_M passengers are assigned to class M, the class with the highest security level, then the n_M passengers with the largest assessed threat values are assigned to class M. Lemma 2 shows that MAP is polynomial solvable for a fixed number of classes.

Lemma 1: Given an instance of MAP with $AT_1 \leq AT_2 \leq \ldots \leq AT_N$ and $L_1 \leq L_2 \leq \ldots \leq L_M$, the optimal way to assign the N passengers to the M classes given that n_i passengers are assigned to class $i = 1, 2, \ldots, M$ is to assign the first n_1 passengers to class 1, the next n_2 passengers to class 2, until the last n_M passengers are assigned to class M.

Lemma 2: MAP is polynomial solvable when there are a fixed number of classes (i.e., M is fixed).

2.1 The Two Class Greedy Heuristic

The Two Class Greedy Heuristic (2GH) obtains the solution to MAP with the highest objective function value using no more than two classes. To describe 2GH, let $C^{N^+} = \left\{i : \left\lfloor \frac{B-FC_i}{MC_i} \right\rfloor \geq N\right\}$, the subset of classes such that the budget is sufficiently large for each class to screen all N passengers, and let $C^{N^-} = \left\{i : \left\lfloor \frac{B-FC_i}{MC_i} \right\rfloor < N\right\}$, the remaining subset of classes. Note that C^{N^-} and C^{N^+} are mutually exclusive and exhaustive subsets of $\{1, 2, \ldots, M\}$. Two trivial special cases are $C^{N^+} = \emptyset$, when there is no feasible solution (i.e., the budget is not large enough to assign all N passengers to the M classes), and $C^{N^-} = \emptyset$, when the optimal solution value is equal to the largest security level (i.e., the budget is sufficient to assign all N passengers to the class with the largest security level).

To find the solution to MAP with the highest objective function value using no more than two classes, 2GH selects the solution from a set of candidate solutions with the highest objective function value. The candidate solutions are selected from the set of feasible solutions to MAP, where the remaining (non-candidate) feasible solutions can be pruned by assigning individual passengers to classes (see Lemma 1). The set of candidate solutions are obtained as follows: 2GH begins by partitioning the classes into C^{N^-} and C^{N^+} and finding the best solution using a single class. The optimal solution using exactly one class assigns all N passengers to the class in C^{N^+} with the largest security level. 2GH then obtains candidate solutions by considering solutions using exactly two classes, with one class in C^{N^-} and one in C^{N^+}, where each such pair of classes results in at most one candidate solution. The best solution using classes $i^+ \in C^{N^+}$ and $i^- \in C^{N^-}$ assigns the maximum number of passengers into the class with the highest security level such that the assignment and budget constraints are satisfied (i.e., the solution is feasible). Once the number of passengers assigned to each class is known, individual passengers are assigned to the two classes (see Lemma 1). Only classes $i^- \in C^{N^-}$ and $i^+ \in C^{N^+}$ such that $L_{i^+} < L_{i^-}$ need to be considered; otherwise the resulting solutions would be no better than the best solution using a single class.

In the 2GH pseudo-code, z_h is the current best objective function value, the vector x of length MN is its associated passenger assignment, and B_h is the interim cost of this passenger assignment. For each candidate solution considered, z'_h is the objective function value, $\lfloor n_{i^-} \rfloor$ and $\lceil n_{i^+} \rceil$ are the number of passengers assigned to $i^- \in C^{N^-}$ and $i^+ \in C^{N^+}$, respectively, and the vector x' is the passenger assignment.

2GH executes in $\mathrm{O}(\max\{N \log N, M^2 N\})$ time. The initial sorting of the assessed threat values and the partitioning of the classes into C^{N^+} and C^{N^-} requires $\mathrm{O}(M + N \log N)$ time. Note that 2GH considers all combinations of two classes such that one of these classes is in C^{N^+} and the other class is in C^{N^-}. Therefore, 2GH obtains $|C^{N^-}||C^{N^+}| \leq M^2$ candidate solutions in the worst case, and requires $\mathrm{O}(N)$ time to compute the objective function value for each candidate solution. If the assessed threat values are identical, then 2GH obtains a solution in $\mathrm{O}(M^2)$ time since the assessed threat values do not need to be sorted and the objective function values are computed in constant time.

Lemma 3 states that 2GH finds an optimal solution to MAP when there are two classes (i.e., $M = 2$). Theorem 2 applies only to the particular case of MAP when passengers are indistinguishable. Theorem 2 shows that in this case, 2GH always obtains solutions which are at least 1/2 of the optimal objective function value.

Lemma 3: MAP is polynomial solvable when there are two classes.

Theorem 2: 2GH always obtains solutions to MAP that are at least 1/2 of the optimal solution value when passengers have identical assessed threat values.

algorithm 1 The Two Class Greedy Heuristic

procedure 2GH
  Sort assessed threat values such that $AT_1 \leq AT_2 \leq \ldots \leq AT_N$
  Partition the set of classes into C^{N^-} and C^{N^+}
  $z_h = \max\{L_i : i \in C^{N^+}\}$
  $x_{ij} = 0,\ i = 2, \ldots, M,\ j = 1, 2, \ldots, N;\ x_{C_1^+ j} = 1, j = 1, 2, \ldots, N$
  $B_h = N \cdot MC_{C_1^+} + FC_{C_1^+}$
  for all $i^+ = C^{N^+}$, $i^- = C^{N^-}$ **do**
    Solve the linear equations:
    (1) $(MC_{i^+})n^+ + (MC_{i^-})n^- = B - FC_{i^+} - FC_{i^-}$
    (2) $n^+ + n^- = N$
    if $0 < n^- < N$ **then**
      compute z'_h, x' based on putting the $\lceil n^+ \rceil$ passengers with the lowest AT values in class i^+ and the remaining $\lfloor n^- \rfloor$ passengers with the highest AT values in class i^-
      if $z'_h > z_h$ **then**
        $z_h = z'_h,\ x \leftarrow x'$
        $B_h = FC_{i^+} + FC_{i^-} + \lceil n^+ \rceil MC_{i^+} + \lfloor n^- \rfloor MC_{i^-}$
      end if
    end if
  end for
  return $x,\ z_h,\ B_h$
end procedure

2.2 Dynamic Programming

This section introduces two dynamic programming algorithms which obtain optimal solutions to MAP in pseudo-polynomial time. For both algorithms, the passengers are sorted such that $AT_1 \leq AT_2 \leq \ldots \leq AT_N$ and the security levels are sorted such that $L_1 \leq L_2 \leq \ldots \leq L_M$. This ensures that the dynamic programming algorithms assign the passengers with the highest assessed threat values to the (nonempty) class with the largest security level (see Lemma 1).

These dynamic programming algorithms assume that the marginal costs, the fixed costs, and the budget assume values that are multiples of a base unit α. For example, the costs may be defined in multiples of \$10, which results in $\alpha = 10$. Moreover, the intermediate budgets $\hat{B}$ may only take on a set of β discrete values such that $\hat{B} = \alpha, 2\alpha, \ldots, \beta\alpha = B$, where the recursion cannot consider other intermediate budget values. Note that if $\alpha = 1$, then $\beta = B$ in the worst case, hence $\hat{B} = 1, 2, \ldots, \beta = B$.

To describe the first dynamic programming algorithm for MAP, let $g_{m,n}(\hat{B})$ denote the optimal solution to the problem defined over the first $m = 1, 2, \ldots, M$ classes, and

$n = 1, 2, \ldots, N$ passengers, with budget $\hat{B} = \alpha_1, \alpha_2, \ldots, \alpha_\beta = B$,

$$g_{m,n}(\hat{B}) = \max\left\{ \frac{1}{\sum_{j=1}^{N} AT_j} \sum_{i=1}^{m} \sum_{j=1}^{n} L_i AT_j x_{i,j} \;\middle|\; \sum_{i=1}^{m}\sum_{j=1}^{n} MC_i x_{i,j} + \sum_{i=1}^{m} FC_i y_i \le \hat{B}; \right.$$
$$\left. \sum_{i=1}^{m} x_{i,j} = 1,\ j = 1, 2, \ldots, n;\ \frac{1}{N}\sum_{j=1}^{n} x_{i,j} - y_i \le 0,\ i = 1, 2, \ldots, m \right\},$$

where $x_{i,j} \in \{0, 1\}$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ and $y_i \in \{0, 1\}$, $i = 1, 2, \ldots, m$. The optimal solution of MAP is given by $g_{M,N}(B)$. Initially, $g_{1,n}(\hat{B}) = L_1 \sum_{j=1}^{n} AT_j$ if $FC_1 + MC_1 n \le \hat{B}$ and $-\infty$ otherwise, $\hat{B} = \alpha_1, \alpha_2, \ldots, \alpha_\beta$, $n = 1, 2, \ldots, N$. Subsequent values of $g_{m,n}(\hat{B})$ are found by the recursion

$$g_{m,n}(\hat{B}) = \max \begin{cases} g_{m-1,n}(\hat{B}), \\ \max_{t=1,2,\ldots,T}\{g_{m-1,n-t}(\hat{B} - FC_m - MC_m t) + L_m \sum_{j=n-t+1}^{n} AT_j\} \end{cases}$$

where $T = \min\{n, \lfloor \frac{\hat{B} - FC_m}{MC_m} \rfloor\}$. At each step of the recursion, either no passengers are assigned to class m, in which case the fixed cost is not assessed against the budget, or between 1 and T unassigned passengers with the highest assessed threat values are assigned to class m. Since $\lfloor \frac{\hat{B} - FC_m}{MC_m} \rfloor = N$ in the worst case, each step in the recursion can call at most N other recursions and the recursion is called at most $M\,N\,\beta$ other times, resulting in a total time bound of $O(N^2 M\,B)$ in the worst case. Storing the values of $g_{m,n}(\hat{B})$ requires space bound $O(M\,N\,B)$.

To describe the second dynamic programming algorithm, let $g_{m,n}(\hat{B})$ be defined as before, and let $f_{m,n}(\hat{B})$, $m = 1, 2, \ldots, M$, $n = 1, 2, \ldots, N$, $\hat{B} = \alpha_1, \alpha_2, \ldots, \alpha_\beta = B$ be the optimal solution over the first m classes and n passengers with budget $\hat{B}$ given that at least one passenger is assigned to class m:

$$\begin{aligned} f_{m,n}(\hat{B}) \;=\; \max &\left\{ \frac{1}{\sum_{j=1}^{N} AT_j} \sum_{i=1}^{m} \sum_{j=1}^{n-1} L_i AT_j x_{i,j} + L_m AT_n \;\middle|\; \sum_{i=1}^{m}\sum_{j=1}^{n-1} MC_i x_{i,j} + \right. \\ &\sum_{i=1}^{m-1} FC_i y_i \le \hat{B} - FC_m - MC_m; \sum_{i=1}^{m} x_{i,j} = 1,\ j = 1, 2, \ldots, n-1; \\ &\left. \frac{1}{N}\sum_{j=1}^{n-1} x_{i,j} - y_i \le 0, i = 1, 2, \ldots, m-1 \right\} \end{aligned}$$

with $x_{i,j} \in \{0, 1\}$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n-1$ and $y_i \in \{0, 1\}$, $i = 1, 2, \ldots, m-1$. The optimal solution of MAP is again given by $g_{M,N}(B)$. Initially, $f_{1,n}(\hat{B}) = g_{1,n}(\hat{B}) = L_1 \sum_{j=1}^{n} AT_j$ if $FC_1 + MC_1 n \le \hat{B}$ and $-\infty$ otherwise. Additionally, $f_{m,1}(\hat{B}) = -\infty$ for $m = 1, 2, \ldots, M$, $\hat{B} < FC_m + MC_m$, and $f_{m,1}(\hat{B}) = L_m AT_1$ for $m = 1, 2, \ldots, M$,

$\hat{B} \geq FC_m + MC_m$. Moreover, $f_{m,n}(\hat{B}) = -\infty$ if $FC_m + MC_m < \hat{B}$ for $m = 1, 2, \ldots,$ $n = 1, 2, \ldots, N$. Subsequent values of $f_{m,n}(\hat{B})$ are found by the recursion

$$f_{m,n}(\hat{B}) = \max \begin{cases} g_{m-1,n-1}(\hat{B} - FC_m - MC_m) + L_m AT_n \\ f_{m,n-1}(\hat{B} - MC_m) + L_m AT_n \}. \end{cases} \tag{3}$$

Subsequent values of $g_{m,n}(\hat{B})$ are found by the recursion

$$g_{m,n}(\hat{B}) = \max\{g_{m-1,n}(\hat{B}),\ f_{m,n}(\hat{B})\}. \tag{4}$$

At each step of (4), either no passengers are assigned to class m, in which case the fixed cost of class m is not assessed against the budget, or at least one passenger is assigned to class m by (3). This dynamic programming algorithm runs in $O(N\,M\,B)$ time. The values $g_{m,n}(\hat{B})$ and $f_{m,n}(\hat{B})$ require $O(N\,M\,B)$ space. When the dynamic programming algorithm is complete, the number of passengers assigned to each class can be determined from iterating back from $g_{M,N}(B)$ and $f_{M,N}(B)$. Once the number of passengers assigned to each class is known, the passenger partitions are determined by Lemma 1.

3 Computational Results

This section provides computational results for MAP. The results incorporate data extracted from the Official Airline Guide (OAG) for the domestic flights of a single airline carrier at distinct stations in the United States. The data provided by the OAG includes the set of flights, the number of available seats on each flight, and the departure time of each flight. It is assumed that all passengers have exactly one checked and one carry-on bag.

A total of 270 scenarios are considered, where these scenarios were designed using three passenger sets (i.e., the number of passengers), three assessed threat distributions, three types of classes, and ten budget values. The three passenger sets are defined based on data extracted from the OAG, which consider 10, 30, and 60 minute windows of time. To find the number of passengers in these time segments, passengers are assumed to arrive randomly according to a uniform distribution between 30 and 90 minutes prior to the departure time of each flight. This arrival interval is recommended by airlines for domestic flights (e.g., see www.nwa.com, www.ual.com, www.aa.com, and www.delta.com). Moreover, each flight is assumed to have an enplanement rate of 80% (i.e., the number of passengers divided by the number of available seats). The data set is chosen based on finding the largest expected number of arriving passengers in 10, 30, and 60 minute windows, resulting in 1230, 3690, and 6200 passengers, respectively. The flights associated with each group of passengers can be identified and grouped together resulting in three subsets of flights for the different scenarios.

The scenarios consider three hypothetical assessed threat value distributions based on information from the TSA. Initially, all passengers are assumed to be identical resulting in a degenerate distribution with all assessed values equal to one. This corresponds to the case when no information is known about the passengers (call this Type I). The next two assessed threat distributions are taken to be exponential distributions with means 1/8 and 1/16, respectively, that are truncated to allow only values less than one (Ross 2000). Call these two cases Type II and Type II, respectively. Note that the Type II and Type III assessed threat distributions result in approximately 80% of the passengers having assessed threat values less than 0.2 and 0.1, respectively. These assessed threat distributions model situation where few passengers are perceived to be potential risks, hence most passengers have low assessed threat values.

Table 1 contains the security screening device data used for all the scenarios. It is divided into three areas: checked baggage, passenger, and carry-on baggage screening devices. The device values are estimated using information available in the public domain (Butler and Poole 2002; Virta et al. 2003). The yearly costs are computed based on the purchase costs, the expected lifetime of the device, and the yearly maintenance costs. In Table 1, the unadjusted fixed cost FC' is the fixed cost per hour per 1000 passengers, and it is based on the yearly fixed costs divided by the hours of operation per year (360 days a year, 6 peak hours per day), normalized by the capacity.

Three cases are considered consisting of three, five, and eight classes. The three-class scenarios are motivated by the TSA CAPPS II description, while allowing all passengers are allowed to board an aircraft (TSA 2003b). The five-class and eight-class scenarios correspond to environments with passengers of varying perceived risk levels using existing security screening devices. The marginal costs, fixed costs, and security level associated with each class for all three class scenarios are summarized in Table 2. The costs given are in (United States) dollars. These are computed from the values

Table 1: Security device data

Device Category	Device Type	False Clear	FC′	MC	Units/hour
Checked Baggage	EDS	0.12	4.167	1.00	125
	Open Bag Trace (OT)	0.15	1.199	0.83	28
Passenger	Metal Detector (MD)	0.30	0.051	0.28	90
	Hand Wand Inspection (HW)	0.20	0.009	1.25	20
Carry-on Baggage	X-ray Machine (XR)	0.20	0.720	0.28	90
	Detailed Hand Search (DHS)	0.20	0	1.25	20
	Open Bag Trace/ Detailed Search (OTDS)	0.15	1.199	1.29	18

associated with each the devices in each class and the number of passengers.

The marginal costs are computed by summing the marginal costs of the devices used by the class. The fixed costs are computed by summing the fixed costs of the devices used by the class and scaling to account for the length of the time window and the number of passengers. The fixed cost associated with open trace is only assessed once, even if both types of open trace are used in the class. Furthermore, if there are M classes, then the fixed cost values are divided by M. This is done to compensate for the fact that not all passengers are screened by all devices and that some classes share security screening devices.

The security levels for these examples are the overall true alarm rate, the probability that a threat is detected given that there is a threat. A passenger who is a threat is assumed to be detected if at least one security device gives an alarm response. Additionally, it is assumed that it is equally likely for a threat to be in a checked bag, carry-on bag, or on a person. This assumption is reasonable since no data exists that suggests a distribution among these means of attack.

Each scenario is addressed using the 2GH implemented in Matlab. All the scenarios are also formulated as integer programming models (IPs) and solved using CPLEX 7.0. All the computational experiments were executed on a Pentium III 550 MHz processor with 1048 MB of RAM. Tables 3, 4, and 5 contain the 2GH and IP values for the scenarios, as well as the CPU times for the IPs. For each IP, CPLEX was halted after 170,000 CPU seconds (approximately 2 CPU days) if it had not found an optimal solution. This value is sufficiently long to give CPLEX a reasonable amount of time to solve the problems. If the IP was not solved to optimality, an asterisk (*) is listed

Table 2: Security class costs and security levels

Class	Devices	FC N=1230	FC N=3690	FC N=6200	MC	L
1	EDS, MD, XR	67.49	202.47	340.19	1.56	0.793
2	EDS, MD, HW, XR DHS	67.62	202.85	340.83	2.81	0.927
3	EDS, OT, MD, HW, XR, OTDS	93.10	279.30	469.29	4.93	0.964
1	MD, XR	18.98	56.94	95.68	0.56	0.500
2	EDS, MD, XR	121.48	364.44	612.35	1.56	0.793
3	EDS, MD, XR, DHS	121.48	364.44	612.35	2.81	0.847
4	OT, MD, HW, XR, DHS	48.70	146.10	245.47	2.64	0.917
5	EDS, OT, MD, HW, XR, OTDS	167.58	502.74	844.72	4.93	0.964
1	MD, XR	23.73	71.18	119.60	0.56	0.500
2	MD, HW, XR	24.01	72.03	121.03	1.81	0.580
3	EDS, MD, XR	151.85	455.56	765.43	1.56	0.793
4	EDS, MD, HW, XR	151.85	455.56	765.43	2.81	0.847
5	EDS, MD, XR, DHS	152.14	456.41	766.87	2.81	0.873
6	OT, MD, HW, XR, DHS	60.87	182.62	306.84	3.89	0.917
7	OT, MD, HW, XR, OTDS	81.35	244.05	410.06	3.93	0.920
8	EDS, OT, MD, HW, XR, OTDS	209.48	628.43	1055.90	4.93	0.964

as its IP value. Therefore, all the IP values listed are optimal values. All 90 three-class scenarios finished in the allotted time, whereas 87 and 72 five-class and eight-class scenarios, respectively, finished in the allotted time. The Type II and Type III solutions always have larger objective function values than the corresponding Type I solution with the same number of passengers and budget value. This suggests that models that incorporate such information about passengers are more effective than models which assume that passengers are indistinguishable. For small budget values, the Type II and Type III solutions have significantly higher objective values than the corresponding Type I solution. As the budget increases, these differences become less pronounced, which suggests that when a large budget is available, it is less critical to distinguish between passengers (since nearly all passengers can then be assigned to the class with the highest (and most costly) security level).

For all of the Type I IPs, the scenarios with indistinguishable passengers finished in under 2 CPU seconds regardless of the number of classes or the number of passengers. The Type II and Type III IPs took no less than 1.95 CPU seconds to complete. Of the 90 three-class IPs, 89 finished in less than one CPU hour, and 81 finished in less than one CPU minute. Of the 87 five-class IPs that finished, 74 finished in less than one CPU hour, and 47 finished in less than one CPU minute. Of the 72 eight-class IPs that

Table 3: Solutions for three-class scenarios

		Type I			*Type II*			*Type III*		
N	B ($)	2GH value	IP value	IP CPU time (sec)	2GH value	IP value	IP CPU time (sec)	2GH value	IP value	IP CPU time (sec)
1230	2202.47	**0.8000**	0.8000	1.33	**0.8199**	0.8199	2.39	**0.8192**	0.8192	2.16
1230	2646.91	**0.8388**	0.8388	0.98	**0.8894**	0.8894	3.27	**0.8868**	0.8868	3.14
1230	3091.36	**0.8774**	0.8774	1.00	0.9172	0.9176	18.9	0.9161	0.9161	32.2
1230	3535.80	**0.9162**	0.9162	0.97	0.9266	0.9346	3.27	0.9265	0.9335	5.45
1230	3980.25	**0.9312**	0.9312	1.03	0.9402	0.9458	2.70	0.9399	0.9447	3.16
1230	4424.69	**0.9375**	0.9375	1.02	0.9511	0.9533	2.83	0.9504	0.9525	3.05
1230	4869.14	**0.9438**	0.9438	0.97	0.9574	0.9583	2.92	0.9569	0.9578	2.94
1230	5313.58	**0.9501**	0.9501	1.02	0.9612	0.9616	2.94	0.9609	0.9612	2.92
1230	5758.03	**0.9564**	0.9564	0.98	0.9632	0.9633	2.28	0.9631	0.9632	2.38
1230	6202.47	**0.9627**	0.9627	0.98	**0.9640**	0.9640	1.95	**0.9640**	0.9640	2.17
3690	6407.41	**0.7942**	0.7942	1.06	**0.7997**	0.7997	8.78	**0.7998**	0.7998	8.52
3690	7762.97	**0.8336**	0.8336	1.00	**0.8827**	0.8827	9.59	**0.8819**	0.8819	13.1
3690	9118.52	**0.8730**	0.8730	1.00	0.9147	0.9158	8.95	0.9143	0.9155	9.37
3690	10474.08	**0.9124**	0.9124	0.98	0.9262	0.9338	8.98	0.9262	0.9334	12.2
3690	11829.63	**0.9310**	0.9310	1.00	0.9399	0.9453	9.63	0.9398	0.9448	8.84
3690	13185.19	**0.9374**	0.9374	1.03	0.9508	0.9529	8.75	0.9506	0.9526	10.1
3690	14540.74	**0.9438**	0.9438	1.03	0.9572	0.9581	10.6	0.9571	0.9580	10.4
3690	15896.30	**0.9503**	0.9503	1.00	0.9612	0.9615	8.08	0.9611	0.9614	10.5
3690	17251.85	**0.9567**	0.9567	0.97	0.9633	0.9633	9.83	0.9632	0.9633	8.63
3690	18607.41	**0.9631**	0.9631	0.98	**0.9640**	0.9640	8.41	**0.9640**	0.9640	8.19
6200	10720.57	**0.7935**	0.7935	1.05	**0.7958**	0.7958	33.9	**0.7960**	0.7960	7359
6200	13042.79	**0.8336**	0.8336	0.95	**0.8822**	0.8822	47.8	**0.8814**	0.8814	105
6200	15365.01	**0.8738**	0.8738	1.02	**0.9148**	0.9148	73.1	**0.9145**	0.9145	75.2
6200	17687.24	**0.9139**	0.9139	1.00	0.9263	0.9333	226.5	0.9263	0.9330	45.2
6200	20009.46	**0.9310**	0.9310	1.03	0.9400	0.9451	67.0	0.9399	0.9447	35.6
6200	22331.68	**0.9376**	0.9376	1.03	0.9509	0.9529	61.3	0.9506	0.9527	30.8
6200	24653.90	**0.9441**	0.9441	1.02	0.9573	0.9582	67.3	0.9572	0.9581	33.9
6200	26976.13	**0.9507**	0.9507	0.98	0.9613	0.9616	54.3	0.9612	0.9615	31.6
6200	29298.35	**0.9572**	0.9572	1.05	0.9633	0.9634	61.0	0.9633	0.9634	24.0
6200	31620.57	**0.9637**	0.9637	1.02	**0.9640**	0.9640	31.2	**0.9640**	0.9640	12.9

Table 4: Solutions for five-class scenarios

		Type I			*Type II*			*Type III*		
		2GH	IP	IP CPU	2GH	IP	IP CPU	2GH	IP	IP CPU
N	B ($)	value	value	time (sec)	value	value	time (sec)	value	value	time (sec)
1230	800	**0.5068**	0.5068	1.19	**0.5326**	0.5326	10648	**0.5328**	0.5328	16884
1230	1400	**0.6358**	0.6358	1.19	**0.7534**	0.7534	86201	**0.7456**	0.7456	79460
1230	2000	**0.7787**	0.7787	1.19	**0.8524**	0.8524	3214	**0.8471**	0.8471	19554
1230	2600	**0.8407**	0.8407	1.19	**0.9005**	0.9005	139	0.8982	0.8983	43.7
1230	3200	**0.8980**	0.8980	1.19	0.9166	0.9222	190	0.9165	0.9206	16.7
1230	3800	**0.9226**	0.9226	1.19	0.9344	0.9406	19.6	0.9340	0.9396	8.19
1230	4400	**0.9326**	0.9326	1.19	0.9503	0.9522	17.6	0.9494	0.9513	26.8
1230	5000	**0.9426**	0.9426	1.19	0.9585	0.9591	1555	0.9580	0.9585	6.7
1230	5600	**0.9526**	0.9526	1.19	0.9626	0.9627	286	0.9624	0.9625	5.6
1230	6200	**0.9626**	0.9626	1.19	**0.9640**	0.9640	6.42	**0.9640**	0.9640	3.14
3690	2400	**0.5070**	0.5070	1.19	**0.5347**	0.5347	6618	**0.5350**	0.5350	3597
3690	4200	**0.6359**	0.6359	1.17	**0.7515**	0.7515	159	**0.7485**	0.7485	469
3690	6000	**0.7789**	0.7789	1.19	**0.8504**	0.8504	240	**0.8492**	0.8492	808
3690	7800	**0.8407**	0.8407	1.19	**0.8997**	0.8997	157692	**0.8988**	0.8988	84619
3690	9600	**0.8982**	0.8982	1.19	0.9166	0.9219	158	0.9165	0.9213	519
3690	11400	**0.9226**	0.9226	1.19	0.9345	0.9404	51.1	0.9344	0.9398	126
3690	13200	**0.9326**	0.9326	1.19	0.9500	0.9519	130	0.9497	0.9515	109
3690	15000	**0.9426**	0.9426	1.19	0.9583	0.9589	1310	0.9582	0.9587	1029
3690	16800	**0.9526**	0.9526	1.19	**0.9626**	0.9626	29.9	**0.9625**	0.9625	35.8
3690	18600	**0.9626**	0.9626	1.19	**0.9640**	0.9640	14.9	**0.9640**	0.9640	10.3
6200	4000	**0.5060**	0.5060	1.19	**0.5307**	0.5307	6934	**0.5310**	0.5310	16624
6200	7033.33	**0.6348**	0.6348	1.19	0.7492	*		**0.7462**	0.7462	1165
6200	10066.67	**0.7782**	0.7782	1.19	0.8488	*		**0.8478**	0.8478	4233
6200	13100	**0.8406**	0.8406	1.17	**0.8991**	0.8991	78617	0.8985	*	
6200	16133.33	**0.8984**	0.8984	1.19	0.9166	0.9216	153615	0.9165	0.9211	353.8
6200	19166.67	**0.9227**	0.9227	1.19	0.9346	0.9402	117	0.9344	0.9397	128.2
6200	22200	**0.9327**	0.9327	1.19	0.9499	0.9517	91.1	0.9497	0.9514	112.2
6200	25233.33	**0.9427**	0.9427	1.19	0.9583	0.9588	85.3	0.9582	0.9586	114.8
6200	28266.67	**0.9528**	0.9528	1.19	0.9626	0.9626	86.8	**0.9625**	0.9625	86.6
6200	31300	**0.9628**	0.9628	1.19	**0.9640**	0.9640	35.7	**0.9640**	0.9640	27.8

Table 5: Solutions for eight-class scenarios

		Type I			*Type II*			*Type III*		
		2GH	IP	IP CPU	2GH	IP	IP CPU	2GH	IP	IP CPU
N	B ($)	value	value	time (sec)	value	value	time (sec)	value	value	time (sec)
1230	800	**0.5033**	0.5033	1.19	0.5133	*		0.5134	*	
1230	1400	**0.6274**	0.6274	1.19	**0.7364**	0.7364	114570	**0.7320**	0.7320	14505
1230	2000	**0.7704**	0.7704	1.19	0.7930	0.8135	740.9	0.7921	0.8105	2053.9
1230	2600	**0.8133**	0.8133	1.19	0.8503	0.8611	132.2	0.8487	0.8578	197.4
1230	3200	**0.8438**	0.8438	1.17	0.8891	0.8968	2074.6	0.8863	0.8927	59.03
1230	3800	**0.8773**	0.8773	1.19	0.9205	0.9233	133.4	0.9173	0.9203	46.9
1230	4400	**0.8972**	0.8972	1.19	0.9405	0.9417	4547.8	0.9385	0.9397	276.7
1230	5000	**0.9200**	0.9200	1.19	0.9534	0.9538	21.64	0.9521	0.9525	47.91
1230	5600	**0.9370**	0.9370	1.19	**0.9606**	0.9606	1075.4	**0.9602**	0.9602	274.4
1230	6200	**0.9590**	0.9590	1.17	**0.9639**	0.9639	7.28	**0.9639**	0.9639	8.19
3690	2500	**0.5060**	0.5060	1.19	0.5305	*		0.5308	*	
3690	4300	**0.6355**	0.6355	1.19	**0.7410**	0.7410	128020	0.7401	*	
3690	6100	**0.7784**	0.7784	1.19	0.7947	*		0.7932	0.8153	83785
3690	7900	**0.8147**	0.8147	1.19	0.8528	*		0.8521	0.8608	4852
3690	9700	**0.8456**	0.8456	1.19	0.8908	0.8974	732.1	0.8896	0.8958	102.3
3690	11500	**0.8784**	0.8784	1.19	0.9207	*		0.9197	0.9225	3945
3690	13300	**0.8984**	0.8984	1.19	0.9406	*		0.9401	0.9412	1793.5
3690	15100	**0.9200**	0.9200	1.19	0.9535	0.9538	2730.6	0.9531	0.9533	2285.1
3690	16900	**0.9382**	0.9382	1.19	**0.9608**	0.9608	107475	0.9606	*	
3690	18700	**0.9603**	0.9603	1.19	**0.9640**	0.9640	26.05	**0.9640**	0.9640	32.52
6200	4000.00	**0.5030**	0.5030	1.19	**0.5126**	0.5126	1445.6	0.5126	*	
6200	7055.56	**0.6275**	0.6275	1.19	**0.7338**	0.7338	799.2	**0.7328**	0.7328	989.8
6200	10111.11	**0.7719**	0.7719	1.19	0.7922	*		0.7922	*	
6200	13166.67	**0.8138**	0.8138	1.19	0.8510	*		0.8502	*	
6200	16222.22	**0.8448**	0.8448	1.19	0.8895	*		0.8882	0.8944	8634.0
6200	19277.78	**0.8781**	0.8781	1.19	0.9197	0.9225	7918.8	0.9189	*	
6200	22333.33	**0.8983**	0.8983	1.19	0.9400	0.9411	3596.7	0.9397	0.9407	6889.5
6200	25388.89	**0.9200**	0.9200	1.17	0.9532	*		0.9529	0.9532	1272.5
6200	28444.44	**0.9386**	0.9386	1.17	**0.9608**	0.9608	14360	**0.9606**	0.9606	1321.0
6200	31500.00	**0.9609**	0.9609	1.19	**0.9640**	0.9640	57.97	**0.9640**	0.9640	108.3

Table 6: Average 2GH CPU times (seconds)

M	N	Type I	Type II	Type III
3	1230	0.0314	0.0328	0.0343
3	3690	0.0954	0.0892	0.0923
3	6200	0.140	0.140	0.148
5	1230	0.0640	0.0672	0.0625
5	3690	0.186	0.183	0.181
5	6200	0.303	0.305	0.308
8	1230	0.144	0.147	0.148
8	3690	0.438	0.434	0.436
8	6200	0.719	0.727	0.727

finished, 60 finished in less than one CPU hour and 39 finished in less than one CPU minute. In general, the IPs with relatively small budget values took longer to solve, while the IPs that had larger budget values took the least amount of computing time, particularly when the budgets were large enough to assign nearly all passengers to the class with the highest security level.

In all 90 Type I scenarios and in 34 of 78 (finished) Type II and 33 of 81 (finished) Type III scenarios, the 2GH solution is identical to the optimal solution. The scenarios where the 2GH value and the IP value matched are in boldface in Tables 3, 4, and 5. Typically, scenarios with either small or large budget values had the same 2GH and IP values, where the optimal solutions used two classes. When the 2GH and IP values were different, the IP solutions never used more than three classes.

Table 6 contains the average CPU time to execute 2GH for each of the sets of scenarios. All of the average 2GH CPU times for a given number of classes, number of passengers, and assessed threat distribution are less than 0.73 CPU seconds, with the 2GH never taking longer than 1.3 CPU seconds to identify a single approximate solution. Moreover, for a given number of classes and passengers, the 2GH CPU time remained approximately the same across the Type I, II, and III assessed threat distributions. This can be contrasted to the corresponding IPs, which were solved quickly for Type I scenarios, and generally took longer for the Type II and III scenarios.

The quality of the 2GH solutions can be measured by the *relative effectiveness measure*

$$\Gamma = \frac{z^h - z_0}{z - z_0}.$$

where z^h is the 2GH solution objective function value, z_0 is the worst possible feasible solution objective function value, and z is the optimal solution objective function value. If the relative effectiveness measure is one, then 2GH obtains an optimal solution, while if is zero, then 2GH obtains the worst possible feasible solution. Figures 1, 2, and 3 depict the relative effectiveness measure of the 2GH solutions for the the three-class, five-class, and eight-class scenarios, respectively. Several relative effective measure values could not be computed for the five-class and eight-class solutions since their corresponding IPs did not finish in the allotted time, hence several points are missing from Figures 2 and 3. Note that the Type I scenarios are omitted since the relative effectiveness measures for these scenarios are all one. The Type II and Type III scenarios often have relative effectiveness measures of one when the budget was either small or large. The relative effectiveness measure is never less than 0.943 for the three-class scenarios, 0.986 for the five-class scenarios, and 0.930 for the eight-class scenarios, which indicates that the 2GH solutions are close to the optimal values when using just two classes.

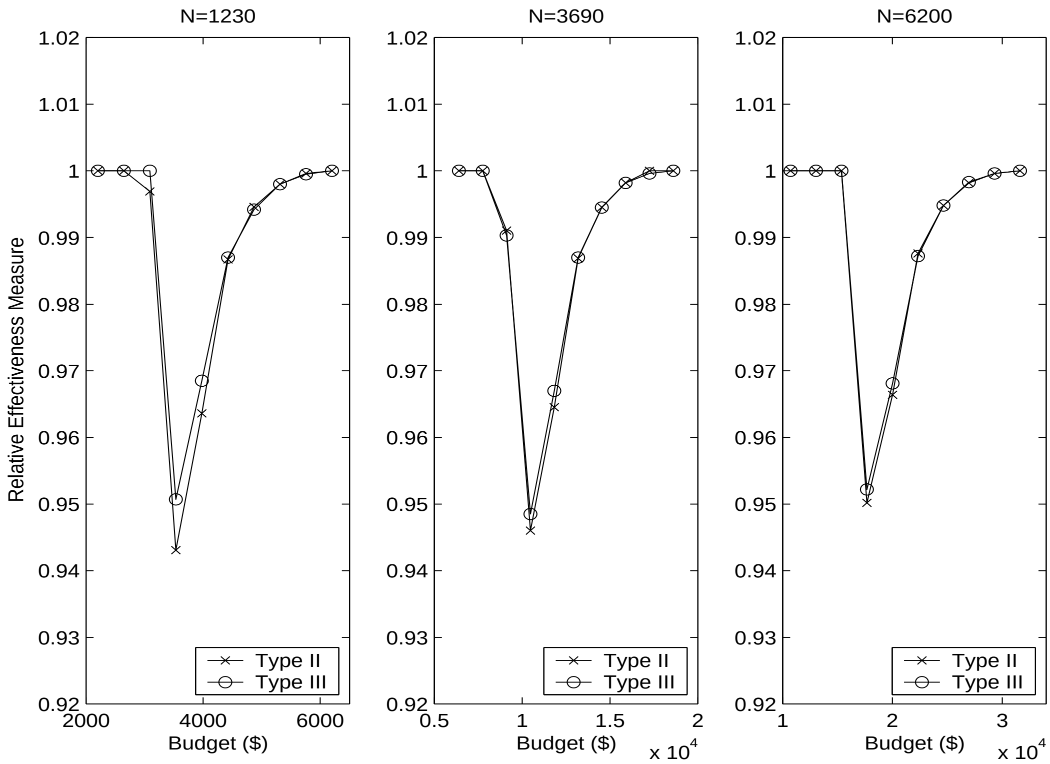

Figure 1: Relative effectiveness measure for three-class examples

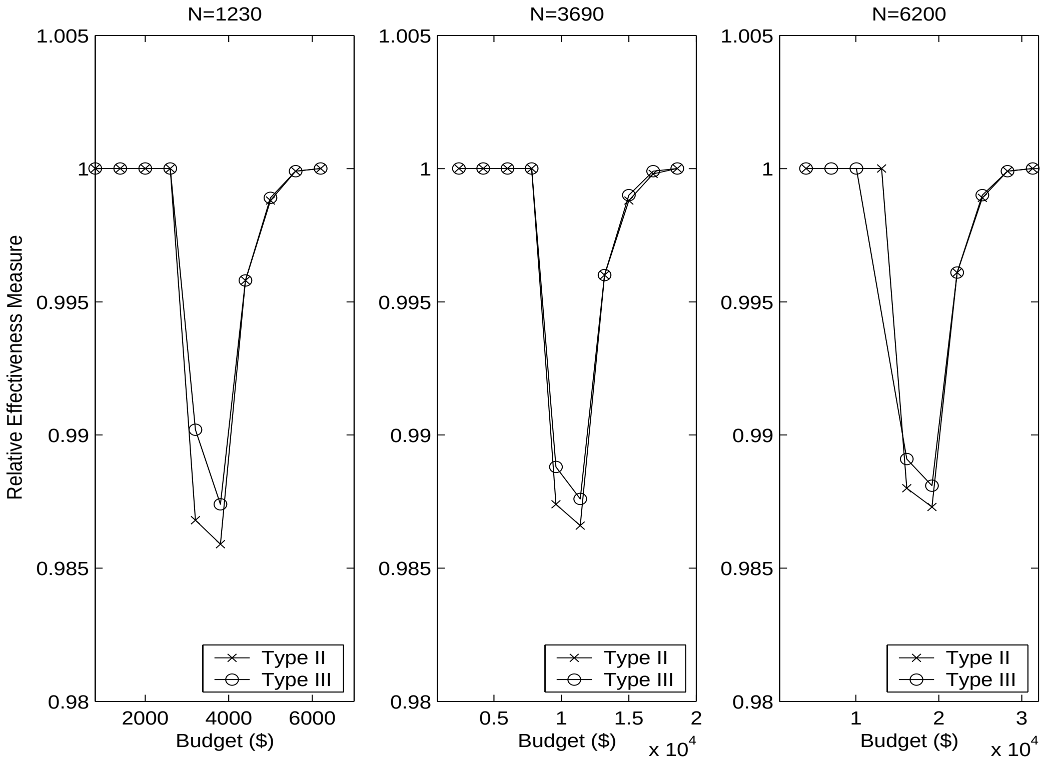

Figure 2: Relative effectiveness measure for five-class examples

4 Conclusions

Passenger and baggage screening is a critical component of any aviation security system operation. This paper introduces the Multilevel Allocation Problem, a framework to model passenger and baggage screening systems. MAP models each security class in terms of its marginal and fixed costs, and security level, where passenger assignment and budget constraints must be satisfied. MAP is formulated as an integer programming model. MAP is shown to be NP-hard, and two dynamic programming algorithms that solve MAP in $O(N^2MB)$ and $O(NMB)$ time and $O(NMB)$ space are presented. Furthermore, the Two Class Greedy Heuristic (2GH) is introduced to obtain approximate solutions to MAP in $O(M^2N)$ time with an additional $O(N \log N)$ time for the initial sorting of passengers.

Data extracted from the Official Airline Guide (OAG) and hypothetical data based on information provided by the TSA was used to construct a total of 270 scenarios for MAP. The optimal (IP) and 2GH objective function values were computed for each of these scenarios, where twenty-one of the IPs did not terminate with an optimal

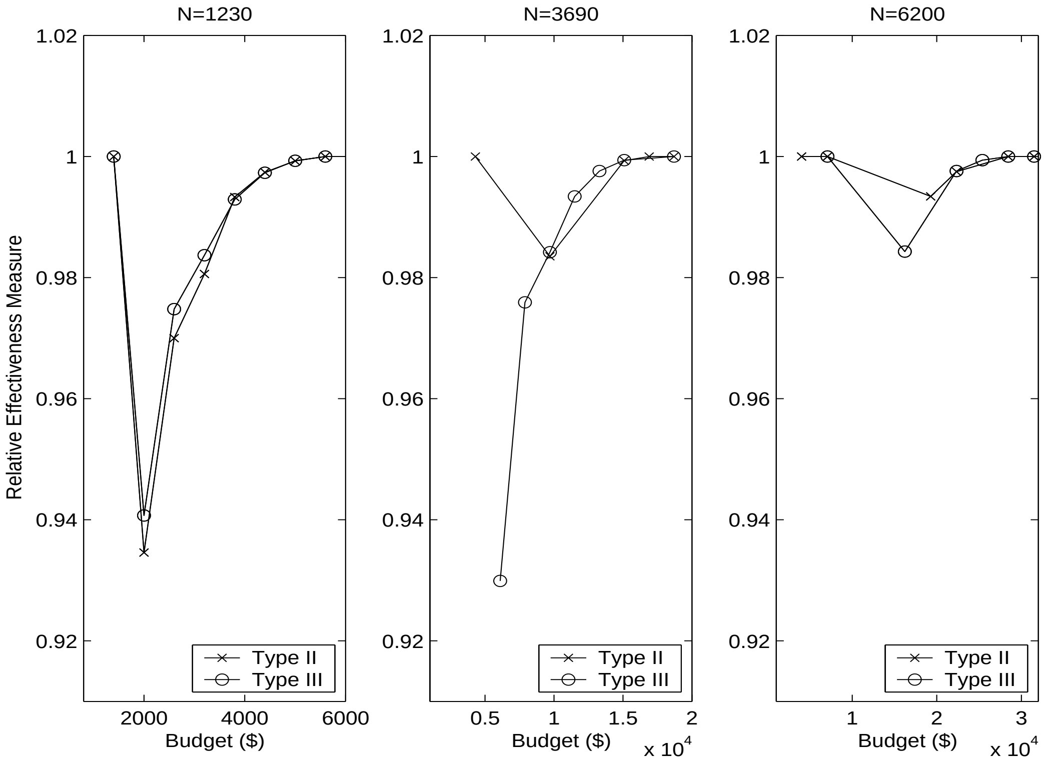

Figure 3: Relative effectiveness measure for eight-class examples

solution in the allotted computing time (three of which are five-class scenarios and eighteen of which are eight-class scenarios). In many cases, 2GH obtained the optimal solution value, including all of the Type I scenarios, 34 of the 78 (completed) Type II scenarios, and 33 of the 81 (completed) Type III scenarios. The relative effectiveness measures for the 2GH are greater than 0.940 in the three-class scenarios, 0.986 for the five-class scenarios, and greater than 0.930 for the eight-class scenarios. One limitation of the 2GH is that it obtains solutions that use no more than two classes. However, the optimal solutions for all of the Type I scenarios use two classes, and none of the completed optimal solutions for the Type II and Type III scenarios use more than three classes. This suggests that even when many classes are available, it may be more effective (from a security standpoint) to use fewer classes. This implication is desirable on a practical level since security personnel need to be trained to be fluent with fewer security procedures. Therefore, any security system design that is simple to implement and easy operate is of added value.

There are several possible directions to extend the research results presented. First, minimizing the overall false alarm rate is of interest because the majority of passengers are not threats, and high false alarm rates are costly for the airlines. This objective suggests the development of the Multicriteria MAP, which simultaneously maximizes the overall true alarm rate and minimizes the overall false alarm rate. Second, the Dynamic MAP, a variation of MAP in which passengers arrive dynamically can be formulated, and algorithms for effectively assigning passengers to classes in real-time would need to be obtained. Another model for passenger and baggage prescreening considers classes after the budget has been used to purchase security devices and hire security personnel. In this case, each class is defined in terms of its associated devices and their capacities, though costs are not included in such a model and the objective is to assign passengers to classes so that the total security is maximized while all devices are operating within their capacities. Work is in progress to design algorithms and heuristics for all these models.

Acknowledgements

This research was supported in part by the National Science Foundation (DMI-0114046, DMI-0114499) and the Air Force Office of Scientific Research (FA9550-04-1-0110). The first author was supported in part by the program in Arms Control, Disarmament, and International Security at the University of Illinois in Urbana-Champaign. The computational work was done in the Simulation and Optimization Laboratory housed within the Department of Mechanical and Industrial Engineering at the University of Illinois. The authors would like to thank Dr. Lyle Malotky, Mr. Michael McCormick and Dr. John Nestor of the Transportation Security Administration, Department of Homeland Security, for their guidance and feedback on the research results reported in this paper.

References

A. Barnett, “The Worst Day Ever,” *OR/MS Today* 28, 28-31 (2001).

A. Barnett, “CAPPS II: The Foundation of Aviation Security?” *Risk Analysis*, to appear (2004).

A. Barnett, M. Abraham, and V. Schimmel, “Airline Safety: Some Empirical Findings," *Management Science* 25, 1045-1056 (1979).

A. Barnett, and M. K. Higgins, "Airline Safety: The Last Decade," *Management Science* 35, 1-21 (1989).

A. Barnett, R. W. Shumsky, M. Hansen, A. Odoni, and G. Gosling, "Safe at Home? An Experiment in Domestic Airline Security," *Operations Research* 49, 181-195 (2001).

R. A. Brualdi, *Introductory Combinatorics* (3rd ed.), Prentice Hall, Upper Saddle River, NJ (1999).

V. Butler and R. W. Poole Jr., "Rethinking Checked-Baggage Screening," *Reason Public Policy Institite, Policy Study No. 297.*, Los Angeles, CA (2002).

S. Chakrabarti and A. Strauss, "Carnival Booth: An Algorithm for Defeating the Computer-Aided Passenger Screening System." *First Monday* 7, (2002), (available at www.firstmonday.org).

FAA, "FAA Forecast Continued Drop in Air Traffic This Year, Strong Recovery in 2003," Report, Federal Aviation Administration, Washington, D. C. (2002).

Hall and DeLollis, "Plan to collect flier data canceled." *USA Today*, 1 (July 14, 2004).

G. H. Hardy, J. E. Littlewood, and G. Polya, *Inequalities.* Cambridge University Press, Cambridge, England (1934).

J. Hilkevitch, "London airport offers security lessons for U.S." Chicago Tribune, 1 (February 17, 2003).

S. H. Jacobson, J. M. Bowman, and J. E. Kobza, "Modeling and Analyzing the Performance of Aviation Security Systems using Baggage Value Performance Measures," *IMA Journal of Management Mathematics* 12, 3-22 (2001).

S. H. Jacobson, L. A. McLay, J. E. Kobza, and J. M. Bowman, "Modeling and Analyzing Multiple Station Baggage Screening Security System Performance." *Naval Research Logistics*, to appear (2005).

S. H. Jacobson, J. E. Virta, J. M. Bowman, J. E. Kobza, and J. J., Nestor, "Modeling Aviation Baggage Screening Security Systems: A Case Study." *IIE Transactions* 35, 259-269 (2003).

G. S. Lueker, "Two NP-Complete Problems in Nonnegative Integer Programming." Technical Report No. 178, Computer Science Laboratory, Princeton University, Princeton, NJ (1975).

L. O. Malotky, "Introduction to the Sixth International Civil Aviation Security Conference," *Journal of Testing and Evaluation* 22, 235-237 (1994).

S. Martonosi, and A. I. Barnett, "Security Profiling of Airline Passengers: How Effective Would It Be?" presented at the INFORMS National Meeting, Atlanta, GA (2003).

L. A. McLay, and S. H. Jacobson, "Integer Knapsack Problems with Set-up Weights," Technical Report, Department of Mechanical and Industrial Engineering, University of Illinois, Urbana, IL (2003).

K. M. Mead, "Key Challenges Facing the Transportation Security Administration," Office of the Inspector General Report Number CC-2002-180, Department of Transportation, Washington, D. C. (2002).

K. M. Mead, "Aviation Security Costs, Transportation Security Administration," Office of the Inspector General Report Number CC-2003-066, Department of Transportation, Washington, D. C. (2003).

National Research Council, *Airline Passenger Security Screening*, National Academy Press Publication NMAB-482-1, Washington, D.C. (1996).

R. W. Poole Jr. and G. Passantino, "A Risk-Based Airport Security Policy," *Reason Public Policy Institute, Policy Study No. 308.*, Los Angeles, CA (2003).

S. M. Ross, *Introduction to Probability Models* (7th ed.) Harcourt Academic Press, San Diego, CA, 591-595 (2000).

Singer, "Life after death for CAPPS II?." *Wired News*, (2004) (available at www.wired.com).

TSA, Personal Communications, TSA Briefing, October 3, 2003, Arlington, VA (2003a).

TSA, "TSA's CAPPS II Gives Equal Weight to Privacy, Security," Press Release, United States Department of Homeland Security, Washington, D. C. (2003b).

J. E. Virta, S. H. Jacobson, and J. E. Kobza, “Outgoing Selectee Rates at Hub Airports,” *Reliability Engineering and System Safety* 76, 155-165 (2002).

J. E. Virta, S. H. Jacobson, and J. E. Kobza, “Analyzing the Cost of Screening Selectee and Non-Selectee Baggage,” *Risk Analysis* 23, 897-908 (2003).

Appendix

Proof of Theorem 1:

MAP is shown to be NP-hard by constructing a polynomial Turing reduction from the Integer Knapsack Problem (IKP) (Lueker 1975). The objective of IKP is to choose the number of each item type to add to the knapsack such that the knapsack is within its capacity and the total profit is maximized. IKP is first formally stated.

Integer Knapsack Problem (IKP)
Given a total of m item types, with weights $w_1, w_2, \ldots, w_m$, profits, $p_1, p_2, \ldots, p_m$, and knapsack capacity, c, assign nonnegative integers $u_1, u_2, \ldots, u_m$ to the m objects such that $\sum_{i=1}^{m} u_i w_i \le c$ and $\sum_{i=1}^{m} u_i p_i$ is maximized.

Given an arbitrary instance of the IKP, construct a particular instance of MAP. Without loss of generality, assume that $w_1 \le w_i$ for $i = 2, 3, \ldots, m$. For simplification, let $\bar{p} = \sum_{i=1}^{m} p_i$. Then any arbitrary instance of IKP can be formulated as a particular instance of MAP as follows: $M = m + 1$ classes, $N = \lfloor c/w_1 \rfloor$ passengers, budget $B = c$, the security level of each class $L_i = p_i/\bar{p}$ for $i = 1, 2, \ldots, m$, $L_{m+1} = 0$, marginal costs $MC_i = w_i$ for $i = 1, 2, \ldots, m$, $MC_{m+1} = 0$, $FC_i = 0$ for $i = 1, 2, \ldots, m + 1$, and $AT_j = 1/N$ for $j = 1, 2, \ldots, N$ (i.e., passengers are indistinguishable). Note that $0 \le L_i \le 1$ for $i = 1, 2, \ldots, m + 1$. This reduction requires $O(m + \lfloor c/w_1 \rfloor)$ time and $O(m)$ space.

To show that an optimal solution to MAP maps to an optimal solution of IKP, suppose that $A_1^*, A_2^*, \ldots, A_{m+1}^*$ is an optimal solution to MAP. Note that $A_1^*, A_2^*, \ldots, A_{m+1}^*$ are mutually exclusive and exhaustive subsets of $\{1,2,\ldots,N\}$, the budget constraint is satisfied (i.e., $\sum_{i=1}^{m+1} MC_i |A_i^*| \le B$), and the total security is maximized (i.e., $\sum_{i=1}^{m+1} L_i |A_i^*|$). The claim is that $u_1^* = |A_1^*|, u_2^* = |A_2^*|, \ldots, u_m^* = |A_m^*|$ is an optimal solution to IKP. Since $w_i = MC_i$ and $p_i = (1/\bar{p})\, L_i$, $i = 1, 2, \ldots, m$, $MC_{m+1} = L_{m+1} = 0$, and $c = B$, then $u_2^*, u_1^*, \ldots, u_m^*$ is a feasible solution to IKP (i.e., $\sum_{i=1}^{m} w_i u_i^* \le c$) with objective function value $\sum_{i=1}^{m+1} L_i|A_i^*| = (1/\bar{p})\sum_{i=1}^{m} p_i u_i^*$.

Suppose that $\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m$ is an optimal solution to IKP such that $\sum_{i=1}^{m} p_i\bar{u}_i > \sum_{i=1}^{m} p_i u_i^*$. Then $|\bar{A}_1| = \bar{u}_1, |\bar{A}_2| = \bar{u}_2, \ldots, |\bar{A}_m| = \bar{u}_m$ and $|\bar{A}_{m+1}| = N - \sum_{i=1}^{m} \bar{u}_i$ is a feasible solution to MAP. To see this, note that $\sum_{i=1}^{M} |\bar{A}_i| = N$, hence, mutually exclusive and exhaustive passenger assignment subsets exist. Since $\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m$ is a feasible solution to IKP, then

$$\sum_{i=1}^{m+1} MC_i|\bar{A}_i| = \sum_{i=1}^{m} MC_i|\bar{A}_i| = \sum_{i=1}^{m} w_i\bar{u}_i \leq c = B,$$

and $\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_{m+1}$ is a feasible solution to MAP. Since $\bar{p}\sum_{i=1}^{m+1} L_i|\bar{A}_i| = \sum_{i=1}^{m} p_i\bar{u}_i$, then

$$\sum_{i=1}^{m} p_i u_i^* = \bar{p}\sum_{i=1}^{m+1} L_i|A_i^*| \geq P\sum_{i=1}^{m+1} L_i|\bar{A}_i| = \sum_{i=1}^{m} p_i\bar{u}_i.$$

and $\sum_{i=1}^{m} p_i\bar{u}_i > \sum_{i=1}^{m} p_i u_i^*$, which is a contradiction. Therefore, $u_1^* = |A_1^*|, u_2^* = |A_2^*|, \ldots, u_m^* = |A_m^*|$ is an optimal solution to IKP. □

Proof of Lemma 1:

Let l_k, $k = 1, 2, \ldots, N$ be the security level of the class passenger k is assigned to and $n_i = |\{k|l_k = L_i\}|$ for $i = 1, 2, \ldots, M$ be the number of passengers assign to class i. Since N is in general much larger than M, each security level value may be duplicated many times. For example, if there are fifty passengers and two classes with $L_1 = 0.5$ and $L_2 = 0.9$, and the first forty passengers are assigned to class 1, then $l_1 = l_2 = ... = l_{40} = 0.5$ and $l_{41} = l_{42} = ... = l_{50} = 0.9$. The MAP objective with passenger assignments $A_1, A_2, \ldots, A_M$ is

$$\frac{1}{\sum_{j=1}^{N} AT_j}\sum_{i=1}^{M}\sum_{j\in A_i} L_i AT_j = \frac{1}{\sum_{j=1}^{N} AT_j}\sum_{j=1}^{N} l_j AT_j.$$

The proof follows from Hardy's Lemma (Hardy et al. 1934), which states that if $x_1 \leq x_2 \leq \ldots \leq x_n$ and $y_1 \leq y_2 \leq \ldots \leq y_n$ are sequences of numbers, then

$$\max_{(i_1,i_2,\ldots,i_n)\in P}\sum_{j=1}^{n} x_{i_j} y_j = \sum_{j=1}^{n} x_j y_j,$$

where P is the set of all permutations of the integers $(1, 2, \ldots, n)$. □

Proof of Lemma 2:

An enumeration algorithm considers all possible passenger assignments and returns the feasible solution with the best objective function value. Since each of the N passengers can be assigned to each of the M classes, then this enumeration algorithm considers M^N total passenger assignments.

However, consider a modified enumeration algorithm for solving MAP which uses Lemma 1 to limit the number of passenger assignments considered. Lemma 1 indicates how to assign individual passengers to classes once the number of passengers assigned to each class is known. Therefore, the total number of passenger assignments to consider is a "combination of multisets" (Brualdi 1999),

$$\begin{pmatrix} N+M-1 \\ N \end{pmatrix} = \frac{(N+M-1)(N+M-2)\cdots(N+1)}{(M-1)!} = O(N^{M-1}).$$

Since O(N) time is required to assign individual passengers to classes and to determine the objective function value, then this modified enumeration algorithm requires O(N^M) time, hence, is polynomial in terms of the number of passengers N for a fixed number of classes M. □

Proof of Lemma 3:

For $M = 2$ classes, 2GH executes in O(N) time with an additional O($N \log N$) time for the initial sort. Without loss of generality, it is assumed that $L_1 \leq L_2$ and that at least one class is in C^{N^+}, hence a feasible solution exists. In order to determine whether 2GH finds an optimal solution, two cases are considered: 1) class 2 is in C^{N^+} and 2) class 1 is in C^{N^+} and class 2 is in C^{N^-}.

When class 2 is in C^{N^+}, 2GH finds the optimal solution when it considers the best solution using one class. In this case, the heuristic value is identical to the optimal objective function value of L_2.

When class 1 is in C^{N^+} and class 2 is in C^{N^-}, then the best solution using exactly one class that 2GH obtains is L_1. Then, 2GH considers solutions using both classes, which leads to solving the following linear equations,

$$MC_1 n_1 + MC_2 n_2 = B - FC_1 - FC_2,$$

$$n_1 + n_2 = N.$$

If $n_2 \leq 0$, then there is no feasible solution using both classes and L_1 is the optimal solution value. If $0 < n_2 < N$, an integer feasible solution is created by adding $\lceil n_1 \rceil$ and $\lfloor n_2 \rfloor$ passengers to class 1 and class 2, respectively. Therefore, $\lceil n_1 \rceil$ and $\lfloor n_2 \rfloor$ are the optimal number of passengers to assign to class 1 and class 2, respectively.

Secondly, Lemma 1 indicates that the optimal way to assign $\lfloor n_2 \rfloor$ passengers to the class with the higher security level is to choose the $\lfloor n_2 \rfloor$ passengers with the highest assessed threat values, which is how the 2GH assigns passengers to class 2. Since both 2GH and the optimal solutions assign the same set of passengers to class 2, then the solutions are identical. □

Proof of Theorem 2:

When all of the passengers have identical assessed threat values, then the particular instance of MAP can be formulated as an instance of the k-item Integer Knapsack Problem with Set-up Weights (kIKPSW) (McLay and Jacobson 2003).

The Integer Knapsack Problem with Set-up Weights (kIKPSW)
Given a total of m types of items, with each type of item having positive integer weight $w_1, w_2, \ldots, w_m$, nonnegative integer set-up weight $s_1, s_2, \ldots, s_m$, nonnegative value $v_1, v_2, \ldots, v_m$, knapsack capacity c, and cardinality k, find nonnegative integers $x_1, x_2, \ldots, x_m$ such that $\sum_{i=1}^{m} x_i = k$, $\sum_{i=1}^{m} w_i x_i + \sum_{i:x_i>0} s_i \leq c$, and $\sum_{i=1}^{m} v_i x_i$ is maximized.

The particular instance of MAP is equivalent to kIKPSW with $m = M$, $k = N$, $c = B$, $w_i = MC_i,\ i = 1, 2, \ldots, m$, $s_i = FC_i,\ i = 1, 2, \ldots, m$, and $v_i = L_i / \sum_{j=1}^{N} AT_j,\ i = 1, 2, \ldots, m$.

The kIKPSW Greedy Heuristic, $H_k^{1/2}$, finds the best solution using no more than two item types, where such solutions are always within at least 1/2 of the optimal solution value (McLay and Jacobson 2003). 2GH is the same as $H_k^{1/2}$ when the passengers have identical assessed threat values. □