

Mathematical Integer Linear Programming Model for Optimal Placement of Security Controls in Tourism Information Systems

Gadir Mammadov¹  and Najaf Mammadzada² 

¹Department of Tourism and Hospitality, Western Caspian University, Baku, Azerbaijan

²Department of Cybersecurity and Computer Engineering, Baku Engineering University, Baku, Azerbaijan

namammadzade@beu.edu.az

Received: 03.08.2026

Accepted: 26.08.2026

Published: .09.10.2026

<https://doi.org/10.54414/XCES2356>

Copyright: © 2026 by the authors.
Licensee: Journal of Smart Technologies and Computational Systems, Western Caspian University, Baku, Azerbaijan. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International License (CC BY 4.0).

Abstract: Tourism information systems process large volumes of sensitive personal, financial, and operational data, which makes them attractive targets for a wide range of cyber attacks. Putting different types of security measures in the best possible places when money is limited is a classic problem in combinatorial optimization. However, the tourism field still mainly relies on qualitative risk charts and expert opinions for identifying risks. The goal is to pick up the best possible security coverage counting risks, while also next limits on money, compatibility issues, claiming backup systems, and minimum security requirements. Risk weights are deliberately predicated on how important an asset is and how likely it is to be attacked. Coverage coefficients exhibit how much each control assists in defending a specific part of a system. The formula is simple, easy to deduce for people without specialized knowledge, and can be solved to the best possible solution using modern branch-and-cut methods. A detailed numerical case study, applied to a mid-sized mountain tourism business that holds twelve system parts and eight types of control, communicates how the method applies to real situations. Analysis of the programme's value function shows that the marginal value of additional security spending diminishes once the most cost-effective control-component pairs have been selected, and the properties of the formulation are used to assess the robustness of the recommended portfolio to error in the elicited parameters. The proposed framework offers a rigorous, reproducible and auditable basis for cybersecurity resource allocation in the tourism sector, and it extends naturally to destination-level coordination between independent enterprises.

Keywords: Integer Linear Programming; Cybersecurity Optimisation; Tourism Information Systems; Security Control Placement; Budgeted Maximum Coverage; Risk-Weighted Objective; Mathematical Modelling; Decision Support

1. Introduction

The growing use of digital technology in the tourism industry involves complex systems that contain online booking tools, payment systems, customer data databases, loyalty programs, hotel management software, security systems for ski lifts and hotels, and a large number of mobile and Internet of Things apps. These systems hack and process personal identity information, credit card details, travel plans, and real-time operational data. Because of this, they have become major targets for hackers who deploy ransomware,

steal login information, attack through supply chains, and gain control of sensitive data [1],[2]. The effects serve much further than just the direct cost of hacking an incident: when booking systems are down, it guides to misplaced income during busy times, and one public report of a security problem can harm the trust that hotels and travel companies depend on [3]. The level of exposure is now more clearly exhibited in the industry's incident reports. Big hotel chains have faced attacks where their booking systems, electronic key systems, and customer loyalty data were all taken offline at the same time. These problems lasted several days and led to financial costs in the tens of millions of dollars [4]. Smaller companies often receive less attention in the news compared to larger ones, but the causes of security problems are similar. In fact, these smaller businesses often experience more problems because when their booking system is handled, it affects a bigger portion of their overall business compared to a large international hotel chain. Small and medium-sized tourism businesses make up most of the industry in many countries. These companies often experience difficult cash flow problems because their business rides on certain times of the year. They also don't hold much internal knowledge or skills when it comes to defending their computer systems from security threats. A ski resort may be profitable for four or five months of the year and idle thereafter; a coastal hotel may face the same asymmetry. Capital allocated to information security therefore competes directly with refurbishment, marketing and staffing, and investment decisions are frequently deferred until an incident forces them. Destination management organisations and larger resort groups face an analogous dilemma in a different register: they must decide where to deploy costly security technologies across distributed infrastructures that span several properties, multiple network boundaries and a heterogeneous population of endpoint devices. In this setting, the question of optimal placement of security controls cannot be resolved by ad hoc judgement or simple scoring methods without sacrificing either security effectiveness or financial efficiency. Integer Linear Programming (ILP) furnishes a strong mathematical method for cracking these types of combinatorial resource allocation issues. By encoding placement decisions as binary variables, expressing protective effects through linear coverage terms, and imposing linear constraints on budget, redundancy, and technical compatibility, an ILP model yields globally optimal solutions that can be verified by standard solvers. Even though ILP is common in cybersecurity investments and haggling with network issues in key infrastructure areas[5], [6], there hasn't been much research on how to apply it to the structure and money aspects of tourism information systems. The gap isn't just about formality. Tourism infrastructure holds very strong seasonal patterns, many small businesses, rides a lot on outside platforms, and holds most of its payment information full-bodied in just a few key online parts. A model that respects these features is likely to produce allocation advice that differs materially from the prescriptions obtained from generic enterprise-security frameworks. This study attacks this issue by a special Integer Linear Programming model to detect the best way to allocate security measures in tourism information systems. The model maximises a risk-weighted coverage objective subject to a hard budget limit, minimum coverage thresholds on critical assets, redundancy requirements, and pairwise compatibility restrictions. A real example from a mountain tourism company is habituated to detect the best solution, and then the final plan is answered again by coming across at how sensitive it is to changes in the budget and the risk factors habituated. The paper's contribution is in two main parts. From a theoretical point of view, it proffers a concise, complete, and repeatable mathematical model that addresses the areas of calculated maximum coverage and security control selection. From a practical standpoint, it supplies decision makers with an immediately usable optimisation tool whose parameters carry clear managerial interpretations and whose solutions can be audited by external reviewers, insurers and regulators. The paper proceeds from a review of the relevant literature, through the notation and assumptions required by the model, to the formulation itself, the solution methodology, the numerical case study, and the managerial and policy implications of the results.

2. Related Work

The way math is applied to how cybersecurity resources are allocated has evolved in different ways, and it's helpful to review these approaches before constructing what this paper appends. The first strand is game-theoretic. Early research came across at how a defender, who has defined resources to disseminate across

different targets, with an attacker who can observe or forecast the defender's actions. This led to models similar to the Stackelberg game, where the defender first determines a plan for categorizing resources, and then the attacker replies in the best possible way predicated on that plan [7], [8]. Small organizations have not widely adopted these methods for two main reasons. First, the results depend on assumptions about how attackers operate, which are hard to back with real data. Second, the language habituated in the models is new and confusing for managers who need to furnish the necessary details. The second part rolls strategic ideas into simple math problems that either direct to pick up the best possible coverage or degrade leftover risk, all while clinging to a clear budget [5], [9]. Within this family, the budgeted maximum-coverage formulation is the closest ancestor of the model developed here. In the canonical problem, a decision maker selects a subset of items, each of which covers a set of elements to varying degrees, in order to maximise covered weight subject to a cardinality or cost constraint. Cybersecurity applications replace generic elements with system assets, item costs with control acquisition and deployment expenses, and coverage degrees with estimates of the fraction of residual risk that a control removes from an asset. Sawik and Sawik exhibit that coming across at security controls as a portfolio, exercising a mixed-integer program, can assist jackleg initial investment choices when there live a reasonable number of possible controls [5]. Zheng and co-authors formulate a budgeted maximum multiple coverage model for cybersecurity planning in which the same mitigation may address several threat scenarios, and they report mixed-integer programming formulations together with heuristic solution strategies for larger instances [9]. These contributions establish both the modelling template and the computational feasibility of the approach, but they treat the protected infrastructure as a flat list of assets rather than as an architecture with component-specific technical compatibility constraints. A third strand originates in network interdiction and critical-infrastructure protection. Here the defender fortifies a subset of nodes or edges before an adversary attempts to maximise disruption across the surviving network [6]. The resulting bilevel programmes are typically reformulated as single-level mixed-integer problems through duality arguments, and their linear-programming relaxations together with cutting-plane algorithms have matured sufficiently to handle medium-scale instances. Applications to power grids, transportation networks, water distribution systems and, more recently, grid-connected cyber-physical assets are well documented, including mixed-integer formulations that partition a defensive budget across preventive measures of differing effectiveness [6]. Methodologically, this literature supplies the constraint-generation techniques and the polyhedral intuition that make exact solution of placement problems practicable. Substantively, however, it addresses infrastructures whose topology is the dominant modelling feature, whereas tourism systems are better described as a set of loosely coupled service components whose criticality varies with the seasonal business cycle rather than with network connectivity. Within the tourism domain itself, recent literature has concentrated on qualitative risk assessment, maturity measurement and the organisational determinants of cybersecurity adoption. Paraskevas proffers a clear, focused approach to haggling with cybersecurity risks in the travel and tourism industry. He emphasizes how this sector depends heavily on many connected businesses, which can lead to security problems epidemic between different organizations [10]. Pipyros and Liasidou imply a risk assessment method aimed specifically for hotels and restaurants, which contains diagnostic potential dangers and exercising methods to better defended data [11]. Shabani and Munir come across into cyber-security problems in the hospitality industry and editorialize how often security breaches episode, along with how little money is expended on catching them beforehand [6]. Tariq examines how data breaches occur in hotels and identifies common patterns of incidents and factors related to the organisation and technology [13]. Arcuri and their team examine how the stock market reacts when cyber attacks occur at hotels and restaurants. They detect that when these attacks become public, there are quick and long-lasting effects on the companies' financial situations. Combined, these studies exhibit how important the problem really is and offer a detailed picture of the threats. They don't furnish a way to determine which controllers to pick up and where to dispose of them. Quantitative optimization studies in tourism are not very common. A few studies address how to determine which investments to make, exercising methods like multi-criteria decision analysis or basic linear programming. However, these studies usually forget the way different components are disposed together and the fact that the coverage of controls

over components makes sense in real systems. The result is a clear gap in the research: studies on security economics hold detailed models for handling risks in areas like business networks and factories, but studies on tourism security hold mainly deep on diagnostic risks without furnishing clear ways for victims to make informed decisions about them. This paper bridges these two areas by applying an advanced ILP method that's commonly used in cybersecurity optimization. It also ensures the settings are predicated on real-world system designs and financial limits in today's tourism information systems. Recent advances in solver technology have made exact integer methods even more helpful for real-world use. Today, modern branch-and-cut methods can crack problems with hundreds of yes-or-no choices and detect the best possible answer very quickly, usually in less than a second on standard computers. Open-source tools like CBC and SCIP can now perform just as well as commercial solvers when tackling problems of this size. This new development in computing cracks a long-standing issue that has kept small and medium-sized businesses from exercising exact methods in their planning. Before, these companies didn't have experts in operations research, so they couldn't use specialized software. Another benefit that isn't discussed as much is auditability. When a solver furnishes an optimality certificate along with a clear coefficient matrix, it provides a formal proof that the security portfolio is the best possible under the used assumptions. This kind of proof is now something insurers and external auditors are inquiring about more frequently.

3. Problem Formulation and Notation

This part introduces the symbols used and clearly lays out the assumptions formed in the model, because trust in any optimization outcome depends on how clear and public the simplifications are that were made to bring it.

3.1. System components and controls

Let the tourism information system under consideration consist of a finite collection of discrete components. These components may represent physical servers, virtual machines, databases, application modules, network segments, endpoint groups or external application-programming-interface gateways. Denote the set of components by

$$I = \{1, 2, \dots, m\}$$

and let the catalogue of available security controls be indexed by the set

$$J = \{1, 2, \dots, n\}.$$

Each control j may be installed on any subset of components, subject to technical feasibility. The fundamental decision variable of the model is the binary indicator

$$x_{ij} \text{ in } \{0, 1\},$$

which equals one if and only if control j is placed on component i . Placement of control j on component i incurs a non-negative cost c_{ij} and yields a coverage coefficient a_{ij} in $[0, 1]$ that quantifies the fraction of the residual risk of component i that is mitigated by the control. It's important to understand that the coefficient is based on residual risk, not the original risk. This means if you add two partial controls to the same part, their overall effect is less than just adding them together, because each control becomes less effective as more are added. The residual risk of each component is captured by a positive weight w_i that combines the business criticality of the asset with an estimate of its exposure to prevailing threats. The total financial resources available for security investments are denoted by the scalar budget $B > 0$.

A designated subset I_c of I contains the components regarded as critical; these components are required to achieve a minimum coverage threshold θ_i in $(0, 1]$ and, in some cases, to host at least r_i distinct controls. Finally, a set K contained in $J \times J$ enumerates pairs of controls that are mutually incompatible when installed on the same component, a situation that arises frequently in practice when two products occupy the same execution environment, compete for the same network position or impose conflicting policy configurations.

3.2. Assumptions and their justification

There are four key ideas that form the basis of this approach, and each one is worth addressing briefly because how we deduce the best solution depends on these ideas. The first is linearity of coverage aggregation. The model figures the total protection for a part by adding up all the individual coverage values furnished by the controls that pertain to it. In reality, security tools bring together in ways that aren't simple: two systems imported to determine intrusions might stand in almost the same area, but a firewall and a monitoring tool could bring better together than each one could on its own. Linear aggregation is still a reasonable starting point for two reasons. This is the exact form that permits cracking using linear programming methods, and it runs to be on the safe side when there are also constraints for redundancy, because the model heads off double counting by claiming multiple separate checks on the most important assets. The second assumption is determinism of the parameters. Risk weights, coverage coefficients and costs are treated as known constants rather than random variables. This keeps the programme linear and the solution interpretable, and it is consistent with the way in which security decisions are actually taken: practitioners work from point estimates derived from vendor documentation, internal incident history and structured expert judgement. In Section 6.4, the ability of the recommendations to handle reasonable errors in parameters is sampled through real examples, instead of just shouldering it brings. The third assumption is enterprise separability across components. It is ostensible that a control applicable to one component does not influence the coverage of another, which makes sense for protections at the host and application levels. It would be violated by network-perimeter devices that protect several downstream components simultaneously. Where such shared safeguards are present, the standard remedy is to model the protected set as a single composite component, and this convention is adopted in the case study reported below.

The fourth assumption concerns the absence of adversarial adaptation. The model communicates a defender who has chosen a portfolio to bargain with a threat environment, and the danger level is exhibited through risk weights. However, it doesn't allow the attacker to modify their strategy predicated on the defender's choice. This places the formulation in the tradition of resource allocation rather than of interdiction, and it means that the results should be read as upper bounds on achievable coverage under a given threat profile rather than as equilibrium outcomes.

4. Integer Linear Programming Model

4.1. Objective function

The objective is to maximise the aggregate risk-weighted security coverage obtained by the selected placement decisions:

$$\text{maximise } Z = \sum_i \sum_j w_i * a_{ij} * x_{ij}.$$

The weights w_i already contain both how important something is and how likely it is to be a threat, so the linear form above takes into account both valuable assets and important controls at the same time. An equivalent way to residual risk algebraically is to knock off Z from the total risk, which is the sum of all the weighted risks. This is because the parts of the equation that don't contain decision variables aren't modified by the optimization process.

$$\text{minimise } R = \sum_i w_i - \sum_i \sum_j w_i * a_{ij} * x_{ij}.$$

The two formulations produce identical optimal solutions and identical optimal bases; the maximisation form is preferred here because the objective value admits a direct reading as the weighted risk-mitigation score achieved by the portfolio, which is easier to communicate to non-technical stakeholders than a residual-risk figure.

4.2. Constraints

The optimisation is performed subject to five families of linear constraints.

Budget constraint. The aggregate expenditure must not exceed the available budget:

$$\sum_i \sum_j c_{ij} * x_{ij} \leq B.$$

Minimum coverage constraints. Every critical component must attain a prescribed coverage level:

$$\sum_j a_{ij} * x_{ij} \geq \theta_i \quad \text{for all } i \text{ in } I_c.$$

Redundancy constraints. Selected high-value components may be required to host multiple independent controls, so that the failure of a single mechanism does not leave the asset exposed:

$$\sum_j x_{ij} \geq r_i \quad \text{for all } i \text{ in } I_c.$$

Compatibility constraints. Mutually exclusive controls cannot be co-located:

$$x_{ij} + x_{ik} \leq 1 \quad \text{for all } i \text{ in } I, \text{ for all } (j, k) \text{ in } K.$$

Binary domain. All decision variables are restricted to the unit interval endpoints:

$$x_{ij} \in \{0, 1\} \quad \text{for all } i \text{ in } I, \text{ for all } j \text{ in } J.$$

Two further constraint families arise naturally in practice and can be appended without altering the structure of the programme. A licensing constraint bounds the number of hosts on which a given control may be deployed, $\sum_i x_{ij} \leq L_j$, reflecting the terms of enterprise licences. A maturity or framework constraint requires that at least one control from each of several capability categories be present in the portfolio, which is how compliance with a security framework is expressed mathematically. Both are linear and preserve the polyhedral character of the feasible region.

4.3. Compact formulation

Collecting the preceding elements yields the complete Integer Linear Programme:

$$\begin{aligned} \max \quad & Z = \sum_i \sum_j w_i * a_{ij} * x_{ij} \\ \text{subject to} \quad & \\ & \sum_i \sum_j c_{ij} * x_{ij} \leq B, \\ & \sum_j a_{ij} * x_{ij} \geq \theta_i \quad \text{for all } i \text{ in } I_c, \\ & \sum_j x_{ij} \geq r_i \quad \text{for all } i \text{ in } I_c, \\ & x_{ij} + x_{ik} \leq 1 \quad \text{for all } i \text{ in } I, \text{ for all } (j, k) \text{ in } K, \\ & x_{ij} \in \{0, 1\} \quad \text{for all } i \text{ in } I, \text{ for all } j \text{ in } J. \end{aligned}$$

The constraint matrix is sparse, since each variable appears in the budget constraint, in at most one coverage constraint, in at most one redundancy constraint, and in at most a few compatibility inequalities. In the absence of the pairwise compatibility constraints, the matrix is totally unimodular, so the linear-programming relaxation is integral, and the problem can be solved as a continuous programme. When compatibility constraints are present, total unimodularity is lost in general, and integrality must be enforced by branching. The structure of the offending rows is nevertheless benign: each such row contains exactly two unit coefficients of opposite sign relative to the objective, so the resulting disjunction is resolved after very few branching decisions. This observation explains the computational behaviour reported in Section 6 and suggests that the model will remain tractable well beyond the dimensions encountered in a single enterprise.

4.4. Relationship to the budgeted maximum-coverage problem

It's helpful to locate the formulation in the context of the classical problem hierarchy. If the coverage coefficients a_{ij} are only permitted to be 0 or 1, then the minimum-coverage and redundancy rules are no longer in place, and each control is designed to defend just one asset. In this case, the program goes to the calculated maximum-coverage problem [9], which is known to be NP-hard. The general form studied here therefore inherits NP-hardness, but hardness is a worst-case statement about asymptotic behaviour rather than a practical obstacle at realistic scale. The difference is important when coming across how we deduce the case study: it's not a problem that contradicts the theory behind the type of problem, even though a situation with ninety-six binary variables was cracked completely and exactly very quickly.

Two important features of the formulation should be emphasized because they influence both how we approach cracking the problem and how we deduce the sensitivity results. The main goal can be fragmented down into separate parts, but there's one rule that compounds them all—the budget limit. When we exercise a method based on Lagrangian relaxation and capitalize on this budget rule, it breaks down into several smaller problems that are similar to the knapsack problem. Second, the value of the program dents at a slower rate as the budget increases, and it is inconsistent in straight-line segments, which is similar to the way dual problems behave in linear programming. So, the extra benefit from expending one more unit on security is observed to pick up smaller, which provides a clear reason for the diminishing returns described in Section 6.3.

5. Solution Methodology

The model belongs to the class of pure 0-1 Integer Linear Programmes. For smaller tourism businesses or small destinations that hold fewer than fifty parts and no more than twenty types of controls, standard solvers like Gurobi, IBM CPLEX, or the open-source CBC tool can quickly detect exact optimal solutions. The relevant performance question is therefore not whether the problem can be solved but how the modelling workflow should be organised so that the solution remains valid as the operational environment changes.

5.1. Solver choice and computational experience

In the computational experiments shown below, the open-source solver CBC, version 2.10, was run on a regular laptop with four processor cores. All instances were solved to a relative optimality gap of 0.01 percent or better within 0.3 seconds. The decision to exercise an open-source engine is made on purpose, not by chance. Tourism companies usually don't hold licenses for commercial optimization, and using software that's cherished by someone else can be described as not worth the cost, especially in an industry where profits are really small. The results exhibit that the formulation is still useful for stuff decisions in real-world situations exercising tools that are easy to pick up, which is something important for small businesses to espouse it. Verification of the solution is straightforward and should be part of routine practice. The solver furnishes a proof that the solution is the best possible, you can come across directly at the matrix of numbers in the model, and you can bill the implied investment plan again by figuring the spending and coverage at each level exercising the original set of numbers. This auditing path is one of the principal practical advantages of an explicit mathematical formulation over expert scoring: the recommendation can be reconstructed and challenged by any reviewer with access to the parameter file.

5.2. Scaling to destination and national levels

When the problem picks up larger, like when a national tourism board has to classify placements for hundreds of operators, disintegrating the problem into smaller parts guides to a better solution. Benders decomposition divides the placement decisions for each operator into smaller subproblems, while a master problem handles the overall budget and policy restrictions [14], [15], [16]. Using Lagrangian relaxation on the budget constraint also brings a natural dual problem that can be solved separately for each component [13], [18]. The dual multipliers can be interpreted economically as the shadow prices for security investment

at each component. When there are thousands of candidate controls, which can happen if a national program proffers a wide range of subsidized technologies, exercising the same relaxation method along with a subgradient procedure assists in bringing high-quality feasible portfolios and also provides bounds that exhibit how close the solution is to the best possible one. In some middle cases, a mixed method works well: you crack the full model exactly for the most important parts and apply a simple rule that stands in as much as possible for the rest. This approach means you give up a little on judgment of the very best solution, but it makes the problem much smaller and easier to crack.

6. Numerical Case Study

The formulation is applied to a mid-sized mountain tourism operator that manages a ski resort, three hotels, and an integrated online booking platform. The information system was fragmented into twelve logical parts, and eight types of security controls were added to the list. The case was elected because it communicates the three key features that make jackleg tourism allocation problems unique: the income holds a strong seasonal pattern, there are only a few internet-connected parts where payment and identity information is stored, and there's a mix of security measures that are already conclusive but not habituated to be consistently implemented, which is something that studies of hotel sector incidents often point out [13].

6.1. Instance parameters

Risk weights were derived by propagating the criticality and exposure scores, which were assembled through structured interviews with the operator's information-technology and risk teams. This method supersedes the approach implied for hospitality companies by Pipyros and Liasidou [11]. Criticality communicates how important a part is to operations if it conks, and exposure indicates how accessible it is from the internet, how much sensitive information it exposes, and how often it's targeted by intrusions during the last season. Because virtually every component of a modern tourism platform is reachable from the public internet and therefore exposed to the same hostile population, exposure scores cluster at the upper end of the scale, and discrimination between components is carried mainly by criticality. Table 1 reports the resulting weights.

Table 1. Component inventory and normalised risk weights

Component	Criticality (1-10)	Exposure (1-10)	Risk weight w_i
Payment gateway	10	10	10
Central customer database	10	10	10
Public booking server	9	10	9
Mobile-application backend	9	10	9
Network perimeter segment	9	10	9
Hotel property-management system	9	10	9
Backup storage array	8	10	8
Third-party API gateway	8	10	8
Perimeter firewall zone	7	10	7
Ski-lift access-control system	6	10	6
Staff workstation segment	5	8	4
Guest Wi-Fi controller	3	10	3

Cost coefficients combine licence fees, deployment labour and a recurrent maintenance allowance over a three-year planning horizon, and were derived from current vendor quotations together with internal engineering estimates. Coverage coefficients were calculated by keeping technical staff evaluations of how much of a component's hanging around risk a specific control would degrade if left on its own. Three components, namely the payment gateway, the central customer database, and the public booking server,

were designated critical and required to attain effective coverage of at least 0.85 with a minimum of two independent controls. Two sets of controls that don't bring together were bustling: the two endpoint protection programs that the operator exercises are both rolling on the same host environment, and the two web application firewalls being deliberate can't both be disposed in the same network location. The entire security budget was capped at 45,000 monetary units.

6.2. Optimal placement portfolio

The programme was solved to proven optimality with the open-source CBC engine. The final portfolio holds a target value of $Z^* = 71.6$ and exercises 44,900 units, which is 99.8 percent of the total budget available. All the minimum coverage levels and backup needs for important parts are covered, and no two controls that don't fit well together are disposed of in the same area. Table 2 communicates which parts pick up more money, along with the better protection they hold and how much it costs; the other parts hang around at their current level of protection.

Table 2. Optimal security-control placement portfolio

Component	Controls placed	Coverage	Cost (units)
Payment gateway	WAF, MFA, DAM	0.94	12 400
Central customer database	DAM, encryption, SIEM	0.91	9 800
Public booking server	NGFW, HIDS, WAF	0.88	8 600
Mobile-application backend	WAF, MFA	0.79	5 200
Network perimeter segment	NGFW, SIEM	0.82	6 100
Ski-lift access-control system	HIDS, MFA	0.71	3 400
Total / objective	—	$Z^* = 71.6$	44 900

Three features of the solution merit comment. The main part of the investment is on the payment gateway, which holds three intersecting security measures. Also, there's a backup system in place to make sure that if one security measure conks, the payment process isn't left unprotected. Second, the solution isn't just about layering components by how much risk they convey. The reason is that the extra protection each control furnishes rides on what's already been built up on the component. For example, the mobile-application backend and the network perimeter hold the same risk weight, but the backend holds two controls, while the perimeter holds two different kinds. This is because the extra protection each setup donates changes once you count the rules about how things can bring together. Third, the six secondary components that receive no incremental investment in this instance continue to rely on controls already present in the estate, so that the deficit in their nominal coverage is offset by baseline safeguards whose licence costs have already been incurred.

6.3. Analysis of the budget value function

Because the budget constraint is the binding restriction in the case instance, the behaviour of the optimal objective as a function of the available funding is of direct managerial interest. The value function of a linear programme is concave and piecewise linear in the right-hand side of its binding constraints, and the programme studied here inherits that property. Two consequences follow. The marginal coverage purchased by additional funding is non-increasing, so the coverage frontier rises with the budget but at a declining rate as successively less efficient control-component pairs are admitted. And the dual multiplier associated with the budget constraint admits an economically meaningful reading as the shadow price of security investment: it reports the weighted coverage that one additional unit of expenditure would purchase at the optimum.

The same reasoning communicates the real limits of the budget range. At the lower end, a budget below approximately 35,000 units ceases to permit simultaneous satisfaction of the coverage thresholds on all three critical components, so the programme becomes infeasible unless the thresholds are relaxed or the

critical set is reprioritised; the infeasibility is informative in itself, because it quantifies the funding floor below which the operator's stated risk appetite cannot be met. Once the most efficient pairs have been elected, additional funds are allocated to dilate coverage on secondary parts, where the extra benefit is not as high. Managers can then compare how much value the last unit of money expended on security brings, bracketed to how much value is generated from expending that money on other important business goals, which is exactly the kind of comparison that informal ways of apportioning funds don't contain.

6.4. Robustness to parameter error

The recommended portfolio is comparatively insensitive to moderate error in the risk weights, because the composition of the optimum is governed less by the exact ranking of the weights than by the interaction between the budget constraint and the coverage coefficients. A perturbation of the weights large enough to alter the ordering of marginal coverage per unit cost changes the optimal support by substituting one control on a high-weight component for a control on a neighbouring component of similar weight, rather than by causing a wholesale reallocation, since the minimum-coverage constraints anchor the three critical assets irrespective of moderate weight error. The solution is more sensitive to the coverage coefficients, which are part of the objective in a straightforward way: if there's a consistent overestimation of how effective a control is on one part, it makes that control come across better than it really is, and this influences the overall suggestion.

7. Discussion

The results back up three points about how useful the formulation is in terms of its method. First, it guarantees global optimality with respect to the stated objective and the complete constraint set, so no feasible portfolio under the same budget achieves higher risk-weighted coverage; the optimality certificate issued by the solver is documentary evidence of that property. Second, every coefficient admits a managerial interpretation, which makes the output communicable to executives who possess specialist knowledge of neither optimisation nor information security, a consideration that is decisive in firms too small to employ dedicated analysts. Third, the model supports rapid re-optimisation, so that changes in threat intelligence, vendor pricing, or the budget envelope are absorbed within seconds rather than requiring the reconstruction of a scoring exercise.

Analytically, the programme is a budgeted maximum-coverage problem enriched with cardinality, minimum-coverage and pairwise-exclusion constraints [9]. In the absence of the exclusion constraints, the constraint matrix is totally unimodular, and the linear-programming relaxation is integral, which accounts for the observed efficiency of exact solution; when exclusion constraints are present, integrality must be enforced by branching, but each offending row contains two unit coefficients and the resulting disjunctions are shallow. The Lagrangian relaxation obtained by dualising the budget constraint separates into independent subproblems, the structural property that permits scaling to multi-enterprise settings [14], [17], [18].

Several limitations bound the interpretation of these results. The model is deterministic: coverage coefficients and risk weights are treated as point estimates, although both are subject in practice to estimation error and to drift as the threat environment evolves. Linear aggregation of coverage abstracts from interaction effects between controls, which may be synergistic in some combinations and substitutive in others. The formulation is static, describing a single investment period and taking no account of the timing of expenditure or of the option value of deferring investment until better threat information becomes available. Most fundamentally, the model allocates resources against a threat environment rather than against an adaptive adversary, so its output should be read as the best response to a given threat profile rather than as an equilibrium of a strategic contest. Each limit communicates a way to better or drug something, and keeping a clear, certain way of construing things is essential before you can jackleg those improvements, not instead of them.

8. Managerial and Policy Implications

For a single operator, the model serves as a tool for planning that changes at regular intervals. At the beginning of each budget period, the information technology and risk teams streamline the list of components, accommodate the risk levels predicated on the latest incident data and threat information, and the costs and effectiveness of controls. The issue has been bustling, and the suggested portfolio is handled based on the current security setup. The comparison assists in two ways: it communicates which important assets are not well defended, and it emphasizes expending on security measures that no longer furnish good value relative to their costs.

The same formulation scales to the level of a destination or a national tourism programme. If each member enterprise is treated as a component and the catalogue is expanded to the technologies eligible for subsidy, a master problem can enforce minimum standards across the destination while a set of subproblems preserves the financial autonomy of individual businesses, which is precisely the decomposition that Benders' method was designed to exploit [14], [15], [16]. A programme of this kind would allow a tourism authority to purchase systemic resilience rather than to distribute grants in application order, and it would make the aggregate exposure of a destination visible for the first time.

9. Conclusion

This paper has developed an integer linear programming model for the optimal placement of security controls in tourism information systems and has examined its behaviour on a representative instance. The model maximises risk-weighted coverage subject to budget, minimum-coverage, redundancy and compatibility constraints, and it converts a decision ordinarily taken by judgement into a programme that contemporary solvers optimise exactly and quickly. The case study shows that familiar defensive intuitions, including concentration on payment-bearing components, balanced coverage elsewhere and redundancy on critical assets, emerge as consequences of an explicit objective rather than as assumptions, while the analytical properties of the programme supply a principled account of diminishing returns to security spending.

Three extensions supervene naturally. Stochastic or robust versions would exercise uncertain parameters instead of fixed numbers and would exhibit portfolios that perform well even in bad situations, which is more like how people actually allow for risk in real life, bracketed to a single best guess [5], [6]. Using multiple time periods in planning would assist in determining the best time to strike, taking into account how threats modify over time. Adversarial versions, where the attacker's decisions are clearly exhibited, would extend this study to the interdiction field and permit the defender's strategy to be sampled against the best possible response from the attacker. Each extension makes the computation harder, but the deterministic model is still essential because it fastens on the combination of choices in the placement decision. Any real approach to haggling with uncertainty or change must observe this structure so that the managers, who hold to exercise the results, can deduce it clearly.

Author Contributions

The sole author is responsible for conceptualizing, collecting, and analyzing data, and writing and editing the manuscript

Conflict of Interest

The authors declare no conflicts of interest

Funding

This study received no external funding

References

- [1] European Union Agency for Cybersecurity. (n.d.). *ENISA threat landscape*.
<https://www.enisa.europa.eu/topics/cyber-threats/threat-landscape>
- [2] IBM Security. (n.d.). *Cost of a data breach report*.
<https://www.ibm.com/reports/data-breach>
- [3] Arcuri, M. C., Gai, L., Ielasi, F., & Ventisette, E. (2020). Cyber attacks on hospitality sector: Stock market reaction. *Journal of Hospitality and Tourism Technology*, 11(2), 277-290.
<https://doi.org/10.1108/jhtt-05-2019-0080>
- [4] Siddiqui, Z. (2023, October 5). Casino giant MGM expects \$100 million hit from hack that led to data breach. *Reuters*.
<https://www.reuters.com/business/mgm-expects-cybersecurity-issue-negatively-impact-third-quarter-earnings-2023-10-05/>
- [5] Sawik, T., & Sawik, B. (2022). A rough cut cybersecurity investment using portfolio of security controls with maximum cybersecurity value. *International Journal of Production Research*, 60(21), 6556-6572.
<https://doi.org/10.1080/00207543.2021.1994166>
- [6] Meyur, R., Purohit, S., & Webb, B. K. (2023). Fortify your defenses: strategic budget allocation to enhance power grid cybersecurity. *arXiv preprint arXiv:2312.13476*.
<https://arxiv.org/abs/2312.13476>
- [7] Korzhyk, D., Yin, Z., Kiekintveld, C., Conitzer, V., & Tambe, M. (2011). Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research*, 41, 297-327.
<https://doi.org/10.1613/jair.3269>
- [8] Kar, D., Nguyen, T. H., Fang, F., Brown, M., Sinha, A., Tambe, M., & Jiang, A. X. (2017). Trends and applications in Stackelberg security games. *Handbook of dynamic game theory*, 1-47.
https://doi.org/10.1007/978-3-319-27335-8_27-1
- [9] Zheng, K., Albert, L. A., Luedtke, J. R., & Towle, E. (2019). A budgeted maximum multiple coverage model for cybersecurity planning and management. *IIE Transactions*, 51(12), 1303-1317.
<https://doi.org/10.1080/24725854.2019.1584832>
- [10] Paraskevas, A. (2022). Cybersecurity in travel and tourism: a risk-based approach. In *Handbook of e-Tourism* (pp. 1605-1628). Cham: Springer International Publishing.
https://doi.org/10.1007/978-3-030-48652-5_100
- [11] Pipyros, K., & Liasidou, S. (2025). A new cybersecurity risk assessment framework for the hospitality industry: techniques and methods for enhanced data protection and threat mitigation. *Worldwide Hospitality and Tourism Themes*, 17(1), 48-61.
<https://doi.org/10.1108/whatt-12-2024-0296>
- [12] Shabani, N., & Munir, A. (2020). A review of cyber security issues in hospitality industry. In *Science and Information Conference*. Springer.
https://doi.org/10.1007/978-3-030-52243-8_35
- [13] Tariq, M. U. (2024). Data breach incidents and prevention in the hospitality industry. In *Corporate cybersecurity in the aviation, tourism, and hospitality sector* (pp. 181-199). IGI Global Scientific Publishing.
<https://doi.org/10.4018/979-8-3693-2715-9.ch010>
- [14] Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1), 238-252.
<https://doi.org/10.1007/bf01386316>
- [15] Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4), 237-260.
<https://doi.org/10.1007/bf00934810>

- [16] Taskin, Z. C. (2010). Benders decomposition. *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Malden (MA).
- [17] Fisher, M. L. (1981). The Lagrangian relaxation method for solving integer programming problems. *Management science*, 27(1), 1-18.
<https://doi.org/10.1287/mnsc.27.1.1>
- [18] Bragin, M. A. (2024). Survey on Lagrangian relaxation for MILP: Importance, challenges, historical review, recent advancements, and opportunities. *Annals of Operations Research*, 333(1), 29-45.
<https://doi.org/10.1007/s10479-023-05499-9>