An Outsourcing Model for Alert Analysis in a Cybersecurity Operations Center

ANKIT SHAH, University of South Florida, USA
RAJESH GANESAN and SUSHIL JAJODIA, George Mason University, USA
HASAN CAM, U.S. Army Research Laboratory, USA

A typical Cybersecurity Operations Center (CSOC) is a service organization. It hires and trains analysts, whose task is to perform analysis of alerts that were generated while monitoring the client's networks. Due to ever-increasing financial and infrastructure burden on a CSOC driven by the rapidly growing demand for security services, it would become prohibitively expensive to continually expand the size of a CSOC to meet the demands in the future. An alternative solution is to outsource the alert analysis process to on-demand analysts, to provide scalable CSOC service to its clients with features, such as (1) higher throughput, (2) higher quality, and (3) more economical service than the current in-house service. The current outsourcing model is not cost effective and an exact optimization model is computationally inefficient. This article presents a novel two-step sequential mixed integer programming optimization method that is used in the development of a new decision-support business model for outsourcing the alert analysis process. It is demonstrated that through this model, a CSOC can effectively deliver its alert management services with the above-mentioned features. Results indicate that the model is scalable, computationally viable, real-time implementable, and can deliver CSOC services that meet the service-level agreement (SLA) between the CSOC and its client. In addition, the article provides valuable insights into the cost of operating the new business process outsourcing model for cybersecurity services.

CCS Concepts: • **Networks** → **Network security**; **Network monitoring**;

Additional Key Words and Phrases: Alert analysis, CSOC, outsourcing, on-demand analysts, optimization

ACM Reference format:
Ankit Shah, Rajesh Ganesan, Sushil Jajodia, and Hasan Cam. 2020. An Outsourcing Model for Alert Analysis in a Cybersecurity Operations Center. *ACM Trans. Web* 14, 1, Article 2 (January 2020), 22 pages.
https://doi.org/10.1145/3372498

1 INTRODUCTION

In a security-as-a-service (SECaaS) model, organizations (clients) pay for the security services by entering into a service-level agreement (SLA) with a cybersecurity operations center (CSOC), who is the service provider. Alert management is one of the primary services provided by a CSOC

Shah, Ganesan, and Jajodia were partially supported by the Army Research Office under Grant No. W911NF-13-1-0421, by the Office of Naval Research under Grant No. N00014-15-1-2007, and by the National Science Foundation under Grants No. IIP-1266147 and No. CNS-1822094.

Authors' addresses: A. Shah, University of South Florida, 4202 E. Fowler Ave. Tampa, FL, 33620, USA; email: ankitshah@usf.edu; R. Ganesan, George Mason University, 4400, Univ Dr., MS 4A6, Fairfax, VA, 22030, USA; email: rganesan@gmu.edu; S. Jajodia, George Mason University, 4400, Univ Dr., MS 5B5, Fairfax, VA, 22030, USA; email: jajodia@gmu.edu; H. Cam, U.S. Army Research Laboratory, 2800 Powder Mill Road, Adelphi, MD, 20783, USA; email: hasan.cam.civ@mail.mil.

1559-1131/2020/01-ART2 $15.00
https://doi.org/10.1145/3372498

[D'Amico and Whitley 2008]. A typical CSOC employs in-house analysts who are scheduled and managed to perform alert analysis. Due to the financial and infrastructure burden on a CSOC driven by the growing demand for security services, it would become prohibitively expensive to continually expand the size of a CSOC to meet the demands in the future. Newer CSOCs will require establishment time to train their employees, which further augments the already growing differences between higher demand for security services and lower supply of service providers. As an alternative, an outsourcing business model of a CSOC for alert management is researched in this article, in which the CSOC selects resources from a large pool of analysts with various expertise levels that are available on-demand. Outsourcing of alert management benefits an organization by reducing the financial and infrastructure burden of maintaining an in-house security group and by allowing the organization to focus on its core competencies. Such core competencies include its ability to deliver security in an economical way (minimize cost of operation and remain profitable), faster response time on alert analysis within a job (higher throughput), scalable services to meet higher demands, and a guaranteed quality of alert analysis compared to an in-house security center. The above motivated our research to build models that would assist a CSOC to outsource some or all of their operations.

In this research, we define an alert analysis job as a group of alerts generated and clustered from a single client over a period of time (usually an hour to a few hours) that can be assigned to one or more analysts. The number of alerts and the type of alerts within a job can vary. Alert analysis requires analysts to perform several steps for investigating an alert, and based on both the alert and the analyst's expertise, the time taken for analysis may vary. The task for the analysts is to analyze all the alerts in the job or a portion of the alerts in a job that is assigned to them, and classify each of them as significant or innocuous. The classified alerts are then sent back to the client organization for further action.

Typically, SLAs are signed to provide a threshold on timeliness and quality of services. With respect to a SECaaS model, the SLA requirements are as follows:

(1) **Timeliness:** All the alerts in a job must be analyzed within the agreed-upon (target) completion time for each client.
(2) **Quality of service:** The average error rate (alerts that are falsely classified as negatives) from alert analysis for each job must not exceed the agreed-upon threshold between the CSOC and the client.

Alert analysis jobs have the following attributes: (a) total number of alerts in each job, (b) threshold on job completion time, and (c) threshold on the error rate. The thresholds are governed by the SLA between the client and the CSOC as described above. Analysts differ from one another in terms of the following attributes: (a) alert analysis rates, (b) quality of analysis, which is measured by the number of false negatives reported (error rates), and (c) hiring costs (base rate per hour).

A CSOC receives a number of alert analysis jobs simultaneously from various clients. Assignment of alert analysis jobs to analysts is a continuous process. In the approach presented in this research, all the alert analysis jobs that arrive in the previous time-stamp (for instance, last hour) are collected at the beginning of the current time-stamp (current hour) and assigned to the selected analysts in real-time (within a few minutes) from the available pool of analysts. The above action to select and assign analyst to jobs is non-trivial because of the numerous combinations that are possible, and the exponential computational time needed to final optimal solutions, which makes the problem NP-Hard. The research objective is to develop an optimized decision-support tool for selecting and assigning on-demand analysts for processing alert analysis jobs such that the SLA requirements are met by minimizing the cost of operation, and the solution is computable and implementable in real-time.

1.1 Modeling Assumptions

(1) Based on our conversations with the CSOC operators, we found that the base rate per hour of hiring an analyst varies based on the number of hours for which an analyst is hired. For instance, analysts offer a discount of 10% if they are hired for more than 4 hours and 15% for 8 or more hours in an on-demand service model. However, it may be unrealistic to expect quality work from security analysts if they are assigned work that takes more than a threshold number of hours in this model. Based on our discussions with the CSOC operators, the maximum time interval for which an analyst could be hired for is no more than 12 hours. Also, for security and accountability reasons, an analyst is allowed to work on only one job at a time, however, a job could be assigned to multiple analysts to reduce analysis time, error rate, or cost of analysis.

(2) Alerts vary in their types and correspondingly need different length of time for investigation. Alerts could also be correlated. The article assumes that job creation is a preprocessing step that has already taken into account the type and quantity of alerts to estimate the threshold time for completing the job, to be used later in the optimization model.

The research presents two models for solving the optimization problem of minimizing the total cost of processing all alert analysis jobs subject to SLA and analyst constraints. The first model is a single exact optimization model, which takes into account individual attributes of all the available analysts and the attributes of all alert analysis jobs waiting for processing, subject to their respective SLA requirements. In the experiments and analysis of results section (Section 5), it is demonstrated that the exact model is computationally very expensive and will not yield decisions in a timely manner for a decision-maker at the CSOC. One of the innovations in this research is to develop a second model that uses a two-step optimization approach by sequentially solving two mathematical models to obtain good solutions in a computationally efficient manner that can then be implemented by a decision-maker at the CSOC. The second model relies on the following property:

1.2 Property of an On-demand Cybersecurity Resources Pool

A CSOC has access to a large pool of analysts in an on-demand SECaaS model, who are carefully vetted and whose performances are monitored continuously. As a result, there exists a strong correlation between the values of the analyst attributes in the outsourcing business model of a CSOC, i.e., higher throughput alert analysis rates and lower error rates strongly correlate with higher hiring cost (base rate per hour). Similarly, lower throughput alert analysis rates and higher error rates correlate with lower hiring base rate per hour. This is an important property that intrinsically groups analysts with similar attribute values into respective expertise-level categories.

The second model takes into account the above property of an on-demand SECaaS model comprising a large pool of resources, where the analyst resources are categorized into expertise-level categories based on the values of their attributes. The optimization problem is decomposed into two steps. First, the minimum number of analysts required and the duration of time (time interval) for which they are hired is determined for each expertise-level j such that the SLA requirements of all jobs are met. In this step, individual analysts are not chosen, instead the number of analysts needed from each expertise-level j (demand per expertise-level j) is determined. Second, individual analysts from the available analysts pool are selected to meet the demand from each expertise-level j for all given time intervals such that the total cost of processing all alert analysis jobs is minimized. Within a given j, the optimization model will select the available analysts with the lowest cost until all the demand per expertise-level j is met.

The contributions of this research are as follows. First, the article demonstrates an exact optimization model for selecting individual analysts from a pool and assigning them to jobs while minimizing the cost, and it shows that it is not a computationally efficient way to perform alert analysis outsourcing. In general, this is a very important insight for organizations who wish to outsource operations that involve selection and assignment of resources to tasks. Second, the article presents an efficient two-step optimization approach that exploits the property of expertise-level categorization, which is used for Step 1: to determine the analyst demand per expertise-level, and in Step 2: to fulfill the analyst demand with individual analysts such that the total cost is minimized. The two-step optimization model is scalable and practical to implement in real-time, and it can be extended to outsourcing problems in other domains where decision-making consists of both selection and assignment of resources to tasks with specific objectives. Third, in the experiments, a significant improvement in the run-time of the model over the first model (exact optimization) is demonstrated along with the trade-off in the total cost. It is shown that the two-step sequential MIP model takes on average less than one second of run-time even for larger instances of problem size, whereas the exact optimization model takes over several hours to days of run-time. The trade-off, however, is the average cost of outsourcing operations using the two-step sequential MIP model, which is slightly higher than the exact optimization model but within 1% of its average cost. Clearly, the gain in computational run-time significantly overweighs the slightly higher cost of using the two-step sequential MIP model. Finally, the article provides valuable insights into the cost of operating a new business process outsourcing model for cybersecurity services.

The article is organized as follows. Section 2 presents the related literature in the field of cybersecurity and security-as-a-service business models. In Sections 3 and 4, the exact and the two-step sequential MIP models are presented. Also, model parameters, mathematical formulation for the optimization models, computational complexity, and algorithms are discussed. Section 5 presents the numerical experiments performed using both the models along with a greedy baseline method and their respective results. Last, in Section 6, the conclusions and future work are presented.

2 RELATED LITERATURE

Clients have SLAs with the CSOC provider for their security needs. While a CSOC offers many services, alert handling is the top service offered by the CSOC [D'Amico and Whitley 2008] to the clients. A CSOC monitors alerts, which are generated from the intrusion detection systems (using signature or anomaly-based techniques [Scarfone and Mell 2007; Crothers 2002; Bejtlich 2005; Rasoulifard et al. 2008]) in the clients' networks. Though machine learning techniques [Sommer and Paxson 2010] and data mining techniques [Barbará and Jajodia 2002] have been utilized for accurate identification of suspicious activities in the form of alerts, the automated filters still generate a large volume of alerts (many of them being innocuous), which require a manual inspection to identify the significant alerts. The significant alerts are categorized under categories 1, 2, 4, or 7 depending upon the severity [CIO 2008].

Strategies for effective CSOCs, in regards to people, processes, and technology, are described in Zimmerman [2014]. A framework for CSOC as a service for cloud computing environments to protect against cyber-attacks and to comply with security and regulatory policies is proposed by Alruwaili and Gulliver [2014]. SECaaS models are optimized mainly from a customer viewpoint in recent literature. For instance, Chaisiri et al. [2015] proposed an optimization model for customers purchasing cloud-based services to optimally allocate security services to SECaaS providers. A stochastic programming model with a three-stage recourse is proposed in that study to balance the costs for service allocation and cyber insurance policies. In a recent study by Liu et al. [2017] it was found that a university with centralized Information Technology decision-making that opts for outsourcing their information security has a lower likelihood of suffering from a cybersecurity breach.

A CSOC hires analysts who have various attributes such as alert analysis rate, hiring cost, and quality of analysis to process the alert analysis jobs [Altner et al. 2017; Ganesan et al. 2017]. Pioneering work by Ganesan et al. [2017] focused on the in-house staffing and scheduling of analysts. The work in Ganesan et al. [2016] and Shah et al. [2018] focused on allocation of additional resources to supplement the in-house cyber workforce. However, operating a CSOC with an in-house team of analysts is prohibitively expensive due to the uncertainty of the demand for alert handling, especially for CSOCs with multiple clients. Issues such as burnout that affect analyst effectiveness in CSOCs are studied in Sundaramurthy et al. [2015]. The following presents some of the well-known examples in scheduling and their differences with this research.

Traditional scheduling heuristics focus on objectives such as minimizing completion time or the weighted tardiness [Pinedo 2009], where release time, processing time, and due dates are given. For example, job shop scheduling is focused on the above objectives and attempt to schedule machines with jobs that have to follow a machine sequence. The article differs from the above scheduling methods in the following manner. The alert management task at a CSOC consists of many domain-specific characteristics such as the attributes of alerts (jobs) and that of the cybersecurity analysts (machines). Alert jobs differ from each other in terms of their importance (captured as target completion time) and sensitivity to error, which is a unique feature unlike traditional job shop settings, while being analyzed (captured as a target error rate). Also, the time taken to analyze a job depends on both the types and the number of alerts in the job. Another important difference is the dynamic setting in which alerts are generated and the critical need to find analysts that are not only cost effective but also have credentials to analyze them. In contrast, job shops in general, have complete information on jobs and machines prior to their scheduling, which make the decision to allocate machines static at a given point in time for the following shift or day of operation. The above makes the outsourcing research problem of allocating alerts to analysts in a cost effective manner challenging and different when compared to traditional scheduling tasks. Applying heuristics at a given point in time to schedule the available jobs and machines is a one-time static and highly myopic decision, which is not suitable to minimize cost in the long run. Hence, the research focuses on mathematical optimization methods, which are in general computationally very hard even for job shop scheduling problems, however, the article goes even deeper in developing a two-stage approach to mitigate the computational complexity and significantly improve the scalability of the approach. To the best of the authors' knowledge, outsourcing decision-making from a cybersecurity provider's viewpoint to minimize the total cost of alert analysis has not been studied in the literature.

Broadcast data delivery systems continuously deliver data from a server to the user community. Broadcast scheduling, which has been an active area of research, determines when and what a server needs to broadcast to the users. Vaidya and Hameed [1999] proposed algorithms that minimize the wait time encountered by the users. Su et al. [1999] formulated the scheduling problem as a deterministic Markov Decision Process (MDP) to minimize the average response time of user's requests. Yeo et al. [2002] proposed a sequential vertex coloring algorithm to minimize the system delay in an ad-hoc network where all the users share a single channel and the time overlap of two or more packet receptions must be avoided. The broadcast scheduling problem is NP-complete and as a result, there are heuristic approaches proposed by researchers. For instance, Wang and Ansari [1997] proposed a mean field annealing-based algorithm for scheduling and Salcedo-Sanz et al. [2003] solved the scheduling problem in two steps, where they first tackle the hardest constraints and then optimize the throughput. Hu et al. [2015] uses a dynamic index strategy for an on-demand data broadcasting schedule, where the focus is to reduce the user's drop ratio and waiting time. Pattanayaka and Kumar [2019] proposed a genetic algorithm for multiple-input multiple-output broadcast scheduling to reduce the complexity of search. More recently, Qiu et al. [2018]

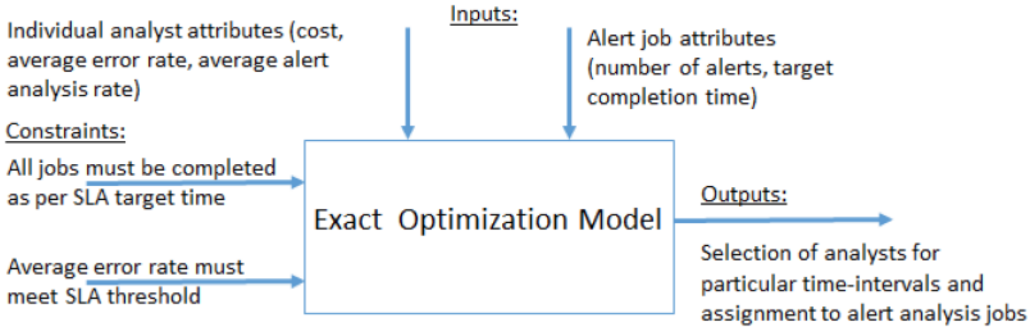
Inputs:
Individual analyst attributes (cost, average error rate, average alert analysis rate)
Alert job attributes (number of alerts, target completion time)
Constraints:
All jobs must be completed as per SLA target time
Average error rate must meet SLA threshold
Exact Optimization Model
Outputs:
Selection of analysts for particular time-intervals and assignment to alert analysis jobs

Fig. 1. Exact optimization model framework.

studied the problem of base station access capacity and a high server concurrency in a real-time on-demand data broadcast scheduling and proposed a method to improve the broadcast efficiency. The research problem presented in this article is different from the broadcast scheduling problems as follows. The broadcast scheduling problems deal with capacity and concurrency issues, whereas in this research problem, we focus on selecting the right types and numbers of resources and pair them with jobs. While the objectives in the broadcast scheduling problems are minimizing the wait-time, minimizing the average response time and maximizing the throughput of broadcasting, the objective of the research problem presented in this article is to minimize the cost of alert management for an on-demand CSOC subject to the SLAs between the CSOC and the clients.

Outsourcing services for profit maximization (or cost minimization) by businesses has been studied in published literature for various fields. Li et al. [2009] proposed a column generation method for making outsourcing decisions for transportation services, which included the in-house pre-scheduling of rides and staffing. A game theoretic model is presented in Zhu [2016] for optimal outsourcing contracts for buyers and suppliers by taking into consideration the three main factors for outsourcing management: cost, quality and time. Co-sourcing decisions by outsourcing calls in a call center to minimize the long-run average costs with linear staffing cost per unit time and linear costs with outsourcing is studied in Koçağa et al. [2015]. A mathematical model for minimizing the total cost of airline in-house and outsourced maintenance is proposed in Bazargan [2016]. Optimizing cost of processing all the alert analysis jobs such that the SLA targets are met for all the clients by selecting from a large pool of resources (analysts) with varying attributes has not been addressed before, to the best of the authors' knowledge.

3 AN EXACT OPTIMIZATION MODEL

An exact optimization model for selection and assignment of on-demand analysts to alert analysis jobs is shown in Figure 1. The inputs to the model takes into account the attributes of all the available analysts, and the unprocessed alert analysis jobs that are awaiting analyst assignment. The SLA requirements are treated as constraints for each of the jobs that arrive at the CSOC from its respective clients. The exact optimization model formulation and its computational complexity are presented next.

3.1 Formulation for Exact Optimization Model

The exact model formulation consists of the objective function, constraints, mathematical model, and outputs, which are explained in detail below. The notations for the parameters of the exact optimization model are described in Table 1.

Table 1. Definitions of Notations

Notation	Definition
Indices	
i	Time-interval index
a	Analyst identity
j	Expertise-level identity
k	Job identity
Inputs	
I	Maximum time interval for which an analyst can be hired
A	Total number of analysts available
J	Total number of expertise levels
K	Total number of jobs
A_j	Total number of analysts available per expertise-level j
N_k	Total number of alerts in job k
$C_{i,a}$	Hiring cost of analyst a for time interval i
$AC_{i,j}$	Average cost of hiring an analyst from level j for time interval i
$M_{i,a}$	Average number of alerts analyzed by analyst a in time interval i
$AM_{i,j}$	Avg. number of alerts analyzed by analysts from level j for time interval i
E_a	Average error rate for alert analysis of analyst a
AE_j	Average error rate for alert analysis of analysts from level j
R_k	SLA threshold for the error rate for job k
T_k	SLA target completion time for job k
Variables	
$x_{i,j,k}$ (Integer)	Number of analysts required from level j for time interval i for job k
$v_{i,j}$ (Integer)	Total number of analysts required from expertise-level j for time interval i
$y_{i,j,a}$ (Binary)	1 if analyst a from level j is selected for time interval i, and 0 otherwise
$z_{i,a,k}$ (Binary)	1 if analyst a is assigned to job k for time interval i, and 0 otherwise

3.1.1 *Input Parameters.*

- Total number of available analysts, A.
- Total number of jobs, K.
- Total number of alerts in job k, N_k.
- Cost of hiring analyst a for time interval i, $C_{i,a}$.
- Average number of alerts analyzed by analyst a in time interval i (obtained from historical data), $M_{i,a}$.
- Average error rate for alert analysis of analyst a (obtained from historical data), E_a.
- SLA threshold for the error rate for job k, R_k.
- SLA target completion time for job k, T_k.

3.1.2 *Decision Variables.*

- $z_{i,a,k} = 1$ if analyst a is assigned to job k for time interval i, and 0 otherwise.

3.1.3 Objective Function. The objective of the exact optimization model is to minimize the total hiring cost of the analysts for processing all the alert analysis jobs by (1) assigning jobs to analysts, and (2) selecting the time interval (duration) for which analysts are hired subject to meeting the

SLA constraints of the jobs. It is defined as follows:

$$w = Min \sum_{i} \sum_{a} \sum_{k} C_{i,a} * z_{i,a,k}. \tag{1}$$

3.1.4 Constraints. The constraints for the exact optimization model are as follows: Each job must be completed, which is given by

$$\sum_{i} \sum_{a} z_{i,a,k} * M_{i,a} \geq N_k \ \forall k. \tag{2}$$

The above constraint translates to assigning analysts such that all the alerts in a respective job are analyzed. Job completion time must meet the SLA target, which is given by

$$\sum_{i>T_k} \sum_{a} z_{i,a,k} = 0 \ \forall k. \tag{3}$$

The above constraint translates to not selecting any analyst for time interval greater than the target completion time of the respective jobs. Average error rate must meet the SLA threshold for the job, which is given by

$$\sum_{i} \sum_{a} z_{i,a,k} * M_{i,a} * E_a \leq R_k * N_k \ \forall k. \tag{4}$$

An analyst must be assigned to, at most, one job, which is given by

$$\sum_{i} \sum_{k} z_{i,a,k} \leq 1 \ \forall a. \tag{5}$$

3.1.5 Outputs. The outputs from the exact optimization model are as follows:

- Total hiring cost of the analysts for processing all the alert analysis jobs.
- Assignment of jobs to analysts.
- Time intervals for which analysts are hired for the respective jobs.

3.2 Algorithm for the Exact Optimization Model

Algorithm 1 provides the implementable steps for the exact optimization model described above.

Theorem 1. *The decision-problem of minimizing the total cost of assigning alert analysis jobs to cybersecurity analysts such that the SLA constraints are satisfied has a complexity of* 2^{A*I*K}.

Proof. In a Generalized Assignment Problem (GAP), let G be the total number of items, and L be the total number of bins available. Each bin l has an upper-bound on capacity u_l. Each pair of bin l and item g, has a weight $w_{l,g}$ and a profit $p_{l,g}$ associated with it. Then, the decision-problem is to find the maximum total profit P that could be attained by assigning items to the bins such that the total sum of the weight of items in each bin does not exceed the capacity of each bin (represented by the upper-bound u_l). This problem is known to be NP-hard. GAP is the closest known algorithm that can be reduced to the problem presented in this article.

In the research problem presented here, G represents the total number of jobs to process, and L represents the total number of analysts available. Each analyst has an alert analysis rate, which determines the average number of alerts that could be analyzed in each time interval for any job. This number is the value for the weight, $w_{l,g}$, for the analyst selected for a particular time interval and the respective job assigned to the analyst. The analyst has varying hiring costs associated with each job, which is based on the number of alerts that are needed to be analyzed in that job and the time interval for which an analyst is hired. The profit $p_{l,g}$ represents this negative cost between the analyst l and the job g. In this problem, there exists a constraint based on reality where an

ALGORITHM 1: Exact Optimization Algorithm for Minimizing Alert Analysis Cost.

```
Input: Total number of available analysts, $A$, Total number of jobs, $K$, Total number of alerts in job $k$, $N_k$,
       Cost of hiring analyst $a$ for time interval $i$, $C_{i,a}$, Expected number of alerts analyzed by analyst $a$ in
       time interval $i$, $M_{i,a}$, Average error rate for alert analysis of analyst $a$, $E_a$, SLA threshold for the
       error rate for job $k$, $R_k$, SLA target completion time for job $k$, $T_k$.
Output: Assignment of jobs to analysts and time intervals for which analysts are hired for the respective
        jobs, $z_{i,a,k}$ $\forall i, a, k$.
/*Initiate a solution search using an integer programming solver */ repeat
    for a set of $z_{i,a,k} = 1$, /*Potential soln. obtained in a search*/ Check for feasibility: do
        $\sum_i \sum_a z_{i,a,k} * M_{i,a} \geq N_k$ $\forall k$ /*Job completion*/ $\sum_{i>T_k} \sum_a z_{i,a,k} = 0$ $\forall k$ /*Job completion time
        SLA*/ $\sum_i \sum_a z_{i,a,k} * M_{i,a} * E_a \leq R_k * N_k$ $\forall k$ /*Error rate SLA*/$\sum_i \sum_k z_{i,a,k} \leq 1$ $\forall a$ /*Analyst to at
        most 1 job*/
    end
    if Feasible then
        $w = Min \ \sum_i \sum_a \sum_k C_{i,a} * z_{i,a,k}$/*Min total cost*/
    end
until Stopping criteria /*Optimal value for w is found*/;
return $z_{i,a,k}$ $\forall i, a, k$.
```

analyst cannot be hired for a time interval greater than I. Hence, u_l represents this upper-bound on the number of alerts that can be analyzed by analyst l. However, unlike a GAP, an alert analysis job can be assigned to more than one analyst by allocating different portions of the job (number of alerts in a job) to different analysts. Hence, by representing the pair of analyst (selected for the respective hired time interval) and the job as a binary decision variable, the time complexity of finding the best (job, analyst) pairs to obtain the minimum cost (maximum profit) of processing all the alert analysis jobs is equivalent to choosing from 2^{H*G} possibilities, where $H = L * I$. □

In this research problem with on-demand resources, it is assumed that a large pool of analysts is available. Hence, even with $I = 12$, $G = 20$, and $L = 60$ (used for one of the experiments), $2^{60*12*20}$ is a large number, and even a relatively small practical instance of the problem remains computationally difficult to solve.

4 A TWO-STEP SEQUENTIAL MIXED-INTEGER PROGRAMMING MODEL

A new two-step approach to solving the research objective presented in this work is detailed in this section. This approach takes into consideration the property of an on-demand service model as explained in the Introduction (Section 1.2) that groups analysts into various expertise-level categories. For each expertise-level category, an average value is assigned to their respective attributes (cost, alert analysis rate, and error rate). The problem is decomposed into two steps: Mixed-integer programming (MIP) models 1 and 2. The two models are solved individually to optimality in a sequential manner. Figure 2 shows the two-step sequential framework presented in this section. The objective of MIP model 1 is to determine the total number of analysts that are required from each expertise-level category for all possible time intervals. The output of MIP model 1 is then provided as an input to the second model (MIP model 2). The objective of MIP model 2 is to minimize the total hiring cost of the analysts for processing all the alert analysis jobs by selecting analysts from the available pool to meet the demand for the number of analysts required from each expertise-level category, as determined by MIP model 1.

4.1 Formulation for MIP Model 1

The notations for the parameters of the model are described in Table 1, and the mathematical formulation is described below.

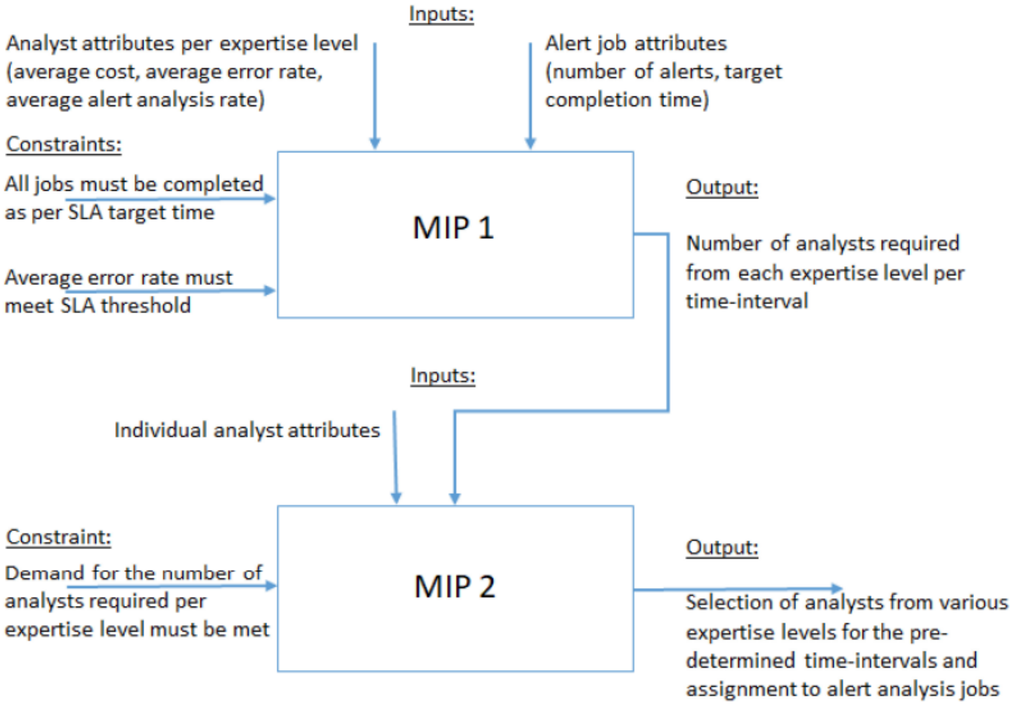
Inputs:
Analyst attributes per expertise level (average cost, average error rate, average alert analysis rate)
Alert job attributes (number of alerts, target completion time)
Constraints:
All jobs must be completed as per SLA target time
Average error rate must meet SLA threshold
MIP 1
Output:
Number of analysts required from each expertise level per time-interval
Inputs:
Individual analyst attributes
Constraint:
Demand for the number of analysts required per expertise level must be met
MIP 2
Output:
Selection of analysts from various expertise levels for the pre-determined time-intervals and assignment to alert analysis jobs

Fig. 2. Two-step sequential mixed-integer programming model framework.

4.1.1 *Input Parameters.*

- Total number of expertise levels, J.
- Average cost of hiring an analyst from expertise-level j for time interval i, $AC_{i,j}$.
- Average number of alerts analyzed by an analyst from expertise-level j for time interval i, $AM_{i,j}$.
- Average error rate for alert analysis of an analyst from expertise-level j, AE_j.
- Total number of alerts in job k, N_k.
- SLA threshold for the error rate for job k, R_k.

4.1.2 *Decision Variables.*

- $x_{i,j,k}$ is an integer variable that represents the number of analysts required from expertise-level j for time interval i for job k.
- $v_{i,j}$ is an integer variable that represents the total number of analysts required from expertise-level j for time interval i.

4.1.3 *Objective Function.* The objective of MIP model 1 is to minimize the expected total hiring cost of the analysts for processing all the alert analysis jobs by selecting the optimal number of analysts that are required from each expertise-level category for all possible time intervals. It is defined as follows:

$$q = Min \sum_i \sum_j AC_{i,j} * v_{i,j}. \tag{6}$$

4.1.4 Constraints. The constraints for MIP model 1 are as follows: Each job must be completed, which is given by

$$\sum_i \sum_j x_{i,j,k} * AM_{i,j} \geq N_k \ \forall k. \tag{7}$$

Job completion time must meet the SLA target, which is given by

$$\sum_{i>T_k} \sum_j x_{i,j,k} = 0 \ \forall k. \tag{8}$$

Expected error rate must meet the SLA threshold for the job, which is given by

$$\sum_i \sum_j x_{i,j,k} * AM_{i,j} * AE_j \leq R_k * N_k \ \forall k. \tag{9}$$

Calculating the number of analysts required per expertise-level for each time interval, which is given by

$$\sum_k x_{i,j,k} = v_{i,j} \ \forall i, j. \tag{10}$$

4.1.5 Output. The output from MIP model 1 is the number of analysts required from each expertise-level to meet the demand of alert analysis jobs per time interval, $v_{i,j}$.

4.2 Formulation for MIP Model 2

The notations for the parameters of the model are described in Table 1, and the mathematical formulation is described below.

4.2.1 Input Parameters.

- Total number of available analysts per expertise-level j, A_j.
- Cost of hiring analyst a for time interval i, $C_{i,a}$.
- Total number of analysts required from level j for time interval i (obtained as an output from MIP model 1), $v_{i,j}$.

4.2.2 Decision Variables.

- $y_{i,j,a} = 1$ if analyst a from level j is selected for time interval i, and 0 otherwise.

4.2.3 Objective Function. The objective of MIP model 2 is to minimize the total hiring cost (actual) of the analysts by selecting the optimal number of analysts to meet the demand from each expertise-level category in the given time interval for processing all the alert analysis jobs. It is defined as follows:

$$p = Min \sum_i \sum_j \sum_a^{A_j} C_{i,a} * y_{i,j,a}. \tag{11}$$

4.2.4 Constraints. The constraints for MIP model 2 are as follows: An analyst must be assigned to, at most, one job, which is given by

$$\sum_i y_{i,j,a} \leq 1 \ \forall j, a. \tag{12}$$

Demand for the number of analysts required from each expertise-level per time interval must be met, which is given by

$$\sum_a^{A_j} y_{i,j,a} \geq v_{i,j} \ \forall i, j. \tag{13}$$

ALGORITHM 2: Two-step Sequential Mixed-Integer Programming Algorithm for Minimizing Alert Analysis Cost.

```
Input: Total number of available analysts, $A$, Total number of jobs, $K$, Total number of alerts in job $k$, $N_k$,
       Cost of hiring analyst $a$ for time interval $i$, $C_{i,a}$, Average error rate for alert analysis of analyst $a$, $E_a$,
       SLA threshold for the error rate for job $k$, $R_k$, SLA target completion time for job $k$, $T_k$, Total number
       of expertise levels, $J$, Average cost of hiring an analyst from expertise-level $j$ for time interval $i$,
       $AC_{i,j}$, Average number of alerts analyzed by an analyst from expertise-level $j$ for time interval $i$,
       $AM_{i,j}$, Average error rate for alert analysis of an analyst from expertise-level $j$, $AE_j$, Total number of
       available analysts per expertise-level $j$, $A_j$.
Output: Selection of analysts from the various expertise levels for the required time intervals, $y_{i,j,a} \forall i, j, a$,
        and assignment to jobs, $x_{i,j,k} \forall i, j, k$.
/* Initiate a solution search using an integer programming solver */repeat
    for a set of $x_{i,j,k}$, /*Potential soln. obtained in a search*/ Check for feasibility: do
        $\sum_i \sum_j x_{i,j,k} * AM_{i,j} \geq N_k \ \forall k$ /* Job completion */ $\sum_{i>T_k} \sum_j x_{i,j,k} = 0 \ \forall k$ /* Job completion time
        SLA */ $\sum_i \sum_j x_{i,j,k} * AM_{i,a} * AE_j \leq R_k * N_k \ \forall k$/* Average error rate SLA */$\sum_k x_{i,j,k} = v_{i,j} \ \forall i, j$ /*
        Number of analysts per expertise-level */
    end
    if Feasible then
        $q = Min \ \sum_i \sum_j AC_{i,j} * v_{i,j}$/* Min expected total cost */
    end
until Stopping criteria /*Optimal value for q is found*/;
/* Initiate a solution search using an integer programming solver */repeat
    for a set of $y_{i,j,a} = 1$, /* Potential soln. obtained in a search */ Check for feasibility: do
        $\sum_i y_{i,j,a} \leq 1 \ \forall j, a$ /* Analyst to at most 1 job */ $\sum_a^{A_j} y_{i,j,a} \geq v_{i,j} \ \forall i, j$ /* Demand for number of
        analysts per expertise-level */
    end
    if Feasible then
        $p = Min \ \sum_i \sum_j \sum_a^{A_j} C_{i,a} * y_{i,j,a}$/*Min total cost (actual)*/
    end
until Stopping criteria /* Optimal value for p is found */;
return $y_{i,j,a} \ \forall i, j, a$ and $x_{i,j,k} \ \forall i, j, k$.
```

4.2.5 Output. The output from the MIP model 2 is the selection of individual analysts from the various expertise-level categories for the required time intervals.

4.3 Algorithm for Two-step Sequential MIP Model

Algorithm 2 provides the implementable steps for the two-step sequential MIP model described above.

4.4 Reduction in Computational Time

The computational complexity of the two-step sequential MIP algorithm presented above is *Max* $(n^{J*I}, 2^{J*I*K})$, where n takes an integer value between 0 and the maximum number of analysts that could be hired, and J is significantly smaller than A. It is to be noted that the actual run-time of a problem instance depends on the input data. By utilizing the property of an on-demand cybersecurity resources pool (as explained in Section 1.2) in the two-step sequential MIP model formulation, the expertise-level attribute values had significant variance among each other. As a result, the MIP model 1 is able to select the near-optimal number of analysts from each expertise-level with a significantly lower computational time unlike the exact model where the variances in the analyst attribute values among certain analysts were very small, which would take computationally longer time to evaluate for a larger set of candidate solutions.

It should be noted that MIP model 2 can also be implemented using a sorting heuristic, however, the article presents only the results from the MIP model 2. In the sorting heuristic, all combinations of (i, j) pairs are represented as bins with all analysts available, and are further sorted in an increasing order of costs in each of the bins. For each expertise-level j, analysts can be picked from the top starting from the (i, j) bin with the largest time interval i until the demand $v_{i,j}$ is met. Once an analyst is selected from a bin, the same analyst cannot be selected from the remaining bins. This process is followed until all time intervals are covered in a decreasing order of expertise levels.

4.5 Convergence and Optimality

The exact optimization model and the two-step approach are modeled as mixed-integer programming models (MIPs), which are proven to converge to optimal solutions [Chen et al. 2010]. However, the exact model is optimal but computationally expensive and time consuming. In the two-step model, the result of MIP model 1 is optimal to the given average values, which are assigned to the respective attributes (cost, alert analysis rate, and error rate). The MIP model 2 is also optimal. However, when taken together, the final solution of the two-step model in comparison to the exact model is only near-optimal to the optimal solution obtained from the exact model but it is computationally faster than the exact model. This trade-off between optimality and computational time savings is due to the average values taken in MIP model 1 as given above.

5 EXPERIMENTS AND ANALYSIS OF RESULTS

In this section, the setup for conducting experiments is described first, followed by the analysis of results, which are obtained by implementing the models developed in this research. Results from the scalability experiments are described later in the section.

5.1 Experimental Setup

An experiment with 20 jobs and 60 analysts is considered for demonstrating the capabilities of the optimization models along with the computational time needed. It is reminded that the alert analysis jobs that arrived in the previous hour are considered, and the decision to select analysts and allocate them to the jobs is done in the current hour. Hence, results from one instance of decision making are shown below, which is repeated for each hour of decision making.

The attribute values for the jobs and analysts in the experiments are selected as follows. The number of alerts in each job is chosen randomly between 1,000 and 6,000 alerts, and the target completion time is chosen randomly between 4 and 12 hours. Table 2 shows a sample of attribute values of the jobs that are selected for a single simulation run. There are three categories (expertise levels) of analysts created: senior, intermediate, and junior. The following attributes are assigned to each expertise-level: an average rate of alert analysis per hour, an average error rate, and an average base rate per hour. It is to be noted that the cost of hiring an analyst for a time interval takes into account the discount on the respective average base rate per hour if hired for more than 4- or 7-hour time interval. By definition of the property given in the introduction, a senior analyst is said to have higher throughput, lower error rate, and higher cost per hour. Likewise, a junior analyst is said to have lower throughput, higher error rate, and lower cost per hour. The intermediate analyst is said to be in-between the senior and junior for their attributes values. Similar to discounts offered in other crowd-sourced service industries, there is a discount on the base rate of analysts if they are hired for more number of hours (10% discount if hired for more than four hours and 15% discount if hired for eight or more hours). These numbers were chosen from literature [Altner et al. 2017; Ganesan et al. 2017] and our conversations with the CSOC operators.

Table 2. Input: Alert Job Attributes for a Single Sample Simulation Run

Job (k)	# of Alerts (N_k)	Target Completion Time (T_k, in hours)
1	1,685	8
2	3,449	6
3	2,755	10
4	4,895	12
5	4,311	10
6	4,543	10
7	2,002	6
8	4,257	7
9	3,890	9
10	2,372	11
11	3,414	12
12	3,015	7
13	1,562	8
14	2,647	7
15	3,724	11
16	4,772	11
17	2,448	11
18	2,789	6
19	2,327	9
20	1,493	7

Table 3. Input: Average Values of Analyst Attributes

Attributes	Senior	Intermediate	Junior
Avg. rate of alert analysis/hour	360 alerts	240 alerts	180 alerts
Avg. error rate	5%	7.5%	9%
Avg. base rate/hour	\$75	\$50	\$38
Discount on base rate (for 5–7 hours)	10%	10%	10%
Discount on base rate (for ≥ 8 hours)	15%	15%	15%

Table 3 shows the average of the attribute values of the analysts for the various expertise levels. All 60 analysts were equally divided among the three expertise levels. Next, the values of the expertise-level attributes were varied for each of the individual analysts by randomly choosing values that were within +/− 5% of the average values, i.e., each analyst from a senior expertise-level is assigned a random value between 342 and 378 (360 +/− 5% of 360) for the attribute value of alert analysis rate per hour, 5 +/− 5% of 5 for the error rate, and 75 +/− 5% of 75 for the base rate per hour. Similar values were randomly drawn for each individual intermediate and junior analyst. The experiments were conducted using the Julia language and Gurobi solver. In what follows, the results from three models are presented: (1) a baseline myopic greedy model, (2) the exact optimization model, and (3) the two-step sequential MIP model. It should be noted that 50 simulation runs were conducted, and the input data on the jobs and analyst attributes for all three models across a single run were kept the same. The input data, however, varied between the runs.

Table 4. Output: Comparison of Alert Analysis Minimum Cost, Computational Time, and Number of Runs with SLA Violations between the Models Across 50 Simulation Runs

Model	Minimum Cost ($)	Avg. Computational Time (s)	SLA violations
Baseline Model 1	11,770.85 (sub-optimal)	0.05	11 of 50 runs
Baseline Model 2	11,720.47 (sub-optimal)	0.07	5 of 50 runs
Exact Optimization	11,636.95 (optimal)	10,376.44 (3 hours)	None
Two-step Sequential MIP	11,674.50 (near-optimal)	0.12	None

5.2 Baseline Models

In this section, we setup two baseline models to compare with the proposed research models: (1) time of arrival-based selection model and (2) target completion time-based selection model with collection of jobs. Next, the two models are explained.

5.2.1 Baseline Model 1: Time of Arrival-based Selection Model. A baseline method is setup for comparison with the models developed in this research. In this method, alert analysis jobs have a time of arrival stamp, and are processed as and when they arrived, i.e., one job at a time (first-in-first-out policy). The decision on selecting the best set of analyst resources and their respective hiring time intervals for a job is taken such that it minimizes the cost of processing the job. The process is repeated for all the jobs, and it is to be noted that such decisions are made in a myopic manner (greedy selection), which does not consider saving the best analyst resources for a later time.

5.2.2 Baseline Model 2: Target Completion Time-based Selection Model with Collection of Jobs. Another baseline method is setup to compare with the models developed in this research. In this method, the alert analysis jobs that arrived in the previous hour are collected and the decision to select analysts and allocate them to the jobs is taken in the current hour. The list of alert analysis jobs is sorted based on the target completion time. Unlike the myopic model, which considers the time of arrival of the jobs as the selection criteria for assignment to the analysts, this method first accumulates the jobs that arrived in the last hour and then assigns analysts by picking one job at a time starting with the shortest target completion time. This method differs from the proposed research methods (exact optimization and two-step sequential MIP) as it considers the selection of analysts for assignment to one job at a time. Once the analysts are assigned to a job, the selected analysts are removed from the available analyst pool and the job is dequeued. This continues until all the jobs are assigned.

5.3 Results of the Baseline Model

Next the results obtained from using the baseline models are presented.

5.3.1 Time of Arrival-based Selection (Baseline Model 1). The experiment was conducted with 50 simulation runs. The input data was generated for each run, which consisted of 20 jobs (see a sample in Table 2), and 60 analysts whose attributes were generated using data in Table 3 with +/− 5% variation about their averages. The minimum cost of processing all the alert analysis jobs obtained from the 50 simulation runs is shown in Table 4 for the baseline models. It is to be noted that 11 out of the 50 simulation runs (22%) did not produce a feasible solution. It was observed that in these 11 experiments, all the analysts with faster analysis rates and lower error rates were assigned to the first few jobs among the 20 jobs that arrived in order of their time-stamp. Hence, when a new job arrived later in the same order that required a faster completion time or a lower error rate, the SLA requirements were violated because the selection of analysts from the remainder available analysts did not have the required attributes to meet the SLA. This demonstrated that the allocation of the best available analysts to the jobs in the order of their arrival based on their time-stamp is not optimal. Hence, it is better to collect the jobs from the previous hour of arrival and allocate the analysts. Next, the results from another baseline method is presented with such a collection of the jobs and assignment of analysts to jobs, which are selected one at a time based on their target completion times.

5.3.2 Target Completion Time-based Selection with Collection of Jobs (Baseline Model 2). Similar experiment to that of baseline model 1 was conducted. The minimum cost of processing all the alert analysis jobs obtained from the 50 simulation runs using the baseline model 2 is shown in Table 4. It was observed that 5 out of the 50 simulation runs (10%) did not produce a feasible solution. Due to the collection of jobs and then processing one job at a time with respect to a selection criterion, it is observed that the minimum cost obtained for processing all of them is slightly lower than that obtained using the baseline model 1, in which the jobs are not accumulated over the last hour. The target completion time violations that occurred using the baseline model 1 were avoided due to the earlier selection of the shortest target completion time jobs from the list of accumulated jobs. It was observed that the analysts with lower error rates were assigned to the earlier jobs, which resulted in violations due to higher error rates from the remaining analysts on the jobs that were selected later. Henceforth, the baseline models are not used for comparisons, because they are expensive, myopic, and result in many infeasible solutions (SLA violations). Next, the results from the exact optimization model are presented.

5.4 Results of the Exact Optimization Model

The results obtained from using the exact optimization model (Algorithm 1) on the experimental data set are presented here. By processing all jobs collected from the previous hour and the available analysts, both taken together in one mathematical model, an optimal solution is obtained for each of the 50 simulation runs. The input data that was generated for each run is the same as that of the baseline model. Table 4 shows the lowest optimal cost over 50 runs of processing all the alert analysis jobs, and the substantially high computational time taken to find the optimal solution for the exact optimization method. The cost was observed to be lower and optimal for the exact model when compared to the baseline myopic greedy model for each of the 50 simulation runs. This is an important observation, which justifies the collection of jobs from the previous hour, and the use of an optimization model over the baseline myopic greedy model for selecting and allocating available analysts to the jobs such that the cost of operation is minimized.

Table 5 shows the number of analysts required per time interval using the exact model. This table corresponds to the input data shown in Table 2, in which the highest target completion time is 12 hours for job #4 and job #11. Jobs are spread over the maximum time interval (12 hours), for which an analyst could be hired as shown in Table 5. By solving for all jobs at once, both analysts and their hiring time intervals are optimally selected such that the SLA constraints of all jobs are met by the exact optimization model. A similar observation was made for each of the 50 simulations; hence, there were no SLA violations. It is to be noted that a large number of analysts are chosen for time intervals that offer the best discount rates (such as 5-hour time-interval with 10 analysts, and 8-hour time interval with 8 analysts). With the hiring rate per hour being maximum

Table 5. Output: Comparison of Number of Analysts Required Per Time Interval

Time Interval (i)	# of Analysts (Exact Model)	# of Analysts (Sequential MIP)
1	11	9
2	1	2
3	1	0
4	0	0
5	10	12
6	6	2
7	1	2
8	8	11
9	3	3
10	1	0
11	0	1
12	0	0
Total	42	42

Table 6. Output: Number of Analysts Required Per Expertise-level Per Time Interval for Exact Optimization Model

Time Interval (i)	# of Senior Analysts	# of Intermediate Analysts	# of Junior Analysts
1	0	6	5
2	0	1	0
3	1	0	0
4	0	0	0
5	6	1	3
6	4	1	1
7	1	0	0
8	4	1	3
9	3	0	0
10	1	0	0
11	0	0	0
12	0	0	0
Total	20	10	12

(without discount) for the first 4 hours, it is observed that a majority of the analysts (11) are chosen for the minimum time interval of 1-hour to minimize the total cost. Furthermore, Table 6 also shows the number of analysts that are hired from various expertise-level categories for the different time intervals. The optimal solution for input data shown in Table 2 consists of 20 senior-level analysts, 10 intermediate-level analysts, and 12 junior-level analysts.

The trade-off in achieving an optimal solution is a substantial increase in computational time. It was observed that the average computational time taken across all 50 runs is close to 3 hours (as shown in Table 4) for the problem size of 20 jobs and 60 analysts at a single decision making instance of the outsourcing problem. Experiments conducted with a larger problem size at a single decision making instance, for example, 50 jobs and 240 analysts, and 100 jobs and 1050 analysts, took many hours to several days for the exact optimization algorithm to reach an optimal solution as shown in Table 7. The scalability results are discussed at the end of this section. Clearly, the

Table 7. Average Alert Analysis Cost and Computational Time of the Exact Optimization Model (50 Simulation Runs)

(Jobs, Analysts)	Avg. Cost ($)	Avg. Time (s)
(20,120)	10,942.86	16,200 (4.5 hours)
(50,240)	27,476.12	39,600 (11 hours)
(100,1050)	53,630.67	691,200 (8 days)

Table 8. Output: Number of Analysts Required Per Expertise-level Per Time Interval for the Two-step Sequential MIP Model

Time Interval (i)	# of Senior Analysts	# of Intermediate Analysts	# of Junior Analysts
1	3	3	3
2	2	0	0
3	0	0	0
4	0	0	0
5	8	2	2
6	0	1	1
7	0	0	2
8	5	4	2
9	1	2	0
10	0	0	0
11	1	0	0
12	0	0	0
Total	20	12	10

substantial increase in computational time renders the exact optimization model impractical to implement in real-time if the decision to select and allocate on-demand analyst to jobs are to be performed for each hour of CSOC operation. This observation motivates the need for the two-step sequential MIP model, whose results are presented next.

5.5 Results of the Two-step Sequential MIP Model

In this section, the results obtained from using the two-step sequential MIP model (Algorithm 2) on the experimental data set are discussed. The model is decomposed into two MIP models. The first model, MIP 1, determines the number of analysts that are needed from each expertise-level category for each time interval. The second model, MIP 2, selects the individual analysts that are available to meet the demand for analysts generated from MIP 1 model. The input data that was generated for each run is the same as that of the baseline model.

Table 4 shows that the lowest total cost of processing all alert analysis jobs across all 50 runs is slightly higher than that obtained from the exact optimization model. However, the cost is lower than that of the baseline model. Table 8 shows the number of analysts that are selected for each time interval. This table corresponds to the input data shown in Table 2. Similar to the selection of analysts in the exact optimization model, it is shown in Table 5 that a large number of analysts are selected for the time intervals, which offer the best discount rate per hour (such as the 5-hour time interval and the 8-hour time interval). It is also observed that the total number of analysts that are required by the two-step sequential MIP model is the same as the one obtained from the exact optimization model for the jobs data in Table 2. However, out of the 42 analysts needed using the

Table 9. Average Alert Analysis Cost and Computational Time of the Two-Step Sequential MIP Model (50 Simulation Runs)

(Jobs, Analysts)	Avg. Cost ($)	Avg. Time (s)
(20,120)	10,956.64	0.15
(50,240)	27,553.06	0.12
(100,1050)	54,120.58	0.58

Table 10. Average Alert Analysis Cost and Average Computational Time Comparison of the Two-step Sequential MIP Model Over the Exact Optimization Model

(Jobs, Analysts)	Avg. Increase in Cost (%)	Avg. Decrease in Computational Time (%)
(20,120)	0.126	very high
(50,240)	0.28	very high
(100,1050)	0.91	very high

sequential MIP approach, 20 of them are senior-level analysts, 12 of them are intermediate-level analysts, and 10 of them are junior-level analysts. One of the reasons for the differences in the total cost among the exact and sequential MIP models is due to the selection of 2 intermediate-level analysts in place of 2 junior-level analysts by the sequential MIP model. Another reason is that the exact optimization model selected only 1 senior-level analyst for the first 4 hours of the 12-hour time interval (being the most expensive base rate per hour), while the sequential MIP model selected 5 senior-level analysts during the same 4-hour time-interval window.

The computational time taken to obtain the solution was under one second. This is a significant saving in computing time over the exact model, which took on average about 3 hours of computing time for 20 jobs and 60 analysts. The results of the sequential MIP model shows that while there is a marginal increase in cost (within 1%), there are significant decreases (several folds) in the computation time over the exact optimization model, which made the sequential MIP model practical to implement. However, to prove that the MIP model can still remain practical for implementation in larger-sized problems (more number of jobs and analysts), the research conducted a scalability test. The results for the scalability experiments are discussed next.

5.6 Scalability Experiments

Several scalability experiments were conducted with larger input sizes to test whether the sequential MIP model is implementable in real-time. The input size for the number of jobs and number of analysts were increased gradually for the experiments, and each experiment was simulated 50 times to evaluate the sequential MIP model. The results for the average total cost and the average computational time to solve the larger-size problems across 50 simulation runs are shown in Table 7 for the exact optimization model, and in Table 9 for the two-step sequential MIP model. It is to be noted that the computational time taken to solve these experiments is very high for exact optimization model and under one second for the two-step sequential MIP model.

Table 10 shows the increase in average cost and the decrease in average computational time using the two-step sequential MIP model over the exact optimization model. Table 10 is created by comparing Tables 7 and 9. It is observed that while there is an increase of less than 1% in average cost among all the cases, there is a substantial decrease in average computational time for the two-step sequential MIP approach over the exact optimization approach. The exact optimization model took several days to solve to optimality in many of the scalability experiments.

6 DISCUSSION

Several useful insights were observed during the experiments, which are presented below.

- It is clear that a myopic greedy approach will result in higher cost in the long run. This is because the method attempts to allocate analysts as and when a job arrives, without any regard to the arrival of jobs in the future that might require more experienced analysts. Hence, a first-in-first-out policy with the allocation of the best analyst is an extremely myopic decision making policy, which will become expensive in the long-run.
- The above issue is resolved by (1) collecting jobs from the previous hour, (2) selecting analysts, and (3) allocating analysts to jobs, all within one exact optimization model. However, while optimal solution is obtained that minimizes the cost, the method is computationally impractical to implement in real-time.
- The above computational time issue is resolved by exploiting an important property. The cybersecurity outsourcing problem presented in this work has a unique feature where the variances in the attributes values of the resources are very small in certain groups. Hence, it takes longer to find an optimal set of analysts and their time intervals in the solution search due to the evaluation of a large number of candidate solutions. It is intuitive that the solution search will be faster if the variances in the attribute values of the candidates are larger. Hence by exploiting the property of pooling the cybersecurity resources (analysts) into fewer expertise-level categories with larger variances in their respective attribute values among the category levels, the sequential MIP model was able to find the solution significantly faster than the exact optimization model.
- There is a tradeoff between finding a solution faster and minimizing cost among the sequential two-step MIP model and the exact model. In general, the sequential MIP model is expected to have a higher cost over the exact model, because the former uses the average base rate per hour for determining the number of analysts per expertise-level (demand), followed by the determination of the exact cost during the selection of individual analysts for meeting the demand of each expertise-level, while the latter uses the exact base rate per hour directly. By doing so, the sequential MIP model introduces two errors in MIP model 1: (a) the use of average base rate instead of exact rate per hour to minimize cost and to calculate the number of individuals needed, and (b) rounding errors for each category of expertise-level (the number of individuals needed per expertise level are rounded to the next highest integer), whereas the exact model picks the exact number of individuals and their duration of hire that minimizes the cost.
- In all our experiments, similar to the average cost, there were no significant differences between the average error rates of the analysts chosen by the exact optimization model and the two-step sequential MIP model. It should be noted that the exact optimization model and the MIP 2 model of the two-step sequential MIP model can be optimized with multiple objectives of both minimizing cost and error rates whose weights can be varied ion the objective function. The focus of this article, however, remained on the minimization of cost.

7 CONCLUSIONS AND FUTURE WORK

The article presented a scalable and real-time implementable two-step sequential MIP model for outsourcing the alert analysis process at a CSOC. The model is shown to significantly outperform an exact optimization model in terms of computational time, although with an increase in cost, which is within 1% deviation from the cost of the exact optimization model. To our knowledge this is the first article that analyzes a business process outsourcing model for alert investigation at a COSC, which is scalable and cost effective, and can be implemented in real-time. Several insights are presented in the article, and the most notable is the exploitation of the property that clusters similar analysts into groups with a certain expertise-level. The property is shown to reduce the computational time of the two-step sequential MIP model, which makes it viable for real-time implementation. Scalability tests demonstrated the ability of the two-step sequential MIP model to handle large number of jobs and analysts, and obtain near-optimal solutions for the selection and allocation of analysts to jobs such that the cost of operation is minimized. As part of future research, the MIP model 2 can be extended to include an incentive or a constraint for the rotation of analysts. A CSOC would benefit by giving alert analysis jobs to analysts who have been idling for some time. Another extension is to optimize multiple objectives such as both cost and error rates of analysts chosen from the on-demand pool while meeting the SLA requirements. The two-step sequential MIP model is a paradigm shift in how future CSOCs can be operated using on-demand analysts to meet the growing demands for providing security to its clients through (1) higher throughput rate of alerts investigated, (2) higher quality of alert investigation (lower error rate), and (3) lower cost of operation.

ACKNOWLEDGMENT

The authors thank Dr. Cliff Wang of the Army Research Laboratory for suggesting this problem to us.

REFERENCES

Fahad F. Alruwaili and T. A. Gulliver. 2014. SOCaaS: Security operations center as a service for cloud computing environments. *Int. J. Cloud Comput. Serv. Sci.* 3, 2 (2014), 87–96.

Douglas S. Altner, Anthony C. Rojas, and Leslie D. Servi. 2018. A two-stage stochastic program for multi-shift, multi-analyst, workforce optimization with multiple on-call options. *J. Schedul.* 21, 5 (2018), 517–531. DOI: https://doi.org/10.1007/s10951-017-0554-9

Daniel Barbará and Sushil Jajodia (Eds.). 2002. *Application of Data Mining in Computer Security.* Advances in Information Security, Vol. 6. Springer.

Massoud Bazargan. 2016. Airline maintenance strategies—in-house vs. outsourced—an optimization approach. *J. Qual. Maint. Eng.* 22, 2 (2016), 114–129.

Richard Bejtlich. 2005. *The Tao of Network Security Monitoring: Beyond Intrusion Detection.* Pearson Education Inc.

Sivadon Chaisiri, Ryan K. L. Ko, and Dusit Niyato. 2015. A joint optimization approach to security-as-a-service allocation and cyber insurance management. In *Proceedings of the IEEE Trustcom/BigDataSE/ISPA*, Vol. 1. 426–433. DOI: https://doi.org/10.1109/Trustcom.2015.403

Der-San Chen, Robert Batson, and Yu Dang. 2010. *Applied Integer Programming.* Wiley, New York, NY.

CIO. 2008. *DON Cyber Crime Handbook.* Dept. of Navy, Washington, DC.

Tim Crothers. 2002. *Implementing Intrusion Detection Systems.* Wiley Publishing Inc.

Anita D'Amico and Kirsten Whitley. 2008. In *Proceedings of the Workshop on Visualization for Computer Security (VizSEC'07).* Springer, Berlin.

Rajesh Ganesan, Sushil Jajodia, and Hasan Cam. 2017. Optimal scheduling of cybersecurity analyst for minimizing risk. *ACM Trans. Intell. Syst. Technol.* 8, 4 (Feb. 2017).

Rajesh Ganesan, Sushil Jajodia, Ankit Shah, and Hasan Cam. 2016. Dynamic scheduling of cybersecurity analysts for minimizing risk using reinforcement learning. *ACM Trans. Intell. Syst. Technol.* 8, 1, (July 2016). DOI: https://doi.org/10.1145/2882969

Wenbin Hu, Cunlian Fan, Jiajia Luo, Chao Peng, and Bo Du. 2015. An on-demand data broadcasting scheduling algorithm based on dynamic index strategy. *Wireless Commun. Mobile Comput.* 15, 5 (2015), 947–965.

Yaşar Levent Koçağa, Mor Armony, and Amy R. Ward. 2015. Staffing call centers with uncertain arrival rates and co-sourcing. *Product. Oper. Manage.* 24, 7 (2015), 1101–1117.

Yihua Li, Xiubin Wang, and Teresa M. Adams. 2009. Ride service outsourcing for profit maximization. *Transport. Res. Part E: Logist. Transport. Rev.* 45, 1 (2009), 138–148. DOI: https://doi.org/10.1016/j.tre.2008.02.006

Che-Wei Liu, Peng Huang, and Henry Lucas. 2017. IT centralization, security outsourcing, and cybersecurity breaches: Evidence from the U.S. higher education. Retrieved from https://aisel.aisnet.org/icis2017/Security/Presentations/1.

Prabina Pattanayaka and Preetam Kumar. 2019. An efficient scheduling scheme for MIMO-OFDM broadcast networks. *AEU Int. J. Electron. Commun.* 101 (2019), 15–26.

Michael Pinedo. 2009. *Planning and Scheduling in Manufacturing and Services.* Springer, New York, NY.

Zhenyu Qiu, Wenbin Hu, and Bo Du. 2018. RPPM: A request pre-processing method for real-time on-demand data broadcast scheduling. *IEEE Trans. Mobile Comput.* 17, 11 (2018), 2619–2631.

Amin Rasoulifard, Abbas Ghaemi Bafghi, and Mohsen Kahani. 2008. Incremental hybrid intrusion detection using ensemble of weak classifiers. In *Advances in Computer Science and Engineering*. Springer, 577–584.

Sancho Salcedo-Sanz, Carlos Bousoño-Calzón, and Aníbal R. Figueiras-Vidal. 2003. A mixed neural-genetic algorithm for the broadcast scheduling problem. *IEEE Trans. Wireless Commun.* 2, 2 (2003), 277–283.

Karen Scarfone and Peter Mell. 2007. *Guide to Intrusion Detection and Prevention Systems (IDPS)*. Special Publication 800-94. NIST.

Ankit Shah, Rajesh Ganesan, Sushil Jajodia, and Hasan Cam. 2018. Dynamic optimization of the level of operational effectiveness of a CSOC under adverse conditions. *ACM Trans. Intell. Syst. Technol.* 9, 5, Article 51 (Apr. 2018), 20 pages. DOI:https://doi.org/10.1145/3173457

Robin Sommer and Vern Paxson. 2010. Outside the closed world: On using machine learning for network intrusion detection. In *Proceedings of IEEE Symposium on Security and Privacy*. 305–316.

Chi-Jiun Su, Leandros Tassiulas, and Vassilis J. Tsotras. 1999. Broadcast scheduling for information distribution. *Wireless Netw.* 5, 2 (1999), 137–147.

Sathya Chandran Sundaramurthy, Alexandru G. Bardas, Jacob Case, Xinming Ou, Michael Wesch, John McHugh, and S. Raj Rajagopalan. 2015. A human capital model for mitigating security analyst burnout. In *Proceedings of the 11th Symposium on Usable Privacy and Security (SOUPS'15)*. USENIX Association, 347–359.

Nitin H. Vaidya and Sohail Hameed. 1999. Scheduling data broadcast in asymmetric communication environments. *Wireless Netw.* 5, 3 (1999), 171–182.

Wang Gangsheng and Ansari Nirwan. 1997. Optimal broadcast scheduling in packet radio networks using mean field annealing. *IEEE J. Select. Areas Commun.* 15, 2 (1997), 250–260.

Jaehyun Yeo, Heesoo Lee, and Sehun Kim. 2002. An efficient broadcast scheduling algorithm for TDMA ad-hoc networks. *Comput. Operat. Res.* 29, 13 (2002), 1793–1806.

Xiaowei Zhu. 2016. Managing the risks of outsourcing: Time, quality, and correlated costs. *Transport. Res. Part E: Logist. Transport. Rev.* 90 (2016), 121–133. DOI:https://doi.org/10.1016/j.tre.2015.06.005 Risk Management of Logistics Systems.

Carson Zimmerman. 2014. *The Strategies of a World-class Cybersecurity Operations Center*. The MITRE Corporation, McLean, VA.

Received October 2018; revised July 2019; accepted November 2019