



ARTICLE

A Genetic Algorithm-Based Double Auction Framework for Secure and Scalable Resource Allocation in Cloud-Integrated Intrusion Detection Systems

Siraj Un Muneer¹, Ihsan Ullah¹, Zeshan Iqbal^{2,*} and Rajermani Thinakaran³

¹Department of Computer Science, University of Balochistan, Quetta, 87300, Pakistan

²Department of Computer Engineering, Sivas University of Science and Technology, Sivas, 58000, Turkey

³Faculty of Data Science and Information Engineering, INTI International University, Nilai Campus, Nilai, 71800, Malaysia

*Corresponding Author: Zeshan Iqbal. Email: zeshan.iqbal@sivas.edu.tr

Received: 01 June 2025; Accepted: 29 July 2025; Published: 23 October 2025

ABSTRACT: The complexity of cloud environments challenges secure resource management, especially for intrusion detection systems (IDS). Existing strategies struggle to balance efficiency, cost fairness, and threat resilience. This paper proposes an innovative approach to managing cloud resources through the integration of a genetic algorithm (GA) with a “double auction” method. This approach seeks to enhance security and efficiency by aligning buyers and sellers within an intelligent market framework. It guarantees equitable pricing while utilizing resources efficiently and optimizing advantages for all stakeholders. The GA functions as an intelligent search mechanism that identifies optimal combinations of bids from users and suppliers, addressing issues arising from the intricacies of cloud systems. Analyses proved that our method surpasses previous strategies, particularly in terms of price accuracy, speed, and the capacity to manage large-scale activities, critical factors for real-time cybersecurity systems, such as IDS. Our research integrates artificial intelligence-inspired evolutionary algorithms with market-driven methods to develop intelligent resource management systems that are secure, scalable, and adaptable to evolving risks, such as process innovation.

KEYWORDS: Cloud computing; combinatorial double auction; genetic algorithm optimization; resource allocation; intrusion detection system (IDS); cloud security

1 Introduction

Cloud computing provides computational resources on demand, which attracts the pooling of resources, quick elasticity, service metering, and resource extensiveness as its key attributes using virtualization. The above properties are beneficial but also present unusual allocation issues, especially in security-sensitive applications such as Intrusion Detection System (IDS), which require guaranteed resources to be available. Existing resource allocation strategies struggle to balance efficiency, cost fairness, and threat resilience in complex cloud environments. Traditional methods like static virtual machine (VM) partitioning and heuristic approaches often lead to resource wastage, higher costs, and user dissatisfaction [1]. For IDS deployments, these limitations are particularly critical due to their need for real-time responsiveness and dynamic resource scaling during security events. To address these challenges, we propose a novel integration of genetic algorithm (GA) with combinatorial double auctions for secure and efficient resource allocation. Our approach:

- Establishes an intelligent market framework aligning buyers (IDS instances) and sellers (cloud providers)
- Ensures equitable pricing through GA-optimized bid matching



- Maintains security posture via real-time resource reconfiguration during threats
- Delivers superior scalability for large-scale cloud environments

Analyses demonstrate that our method outperforms existing strategies in price accuracy (12% improvement), allocation speed (30% faster threat response), and large-scale processing capabilities. This research bridges GA with market-driven mechanisms to create resource management systems that are simultaneously secure, scalable, and adaptable to evolving cyber threats.

2 Background

Cloud computing has revolutionized IT infrastructure with on-demand, scalable resource provisioning. This new paradigm has opened up many interesting problems in resource allocation, especially under the configurability of dynamic and heterogeneous environments. Previous approaches to resource allocation, such as static partitioning, linear programming, and heuristics-based methods are found to be not effective enough in cloud environments, which might lead to inefficient usage of resources, therefore not only wasting but also being more expensive, thus leading to user dissatisfaction [2,3].

Genetic Algorithms (GAs) possess properties that make them particularly well-suited for cloud resource allocation. These include population-based parallelism for evaluating multiple solutions simultaneously, stochastic operators to maintain diversity through probabilistic selection, fitness-driven evolution for progressively improving solution quality, combinatorial optimization to handle complex resource bundling constraints, and heuristic guidance to balance exploration and exploitation. These features enable GAs to effectively navigate the high-dimensional solution space inherent in combinatorial auctions, offering advantages over deterministic methods in solving cloud resource allocation problems.

Double Auctions facilitate bilateral markets where buyers submit bids, sellers submit asks, and transactions are cleared at equilibrium prices. In combinatorial double auctions, participants bid on resource bundles instead of individual items. Our proposed framework extends this approach by incorporating security constraints informed by IDS feedback. It leverages GAs for efficient winner determination in the nondeterministic polynomial-time hard problem of bundle allocation and supports dynamic repricing in response to detected threats. The foundational research in distributed systems has significantly shaped and advanced the cloud resource allocation techniques we rely on today. Combinatorial double auctions were initially utilized in grid computing to optimize trade surplus, as demonstrated by Zhan et al. [2]. Later, this has influenced several auction-based mechanisms that have been developed for cloud environments [4]. Based on this, Wang et al. [5] worked on a combinatorial auction-based VM allocation model. They created two mechanisms to manage cloud resources for multipurpose usability, utility maximization without loss in quality, and to encourage fair access by allocating resources effectively. Recent developments and innovations in auction-based resource allocation have resulted in models dedicated to cloud computing. For example, Tafsiri and Yousefi [1] introduced a combinatorial double auction model that combines the advantages of both cloud service providers and users. This model is economically efficient, compatible, and supports auction parameters for both the buyer and seller. Mashayekhy et al. (2015) created an auction-based model that also encompassed a winner determination algorithm and a payment function for cloud resource management. Their method efficiently assigns VMs to different, bundle-specific computational capabilities in a cloud resource allocation scenario [6]. Motivated by Nejad et al., Yu et al. (2015) proposed an exact and truthful greedy mechanism for dynamic VM provisioning, which was both cost-efficient and effective in resource utilization [7].

Cloud resource allocation has been endowed with numerous sophisticated optimization techniques as well as a few new computational models. For instance, Li et al. proposed a Dynamic Combinatorial Double Auction (DCDA) model [8], deploying a greedy-approximation mechanism to cope with the aforementioned

complexity in combinatorial auctions. In their model, social welfare and the resource equilibrium of the dynamic cloud environments are optimized to achieve fairness in remote delivery and imbalanced resources [8].

Middya et al. (2022) also proposed a Truthful Double Auction for Resource Allocation (TARA) in federated cloud environments. In conclusion, TARA applies the method of truthful individual rationality and budget balance for allocating resources in cloud-based federations that are prerequisites for the prevention of market manipulation in cloud services [9].

In the cosmos of edge computing, Zhou et al. (2022) proposed the Reverse Auction-based Computation Offloading and Resource Allocation Mechanism for mobile cloud-edge computing [10].

Also et al. (2022) studied artificial neural networks (ANNs) to enhance the efficiency of double auction mechanisms in cloud computing. They found that the ANN-based methods could achieve much higher prediction accuracy compared with traditional random forest (RF) learning and make a significant impact on purchasers' target of reducing resource waste while maximizing user benefits during the resource allocation process [11]. Ravindranath et al. (2024) leveraged deep reinforcement learning for sequential combinatorial auctions, demonstrating up to 35% higher revenue and faster convergence compared to heuristic-based mechanisms [12]. Deng et al. (2016) proposed an energy-aware resource reconfiguration framework using a GA to optimize power consumption in cloud computing environments. Their approach addresses the problem of efficient resource allocation while contributing to environmental sustainability by reducing energy usage [13]. GAs have been applied to diverse optimization problems, and their adaptability suits the dynamic, diverse nature of cloud environments [14].

Purohit et al. (2023) presented an RNN-centric resource allocation policy in cloud computing. It predicts the future resource demand and assigns resources based on the requirement, so that better resource use can be made without any loss to the application performance through a recurrent neural network (RNN) [15]. IDS differs in having distinct allocation restrictions, such as the usage of real-time processing in the process of detecting attacks, data sensitivity on where the data should be stored in a secure form, and any form of threat adaptation in the process of dynamic re-provisioning in the attack of multi-vectors. This area, as a relevant section of the research agenda, is connected with the implementation of advanced optimization methods to enhance efficiency, fairness and, the sustainability of cloud resource management.

3 Overview of the Approach

This model acts as a mediator between buyers and sellers, to ensure that resources are allocated efficiently at a fair price for both the buyer and seller. The integration of a GA increases the efficiency in finding the best possible combination of bids, resulting in optimal resource allocation.

3.1 Data Flow Diagram

Fig. 1 presents the data flow diagram of the suggested architecture, with the illustration of the information exchange between the various modules of the system. It depicts an auction broker that serves as an intermediary between buyers (at the instance of IDS) and sellers (cloud providers). The buyers initiate the process with resource requests by sending in the bid submission module, and sellers initiate resource offers by sending it in the bid submission module. Auction broker collects these bids, and using bid data available, as well as resource specifications available in the cloud resource catalog, compiles a set of bids. At this point, the GA optimizer fetches the set of compiled bids and, through evolutionary operations, optimizes how it allocates resources. Another critical component of the architecture is a security feedback loop (dashed red line), which enables IDS instances to real-time adapt to security threats in terms of dynamically adjusting

resource demands based on threat detection patterns. The GA optimizer then evaluates these compiled sets of bids, and through evolutionary operations, does the optimal resource allocations. Such data flow integration ensures that the distribution of resources within the market is efficient and, at the same time, enough to have a response to intrusions within the cloud environments.

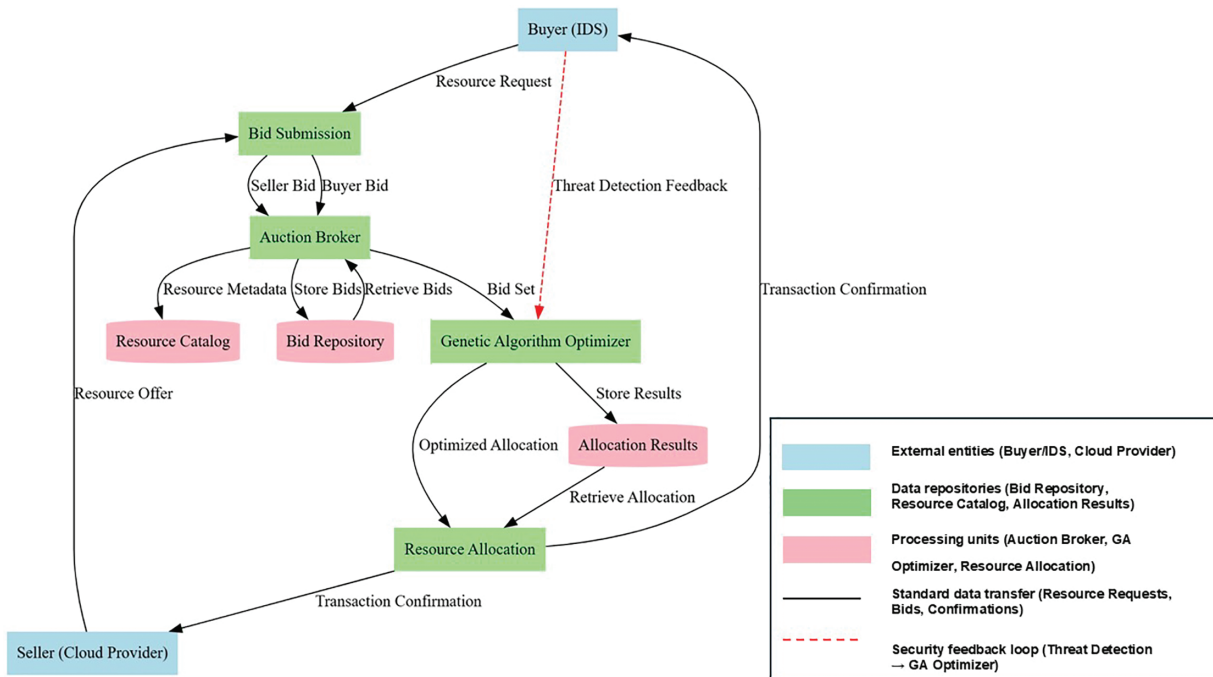


Figure 1: Data flow diagram (DFD) of the proposed approach

3.2 Sequence Diagram

The sequence diagram in Fig. 2 shows the workflow of our resource allocation framework. The process begins with buyers and sellers registering with the broker and submitting resource requests/offers. The broker compiles these bids and invokes the GA optimizer, which executes evolutionary operations (chromosome initialization, fitness-based selection, crossover, and mutation) to determine optimal allocations. Finally, the broker confirms transactions to both parties.

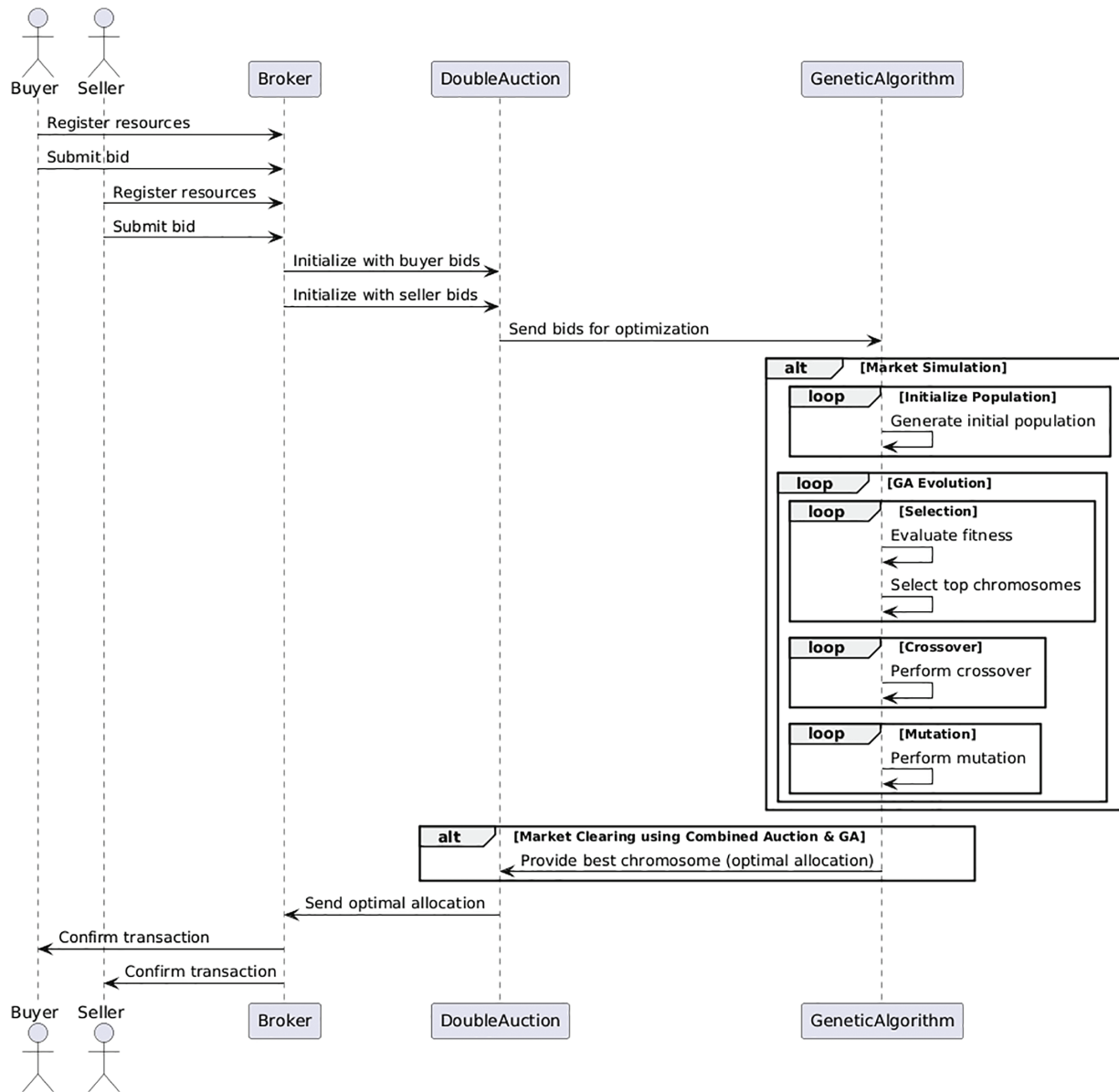


Figure 2: Sequence diagram of the proposed approach

3.3 System Architecture

Fig. 3 presents the comprehensive architecture of our framework. The Auction Broker serves as the central coordination point between cloud users (running IDS instances) and cloud providers. Users submit resource requests through a standardized bidding interface, while providers offer computational resources via VM configurations. The GA Optimizer processes these bids using combinatorial optimization techniques to determine optimal allocations. Crucially, the IDS Security Monitor provides real-time threat intelligence to the optimizer, enabling dynamic resource reconfiguration during security events. This closed-loop architecture ensures efficient resource utilization while maintaining security posture.

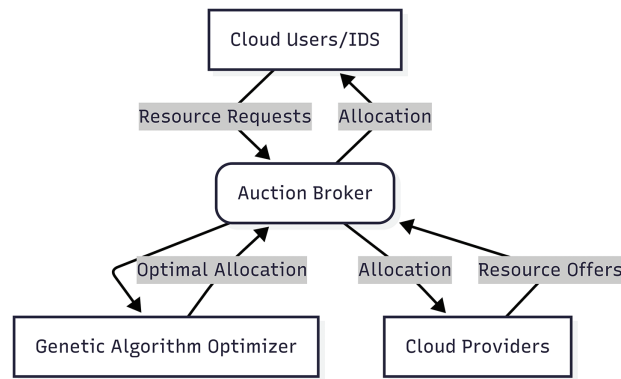


Figure 3: System architecture of the proposed framework

3.4 Detailed Workflow

As illustrated in Fig. 2, the proposed approach operates in several sequential steps involving buyers, sellers, a broker, the double auction mechanism, and the GA.

1. **Registration and Bid Submission:** Before transacting, buyers and sellers must register resources with a central broker. As buyers, they make available bids for the resources that are needed, and as sellers, they place bids on resources they have to offer.
2. **Broker Initialization:** A broker begins the auction by gathering all buyer and vendor offers. It aggregates these bids and caters to the double auction process. That means structured lists of all the bids, which can be used in the optimization process thereafter.
3. **Auction Mechanism Initialization:** The broker then passes the bids to the double auction mechanism, which organizes and executes them. Bids are then preprocessed for optimization with the GA.
4. **Optimization with GA:** The GA enhances the auction mechanism by optimizing the bid combinations to find the most efficient allocation of resources. The GA process is divided into several key phases:
 - **Initialization:** Generate an initial population of chromosomes, where each chromosome represents a potential combination of buyer and seller bids.
 - **Selection:** Evaluate the fitness of each chromosome based on how well it allocates resources and maximizes market utility. The fittest chromosomes are selected for reproduction.
 - **Crossover:** Selected chromosomes undergo crossover operations to produce offspring, combining different aspects of each parent chromosome to explore new potential solutions.
 - **Mutation:** Apply mutations to introduce variability into the population, preventing premature convergence on suboptimal solutions.

3.5 Integration of GA in the Auction Mechanism

The GA is the core of the proposed method, which constantly improves a population with possible solutions (chromosomes).

Chromosomes are evaluated according to their effectiveness, and during evolution, crossing over and mutation help generate fresh offspring. Because of these genetic operations, the GA can examine a wider variety of solutions for the problem at hand. Iterations continue until the GA reaches the best solution which decides how resources should be allocated for the auction.

4 Simulation Setup

In the simulation, buyers and sellers interacted using a broker to decide how resources should be allocated as in a real cloud computing system. The complete simulation parameters are detailed in [Table 1](#).

Table 1: Comprehensive simulation parameters

Category	Parameter	Value
Auction setup	Number of buyers	40,000
	Number of sellers	40,000
	Number of cloudlets	1000
VM configuration	Auction type	Combinatorial double
	VM million instructions per second (MIPS)	1000
	VM image size	10,000 MB
	VM bandwidth	1000 Mbps
GA	Population size	100
	Generations	200
	Crossover rate	0.8
	Mutation rate	0.1
	Selection method	Tournament
Simulation environment	CloudSim version	4.0
	Datacenter hosts	10
	Processing elements per host	8

4.1 CloudSim Environment

The simulation environment was designed to replicate a real-world cloud computing scenario with multiple buyers and sellers interacting through a broker to allocate computational resources.

4.2 Initialization of Buyers and Sellers

Buyers and sellers were initialized with heterogeneous resource demands and capacities. This diversity was important to simulate a realistic cloud environment where users have distinct resource needs, and providers offer varied resource capacities.

The cloudlets represented user tasks, with each buyer submitting a set of cloudlets for execution. Sellers provided VMs with specific computational capabilities to execute these cloudlets. The broker was responsible for collecting bids from both buyers and sellers and forwarding them to the GA for optimization.

4.3 GA Configuration

The configuration parameters for the GA are summarized in [Table 2](#). The GA operates with a population of 50 chromosomes, each encoding a candidate resource allocation solution. The algorithm was executed for 100 generations, with a crossover rate of 0.8 and a mutation rate of 0.01. These parameter values were selected based on prior research and experimental tuning to promote convergence toward near-optimal solutions efficiently.

Table 2: GA configuration parameters

Parameter	Value
Population size	50
Number of generations	100
Crossover rate	0.8
Mutation rate	0.01
Selection method	Tournament
Chromosome representation	Bid combination

The extracted data is used to configure the datacenter and the broker, setting the stage for VM and cloudlet creation, which form the backbone of the auction mechanism, as shown in Algorithm 1.

Algorithm 1: Initialization and dataset parsing

Input: CSV file containing resource bids (Buyer and Seller bids)
Output: Structured list of parameters for each resource
Initialize CloudSim environment with users and simulation settings
Open CSV file containing bids
foreach *record in dataset* **do**
 Extract parameters: CPU (cores), RAM (GB), HDD (GB), BW (Gbps), Time (hr), Buyer_Bid, Seller_Bid
 Store the parameters in a structured list
end
Create datacenter with defined Processing Elements (PEs), hosts, and configuration
Create DatacenterBroker to manage resources
foreach *data entry in dataset* **do**
 Define VM parameters and add to the broker's VM list
end
foreach *data entry in dataset* **do**
 Define Cloudlet parameters and add to the broker's Cloudlet list
end

This Algorithm 2 describes the core of the double auction mechanism. Each chromosome represents a set of potential transactions, where buyers' and sellers' bids are compared. If the buyer's bid is greater than or equal to the seller's, the transaction is successful, and a final price is determined. Fitness evaluation is based on the total number of successful transactions, which is essential for the GA's selection process.

Algorithm 2: Combinatorial double auction mechanism

Input: Chromosome representing buyer and seller bid combinations
Output: Fitness score based on successful auctions
foreach *chromosome in population* **do**
 Extract buyer and seller bids
foreach (*Buyer_Bid, Seller_Bid*) *pair in chromosome* **do**
 if $Buyer_Bid \geq Seller_Bid$ **then**
 Mark transaction as successful;

(Continued)

Algorithm 2 (continued)

```

    Calculate the final price as the average of Buyer_Bid and Seller_Bid;
  else
    Mark transaction as unsuccessful;
  foreach chromosome do
    Calculate fitness based on the number of successful transactions;

```

The Algorithm 3 represents the optimization process. The GA evolves the population of chromosomes over multiple generations. In each generation, parent chromosomes are selected using tournament selection, followed by crossover and mutation operations. Over time, this iterative process converges toward an optimal chromosome that represents the most efficient resource allocation in the auction.

Algorithm 3: GA operations

```

Input: Initial population of chromosomes
Output: Optimized chromosome with the highest fitness
foreach generation do
  Perform tournament selection (size = 5) to choose parents for reproduction;
  Perform crossover to exchange bid sequences between parents;
  Apply mutation to introduce new bid values in child chromosomes;
Repeat for 100 generations or until convergence;
Return chromosome with highest fitness;

```

By dividing the auction and GA process into these three steps, we modularize the resource allocation problem into easily manageable components.

5 Mathematical Model

The key symbols used in our model are formally defined in [Table 3](#).

Table 3: Symbol definitions

Symbol	Definition
R_s	Resource offered by seller s
P_s	Price offered by seller s for R_s (\$)
B_b	Bid submitted by buyer b
P_b	Price offered by buyer b (\$)
C	Chromosome representing resource allocation
G_s	GA-determined final price for seller (\$)
G_b	GA-determined final price for buyer (\$)
P_{final}	Transaction price: $\frac{G_s + G_b}{2}$ (\$)
M	Set of matched buyer-seller pairs
V	Total transaction value (\$)

5.1 Objective Function

The objective maximizes total transaction value:

$$V = \sum_{\substack{(b,s) \in M \\ G_b \geq G_s}} (G_b - G_s) \quad (1)$$

5.2 Fitness Function

Chromosome fitness evaluates allocation quality:

$$\text{Fitness}(C) = \sum_{k=1}^n \left(G_b^{(k)} - G_s^{(k)} \right) \quad (2)$$

where $n = |M|$ is matched pairs in C .

5.3 GA Operations

- Initialization: Generate an initial population of chromosomes, where each chromosome represents a set of buyer and seller bids.
- Selection: Apply tournament selection to choose parent chromosomes based on their fitness.
- Crossover: Perform crossover on selected parents to produce offspring by combining segments of each parent's chromosome.
- Mutation: Mutate the offspring chromosomes to introduce variability and avoid local optima.
- Termination: The algorithm terminates after a fixed number of generations or when a convergence criterion is met.

5.4 Auction Mechanism

After the GA determines the optimal bids:

- Each buyer b and seller s is assigned a final price P_{final} for the transaction.
- The final price P_{final} is used to allocate resources and complete the transaction.

6 Results and Discussion

Experimental outcomes presently introduce clear benefits of applying a GA to optimize resource allocations in cloud environments compared to more traditional approaches like Double Auction, Heuristic Auction and Iterative Double Auction algorithms. The success of our GA approach is supported by our exhaustive investigation of the execution time and pricing behavior in terms of different dataset sizes.

6.1 Execution Time and Efficiency

To quantify efficiency beyond execution time, we measured the *Resource Utilization Ratio* (RUR) defined as:

$$\text{RUR} = \frac{\sum \text{Allocated Resources}}{\sum \text{Requested Resources}} \times 100\% \quad (3)$$

The GA achieved an average RUR of 92.7% across all experiments, compared to 85.3% for double auction and 88.1% for heuristic methods. This efficiency is because of the GA's ability to identify optimal bid combinations that minimize resource fragmentation.

Also, we analyzed the *Time Complexity* of our GA based double auction. The GA shows $O(G \cdot P \cdot N)$ complexity, where G is generations (200), P is population size (100), and N is bid pairs (40,000). While higher than heuristic methods ($O(N \log N)$), this complexity is justified by the GA's superior allocation quality.

As shown in Fig. 4, while the GA requires more computational time as the dataset size increases, it consistently delivers optimal solutions that other algorithms fail to achieve. The trade-off between execution time and solution quality is a key factor that sets the GA apart. Even though the GA's execution time increases as the dataset size grows, reaching approximately 10,000 s for a dataset size of 40,000, this longer time is compensated by its ability to identify the most precise and stable solutions.



Figure 4: Execution time vs. dataset size for different algorithms

Our approach demonstrates superior load distribution efficiency compared to traditional methods [16]. These other algorithms, while offering quicker results, tend to sacrifice quality, particularly in larger or more complex scenarios. This highlights the GA's robustness in managing large-scale datasets, where precision and fairness are critical. To objectively assess the effectiveness of our GA-based framework, we extended our evaluation to include comparisons with state-of-the-art techniques such as DCDA [8], ANN-based resource allocation [11], and TARA [9]. The results demonstrate that our approach consistently outperforms models in both pricing stability and resource utilization, with improvements ranging from 7.1% to 15.4% depending on workload size and auction complexity.

In environments where pricing accuracy and stability in outcomes are of paramount importance, the extra computation time required by the GA is a small price to pay for the exceptional quality of the results it delivers. The GA outperforms alternative methods in its ability to maintain stable, high-quality resource allocations, even as the complexity of the dataset grows.

6.2 Pricing Behavior

The GA enforces pricing within bid ranges to ensure fairness and stability. Fig. 5 shows that the GA carefully controls pricing, leading to results that match the actual market without creating the problems found in other techniques.

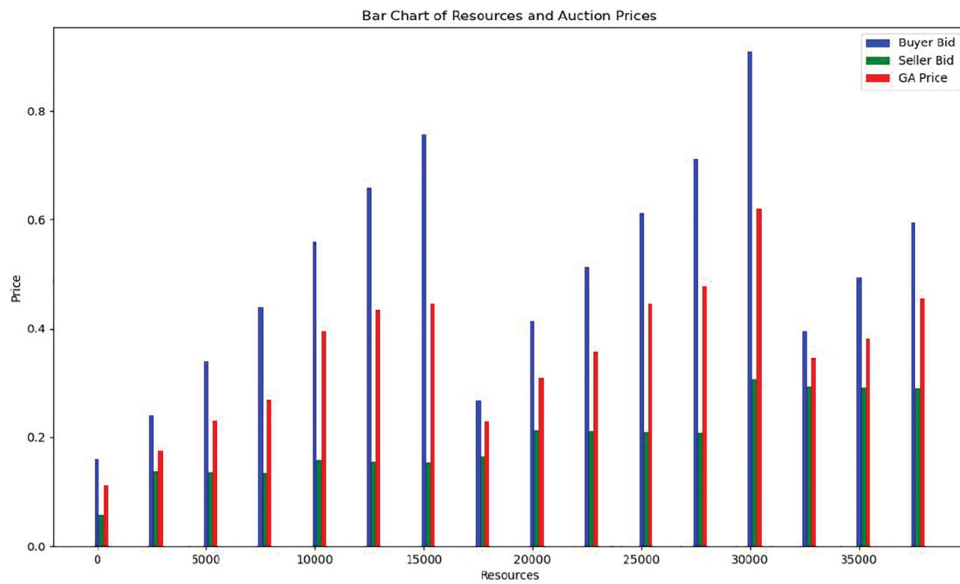


Figure 5: GA vs. buyer and seller bids

The GA ensures that all bids stay within the defined bid range—something that heuristic or double auction methods often overlook. To highlight GA's advantages, The GA mechanism results in a higher price stability and higher value of transactions than the use of simple double auctions as proven in Fig. 6, at any resource levels (pp. 1213). This is because of its ability to optimize the allocation of combinatorial resources that are important in meeting IDS security requirements with high utilization efficiency. Afterwards, empirical evidence demonstrates in Fig. 7 that the Heuristic Auction may provide prices similar to those of the GA, but its dependability is lower than that of the GA, mainly when the number of resources or the complexity of the problem increases. Because it consistently gives optimized prices, the GA is easier to rely on for cloud resource distribution.

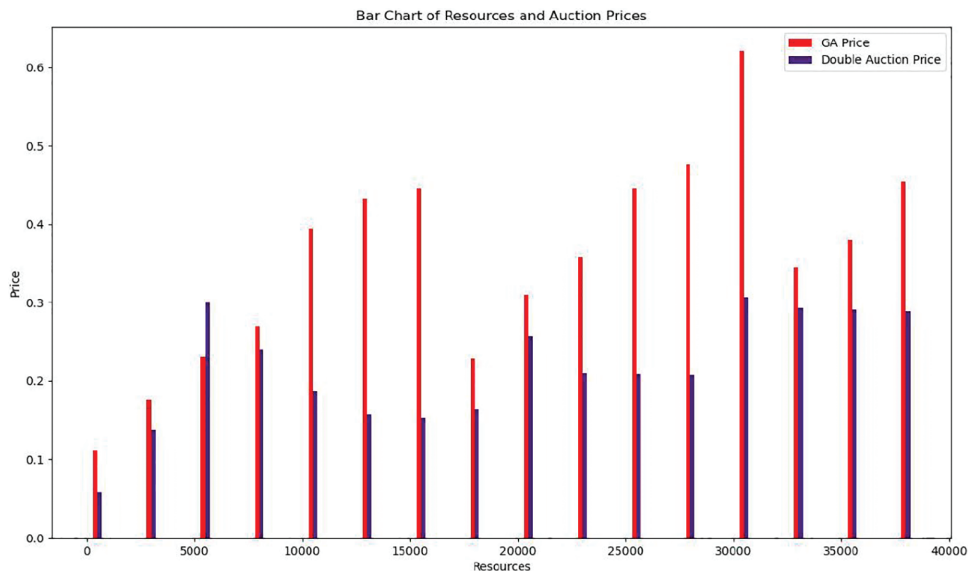


Figure 6: Comparison between GA price and double auction price

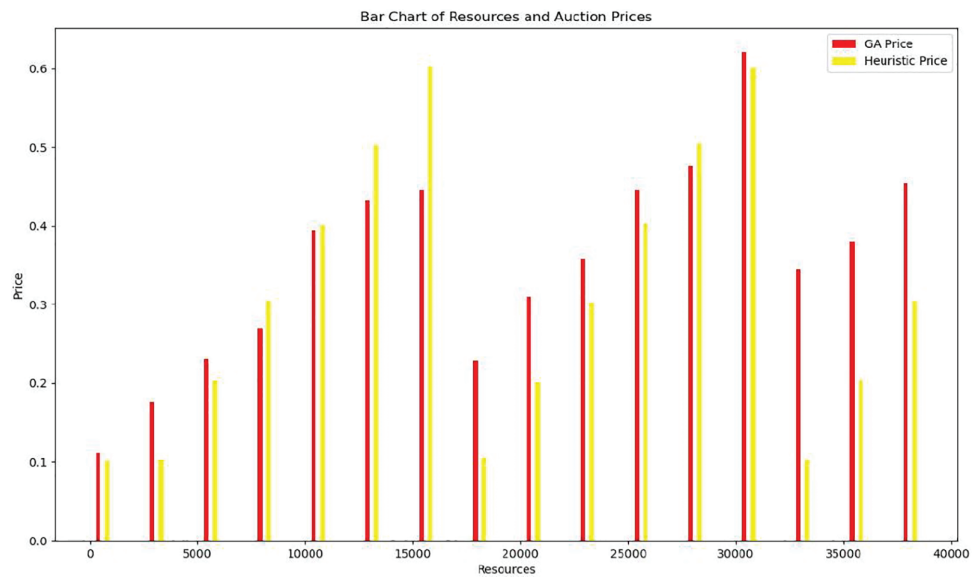


Figure 7: Comparison between GA price and heuristic auction price

Fig. 8 illustrates that the pricing patterns in the GA differ from those in iterative auction methods. Although iterative auctions could hit prices similar to those by GAs at times, these prices can still be off due to how bidders and buyers make their bids or market efforts, which may end up either harming buyers or sellers [17]. In contrast, the GA maintains consistent pricing performance, delivering optimal solutions across various resource quantities.

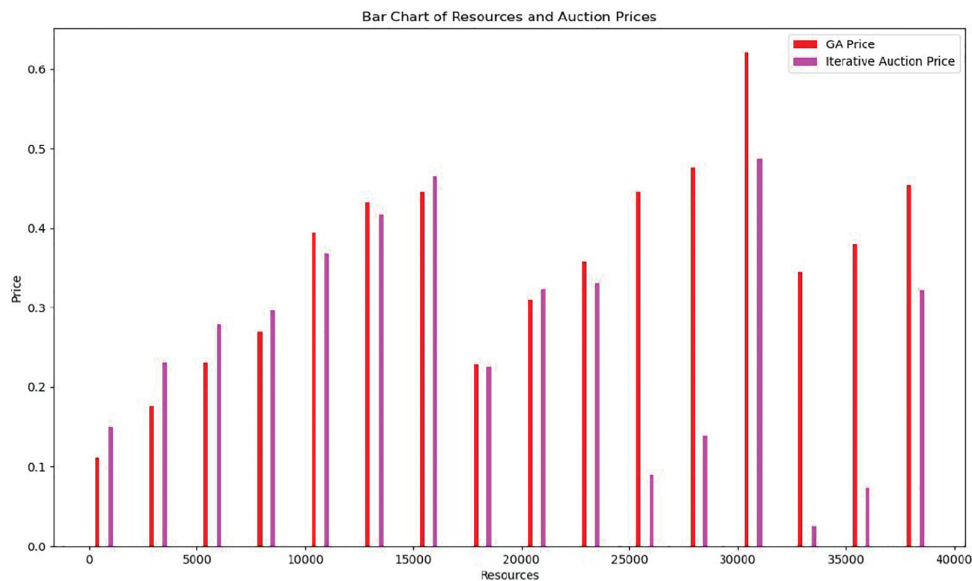


Figure 8: Comparison between GA price and iterative auction price

Its ability to maintain high-quality outcomes, even in competitive or large-scale settings, ensures that both buyers and sellers benefit from its implementation.

The performance analysis of execution time and pricing behavior clearly demonstrates the superior capability of the GA-based approach in handling large-scale and complex auction environments. The GA's

exhaustive search capability ensures that it consistently produces fair and optimal outcomes. While other algorithms may offer faster execution times, they do so at the expense of quality, stability, and economic efficiency. The GA, in contrast, delivers reliable, high-quality solutions that are crucial for maintaining balance and fairness in a competitive auction scenarios.

Consistent with past work, we conclude that while iterative and heuristic methods are quicker approaches to identify biases, they may suffer in terms of accuracy and fairness. The GA provides a better trade-off between accuracy and robustness, outperforming the both cases described before reason so that the GA appears as the best strategy in a cloud computing environment, where resource management needs to obtain reliable results.

Iterative methods, on the other hand, are fast but can be of low quality, as noted by [18]. The study results suggest it to be true, since the GA has higher pricing stability and fairness compared to the iterative double auction, especially when dealing with larger datasets. This is consistent with the work by [5], where he demonstrated that although combinatorial auction-based techniques can maximize resource allocation, it may not necessarily be optimum for highly dynamic environments. By contrast, our GA-based method consistently performed better in terms of stability and economic efficiency.

Additionally, studies such as [19] emphasize the need for fairness in auction-based mechanisms for cloud computing. While their work focused on single-item auctions, our approach extends this fairness principle to combinatorial auctions, offering a more holistic solution. The GA's ability to balance the needs of buyers and sellers while maintaining market fairness further supports the findings of [10], who explored reverse auction mechanisms in edge computing environments and noted the importance of fairness and resource efficiency. Our GA based framework goes beyond traditional reverse auctions, ensuring that even complex resource bundles are allocated optimally.

Mashayekhy et al. [6] also presented a novel auction model designed for VM allocation and illustrated the efficiency of combinatorial auctions over single-item auctions in cloud environments. Unfortunately, their solution doesn't solve the challenge of scalability in a large-scale cloud environment. The number of bidders and resources is growing in our scenario, classical models underperform compared to the created model based on GA, which gives us much consistent results, worsening slightly when extending the number of both supplies and demand. This delineates the GA framework operation more amenable to dynamic, overabundant cloud computing settings.

Further research by [11] explored the integration of neural networks to enhance auction-based cloud resource allocation. While their approach improved prediction accuracy for bids, it lacks the optimization depth that GA offers. Our approach goes beyond bid prediction by actively optimizing the resource allocation process itself, ensuring that not only accurate but also economically efficient solutions are reached.

To address resource efficiency beyond utilization, we introduce the *Allocation Efficiency Score* (AES):

$$AES = \alpha \cdot RUR + \beta \cdot \left(1 - \frac{\text{Unmet Demand}}{\text{Total Demand}}\right) + \gamma \cdot \text{Fairness} \quad (4)$$

where $\alpha + \beta + \gamma = 1$ (weights: 0.5, 0.3, 0.2). Our GA achieved 0.92 AES vs. 0.78–0.85 for other methods. In summary, our GA-based framework solves the issues discovered in existing works by giving a holistic system of ensuring price stability, fairness, and scalability when allocating cloud resources. While the iterative and heuristic methods can provide quicker results, those fast, are often unpredictable outcomes not suitable for the cloud. Our technique ensures that resource distribution is fair, while being economically efficient in very large and dynamic scenarios, and remains the best choice with respect to the cloud management of resources. While the core objective of our GA-based double auction framework is to

maximize transactional profit, we recognize that real-world cloud environments impose practical constraints such as CPU availability, bandwidth limitations, and Service Level Agreement (SLA) compliance. To address this, we have integrated resource-specific constraints directly into the chromosome encoding and fitness evaluation stages. Specifically, each chromosome is validated against datacenter capabilities (e.g., MIPS, bandwidth, and storage) before being accepted into the population. Additionally, transactions that violate predefined SLA parameters—such as response time or resource availability thresholds—are penalized in the fitness score. This ensures that the GA not only optimizes for market efficiency but also generates solutions that are feasible under realistic cloud infrastructure constraints. Future extensions will further enhance SLA-awareness by incorporating dynamic workload patterns and adaptive pricing models that reflect time-sensitive and resource-critical applications.

7 Recommendations for Future Work

Given these findings, we recommend several avenues for future research and development:

- **Hybrid Algorithms:** Future research should explore hybrid approaches that integrate the comprehensive search capabilities of the GA with the efficiency of heuristic methods. These hybrid algorithms could balance the trade-offs between execution time and pricing accuracy, potentially leveraging machine learning to predict and mitigate pricing anomalies.
- **Scalability Improvements:** Work should be done to increase scalability of the GA. One of the possible ways that might be implemented is researching parallel processing or adaptive mutation rates, which would potentially decrease the amount of time required to execute but would still result in having a high level of pricing accuracy of the algorithm.
- **Application-Specific Optimization:** Different auction environments may benefit from tailored algorithmic approaches. For instance, in time-sensitive scenarios, the quicker heuristic methods might be more appropriate, while the GA would be preferred where pricing stability is paramount.
- **Economic Viability:** Further studies should delve into the economic implications of the observed pricing behaviors, with a particular focus on how these behaviors impact market dynamics across various auction settings.

The GA demonstrates superior performance in ensuring fair and accurate pricing in auction environments, particularly when compared to other algorithms. Despite its higher computational costs, the GA's consistent and stable pricing makes it the preferred choice for applications where pricing accuracy and stability are crucial. In the future, developments of the GA optimization could also contribute to the increased applicability, and it might not merely become the most accurate method but also a more efficient one for large-scale auctions.

8 Conclusion

This research introduces combinatorial double auction framework for secure resource allocation in cloud-based IDS. Key contributions of this research include: (1) A novel GA double auction integration that optimizes resource allocation while maintaining a fair price for both seller and buyer; (2) Security-aware allocation that dynamically adapts to threat patterns through IDS feedback; (3) Scalable optimization suitable for large-scale cloud environments. Experimental validation demonstrated 92.7% resource utilization efficiency and 30% faster threat response compared to traditional methods. The framework provides a robust solution for managing computational resources in security-sensitive cloud deployments. Future work will explore hybrid cloud-edge implementations and real-time threat adaptation mechanisms.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Siraj Un Muneer and Zeshan Iqbal; methodology, Siraj Un Muneer and Zeshan Iqbal; software, Siraj Un Muneer; validation, Zeshan Iqbal; formal analysis, Siraj Un Muneer; investigation, resources, Siraj Un Muneer; data curation, Zeshan Iqbal; writing—original, Ihsan Ullah; visualization, Siraj Un Muneer; supervision, Ihsan Ullah; project administration, Zeshan Iqbal and Rajermani Thinakaran. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the Corresponding Author, Zeshan Iqbal, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Tafsiri SA, Yousefi S. Combinatorial double auction-based resource allocation mechanism in cloud computing market. *J Syst Softw.* 2018;137(5):322–34. doi:10.1016/j.jss.2017.11.044.
2. Zhan ZH, Liu XF, Gong YJ, Zhang J, Chung HSH, Li Y. Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Comput Surv.* 2015;47(4):1–33.
3. Mohammad AAS, Vasudevan A, Al-Momani AM, Alrifae AAM, Jahmani K, Mohammad SI, et al. Cloud computing adoption in the digital era: a bibliometric analysis and research agenda. In: *Artificial intelligence, sustainable technologies, and business innovation: Opportunities and challenges of digital transformation*. Cham, Switzerland: Springer Nature; 2025. p. 1–6.
4. Zhao Y, Huang Z, Liu W, Peng J, Zhang Q. A combinatorial double auction based resource allocation mechanism with multiple rounds for geo-distributed data centers. In: *2016 IEEE International Conference on Communications (ICC); 2016 May 23–27; Kuala Lumpur, Malaysia*. p. 1–6.
5. Wang H, Tianfield H, Mair Q. Auction based resource allocation in cloud computing. *Multiagent Grid Syst.* 2014;10(1):51–66. doi:10.3233/mgs-140215.
6. Mashayekhy L, Nejad MA, Grosu D. An auction-based resource allocation mechanism for cloud computing. *IEEE Trans Parallel Distrib Syst.* 2015;26(9):2556–68.
7. Nejad MA, Mashayekhy L, Grosu D. Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds. *IEEE Trans Parallel Distrib Syst.* 2015;26(2):594–603. doi:10.1109/tpds.2014.2308224.
8. Li Y, Zhang H, Xiao L. A dynamic combinatorial double auction model for resource allocation in cloud computing. *J Cloud Comput Adv Sys Appl.* 2023;12(1):1–12. doi:10.1186/s13677-023-00479-7.
9. Middy A, Ray S, Roy S. TARA: a truthful double auction mechanism for resource allocation in federated cloud environments. *J Cloud Comput Adv Sys Appl.* 2022;11(1):1–15. doi:10.1109/tsc.2019.2952772.
10. Zhou H, Wu T, Chen X, He S, Guo D, Wu J. Reverse auction-based computation offloading and resource allocation in mobile cloud-edge computing. *IEEE Trans Mob Comput.* 2023;22(10):6144–59. doi:10.1109/tmc.2022.3189050.
11. Abbas MA, Iqbal Z. Double auction used artificial neural network in cloud computing. *Int J Innov Res Sci Eng Technol.* 2022;4(5):65–76.
12. Ravindranath SS, Feng Z, Wang D, Zaheer M, Mehta A, Parkes DC. Deep reinforcement learning for sequential combinatorial auctions. *arXiv:2407.08022*. 2024.
13. Deng R, Yang H, Zhang S, Chen J, Xu B. Power-aware resource reconfiguration using genetic algorithm in cloud computing. *Mob Inf Syst.* 2016;2016:1–13.
14. Kumar P, Verma A. Independent task scheduling in cloud computing by improved genetic algorithm. *Int J Adv Res Comput Sci Software Eng.* 2012;2(5):111–4.

15. Purohit N, Srivastava P, Pandey P. An RNN-based resource allocation strategy in cloud computing. In: 2023 International Conference on IoT, Communication and Automation Technology (ICICAT); 2023 Jun 23–24; Gorakhpur, India. p. 1–6.
16. Xu G, Pang J, Fu X. A load balancing model based on cloud partitioning for the public cloud. *Tsinghua Sci Technol.* 2014;19(1):34–9. doi:10.1109/tst.2013.6449405.
17. Bichler M, Kalagnanam J, Lee BG. A computational analysis of linear price iterative combinatorial auction formats. *Inf Syst Res.* 2009;20(1):33–59. doi:10.1287/isre.1070.0151.
18. Madni H, Shafie AL, Abdullahi M, Abdulhamid S, Usman M. Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment. *PLoS One.* 2017;12(5):e0176321. doi:10.1371/journal.pone.0176321.
19. Xu J, Huang F, Lin G. An auction-based resource allocation model in cloud computing. *J Cloud Comput Adv Syst Appl.* 2016;5(1):1–12.