Journal of Infrastructure, Policy and Development 2024, 8(16), 9482.
https://doi.org/10.24294/jipd9482

Article

Knowledge refinement mechanism in agency using adaptive automata and genetic algorithms

Wasim Ahmad Khan[1], Muhammad Ayub[2], Muhammad Umer Quddoos[3], Liaqat Ali Waseem[4,*], Muhammad Abdul Basit Ur Rahim[5,*], Muzaffar Hameed[6], Muhammad Adeel[7], Arslan Ahmad Siddiqi[8], Muhammad Furqan Khan[9]

[1] Department of CS & IT, Baba Guru Nanak University, Nankana Sahib 39100, Pakistan
[2] Department of Management Sciences, COMSATS University Islamabad, Attock Campus 43600, Pakistan
[3] Department of Commerce, Bahauddin Zakariya University, Multan 60800, Pakistan
[4] Department of Geography, Government College University Faisalabad, Faisalabad 38000, Pakistan
[5] Department of Computer Engineering and Computer Science, California State University, Long Beach, CA 90840, USA
[6] Department of CS & IT, Bahauddin Zakariya University, Multan 60800, Pakistan
[7] Department of Commerce and Finance, Government College University, Lahore 54000, Pakistan
[8] Institute of Industrial Control Systems, Karachi 24700, Pakistan
[9] Washington University of Science and Technology, VA 22314, USA
*** Corresponding authors:** Muhammad Abdul Basit Ur Rahim, m.basit@csulb.edu; Liaqat Ali Waseem, drliaqataliwaseem@gcuf.edu.pk

CITATION

Khan WA, Ayub M, Quddoos MU, et al. (2024). Knowledge refinement mechanism in agency using adaptive automata and genetic algorithms. Journal of Infrastructure, Policy and Development. 8(16): 9482. https://doi.org/10.24294/jipd9482

ARTICLE INFO

Received: 8 October 2024
Accepted: 20 December 2024
Available online: 28 December 2024

Abstract: Recent advancements in data mining techniques showcase expansive and effective insights. Yet, these insights often remain incomplete. In situations demanding optimal results, such incomplete knowledge from various agents falls short. This paper delves into the role of mobile agents engaging in data mining within a multi-agent environment. Each agent is tailored with distinct goals, mining data in line with its identified problem set. Mobile agents understand insights from relevant agents equipped with specialized data knowledge. Outcomes from diverse agents converge, serving as foundational data for mobile agents. This integration is facilitated by integrating adaptive automata and genetic algorithms, enhancing the mobile agent's expertise on a particular task.

Keywords: adaptive automata; genetic algorithm; mobile agent
JEL Classification: I15; I18; I31; O15

1. Introduction

In the realm of information technology, the endeavour to unearth knowledge from expansive data sets using data mining techniques has garnered significant attention. Each data mining method caters to a unique knowledge requirement. A noteworthy strand of contemporary research delves into utilizing multi-agent-based data mining to address intricate challenges. In this modality, an array of agents deploys diverse data mining methods to cull valuable, albeit partial, insights from data. An essential limitation of this approach emerges when agents combine their gleaned knowledge, potentially introducing inconsistencies or unnecessary data.

Presented herein is an innovative framework centered on mobile agent-based learning. This model synergizes multi-agent-based data mining, adaptive automata, and genetic algorithms. In contrast to conventional approaches that aggregate knowledge from all agents, mobile agents engage in insightful adoption of insights from static agents.

Here, static agents represent entities pre-armed with segmented knowledge derived from data. The core architecture of this mechanism encompasses multiple static agents complemented by one or more mobile agents. These mobile agents undergo an intelligent learning trajectory, clustering static agents through adaptive automata (Abu-Qadri et al., 2018). Classifying those endowed with significant knowledge using genetic algorithm strategies. Through this meticulous process, mobile agents acquire enhanced and streamlined knowledge. As they tackle specific challenges, these enriched mobile agents emerge as the torchbearers of solutions. The intricate workings of this proposed model are elaborated upon in the ensuing sections.

Conventional knowledge refinement techniques are neither flexible, scalable, nor resilient in dynamic settings because they are usually built on static, rule-based systems. To get over these restrictions, we suggest a Knowledge Refinement Mechanism that works in conjunction with Genetic Algorithms and Adaptive Automata. While genetic algorithms maximize the refinement process by evolving solutions over iterations, adaptive automata allow for dynamic modeling and real-time learning by developing state transitions based on environmental changes. Together, they produce a feedback loop in which genetic algorithms guarantee effective and reliable solution space search while adaptive automata give a learning structure. By addressing the drawbacks of conventional methods, this innovative framework provides a scalable and adaptable solution for intelligent systems, dynamic decision-making, and autonomous robots.

The constraints of conventional knowledge refinement techniques are addressed by the suggested framework, which makes use of adaptive automata and genetic algorithms. The foundation is made up of adaptive automata, which dynamically alter their state transitions in response to input from the environment, allowing for real-time learning and decision-making. For instance, autonomous vehicles can change routing methods to reduce delays. To provide robust and scalable adaptation, genetic algorithms optimize the automaton’s rule set through evolutionary processes like as crossover, mutation, and selection. In logistics, for instance, GAs adjust routing choices to reduce delivery times in a variety of scenarios. Adaptive automata and GAs work together to establish a feedback loop in which GAs guarantee long-term optimization and automata give structure to dynamic responses. This makes the framework flexible and useful for applications such as game AI, dynamic decision-making, and disaster response.

Data mining serves as a conduit to unearth hidden correlations within vast data sets, offering pivotal insights instrumental for informed decision-making. Such techniques, as illustrated in **Figure 1**, necessitate categorization based on both their objectives and the nature of their outcomes. At the crux of this categorization lies the distinction between methodologies yielding overarching statements about the data - akin to summaries that specify global patterns. Producing localized perspectives that capture nuances and anomalies in data relationships. **Figure 1** depicts the hierarchical phases of data mining techniques, from raw data to generalization. These challenges focus on covering the gap between preprocessed data to thought-provoking conclusions, specifically when handling large-scale databases. This study resolves the challenge by providing techniques that not just cover hidden patterns but also data-driven transformations using machine learning and genetic algorithms.

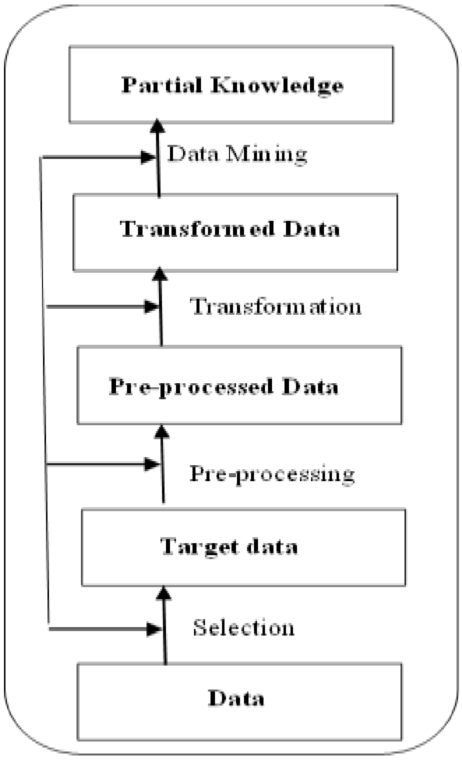

Figure 1. Data mining techniques (key stages of refining raw data into actionable knowledge).

The crux of data mining recommendations hinges on the examination of prevailing databases. This involves cataloging pertinent resources and suggesting resources resembling those previously analyzed or rated. Collaborative and content-based filtering, integral to data mining, exhibit inherent limitations. Collaborative filtering, while powerful, grapples with efficiently integrating novel data or features, with its efficacy being tethered to a rich historical data corpus. On the other hand, content-based filtering faces challenges when venturing into uncharted resource categories, distinct from past evaluations. Moreover, this method mandates the expertise of a domain specialist for effective knowledge integration. Recognizing these inherent challenges, hybrid systems have emerged as a beacon of innovation. By amalgamating the virtues of collaborative filtering with content-based approaches, these systems optimize the recommendation process, catering seamlessly to intrinsic information requisites (Carlo and Andrea, 2023). Data mining has been widely used as a fundamental tool for knowledge discovery from manufacturing databases. The necessary data to be analyzed can be gathered throughout ordinary manufacturing operations (Chehri et al., 2024). To address this problem, a great deal of research effort is being put into creating algorithms that can autonomously change the algorithm parameters without the need for an external agent. Numerous adaptation strategies have already been documented in the literature; nonetheless, they can be divided into three groups: self-adaptive, adaptive, and deterministic (Dogan and Briant, 2021).

An agent can be intelligent but it does not have to be. An agent may be an individual, a household, or even a system that makes decisions. The broad applicability of ABM does not do them any favours either as the concept of an "Agent" may differ depending on the field of research. Fortunately, several features are prevalent in all agent definitions (Gao and Wang, 2024). According to Carlo and Andrea (2023), To solve search and optimization problems, genetic algorithms (GAs) are a bio-inspired approach based on Darwin's evolutionary theory. The search process in GAs is guided by genetic operators, which eliminates the need to specify the steps that lead to a result and gives the algorithm robustness and flexibility. The search process itself is based on the definition of the representation and evaluation of a potential solution to the problem (Dogan and Briant, 2021; Goldberg and David, 1989). Data-driven systems, specifically in manufacturing and optimization, face different challenges in getting thoughtful insights from large-scale datasets. Previous recommendation models demonstrate constraints in scalability and contextual adaptability to handle data. These limitations become more known in industrial environment systems. The key problem discussed in this study is the inability of data mining techniques and recommendation systems to adaptively react to dynamic data and optimize the discovery of knowledge in large-scale databases of manufacturing. These systems generally lack mechanisms for autonomous refinements.

Additionally, in previous studies, the applications of Automata and genetic algorithms have left a gap in their integrations. To bridge this gap. Introduce a framework that integrates adaptive automata and genetic algorithms. This framework incorporates adaptive automata to continuously evaluate, ensuring the performance even in scenarios with limited data inputs while genetic algorithms optimize the feature selection and decision-making processes to uncover patterns in large-scale datasets.

Multi-agent systems now becoming the increasing framework for handling complex data mining issues, specifically, in distributed environments. The work of Hossain et al. (2009) explored the multiagent systems in manufacturing data mining for fault detection. These systems optimize agents to analyze sensor data, improving fault prediction accuracy and precision. This approach also had some limitations in handling datasets, the issue is also pointed out by Bensalem et al. (2015).

The proposed study significantly fills the gaps left by previous studies by integrating genetic algorithms with adaptive automata in multi-agent systems. This approach covers many critical challenges in the field, unlike existing multiagent systems that use predefined models, this framework integrates adaptive automata that self-regulate to real-time data transformation, enhancing scalability and performance. By incorporating Genetic algorithms, the model enhances the performance of data mining and allows for efficient search of solutions, improving insights and computational flexibility.

This paper introduces the model that connects the Genetics algorithm and adaptive automata to manufacturing DB (database). The proposed study combines mining with innovative computations to discuss the specific constraints in collaborative and content filtering. By utilizing the adaptive automata for dynamic system refinements and Genetic algorithms for comprehensive search optimization, the model develops cutting-edge methodologies in adaptive recommendation systems. The collaborative integration provides enhancements in scalability and adaptive contextualization, providing significant contributions in the domain of knowledge and decision-making.

2. Proposed work

The intricate journey of knowledge discovery and data mining encompasses multifaceted phases. This trajectory initiates with Data Collection, meanders through Integration and Cleaning, delves into Data Exploration and Preparation, and then transitions to the Selection and Application of specific data mining tasks and algorithms (AMERSHI and CONATI, 2009). Due to its explorative essence, the Data Mining process often exhibits a cyclical pattern. At junctures where the analytical results appear less than satisfactory, revisiting previous steps becomes a necessity. Within this framework, multi-agents emerge as potent entities, imbibing knowledge derived from the knowledge discovery and data mining spectrum. Subsequently, mobile agents undertake a learning endeavour (Russell and Norving, 2003), assimilating insights from the collective intelligence of multi-agents. This orchestrated convergence culminates in the formulation of the Refined Knowledge Base (RKB), as shown in **Figure 2**, a repository brimming with polished and honed knowledge segments.

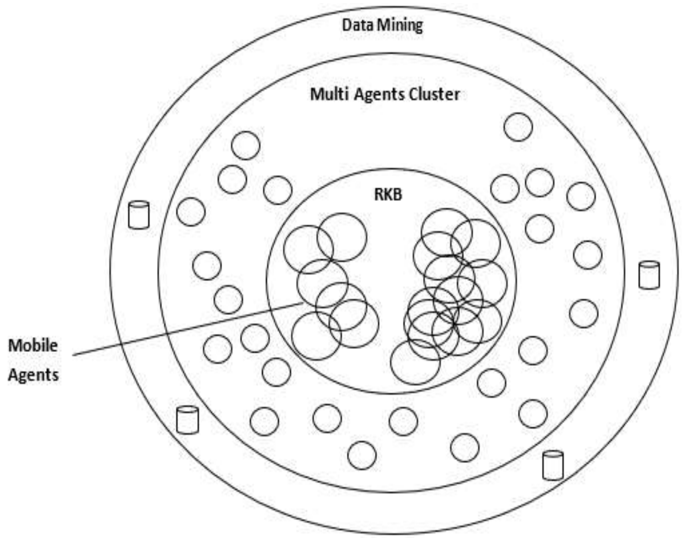

Figure 2. Knowledge refinement mechanism- the process of refining and integrating knowledge into a polished knowledge base.

In the intricate landscape of knowledge refinement, each agent possesses multi-objective attributes tailored to its specific domain. Within this environment, when a mobile agent enters, it acts as an input problem set, prompting the emergence of a relevant agent cluster orchestrated by adaptive automata. This automata mechanism uniquely boasts the inclusion of self-modifying features known as adaptive actions. These actions undertake the pivotal role of meticulously inspecting, adding, and deleting transitions. Consequently, reshaping the adaptive automaton post-execution, as depicted in **Figure 3**.

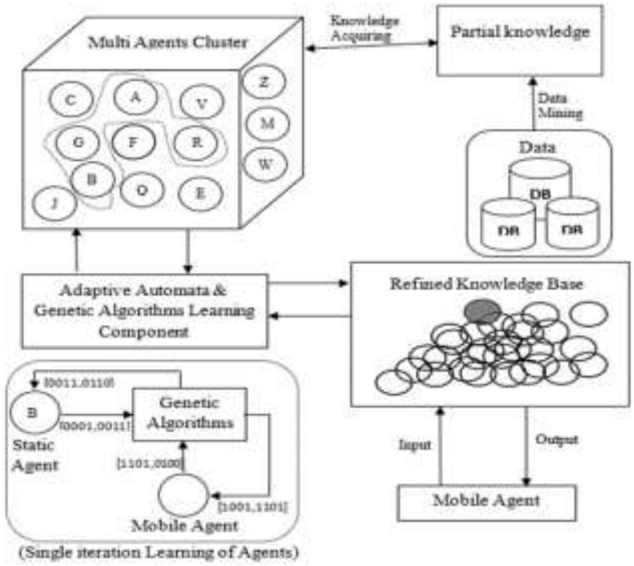

Figure 3. Proposed framework for knowledge refinement by mobile agents focusing on reshaping the adaptive.

The mobile agent, capitalizing on the inherent characteristics of the Adaptive Automata, integrates itself as a transition. Upon encountering the initial pertinent agent, the mobile agent embarks on a knowledge-enhancing journey by genetic algorithm methodologies. Particularly the processes of cross-over and mutation. As this agent traverses, encountering subsequent relevant agents, its repository of learned knowledge amplifies. Concluding the learning trajectory, the enriched mobile agent finds its place within the Refined Knowledge Base, contributing to the progressive evolution of refined knowledge. The correlation between Genetic algorithms, adaptive automata, and their techniques is the foundation of the proposed study. Allowing real-world refinements to a dynamic knowledge environment. In this framework, adaptive automata ensure that the system's learning capabilities improve by assessing and responding to transforming data patterns. GAs enhance this process by introducing comprehensive search processes and improving system parameters through transforming techniques. These techniques develop a comprehensive model for system intelligence using adaptive automata and fine-tuning performance standards by genetic algorithms.

Adaptive automata are the key part of the proposed work, allowing real-time adjustments to change the data environment. Automation is the computational agent that provides decision-making processes based on feedback. The self-modifying features of adaptive automata ensure the system captures real-time data patterns. Consider a manufacturing system where sensors introduce data on production techniques. Adaptive automata continuously evaluate the product quality produced by analyzing sensor data. If an anomaly is detected such as a lack of product efficiency, then adaptive automata refine its parameters to improve system accuracy in detecting anomalies. However, the genetic algorithms could improve by adjusting their parameters. Through various stages of selection, mutation, and crossover, the GA would develop a set of parameters that increase the performance in detecting anomalies.

3. Adaptive automata

Adaptive automata encompass a unique mechanism that permits self-modification. This is accomplished when adaptive actions, which are linked to their state-transition rules, are invoked at the moment of transition application. This form of automata is at times referred to as structured pushdown automata, a variant of the classical pushdown automata (Shultz et al., 2006). Within this, states are intricately organized into mutually recursive finite-state submachines. Though the structured pushdown automata and the classical pushdown automata share equivalency in functionality, one cannot overlook the occasional intricacy of the adaptive automata. This complexity can, at times, challenge comprehensibility and manageability.

For Adaptive automata to perform self-modification, adaptive actions attached to their state-transition rules are activated whenever the transition is applied. Adaptive automata are also known as structured pushdown automata. Structured pushdown automata are a variant of classical pushdown automata (Shultz et al., 2006), in which states are clustered into mutually recursive finite-state submachines. Structured pushdown automata are fully equivalent to classical pushdown automata. However, despite these factors, sometimes lack adaptive automata simplicity (Zorzo et al., 2011), making them difficult to understand and maintain.

4. The adaptive automata mechanism

The crux of the adaptive step lies in altering the Mobile agent's behavior to better correspond with the evolving set of regulations that define its essence. Within the realm of Adaptive Automata, a dual-phase process ensues. The preliminary phase necessitates the establishment of rules for the adaptive transition by electing a Multi-agents Cluster before rule execution. Subsequently, the utilization of the attached transition rules materializes. Addressing this dual process, it's pertinent to assign a duo of adaptation measures, one pre-transition, and the other post-implementation, to cater to the non-adaptive regulations. In the action of adaptation, a specific notation marked by the plus sign becomes paramount to append to the existing transition list. Such modus operandi, incorporated within an automaton possessing adaptivity, sanctions the dynamic evolution of the rule set.

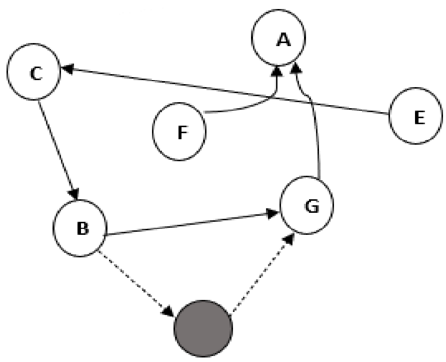

Figure 4. Multi-agents cluster.

As shown in **Figure 4**, the construct of different agents in multi-agent clusters, the properties of adaptive automata apply to a Mobile agent for mapping the best relevant agent. The Mobile agent represented as a shaded state in **Figure 4** represents the Transition Added in such a group of agents. When the Mobile Agent meets the relevant static agent then Genetic Algorithm rules apply and learn new things and then move further relevant static agent. Mobile Agent is learning one by one from every specific agent and evolves itself knowledge, and the whole sequence shows a proper cluster of the appropriate agents. The overall process of the movement of Mobile agent is linearly represent in **Figure 5**.

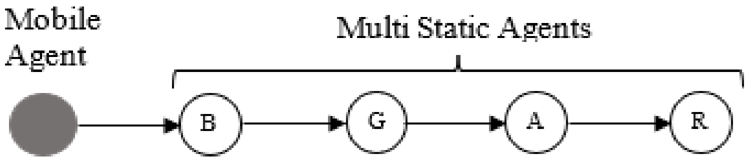

Figure 5. Multi-static agents cluster in sequence.

5. Genetic algorithms

The majority of evolutionary algorithms are adaptive heuristics search algorithms known as genetic algorithms (Denny, 2013). Drawing inspiration from the principles of natural selection and evolution, genetic algorithms offer an array of potential solutions to a designated problem. Typically, these algorithms thrive in environments abundant in candidate solutions. While genetic algorithms exhibit versatility across diverse environments. Specific algorithms tailored to particular scenarios might overshadow their efficacy, especially in the realm of straightforward search tasks. An observable limitation is the extended computational duration required by these algorithms, rendering them unsuitable for immediate real-time applications. Despite these constraints, genetic algorithms stand out as proficient techniques, delivering high-caliber solutions within a reasonable timeframe. In the context of this research, the Mobile agent harnesses the power of genetic algorithms for its learning trajectory. Crucial components of the Genetic Algorithm encompass Random Population, Fitness, Selection, Crossover, Mutation, and Acceptance (Ghnemat et al., 2007).

An overarching perspective presents the genetic algorithm's functionality as an evolutionary process, engaging the entire population of mobile agents. Within each interaction, the Mobile Agent, upon encountering a relevant static agent within the Multi Static Agents Cluster, employs the genetic algorithm for knowledge acquisition. Sequentially engaging with each pertinent static agent, the Mobile Agent undergoes transformative phases of crossover and mutation. Upon concluding this cycle, the Mobile agent reverts to the RKB, leading to knowledge refinement (Won et al., 2012) and subsequent output generation. An in-depth exploration of the genetic algorithm will be presented in the ensuing case study section.

6. Case study-1

This research delves into the simulation of the Hospital Management Hierarchy, aiming to refine and enhance the strategic process for optimal efficiency and reliability within the healthcare setting. The medical domain characterizes the Hospital Management Hierarchy with the Medical Superintendent (MS) at the pinnacle, serving as the key decision-maker and overseer of the entire hospital staff. Beneath the MS, a tier of Additional Medical Superintendents (AMS) emerges, tasked with overseeing various departments and holding a pivotal role within the hierarchical structure.

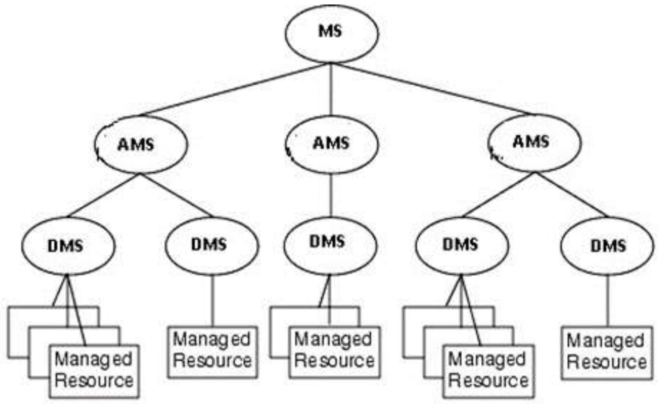

Figure 6. Evaluation of the initial population (AMS & 1st DMS).

Positioned a step below, the Deputy Medical Superintendents (DMS) operate under the guidance of the AMS, each directing a specific department's managerial operations. It's noteworthy to mention that the DMS layer is symbolic of data mining across diverse segments, interfacing directly with the foundational data. For illustrative purposes, consider the depiction in **Figure 6**.

The mobile agent exemplifies the AMS, whereas the multi-agent symbolizes the DMS. Envisage a scenario where the MS mandates the collection of data concerning Dengue—a prevalent epidemic. The AMS, upon receipt of this directive, liaises with the corresponding DMS in charge of the department attending to such epidemics. This DMS then coordinates with various stakeholders, from physicians and nurses to data entry operators, each holding a piece of the puzzle regarding dengue patients. These granular data points encompass disease specifics, medication schedules, economic implications, and socio-environmental conditions. The synthesis of this data by the DMS epitomizes the process of refining fragmented knowledge. Once collated, this data is channeled upwards to the MS, crystallizing as consolidated insights. The overarching objective is to harness this data, ensuring enhanced patient care and strategic countermeasures against the epidemic's spread.

Table 1. Implementation of crossover and mutation functions.

String No ***i***	**Population (Genotypes)**	***X* value (Phenotypes)**	**Fitness** $f(x) = x2$	**Probability of *i***	**Expected count**
1	1101	13	169	0.867	3.48
2	0001	1	1	0.005	0.02
3	0100	4	16	0.082	0.32
4	0011	3	9	0.046	0.18
Sum			195	1	4.00
Average			48.75	0.25	1
Max			169	0.867	3.48

Transitioning to the computational facet of this study, the learning mechanism of AMS is evaluated using genetic algorithms, as shown in **Table 1**. Here, the knowledge base of both AMS and DMS (derived from the problem set obtained from MS) is explored. Imagine that both AMS and DMS possess knowledge represented as quad-binary digits. Such binary configurations allow a clear visual of the crossover and mutation processes. The spectrum of a 4-digit binary can span 16 values, ranging from 0 to 15. For instance, take randomly generated genotype solutions such as 1101, 0001, 0100, and 0011, as shown in **Figure 7**.

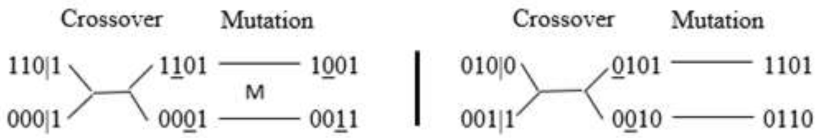

Figure 7. Hospital management hierarchy.

While the first and third strings signify AMS knowledge, the latter two are indicative of DMS insights. Employing a fitness decoding mechanism, each binary string is translated into an integer (denoted as phenotypes), presented in **Table 2**. The fitness evaluation is ascertained using the equation f (x) = x^2, also elaborated in **Table 2**. The Expected Count is derived from the equation: Expected count = N × Probability, with N symbolizing the population size, set at 4 in this instance.

The next phase involves intricate genetic algorithm computations, aiming to ascertain the succeeding generation's adaptability to the future environment. Initial steps encompass the selection of random 4-digit strings from a given population. Post selection, a specific mathematical criterion is employed to gauge the fitness of these strings. Subsequently, survival probabilities for each string are calculated, ranging between 0 and 1. This probability further aids in computing the survival chance for each string within the sampled population scaled between 0 and 4.

The crossover mechanism, employing a singular cut point, is then initiated. The first two strings are interchanged, followed by a similar crossover for the subsequent pair. The offspring resulting from this exercise is showcased in **Table 2**. A subsequent mutation is performed on these offspring, albeit with minimal alterations. The outcome is a new generation of populations: 1001, 0011, 1101, 0110, which supplant the original sets (1101, 0001, 0100, 0011) respectively.

(1). 1101 ≥ 1001 (2). 0001 ≥ 0011

(3). 0100 ≥ 1101 (4). 0011 ≥ 0110

Table 2. Evaluation of the next population.

String No ***i***	**Population (Genotypes)**	***X* value (Phenotypes)**	**Fitness** $f(x) = X^2$	**Probability of *i***	**Expected count**
1	1001	9	81	0.216	0.864
2	0101	5	25	0.067	0.268
3	1101	13	169	0.451	1.804
4	1010	10	100	0.266	1.064
Sum			375	1	4.00
Average			93.75	0.25	1
Maximum			169	0.451	1.804

Hence, the total fitness has gone from 195 to 375 after a single generation, as shown in **Table 2**. This process runs again and again with new knowledge of the next DMS until all the relevant DMS are completed through adaptive automata.

7. Case study-2

This research delves into a simulation of efficient room allocations. It introduces a significant challenge for organizations such as Hospitals, Businesses, and Universities, where the need for space should be balanced. This case study is dedicated to simulating a Room Allocation System (RAS) implementing genetic algorithms to improve allocation strategies. RAS defines the allocation issues, which involves assigning unoccupied rooms to many users while addressing key limitations such as room capacity, scheduling and user preferences unlike traditional systems that often depend on static heuristics, Genetic algorithms-based techniques address dynamically evolving requirements. This makes it specifically useful for larger departments, where quick refinements and opposing demands are frequent. Each solution provides a potential room allocation structure. For example, the solution might encode which department is allocated to which room, considering different parameters such as the number of people and size of the room. This encoding develops the foundation of genetic operations such as mutation, crossover, and selection. The result minimizing unused space, overcapacity, and customization of adjacency. The Genetic Algorithms perform repeatedly, generating a random population of feasible solutions. High-quality solutions are selected for their fitness scores.

The crossover operations integrate the components of parent solutions to develop offspring and enable the discovery of new solutions. Additionally, mutation operations define some random changes in the solution, which leads to exit local optima and discovery of new frontiers.

The test was conducted to evaluate the system's efficiency. The study includes allocating 50 rooms to 20 departments at a medium-sized institute. Each department had many students and specific requirements, such as the adjacency of laboratories and offices. The solution successfully handles limitations and avoids scheduling conflicts. The genetic algorithm illustrated its efficiency by improving dynamic changes such as quick adjustments in room allocation within 30 seconds as shown in **Table 3**.

Table 3. Room allocation comparison.

Constraints	Manual Allocation	Genetic algorithms-based
Room capacity conflicts	18%	10%
Scheduling	15%	0%
Adjacency preferences	60%	90%
Unoccupied space	25%	5%

The system uses GAs to allocate rooms, considering students' room preferences and other parameters such as age, department demands, and adjacency requirements. The detailed methodology of the strategic planning and frameworks for the proposed solution. A visual representation of the framework is divided in **Figure 8**.

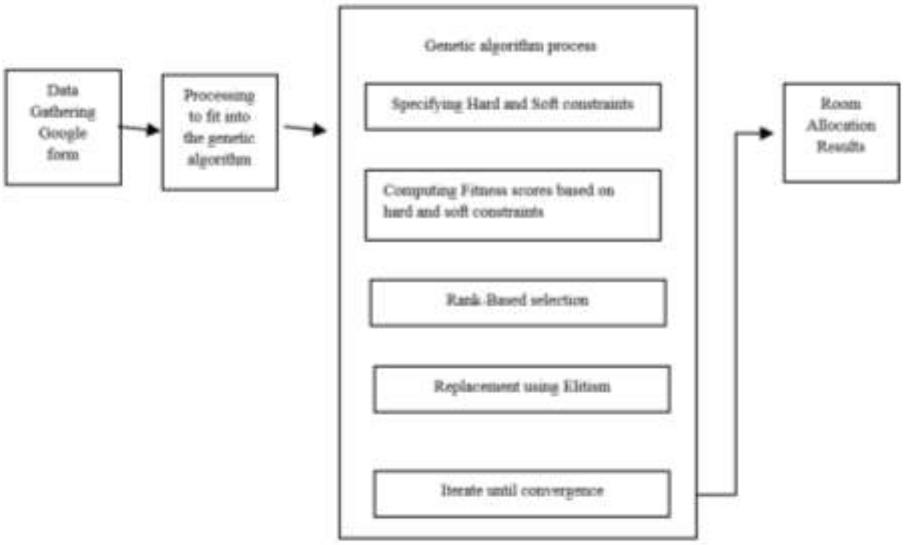

Figure 8. Framework for development of room allocation system.

Data collection is the basic part of the system, as it captures the students and room requirements for room allocation. Surveys were conducted by Google Forms to gather data from students living in two hostels one is male and one is female hostel. A total of 110 responses were collected. After the data collection process, preprocessing stages were conducted to prepare data for genetic algorithms such as data merging, cleaning, and enrichment. The study consists of solving the limitations of optimization issues with both hard and soft constraints. Hard constraints are non-negotiable and must be satisfied for the preferences. For example, age and room constraints. Soft constraints are not so necessary but recommended. For example, department, wing, and entrance preferences.

The genetic algorithms validate the fitness scores to check how well they satisfy the specific constraints. After getting a solution, each solution is ranked according to its score. Solutions that resolve more constraints are ranked higher. In crossover, room allocation from parent solutions is integrated to develop offspring solutions. For example, Crossover must involve exchanging room allocations between different students who can share the same department.

The mutation process introduces random changes to room allocations to prevent convergence. Ensures that genetic algorithms explore new solutions, specifically improving the overall score fitness of the population. The elitism method is used for best-performing solutions. Ensures that solutions are conserved, and provides the techniques for improvements.

Resultantly, the algorithm maintains the effective room allocation without any changes. The genetic algorithm is repetitive. After each iteration, the population handles fitness evaluation, crossover, mutation, elitism, and ranked-based selection. The process continues until population convergence. Convergence means that an optimal solution has been addressed. Results are depicted in **Table 4**.

Table 4. Room allocations.

Emails	Hostel	Retain Current Room	Current Room	New room
Yekesman50@gmail.com	Bethal splendor	NO	A6	C9
Wuwepraman@gmail.com	Bethal splendor	NO	A22	B23
Nkelsmken50@gmail.com	Bethal splendor	NO	B14	B33
Dessyliam@gmail.com	Bethal splendor	NO	B14	D31
Chimanwoguwguw3938@gmail.com	Felicia Adebisi dada	YES	D31	D20
Yulongmeimei@gmail.com	Felicia Adebisi dada	YES	E5	E5
Yekesman50@gmail.com	Felicia Adebisi dada	YES	E26	E26

8. Conclusion and future research agenda

Multi-agents are mostly used for collaborative problem-solving in distributed environments. Many of these environment-dependent applications deal with empirical analysis and mining of data. This paper describes a mechanism for solving a problem by learning mobile agents and combining adaptive automata and genetic algorithms to achieve reliability, automation, and efficiency. Multi-agent-based data mining is a time-consuming and complicated procedure. Still, by making use of this combination we can minimize the time consumption, eliminate deadlocks, and automate the whole process of mining. This paper has defined knowledge-based learning of mobile agents by using partial knowledge of multi-agents and the output of this correspondence is problem-specific refined knowledge. This mechanism will be an efficient approach to extract reliable knowledge in many fields. It will be modeled as a system and validated by simulation. Finally, it can be implemented in real-time scenarios to get the required refined knowledge.

Author contributions: Conceptualization, WAK and MFK; methodology, WAK; software, WAK; validation, MA (Muhammad Ayub), MABUR, LAW and MH; formal analysis, WAK; investigation, WAK and FAK; resources, MABUR; data curation, WAK; writing—original draft preparation, WAK; writing—review and editing, MA (Muhammad Adeel), MUQ and AAS; visualization, WAK; supervision, MH and AAS; project administration, MUQ, MA (Muhammad Adeel) and AAS; funding acquisition, MABUR. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors declare no conflict of interest.

References

Abu-Qdari, S. A., & Faris, H. (2018). An Improved Genetic Algorithm with a New Initialization Mechanism Based on Regression Techniques.

Carlo Poloni, & Andrea Giurgevich. (2023). Hybridization of a multi-objective genetic algorithm, a neural network, and a classical optimizer for a complex design problem in fluid dynamics.

Chehri, A., Jakimi, A., & Saadane, R. (2024). Hyperparameter Optimization with Genetic Algorithms and XGBoost: A Step Forward in Smart Grid Fraud Detection.

Dogan and D. Birant, "Machine learning and data mining in manufacturing," Expert Syst. Appl., vol. 166, p. 114060, Mar. 2021, DOI: 10.1016/j.eswa.2020.114060.

Gao, Q., & Wang, T. (2024). Self-Regulating Automata for Knowledge Refinement in Dynamic Environments. International Conference on Machine Learning (ICML).

Goldberg and E. David, Genetic Algorithms in search, Optimization and Machine Learning, Pearson Education, Inc., 1989.

H. Denny, "Genetic Algorithm for Solving Simple Mathematical Equality Problem," arXiv preprint arXiv:1308.4675, p. 10, 2013.

Jiaxin Zhang, Wendi Cui, Yiran Huang, Kamalika Das, & Sricharan Kumar (2024). Synthetic Knowledge Investigation Towards Knowledge Refinement and Injection for Enhancing LLMs.

Jin Du, Yuhong Yang, & Jie Ding (2024). Adaptive Continuous learning: Knowledge Refinement.

L. M. Fraga, G. M. B. de Oliveira, and L. G. A. Martins, "Adjustment of an Epidemiological Cellular Automata-based Model using Genetic Algorithm," in 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), Nov. 2020, pp. 589–594. DOI: 10.1109/ICTAI50040.2020.00096.

Liu, C., Xu, L., & Wang, Y. (2023). A hybrid genetic algorithm for optimizing complex network parameters.

Ma, H., & Yu, L. (2023). Query Optimization for Parametric Knowledge Refinement in Retrieval-Augmented Large Language Models.

Miele, M., & D'Innocenzo, L. (2022). Genetic Algorithm for Machine Learning.

N. Dey, Applied Genetic Algorithm and Its Variants: Case Studies and New Developments. Springer Nature, 2023.

Neto, J. J., & Pariente, C. A. B. (2002). Adaptive automata—a reduced complexity proposal. In Conference on Implementation and Application of Automata-CIAA, 161-170.

P. Frasconi & M. Gori. (2024). A general framework for adaptive processing of data structures.

Pandey, T. N., Panda, N., & Ku, P. (2012). Improving the performance of distributed data mining (DDM) with a multi-agent system. International Journal of Computer Science Issues (IJCSI), 9(2).

R. Ghnemat, C. Bertelle and G. H. E. Duchamp, "Adaptive Automata Community Detection and Clustering: A generic methodology," In World Congress on Engineering, pp. 25-30, 2007.

S. AMERSHI and C. CONATI, "Combining Unsupervised and Supervised Machine Learning to Build User Models for Exploratory Learning Environments.," Journal of Educational Data Mining, vol. 1.1, pp. 18-71., 2009.

S. D. Zorzo, P. R. M. Cereda and R. A. Gotardo, "Adaptive Automata in Recommendation Systems," Latin America Transactions, IEEE (Revista IEEE America Latina), vol. 9(2), pp. 152-159, 2011.

S. Russell and P. Norving, Artificial intelligence: a modern approach, Vol. 2, Upper Saddle River, New Jersey: Prentice Hall, 2003.

T. A. El-Mihoub, A. A. Hopgood, and L. Nolle, "Self-adaptive learning for hybrid genetic algorithms," Evol. Intell., vol. 14, no. 4, pp. 1565–1579, Dec. 2021, DOI: 10.1007/s12065-020-00425-5.

T. R. Shultz, F. Rivest, L. Egri, and Thivierge, "Knowledge-based learning with KBCC. Proceedings of the Fifth International Conference on Development and Learning ICDL," Department of Psychological and Brain Sciences, Indiana University, 2006.

Taha, Z. Y., Abdullah, A., & Faris, H. (2024). Optimizing feature selection with genetic algorithm: A review of methods and applications.

Won, Chaehwan, K. Jinhwa, and B. K. Jae, "Using genetic algorithm-based knowledge refinement model for dividend policy forecasting," Expert Systems with Applications, 2012.

Youan Cong., Cheng Wang., & Pritom Saha Akash. (2024). Query optimization for parametric knowledge refinement in retrieval-augmented LLMs.

Yuhao Wu, Tongjun Shi. (2023). Online Continual Knowledge Learning for Language Models.

Yuhong &, Jie. (2024) Adaptive Continual Learning: Rapid Adaptation and Knowledge Refinement.

Yuqing Zhao. (2023). AdaptCL: Adaptive Continual Learning for Tackling Heterogeneity in Sequential Datasets

Z. A. Dahi, E. Alba, and G. Luque, "A takeover time-driven adaptive evolutionary algorithm for mobile user tracking in pre-5G cellular networks," Appl. Soft Comput., vol. 116, p. 107992, Feb. 2022, DOI: 10.1016/j.asoc.2021.107992.

Zhang, F., & Li, X. (2023). Adaptive Continuity and Knowledge Refinement in Continual Learning Systems. ICLR.

Zhao, X., & Sun, X. (2023). Plan-on-Graph: Self-Correcting Adaptive Planning of Large Language Model on Knowledge Graphs.