aerospace

MDPI

Article

Organization Preference Knowledge Acquisition of Multi-Platform Aircraft Mission System Utilizing Frequent Closed Itemset Mining

Yuqian Wu [1], Miao Wang [1,*], Wenkui Chu [2] and Guoqing Wang [1]

[1] School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai 200240, China
[2] Aeronautics Engineering College, Air Force Engineering University, Xi'an 710100, China
* Correspondence: miaowang@sjtu.edu.cn

Abstract: Organization preference knowledge is critical to enhancing the intelligence and efficiency of the multi-platform aircraft mission system (MPAMS), particularly the collaboration tactics of task behaviors, platform types, and mount resources. However, it is challenging to extract such knowledge concisely, which is buried in massive historical data. Therefore, this paper proposes an innovative data-driven approach via frequent closed itemset mining (FCIM) algorithm to discover valuable MPAMS organizational knowledge. The proposed approach addresses the limitations of poor effectiveness and low mining efficiency for the previously discovered knowledge. To ensure the knowledge effectiveness, this paper designs a multi-layer knowledge discovery framework from the system-of-systems perspective, allowing to discover more systematic knowledge than traditional frameworks considering an isolated layer. Additionally, the MPAMS's contextual capability reflecting the decision motivation is integrated into the knowledge representation, making the knowledge more intelligible to decision-makers. Further, to ensure mining efficiency, the knowledge mining process is accelerated by designing an itemset storage structure and three pruning strategies for FCIM. The simulation of 1100 air-to-sea assault scenarios has provided abundant knowledge with high interpretability. The performance superiority of the proposed approach is thoroughly verified by comparative experiments. The approach provides guidance and insights for future MPAMS development and organization optimization.

Keywords: multi-platform aircraft mission system; organization preference knowledge; data mining; frequent closed itemset mining; wargame

check for updates

Citation: Wu, Y.; Wang, M.; Chu, W.; Wang, G. Organization Preference Knowledge Acquisition of Multi-Platform Aircraft Mission System Utilizing Frequent Closed Itemset Mining. *Aerospace* **2023**, *10*, 166. https://doi.org/10.3390/aerospace10020166

Academic Editors: Carlos Insaurralde and Francesco Dell'Olio

Received: 20 November 2022
Revised: 9 February 2023
Accepted: 9 February 2023
Published: 10 February 2023

1. Introduction

1.1. Motivation

The multi-platform aircraft mission system (MPAMS) has been the promising trend in avionics with the development of the system-of-systems (SoS) concept, which undertakes many decision-making functions [1–3]. The key feature of the MPAMS is a shift of organization mode from the traditional discrete resource organization within individual avionics platforms to the mission-based collaborative multi-platform organization of tasks, platforms, and mount resources, enhancing the comprehensive performance of different avionics [4]. Specifically, valuable organizational knowledge (MPAMS organizational knowledge includes the valuable collaboration modes of tasks, platforms, and mounts, such as which platform collaboration mode is most efficient for completing a specific portfolio of tasks; this concept is further explained in Section 2.1) in the MPAMS has sparked widespread research attention [5] since it is critical for optimizing the MPAMS organization efficiency. However, acquiring such knowledge is challenging due to its unintuitive nature, high dependency on human experience, and hiddenness in massive combat data.

The methodological paradigm [6] for discovering the above-mentioned knowledge has transferred from knowledge-based reasoning [7–10] to data-driven learning. The latter paradigm could automatically and adaptively acquire knowledge from massive data [11–14], which possesses a wider application prospect than the former paradigm considering the inefficient manual knowledge modeling and deterministic behavior rules [15].

Particularly, frequent closed itemset mining (FCIM) [16] is a data-driven method, that outperforms other data-driven methods in discovering heterogeneous and global knowledge, benefiting from flexibly searching correlations within diverse dimensions, which is promising for practical applications. However, how to successfully apply FCIM to deal with complex MPAMS organizational data is a new and interesting problem, especially what type of preference information should be extracted to reflect useful knowledge from the knowledge effectiveness perspective and how to extract knowledge faster from the knowledge efficiency perspective.

Our motivation lies in discovering valuable and systematic organizational knowledge from massive MPAMS organizational data via FCIM, thus contributing to the optimization of the MPAMS organization.

Furthermore, it is notable that the datasets of MPAMS organization are different from general datasets, which possess more dense and complex input features and a special output itemset format (further refer to Section 2.2). To discover knowledge efficiently, it is necessary to design new itemset structures and corresponding pruning strategies to accelerate the mining process.

1.2. Related Works and Gaps

First, the latest progress in MPAMS knowledge mining methods, regarding knowledge effectiveness, is summarized. Then, the advancement of FCIM algorithms, with respect to efficiency of knowledge mining, is introduced. The criteria to evaluate the research lie in the knowledge type and knowledge interpretability, as shown in Table 1.

(1) MPAMS organizational knowledge mining methods-knowledge effectiveness issue.

Knowledge discovery on MPAMS has evolved from early knowledge-based reasoning approaches, such as expert systems [7], behavior trees [8,9], and state machines [10], to data-driven learning approaches. As data-driven ones could take advantage of massive historical data or wargame simulation [17,18] to extract valuable organizational patterns. Thus far, clustering [19–21], decision trees [22], deep learning [23], etc., have been explored to obtain this knowledge. The literature on data-driven approaches primarily focuses on platform or task behavior collaborations [24]. The research topic includes typical team task behavior strategies [19–21], the team composition and interaction patterns [22], and the synergistic platform recommendation [23].

The problem, however, is the dimensionality of the extracted knowledge type. Current advancements tend to construct specific homogeneous collaboration modes toward the targeted goal, which limits the knowledge representation for heterogeneous feature synergy, as well as the transferability to other scenarios. In fact, the organization of tasks, platforms, and mounts in a MPAMS is quite complex, and the knowledge to be organized may be heterogeneous and mutually influenced.

Compared to other data-driven learning algorithms, FCIM can directly mine homogeneous or heterogeneous collaboration relationships in the MPAMS organization context, which is more flexible to capture abundant types of knowledge.

For the knowledge effectiveness issue, namely what information should be selected to effectively denote knowledge, the basic idea is to find intrinsic dependencies between features of the MPAMS organization considering task behaviors, platform type, and mount usage. Lin et al. [25] proposed a decision-support scheme containing FCIM to analyze collaboration probabilities among platforms. Schwartz et al. [26] applied FCIM to discover valuable task behavior sequences among current states, occurring events, and user actions from wargame logs. Nevertheless, a single type of correlation between organizational features cannot cover the strike performance of a certain preference mode, so knowledge effectiveness cannot be guaranteed.

Therefore, some advanced studies have aimed to extract causal correlations between various organizational features and striking effectiveness, which renders valuable collaboration modes with excellent performance, thereby contributing to MPAMS organization optimization. Cao et al. [27] introduced FCIM to analyze typical correlations between surprise defense efficiency and various strike modes. Li et al. [28] used a growth FCIM to extract associative target selection rules of electronic countermeasures, rending higher organizational rationality. Xing et al. [29] analyzed the effectiveness of various platform and mount collaboration modes against opponents in terms of strike performance causality.

Still, the discovered knowledge remains ineffective in the MPAMS organization. The limitations are as follows.

- Limited comprehensive level. The discovered knowledge cannot lead to a systematic and overall understanding of the MPAMS organization. Since current research has only focused on isolated organizational decisions at the separate task, platform, or mount layer. It overlooks the practical correlated multi-layer collaboration modes of the MPAMS architecture in the SoS context, which limits its applicable value.
- Poor interpretability. The discovered knowledge shows only "what to do" but overlooks "why to do". Such as the knowledge indicates which weapon combinations will produce favorable results, but does not mention the situation background [30,31], i.e., the motivation for this decision. As a result, the isolated knowledge assertion lacks both intrinsic interpretability for the commander and rationality to migrate to similar scenarios.

(2) Frequent closed itemset mining-knowledge efficiency issue.

Knowledge mining efficiency has been the most important aspect of evaluating FCIM performance, as measured by the time spent and the memory occupied to produce knowledge. Various improvements to the itemset storage structure and pruning strategies have been proposed to improve efficiency [32]. Among them, intersection-based FCIM algorithms [33–39] have served as an outstanding branch, that uses intersection to compute new itemsets in an incremental manner. *Ciclad* [37], *CloStream* [38], and *Moment* [39] are some typical algorithms that effectively improve mining efficiency by designing new itemset storage structures or landmarks. Nonetheless, current efficiency advances leave gaps in MPAMS organizational knowledge mining. The limitation lies in:

- Inadequate speed. It is challenging for FCIM to deal with large-scale MPAMS datasets, which possess a complex and dense item distribution, leading to a knowledge construction dilemma with large time consumption. Therefore, the candidate itemset scale, that is, the redundant information in the mining process, should be further reduced to accelerate knowledge discovery, making it capable of handling more complex and large-scale datasets.

Table 1. Research status and gaps of knowledge mining effectiveness and efficiency issues.

Dimension	Method Type		Research Topic	Feature	Drawback
Knowledge effectiveness	Other data-driven methods such as clustering [19–21], decision tree [22], and deep learning [23].		Task behavior strategies [19–21], team composition patterns [22], and synergistic platform recommendation [23].	Knowledge type: Based on the target problem, mainly heterogeneous features. Knowledge interpretability: Low, due to black box mode and scarce background information.	(1) Limited knowledge representation flexibility for heterogeneous features. (2) Must design a specific algorithm to fit the target problems, not flexible to transfer.
	FCIM	Basic	Discover intrinsic correlations within platforms [25], mount features [25], or task behaviors [26].	Knowledge type: MPAMS organizational features. Knowledge interpretability: Low, due to vague strike effectiveness feature.	(1) Localized knowledge composition at separate layers, which neglects multi-layer collaboration modes in MPAMS. (2) Limited understanding of the decision motivation, which restricts the knowledge transferability
		Causality oriented	Discover correlations between strike effectiveness and above-mentioned task behaviors [27], platforms [28,29], or mount features [29].	Knowledge type: Strike effectiveness + MPAMS organizational features. Knowledge interpretability: Higher, by incorporating causality correlation.	
Knowledge efficiency	FCIM		Design new item storage structures and pruning strategies [33–39].	Reduce searching space and accelerate mining speed.	Cannot adapt to the features of MPAMS datasets.

1.3. Contributions

This study focuses on discovering valuable MPAMS organizational knowledge from historical data and strengthening the effectiveness and efficiency of this knowledge by proposing an efficient knowledge discovery method based on the FCIM. The proposed method can enhance the organizational certainty of MPAMS for improved strike performance.

As shown in Figure 1, the main contributions of this study are as follows:

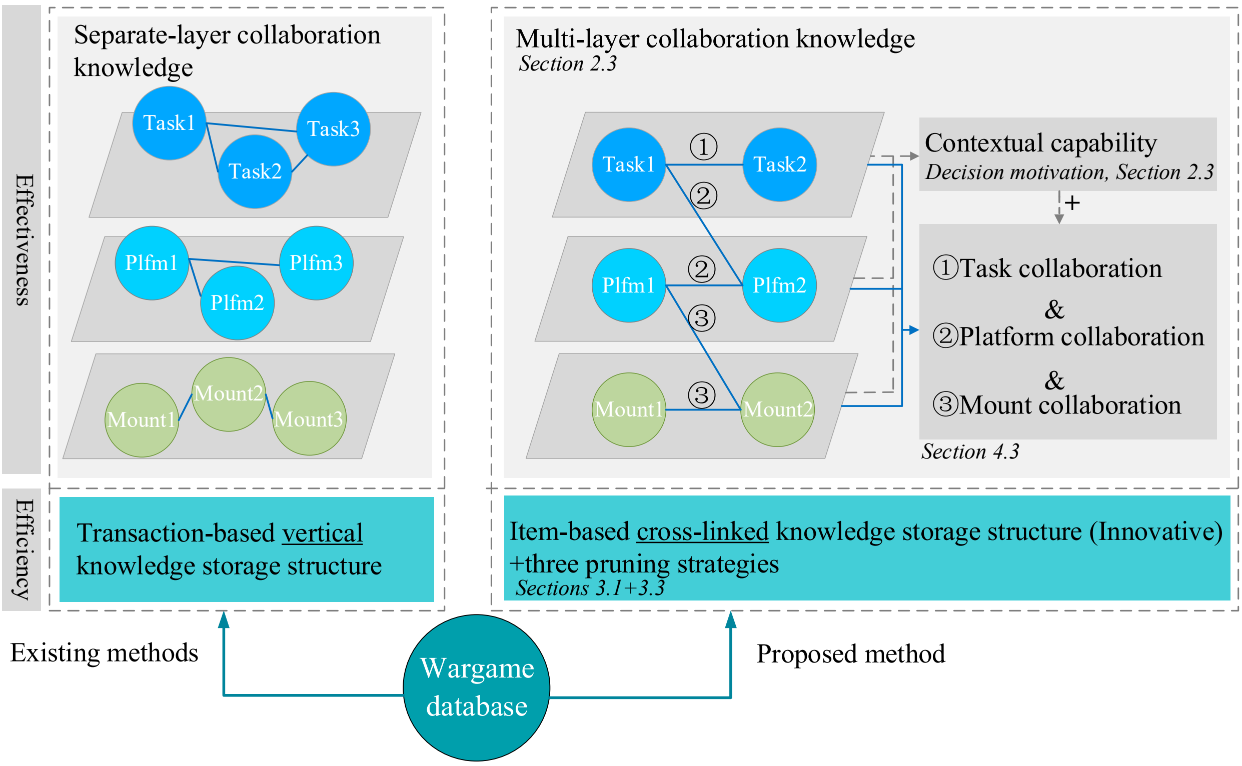

Figure 1. Comparisons between the existing knowledge discovery methods and the proposed method. (Plfm: Platform).

(1) Knowledge Effectiveness: A data-driven multi-layer organization preference knowledge discovery framework, which fits the MPAMS's hierarchical architecture better, is designed from the SoS perspective, thus discovering more systematic heterogeneous knowledge than traditional homogeneous knowledge using isolated-layer frameworks of MPAMS. That greatly enhances the global awareness of the MPAMS organizational patterns.

Moreover, the contextual capability of MPAMS is first integrated into knowledge representation, which reveals the decision motivation, thus making the knowledge more interpretable and credible to the commander, and elevating the knowledge's practical value to apply and transfer to different scenarios (Section 2.3).

(2) Knowledge Efficiency: An innovative cross-linked itemset storage structure and three pruning strategies that adapt to the data distribution of MPAMS are developed, which can further reduce the itemset search space and speed up the knowledge mining process. Therefore, the proposed method is promising to analyze large-scale MPAMS datasets for complex knowledge (Sections 3.1 and 3.3).

(3) Valuable MPAMS organizational preference knowledge embedded with contextual capability factors is acquired from massive data on air-to-sea assault scenarios, which reflects multi-layer collaborative tactics of tasks, platforms, and mounts. The knowledge could advance the understanding of MPAMS organization, and contribute to the MPAMS organization optimization for better strike performance (Section 4.3).

1.4. Organization of the Paper

The paper is organized as follows: Section 2 provides relative concepts and then outlines the knowledge mining framework. Section 3 introduces the proposed FCIM algorithm called *CrossFCI* for knowledge mining, where an innovative cross-linked itemset storage structure together with three pruning strategies is depicted. Section 4 presents experiments on both public datasets and MPAMS datasets to verify *CrossFCI*'s performance superiority and extract valuable and systematic MPAMS organizational knowledge. Section 5 concludes the paper.

2. Proposed MPAMS Organizational Knowledge Mining Method

2.1. Preliminaries

This subsection introduces the concepts of MPAMS, MPAMS organizational knowledge, and FCIM. As MPAMS organizational knowledge is tightly embedded in MPAMS's organization modes, and FCIM serves as a reliable tool to extract this knowledge.

(1) MPAMS

An aircraft mission system is the future trend of avionics in the military field. The concept was first proposed for F-35, highlighting the decision-making support function of avionics [2]. Then the concept of the multi-platform aircraft mission system has been further proposed from the avionics architecture perspective to meet the requirements for flexible mission organization of systematic warfare.

The core feature of a MPAMS is to distribute the capabilities that are traditionally deployed within a single flight platform, such as reconnaissance, electronic attack, and strike, to several distributed and easily upgradable flight platforms. The MPAMS organization structure is presented in Figure 2, where it can be seen that it is dominated by combat missions and operational environments. Then, multiple operational tasks are accomplished by integrating capabilities scattered on different platforms and mount resources to enhance the joint capability.

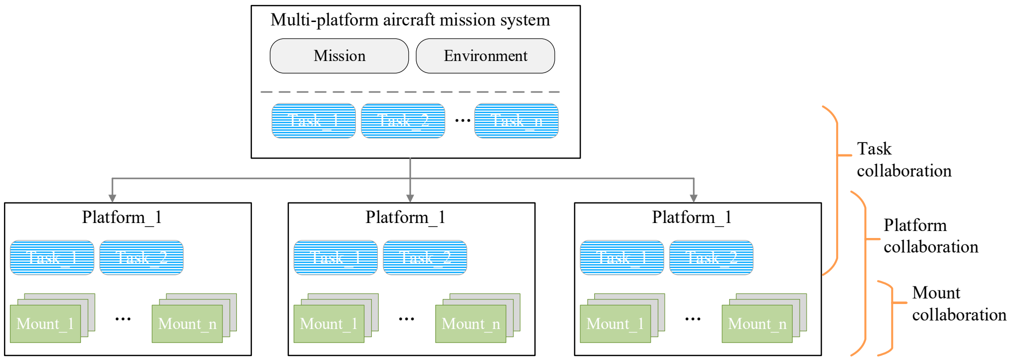

Figure 2. The block diagram of the MPAMS organization.

(2) MPAMS organizational knowledge

The organizational knowledge of a MPAMS represents the collaboration pattern consisting of tasks, platforms, and mounts in the MPAMS architecture. As combat platforms include various operational objectives in an increasingly complex battlefield, the ability to organize and collaborate tasks, aircraft platforms, and mounts in the MPAMS architecture has become the main technology of modern warfare. These collaborations reflect valuable and practical tactics for optimizing MPAMS performance.

In the SoS concept, the aforementioned knowledge is no longer limited to the inner correlations of an independent layer discovered by the existing methods, for instance, the collaboration of air early warning and fighters has a good strike effect; but more attention is mainly devoted to the multi-layer collaboration, as shown by Equation (1). As the MPAMS organization represents an SoS engineering problem, where elements of different layers are crosslinked, decisions must be made from the global view perspective to improve the comprehensive performance of the MPAMS architecture.

The MPAMS organizational knowledge can be expressed as follows:

$$\begin{aligned} Knowledge < task, platform, mount > \Leftrightarrow \ & \{task \leftrightarrow task, task \leftrightarrow platform, \\ & platform \leftrightarrow platform, platform \leftrightarrow mount, \\ & mount \leftrightarrow mount, task \leftrightarrow platform \leftrightarrow mount\} \end{aligned} \tag{1}$$

The above expression demonstrates which task organization mode is preferred, how to organize platforms to execute certain task collaborations most efficiently, and which mount resources perform best with platforms to complete tasks.

(3) FCIM

FCIM is one of the major techniques of data mining, which scans the data collection for frequent and closed itemsets. Let $I = \{i_1, i_1, \cdots, i_n\}$ represents a set of items, $X \subseteq I$ denotes a subset X within I. D denotes all the transactions in the database. Each transaction is stored via (tid, *itemset*), which is uniquely identified by a tid. The left part of Table 2 gives an example of a transaction database.

Table 2. Examples of the transaction sets and the derived frequent closed itemsets.

Transaction Sets *D* in the Database		**Frequent Closed Itemset Family for *D***		
		1-*abcd*:2	9-*a*:5	
(1, *abcdefgh*)	(6, *abdefh*)	2-*abd*:3	10-*bcfgh*:2	17-*cg*:3
(2, *abef*)	(7, *abcd*)	3-*ab*:4	11-*bc*:4	18-*c*:5
(3, *bcfgh*)	(8, *bc*)	4-*acdg*:2	12-*befh*:3	19-*d*:5
(4, *befgh*)	(9, *d*)	5-*acd*:3	13-*bef*:4	20-*gh*:4
(5, *acdg*)	(10, *gh*)	6-*adfh*:2	14-*bfh*:4	21-*g*:5
		7-*ad*:4	15-*bf*:5	22-*h*:5
		8-*aef*:3	16-*b*:7	

Assuming D comprises a tidset Y, $\iota_D(Y) = \cap\{Z|(j,Z) \in D, j \in Y\}$ represents itemset intersections Z within different transactions in Y. For short, $\iota(j) = Z$ iff $(j,Z) \in D$, omitting subscript D. Moreover, the transaction tidsets that cover itemset X could be formulated via $\tau_D(X) = \{j|(j,Z) \in D, X \subseteq Z\}$. For example, $\tau_D(ab) = \{1,2,6,7\}$ and $\iota_D(\{1,6,7\}) = abd$. The itemset significance is evaluated by the frequent itemset and closed itemset, which exhibit universality and representativeness of the itemset, respectively.

Definition 1. *Frequent itemset (FI): an itemset whose support is greater than* ς.

The support value of the itemset X is $\sigma(X) = |\tau(X)|$, and the minimal support threshold is denoted as ς, defining the binary criterion of FI.

Definition 2. *Closed itemset (CI): an itemset whose* $\sigma(\cdot)$ *differs from that of all superset itemsets.*

The CI means an itemset has enough important correlations, and the content is different from other superset CIs, which could be measured by support value, corresponding to the anti-monotony principle of support value. For instance, {*abcd*} is a CI in D while {*abc*} is not. As {*abcd*} is a superset of {*abc*}, and $\tau(abcd) = \tau(abc) = \{1,7\}$. Moreover, X_D denotes the CI of X. The CIs in D are illustrated by C(D). Then, following the idea of formal concept analysis [40], a closure operator κ is introduced to describe C(D) via the composition of the operator.

Property 1. *Assuming* $X \subseteq I, \kappa(X) = \iota(\tau(X))$.

For example, $\kappa(abc) = abcd$, as $\iota(\tau(abc)) = \iota(\{1,7\}) = abcd$ and $abcd = \max(abc_{\mathrm{D}})$. Moreover, $X = \kappa(X)$ iff X is closed. All the FCIs of D are outlined in the right part of Table 2, with the support threshold set as 1. Each FCI is appended with a support value. For instance, {*bc*} is a member of the FCI family with a support value of four as $bc = \cap\{1,3,7,8\}$.

2.2. Problem Description

This subsection begins with the introduction of MPAMS datasets in the form of transactions, and conceptual MPAMS organizational knowledge representation in the form of FCI. Then the problem of knowledge mining by FCIM is described through an example. Furthermore, the special features of MPAMS datasets are depicted in Figure 3.

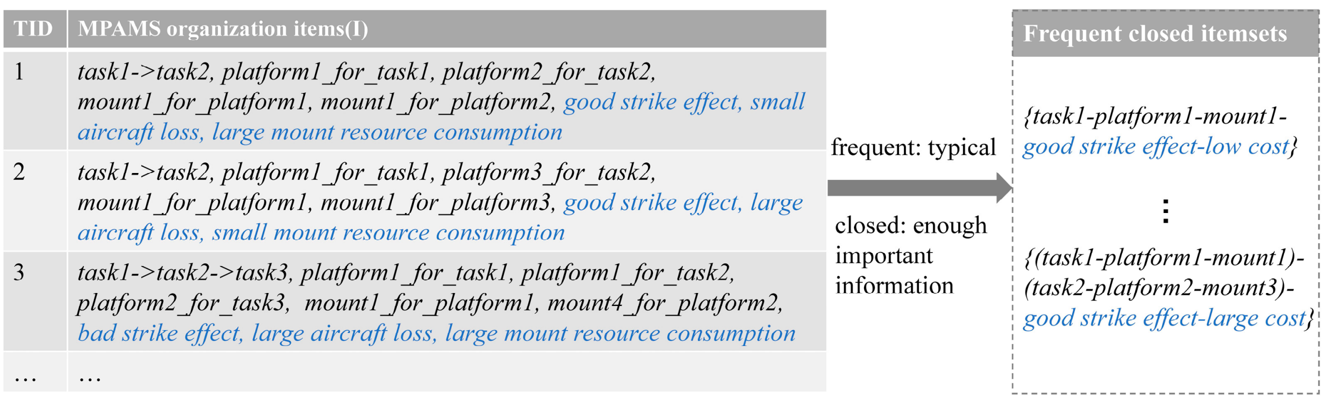

Figure 3. Problem description of the MPAMS organizational knowledge mining.

The MPAMS organizational knowledge represents an efficient collaboration mode with a good striking effect, which also implies the inherent thinking styles and intelligence of a commander. The knowledge is reflected in the correlations of multi-layer organizational paths of tasks, platforms, and mounts in the MPAMS architecture. The purpose of applying the FCIM to a MPAMS is to extract valuable collaboration modes by identifying strongly correlated organizational features from massive historical data of various scenarios. This knowledge mining problem is represented as follows:

$$f(\sum_i \tau_i(\iota_1, \iota_2, \cdots, \iota_k)) \Leftrightarrow FCI\{\iota_1, \iota_2, \iota_3\}, FCI\{\iota_2, \iota_3, \iota_4\}, \cdots \quad (2)$$

where transaction τ_i indicates a MPAMS organization scenario in the confrontation, which includes multiple items; ι_k reflects the item element to be correlated, which involves two main categories: the MPAMS organization element category and the firepower effectiveness category. The former refers to the heterogeneous task-, platform-, and mount-related features. For instance, the mount-related features can represent different types of missiles and electronic warfare weapons. The latter could include the integrity of an enemy's target, friendly aircraft loss, and friendly mount consumption. Then, FCIM is used to discover the correlations of various items, deriving a set of FCIs. A FCI denotes a valuable MPAMS collaboration pattern, which consists of the above-mentioned correlated items. Notably, the collaboration pattern must be typical, which is higher than the support threshold (FI) and contains enough important information (CI). For instance, FCI{*feint task, UAV, fighter, close-range air-to-sea missile, good effect, low cost*} denotes a collaboration mode, namely an unmanned aerial vehicle (UAV) and a fighter equipped with close-range air-to-sea missiles to conduct the feint task, is effective in the MPAMS organization, which could ensure a good strike effect at a low cost.

Moreover, as shown in Figure 3, the transactions for the MPAMS knowledge mining issue are different from traditional transactions, with more dense and complex input features and a special output itemset format. These features aggravate the difficulty of knowledge discovery. Therefore, the item storage structure and pruning strategies should be further designed for FCIM to accelerate knowledge discovery speed.

As mentioned in Section 1, the critical problems for applying FCIM to the MPAMS organizational knowledge discovery task are how to organize the information into meaningful itemsets to effectively reflect the character of MPAMS organization, and how to efficiently extract valuable collaboration knowledge, namely FCIs, from the original itemsets. The solution to the first problem is described in Section 2.3, and the solution to the second problem about efficiency is introduced in Section 3.

2.3. MPAMS Organizational Knowledge Discovery Framework

The proposed multi-layer knowledge discovery scheme is presented in Figure 4, where it can be seen that it includes five main phases. First, the configuration of a platform and mount features is extracted from a wargame replay database. Then, the attacking and defending sides' capabilities are encoded based on the above-structured features, and the strengths and weaknesses of both sides are evaluated further. Further, confrontation-related tasks are organized and serialized according to the combat platform actions. In addition, the firepower effectiveness is determined based on the platform's health status. The aforementioned causal features are combined into mineable confrontation feature itemsets. Finally, the FCIM algorithm is adopted to acquire correlations between diverse organizational features, rendering comprehensive organization preference knowledge considering task, platform, and mount collaboration.

Step 1: Construct aircraft platform and mount features

Aircraft platform and mount features are fundamental to represent the MPAMS architecture configuration, which are captured from raw wargame data. Particularly, aircraft platform features not only include basic information on the type and affiliated side, but also the information on task-related configurations of mobility, sensors, electronic countermeasures, defense, and data transfer. In addition, mount features include the striking range, hit probability, and damage point of a certain mount type. Then, the above-mentioned features are structurally encapsulated for mapping with the MPAMS capability in the following.

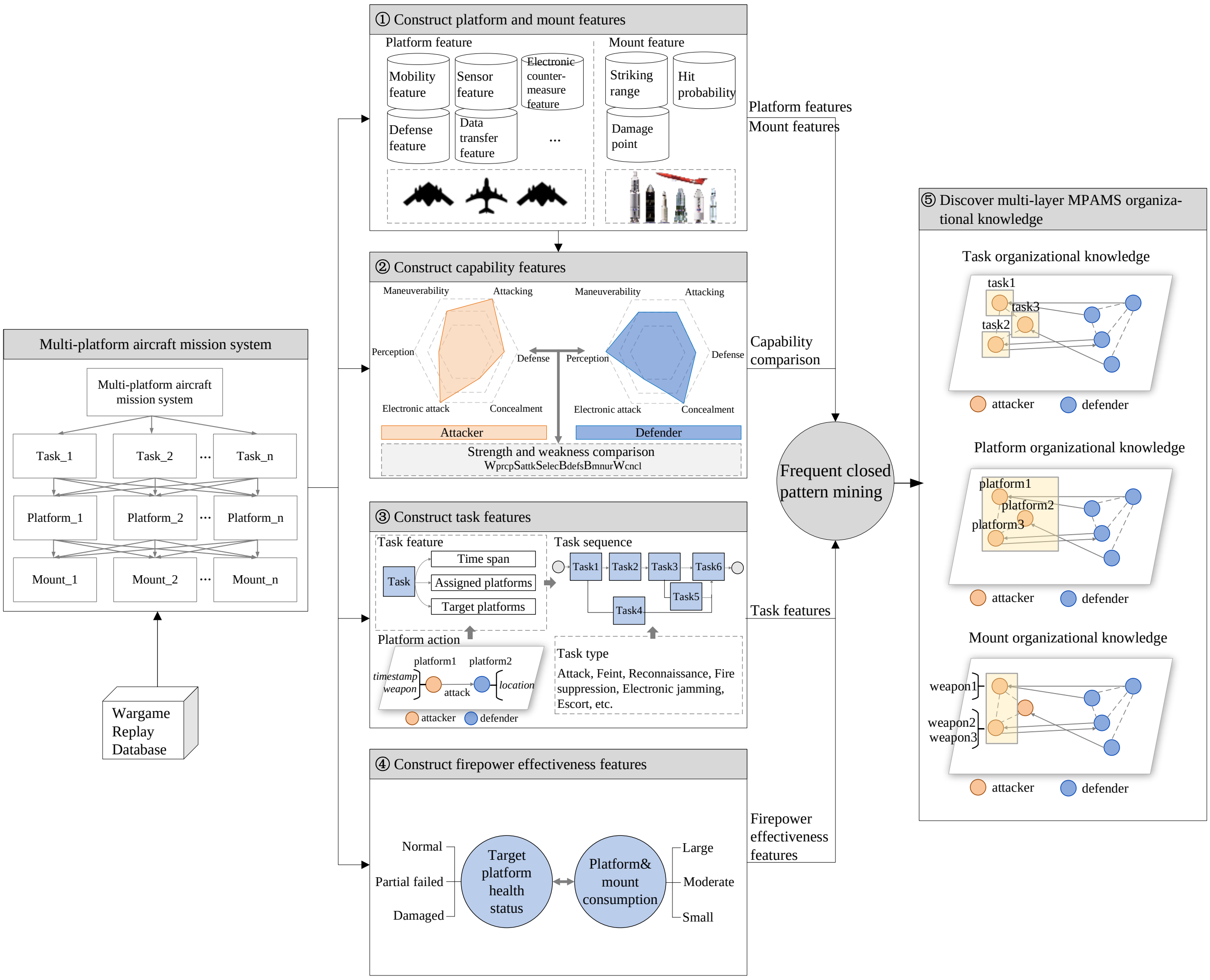

Figure 4. Framework of the MPAMS organizational knowledge discovery. (W: weak, B: balance, S: strong in step 2).

Step 2: Construct capability features

Capability features denote inherent features of one side to defeat the opponent, which is essential for MPAMS's striking strategy development. The MPAMS capabilities can be roughly categorized into six meta-groups based on practical decision-making processes [41–43]. In Table 3, each capability is encoded via the above-mentioned platform and mount features.

Table 3. Capability feature description.

Category	Symbol	Feature	Calculation Formula	Integration
Perception	C_{prcp}	$f(Range, P)$	detection range × detection probability	Max
Attacking	C_{attk}	$f(Range, DamagePoint, P, num)$	(damage point × range) × hit probability × number	Sum
Electronic attack	C_{elec}	$f(Range, DamagePoint, P, num)$	(damage point × range) × hit probability × number	Sum
Defense	C_{defs}	$f(DamagePoint, EscapeTime^{-1})$	damage point/escape time	Sum
Maneuverability	C_{Mnur}	$f(Velocity, Range)$	speed × flight range under that speed	Avg
Concealment	C_{cncl}	$f(D_{visual}^{-1}, D_{Infrared}^{-1}, S^{-1})$	(visible discovery distance × infrared discovery distance × radar reflective area)$^{-1}$	Min

The first column of Table 3 presents the capability domain. In particular, perception capability is the availability of access to target information and intelligence. Attacking capability is the comprehensive damage level of weapon systems. Electronic attack capability is the degree to which an object can utilize the electromagnetic spectrum for the attack. Defense capability represents how well a set of objects can defend against firepower attacks. Maneuverability is the degree of the object's flexibility considering speed and mobility. Concealment capability measures the extent to which an object may be detected or tracked by others [41]. The third and fourth columns of Table 3 depict the supporting features used to calculate the capability. In turn, the capabilities of the whole MPAMS are constructed by integrating every platform through the mechanisms in column 5. Then the comparative strengths and weaknesses of both sides are derived, including the labelled code of each capability: W is for weak, B is for balanced, and S is for strong.

Step 3: Construct task features

The task feature module is responsible for extracting task features from a set of combat platform actions in the replay data in a bottom-up manner. In particular, the strike action takes the dominant role among various actions. This action constitutes the attacking platform, attacked target, weapon used by the attacker, fire timestamp, and attacked target location. As shown in Figure 5, task features cluster collaborative actions according to their similarities in time and location. For instance, two actions *Action1* and *Action2*, which satisfy the constraints given by Equation (3), will be integrated into the same task, which is highlighted by a rectangular box. Task features comprise the time span, assigned platforms, and target platforms. Then, task sequences are delivered in an encoded form, considering task types [43] and collaboration relationships.

$$\begin{cases} ||fire\_timestamp_1 - fire\_timestamp_2|| \leq T_\Delta \\ ||location_1 - location_2|| \leq D_\Delta \end{cases} \tag{3}$$

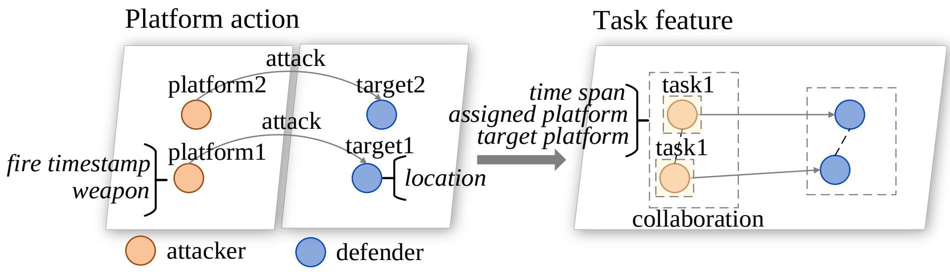

Figure 5. Task feature construction process based on platform actions.

Step 4: Construct firepower effectiveness features

The firepower effectiveness measures whether an operation achieves the expected objective at the end of a wargame scenario. Effectiveness is reflected by the integrity status of enemy's targets, including undamaged, partially failed, or damaged. In addition, the platform and mount consumption of a MPAMS is another way to measure effectiveness.

Step 5: Knowledge mining for MPAMS organization

The organization preference knowledge of MPAMS is acquired by applying the FCIM to confrontation itemsets. Particularly, the theoretical problem of organization preference knowledge mining can be interpreted as a correlation extraction assignment of different organizational features that often appear together under different circumstances, which matches a FCIM problem, as explained in Section 2.2. In particular, each wargame simulation is set as a transaction, and all organizational features are coded as items within the transaction. Then, the MPAMS organization preference knowledge, namely the frequent (typical) closed (significant) itemsets composed of highly correlated organizational features, can be derived from massive simulation transaction records, which are described in detail from the algorithmic perspective in Section 3.

3. Proposed MPAMS Organizational Knowledge Mining Algorithm-*CrossFCI*

To realize MPAMS organizational knowledge mining, the FCIM algorithm should be designed to adapt to the MPAMS dataset's special feature of dense itemset distribution, which exacerbates the analysis difficulty. Therefore, the algorithm should emphasize efficiency optimization to solve that problem.

The latest intersection-based FCIM algorithms such as *Ciclad* [37] store itemsets in a vertical structure and incrementally traverses each transaction to update the structure with better analysis performance. Nonetheless, *Ciclad* uses the anti-monotonicity principle of CI to prune the uncritical candidate itemnodes in the structure, while overlooking the redundant itemnodes from the FI perspective, which leaves room for further analysis scale reduction. Thus, this paper proposes *CrossFCI*, an innovative algorithm that includes a cross-linked itemset storage structure instead of the traditional vertical one, which could traverse each item for more efficiently structure updating. Moreover, benefiting from three proposed pruning strategies, the proposed method could more effectively prune redundant itemnodes in storage space based on both the minimum support constraint of FI and the anti-monotonicity principle of CI. Therefore, it could further reduce mining time and better fit the analysis requirements.

3.1. Proposed Itemset Storage Structure

CrossFCI is made up of two structures for itemset storage:

(1) Item-mediated linked list

The item-mediated linked list G converts horizontally stored transactions in the database into vertically stored items, which are presented in Figure 6a by a series of graphnodes. Each graphnode g contains a primary itemnode as well as a collection of transactions in which it appears. Given transactions 1–10 in Table 2, itemnodes {*a*,*b*,*c*,*d*,*e*,*f*,*g*,*h*} are sequentially encoded to graphnodes. For example, itemnode d is added to G attached by the transaction {*1*,*5*,*6*,*7*,*9*}. Afterward, the graphnodes will be traversed sequentially to update the FCIs in the cross-linked structure. Notably, transactions are only reserved once in G and will not appear in the cross-linked structure, thus reducing the memory usage for the subsequent FCI mining process.

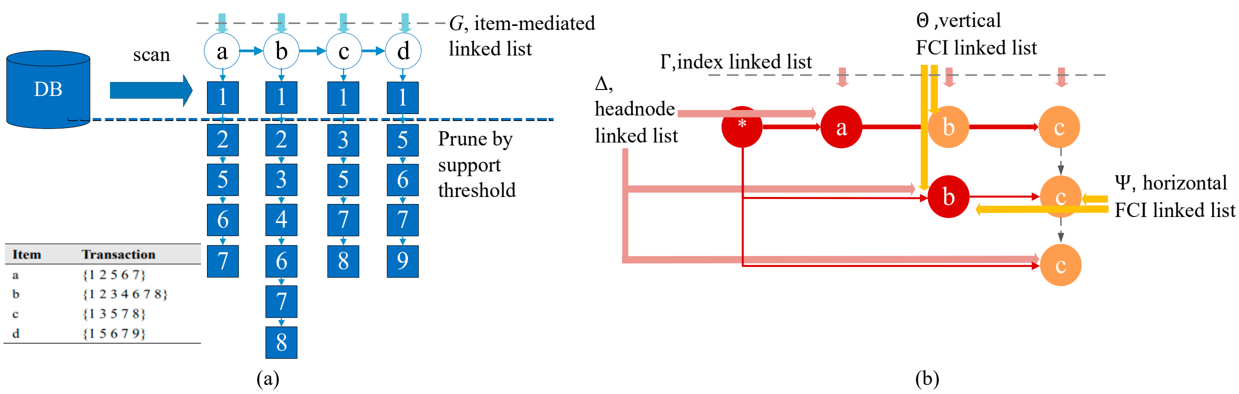

Figure 6. Structures for itemset storage and updating. (**a**) Structure of the item-mediated linked list. (**b**) Structure of the cross-linked structure.

(2) Cross-linked structure

Figure 6b visualizes the cross-linked structure C which stores the FCI in a tree-like structure, given by

$$C = (H, V) \tag{4}$$

where H denotes a horizontal linked list, V stands for a vertical linked list. Particularly, H reserves the entire structure of each FCI as a chain, which is composed of

$$\begin{gathered} H = (\Delta, \Psi) \\ \forall \delta \in \Delta, \delta.\iota = itemnode;\ \forall \varphi \in \Psi, \varphi.\iota = itemnode;\ \exists \Psi \in \delta \text{ s.t. } \delta.\iota = \Psi(0).\iota \equiv \varphi.\iota \end{gathered} \tag{5}$$

where Δ means a headnode linked list to locate the starting position of certain FCI branches by itemnode name. Each headnode δ in Δ is attached to several Ψ, namely FCI branches, which have the same primary itemnode name. Moreover, Ψ indicates a horizontal FCI linked list, to direct each frequent closed itemnode φ in a FCI branch.

Specifically, the tail frequent closed itemnode of each FCI chain is highlighted in orange and directed by a FCI flag pointer φ, containing tags for the support value and transaction collections of this FCI. Moreover, this FCI chain's intermediate itemnodes can also be marked as the tail frequent closed itemnode of other FCIs. E.g., the frequent closed itemnode {*a*} and {*b*} in Figure 7 represent for FCI {*a*} and {*a-b*} with an attached transactions of {*1, 2, 5, 6, 7*} and {*1, 2, 6, 7*}, respectively.

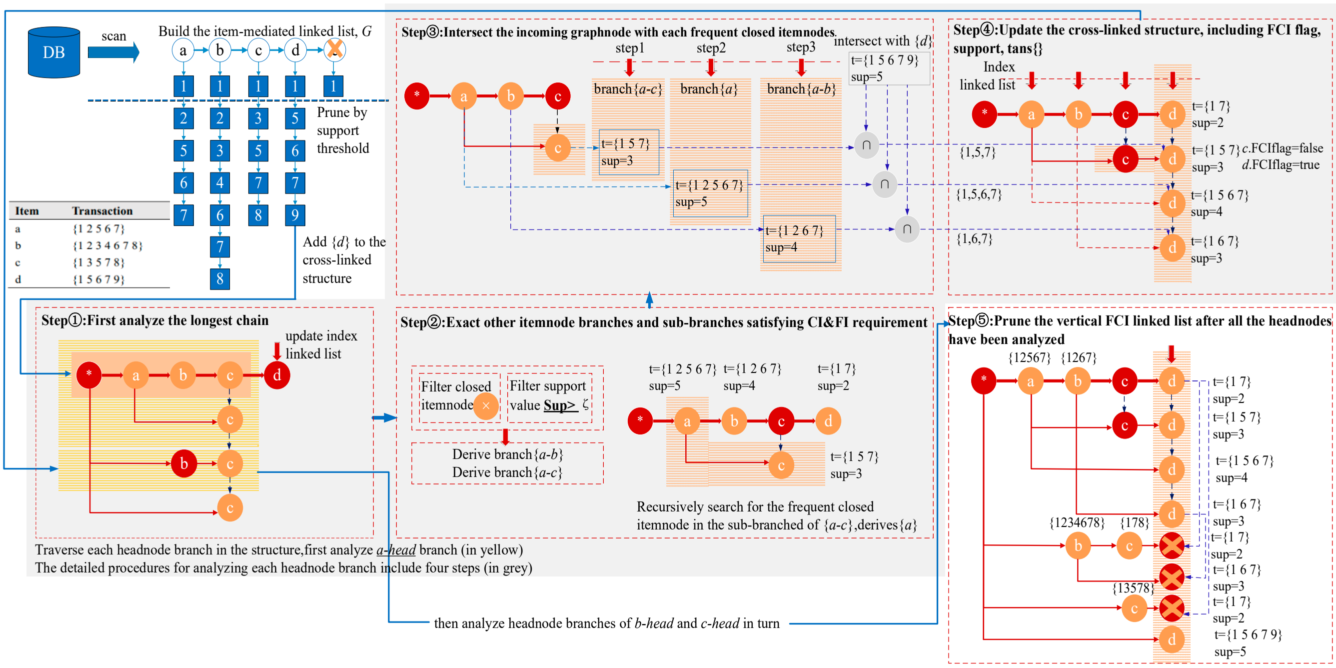

Figure 7. Detailed procedures for the cross-linked structure updating mechanism. (sup: support value, t: transaction set).

The vertical linked list V locates and links identical itemnodes in various horizontal linked lists, which constitutes

$$\begin{gathered} V = (\Gamma, \Theta) \\ \forall \gamma \in \Gamma, \gamma.\iota = itemnode;\ \forall \theta \in \Theta, \theta.\iota = itemnode;\ \exists \Theta \in \gamma \text{ s.t. } \gamma.\iota = \Theta(0).\iota \equiv \theta.\iota \end{gathered} \quad (6)$$

where Γ implies an index linked list to target the specific itemnode by name. Each node $\gamma \in \Gamma$ is appended by an item-based vertical FCI linked list Θ, which connects the same itemnode belonging to different horizontally stored FCIs, enabling further superset itemset comparison of that itemnode.

3.2. Knowledge Mining Procedure

The cross-linked structure updating procedures in the MPAMS organizational knowledge mining process are depicted in Figure 7. First, the database is scanned to construct an item-mediated linked list G. Subsequently, the cross-linked structure C grows along with the iteration over graphnodes of G. When a new graphnode is added, the cross-linked structure is updated in five steps:

Step 1: Analyze all the attached branches Ψ of each headnode g. In the case of *a-head*, branches {*a-b-c*} and {*a-c*} should be covered, and the longest chain {*a-b-c*} will be analyzed first.

Step 2: Recursively assess the sub-branches under each branch that satisfy the CI definition and minimal support requirements of FI, such as subbranch {*a-b*} under branch {*a-b-c*}, and subbranch {*a*} under branch {*a-c*}.

Step 3: Intersect the transactions between the incoming graphnode and existing frequent closed itemnodes. E.g., intersect the transactions of {*d*} with branch {*a-c*}, {*a*}, and {*a-b*} in turn, attaining new frequent closed itemnodes to be updated.

Step 4: Update the cross-linked structure, including the FCI flag pointer, the support value tag, and the corresponding transactions.

Step 5: Evaluate the subsequent branches of *b-head*, *c-head*, and *d-head* in Δ after analyzing all the branches of *a-head*. Once all the headnodes of the cross-linked structure have been assessed horizontally, the FCI nodes are rejudged and pruned according to the inverse monotonicity principle of the vertical linked list Θ, which further removes redundant candidate itemnodes from the cross-linked structure.

Ultimately, the updated cross-linked structure is attained, which includes each frequent closed itemnode along with the corresponding transactions and a support value. This structure enables extracting and pruning CIs and FIs simultaneously.

3.3. Proposed Pruning Strategies

Additionally, the structure updating process covers three pruning strategies, which facilitate fast intersection computation among FCIs while maintaining low storage space.

Theorem 1. *For every φ in Ψ, the support value φ.*sup *should be no less than ς.*

Proof of Theorem 1. $\varphi \in \Psi$ iff $\varphi.\text{sup} \geq \varsigma$. If $\{\exists\varphi.\text{sup} < \varsigma | \varphi \in \Psi\}$ then the updated $\varphi^+ \triangleq \tau(\varphi) \cap \tau(\iota_{new})$ by intersecting φ with the incoming node ι_{new}, with $\varphi^+.\text{sup} = \min\{\varphi.\text{sup}, \iota_{new}.\text{sup}\} < \varsigma$, conflicting with the definition of Ψ.□

Pruning rule 1. *Remove itemnode φ and ι_{new} from Ψ if neither φ.*sup *nor ι_{new}.*sup *is less than ς.*

Theorem 2. Ψ *only contains φ whose transactions is different from those of previous itemnodes $\varphi.pre$.*

Proof of Theorem 2. $\varphi \in \Psi$ iff $\kappa(\varphi) = \varphi$. If $\{\exists\tau(\varphi.pre) = \tau(\varphi) | \varphi, \varphi.pre \in \Psi\}$, then the updated $\varphi^+ \triangleq \{\tau(\varphi.pre) \cap \tau(\iota_{new})\} \cup \{\tau(\varphi) \cap \tau(\iota_{new})\} = \varphi$ or ι_{new}. Hence, the information of $\varphi.pre$ is redundant.□

Pruning rule 2. *Remove $\varphi.pre$ from Ψ for any $\tau(\varphi.pre) = \tau(\varphi)$.*

Theorem 3. *If there is a Θ for ι_{new} in Γ, then every itemnode in Θ possesses a different support value.*

Proof of Theorem 3. $\theta \in \Theta$ iff $\kappa(\theta) = \theta$. *If* $\exists\{\Theta \in \gamma$, s.t. $\gamma.\iota = \iota_{new} | \gamma \in \Gamma\}$, *then for any θ_1 and its superset θ_2, if* $\exists\{(\theta_1.\text{sup} = \theta_2.\text{sup})\ \&\&\ (\tau(\theta_1) \subseteq \tau(\theta_2)) \mid \theta_1, \theta_2 \in \Theta\}$, *which implies* $\kappa(\theta_1) = \kappa(\theta_2) = \theta_2 \neq \theta_1$, *conflicting with the definition of* Θ. □

Pruning rule 3. *Remove θ from Θ if any superset of θ possesses the same support value.*

3.4. Algorithm Implementation

The algorithm implementation consists of four functions. Function 1 is the initial step for transferring the original database to the item-mediated linked list. Function 2 updates the cross-linked structure from a macro perspective based on the derived graphnode of function 1. Function 3 is the recursively invoked micro updating mechanism for a specific headnode branch, which is embedded in function 2. Function 4 integrates the analyzed result into the existing cross-linked structure to be updated after function 2 is completed.

Function 1: Construct the item-mediated linked list to store each item's transactions.

The transaction database is iterated to create a graphnode for each item in the item-mediated linked list, each attached with a set of matching transactions represented by transnodes (lines 2–6). The new graphnode will be connected to the old graphnode if the one with the same name already exists (line 7). Finally, graphnodes that violate the support threshold are deleted (line 8). For an item-mediated linked list including {*a*,*b*,*c*,*d*} in Figure 7, four items are stored as independent circular graphnodes, and the attached square nodes denote the transaction index.

Function Conwg()

Input: original transaction database
Output: item-mediated linked list G with each graphnode attached with a set of transnodes
1: **While**(!transaction database)
2: { **foreach** itemnode ι in a certain transaction
3:     **if** FindSameItem$(\iota, g) = \varnothing$ **then**
4:       create graphnode g_{new} s.t. $g_{new}.\iota = \iota$
5:       Transnode* *translist* = SearchMatch(database, g_{new})//read matched transactions in sequence
6:       $g_{new}.\tau \leftarrow translist$, $G(tail)$.next $\leftarrow g_{new}$
7:     **else** link ι to the matched $g \in G$ }
8: Delete $g \in G$ s.t. g.sup $< \varsigma$
End function

Function 2: Update the cross-linked structure according to each incoming graphnode.

Each incoming graphnode of function 1 is merged with the existing cross-linked structure that stores all the FCIs to formulate updated FCIs. In particular, existing subordinate branches Ψ of all the headnodes $\delta \in \Delta$ in the structure are iterated. For each headnode, the longest chain within the branches is first analyzed to facilitate the following intersections. Subsequently, the *MiningItem* function is called to figure out novel FCI candidates by integrating the incoming graphnode with the other frequent closed itemnodes in the branches (lines 1–6). Moreover, a new headnode branch is created for the graphnode after analyzing all the headnodes (line 7). After that, if there is no itemnode $\gamma \in \Gamma$ matches this graphnode, it will be directly attached to the existing linked list (line 9). Alternatively, if there is no superset with the same support in the vertical FCI linked list Θ of this node, i.e., the pruning strategy 3 is satisfied, the graphnode will be linked to the structure tail as a new FCI node (lines 11–12).

As shown in Figure 7, when graphnode {*d*} is added to the structure, all the headnodes ranging from {*a*} to {*c*} are analyzed. The headnode branches are successively evaluated by function 3 in the following orders: {*a-b-c*, *a-c*}, {*b-c*}, and {*c*}. Then branches {*b-c-d*}, {*b-d*}, and {*c-d*}, whose supersets are {*a-b-c-d*}, {*a-b-d*}, and {*a-b-c-d*}, respectively, are dropped out via the pruning strategy 3.

Function Mining()

Input: each graphnode g in the item-mediated linked list G
Output: updated cross-linked structure
1: Initial()
2: **foreach** $g \in G$
3: { **foreach** $\delta \in \Delta$
4:   { mining the longest chain of the branches firstly
5:     mining the itemnode $\varphi \in \Psi$ s.t. $\Psi \in \delta$ by **MiningItem(**$\varphi, \varphi.next, g, 0$**)**
6:     $\delta \leftarrow \delta.next$ }
7:   **Link**(*null*,Γ, Δ, *null*, *g*, *0*, *3*)//create independent headnode branch for g
8:   //search index itemnode $\gamma \in \Gamma$ which matches g
9:   **if** $\forall \gamma \in \Gamma$ s.t. $\gamma.\iota \neq g$ **then Link**(*null*, Γ, Δ, *null*, *g*, *0*, 2)//create a new headnode branch for g
10:   **else**//conduct pruning strategy 3
11:     **if** $\forall \theta \in \gamma.\Theta$ s.t. $(\theta.\tau \supseteq g.\tau)$&&$(\theta.$sup $\neq g.$sup$)$ **then Link**(*null*,Γ, Δ, γ, Θ,*g*, *0*,*4*)
12: $g \leftarrow g.next$ }
End function

Function 3: Generate new FCI candidates for each headnode branch.

The *MiningItem* function is the atomic procedure of FCI analysis. The FCI chain's tail node is first processed. The graphnode is linked to the branch tail if no superset has the same support value (lines 1–4). After that, the incoming graphnode and each intermediate frequent closed itemnode in the chain is intersected in response to the pruning strategy 2. The frequent closed itemnode flag is transmitted to the graphnode if the transaction intersection is the same (lines 5–8). Otherwise, sub-branches under the intermediate node are assessed recursively which satisfied the pruning strategy 1. The recursion would not end until there are no more sub-branches (lines 9–11).

For branches {*a-b-c*, *a-c*} under *a-head* in Figure 7, {*a-b-c*} is first evaluated with {*d*} appended to the end of the chain. Then the intermediate frequent closed itemnodes {*a*} and {*b*} in {*a-b-c*} are assessed in order. The analytical order for {*a*} is {*a-c*} + {*d*} and {*a*} + {*d*}. In terms of {*b*}, the pair of {a-*b*} + {*d*} is examined.

Function MiningItem(φ_0,φ_1, *g*, *k***)**

Input: φ_0 and φ_1: adjacent itemnodes in Ψ, *g*: graphnode, *k*: the sub-branch index of Ψ
Output: updated FCI branches in the FCI linked list
1: **if** (φ_1 = null) **then**//imply the tail itemnode of the certain FCI chain
2:     **if** $\forall\gamma \in \Gamma$ s.t. $\gamma.\iota \neq g$ **then Link(**Ψ, Γ, Δ, null, g, k, 2**)**//target$\gamma.\iota$ that matches g
3:     **else**//conduct pruning strategy 3
4:         **if** $\forall\theta \in \gamma.\Theta$ s.t. $(\theta.\tau \supseteq g.\tau)$&&$(\theta.\text{sup} \neq g.\text{sup})$ **then Link(**$\Psi, \Gamma, \Delta, \gamma, \Theta$, g, k, 4**)**
5: **else**//imply intermediate itemnode of the certain FCI chain
6:     **foreach** $\varphi_1 \in \Psi$ s.t. $\varphi_1.clsflag = true$&& φ_1.next[0] $\neq null$//conduct pruning strategy 2
7:     { **if** $(\tau(\varphi_1) \cap \tau(g)) = \tau(\varphi_1)$ **then**
8:             $\varphi_1.clsflag \leftarrow$ false , $g.clsflag \leftarrow$ true
9:         **else**//conduct pruning strategy 1
10:             **foreach** sub-branch *sub* of φ_1.next[] s.t. (φ_1.next[*sub.index*].sup $> \varsigma$)
11:                 **MiningItem**(φ_1, *sub*, *g*, *sub.index*)
End function

Function 4: Update the cross-linked structure.

The Link function presents the detailed updating mechanism of the cross-linked structure, which is divided into four cases according to the relationship between the incoming graphnode and the current structure: Initialization of the entire structure by creating a new headnode δ and vertical node γ(case 1), connecting the incoming graphnode to the end of Γ (case 2), creating a new headnode branch for this graphnode in Δ (case 3), and updating the intermediate FCI nodes in the horizontal Ψ and vertical Θ of the structure (case 4).

Function Link($\Psi, \Gamma, \Delta, \Theta$, *g*, *k*, *mode***)**

Input: Ψ: horizontal FCI linked list, Γ: vertical index linked list, Δ: headnode linked list, Θ:vertical FCI linked list, *g*: graphnode, *k*: the sub-branch index of Ψ, *mode*: the structure updating mechanism mode
Output: updated $\Psi, \Gamma, \Delta, \Theta$ attached with new itemnode
Create new φ, $\varphi.\iota \leftarrow g$, $\varphi.clsflag \leftarrow true$
Switch(*mode*)
Case 1://neither headnode nor indexnode exists, applicable to the initial itemnode of the structure
Create new $\gamma, \gamma.\iota \leftarrow g$, create $\delta, \delta.\iota \leftarrow g, \Delta(tail)$.next $\leftarrow \delta, \Gamma(tail)$.next $\leftarrow \gamma$
Case 2://no indexnode exists, applicable to updating the longest branch of δ
Create new $\gamma, \gamma.\iota \leftarrow g, \Gamma(tail)$.next $\leftarrow \gamma, \Psi(tail)$.next$[k] \leftarrow \varphi$
Case 3://no headnode exists, construct an independent headnode branch for g
Create new $\delta, \delta.\iota \leftarrow g, \Delta(tail)$.next $\leftarrow \delta, \Theta(tail)$.next $\leftarrow \varphi$
Case 4://both headnode and indexnode exist, applicable to analysis intermediate branches of Ψ
$\Psi(tail)$.next$[k] \leftarrow \varphi, \Theta(tail)$.next $\leftarrow \varphi$
End function

---

4. Results

This section discusses experiments implemented on both public datasets and replay wargame simulation datasets to verify the efficiency of *CrossFCI* in the MPAMS organizational knowledge mining task. The objective lies in:

- How effectively could *CrossFCI* mine FCIs from the public quantitative datasets?
- How well does *CrossFCI*'s performance benefit from the proposed three pruning strategies?
- What valuable organization preference knowledge in MPAMS organization has been discovered by *CrossFCI*?
- How efficiently could *CrossFCI* adapt to huge MPAMS datasets of different sizes?

4.1. Introduction to Datasets and Algorithms

4.1.1. Public Datasets

As indicated in Table 4, four public datasets with different transaction sizes and densities were used. *Mushroom* is a well-known dense dataset that depicts mushroom information with identically sized transactions. *Retail* is a popular sparse dataset of market basket information. *Synth* and *Synth2* are constructed via the SPMF platform, including synthetic transaction datasets on a medium and small scale.

Table 4. Description of public datasets.

Dataset	Transaction	Avg (t)	Item	Density
Mushroom [1]	8124	23	119	19.3%
Retail [1]	88,163	10.4	16,470	0.06%
Synth [2]	100,000	25.4	10,000	0.25%
Synth2 [2]	10,000	25.5	1000	2.5%

[1]: http://fimi.ua.ac.be/data/, [2]: http://www.philippe-fournier-viger.com/spmf/ (accessed on 19 November 2022).

4.1.2. Compared Algorithms and Experiment Settings

CrossFCI was compared to *Ciclad* [37], *CloStream* [38], and *Moment* [39], which are all intersection-based itemset extraction algorithms. *Ciclad* is one of the latest FCIMs with two folds, which can access the itemset structure quickly. The structure grows through the item-wise intersection and drops out those uncritical items, thereby accelerating the analysis efficiency. *CloStream* boosts efficiency by introducing landmarks. In particular, an inverted list is proposed to create intersections and filter itemsets. *Moment* uses a CE tree to adapt to the itemset enumeration. The itemsets are refreshed by interacting the sibling nodes with some infrequent itemnodes and promising closed itemnodes, deriving the support value of each itemset.

All the algorithms were programmed in C and set in identical conditions. The experiments were conducted on a personal computer with Windows 10, 16 GB of RAM, and Intel i7-8700 CPU. The CPU time consumption throughout the mining process was measured, with a maximum limit of 10,000 s.

4.2. Results on Public Datasets

This subsection evaluates *CrossFCI*'s performance using three tests: a performance test considering mining time, a scalability test regarding adaption to varied-size datasets, and a pruning strategy efficiency test to confirm the function of three pruning strategies.

4.2.1. Efficiency Comparison

Figure 8a,b depict the mining results for dense datasets of *Mushroom* and *Synth2*. It is observed that the mining time of each method rises with the decreased support threshold. Figure 8a shows that *CrossFCI* performs best on *Mushroom*, which takes less than 10 s when the support threshold is 0.01, outperforming *Ciclad* and *Moment* by an order of magnitude and *CloStream* by two orders of magnitude. As for *Synth2*, Figure 8b describes that *CrossFCI*'s time consumption ranges from 8 to 12 s, which also yields numerical superiority to *Ciclad* and *CloStream*. Moreover, *Ciclad* shows a slight time advantage over *CloStream* of around 10 s. In addition, the analysis time of *Moment* increments sharply as the support threshold declines. The peak time consumption is more than 150 s when the support threshold is 0.02. The efficiency advantage of *CrossFCI* for dense datasets stems from the well-designed itemset storage structure, which reduces the workload of processing redundant or unessential itemsets.

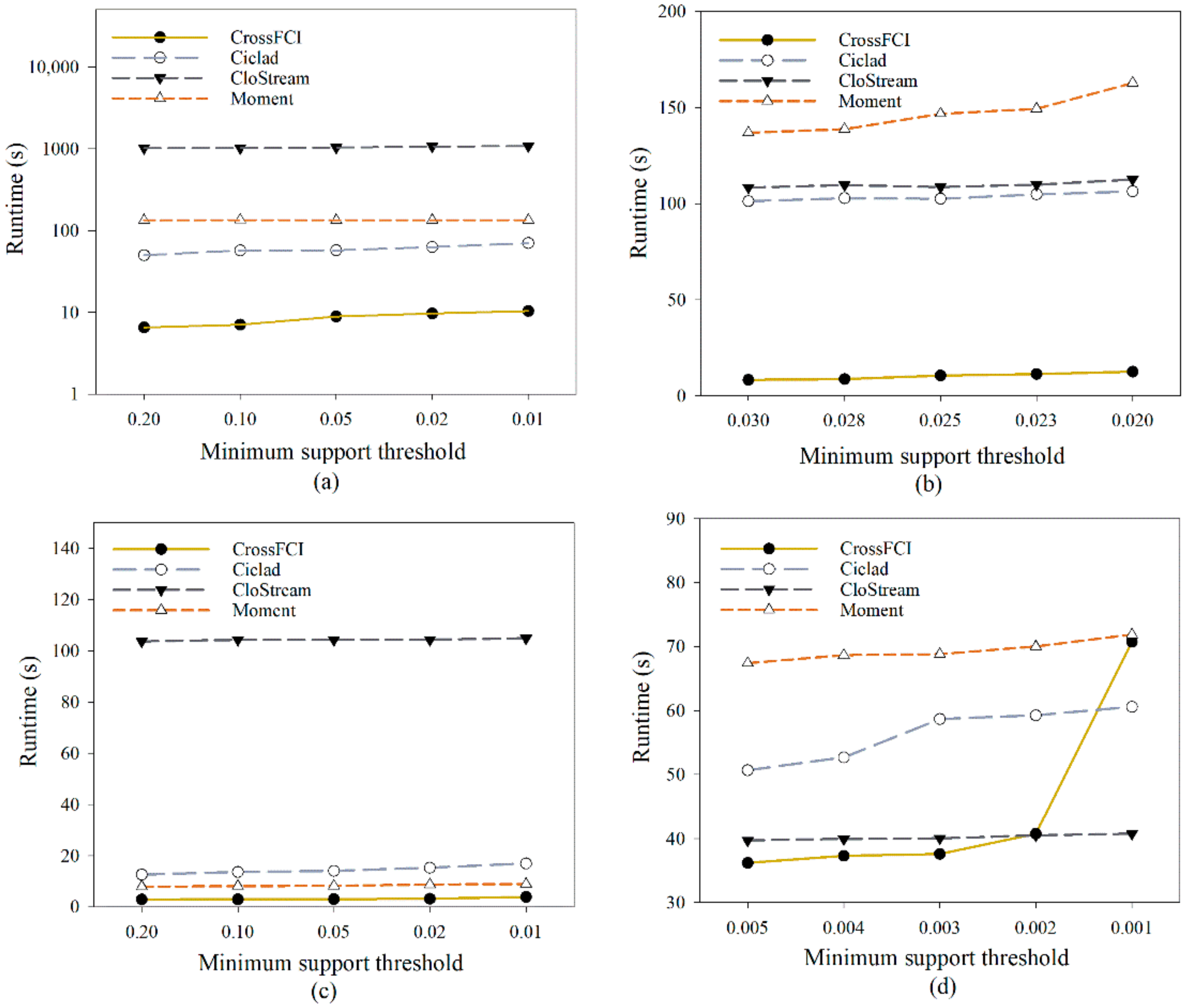

Figure 8. Efficiency comparison of different FCIM algorithms. (**a**) Results on *Mushroom* dataset. (**b**) Results on *Synth2* dataset. (**c**) Results on *Retail* dataset. (**d**) Results on *Synth* dataset.

For sparse datasets, which comprise *Retail* and *Synth*, the FCIs can be effectively mined under relatively sparse conditions. As seen from Figure 8c, for *Retail*, *CrossFCI*'s time consumption fluctuates gently at 3 s, and the growth rate is smooth when the support threshold decreases, which is slightly less than *Moment* and *Ciclad*. *CloStream* lags with more than 100 s on *Retail* dataset. However, for the very sparse dataset *Synth*, the mining time in Figure 8d indicates *CrossFCI* efficiency dips under a low support threshold. For instance, there is a clear rising trend in the threshold span [0.001, 0.002], which performs worse than *Ciclad* and *CloStream*, yet holds a small gap to *Moment*. The underlying reason is *CrossFCI*'s item-based structure for FCI updating. The sparse dataset possesses numerous item categories, while the item density of each transaction is small. Thus, each transaction cannot provide adequate valuable item information for structure updating, resulting in slow structure iteration speed and high search cost when an itemnode arrives. As a result, *CrossFCI* did not achieve the expected efficiency on very sparse itemsets.

4.2.2. Scalability Assessment

In addition, *Mushroom* dataset of different sizes, including 10 k, 100 k, 200 k, and 500 k, were analyzed to verify the scalability of *CrossFCI*. As illustrated in Figure 9a, each algorithm completes the assignment in 1 s under the premise of a small dataset. When the data scale grows, *CrossFCI* can still acquire results in less than 10 s, whereas the time cost of other algorithms apparently ascends. For instance, the time cost of *CloStream* gradually approaches the highest point of 1000 s on the dataset of 500 k (see Figure 9a).

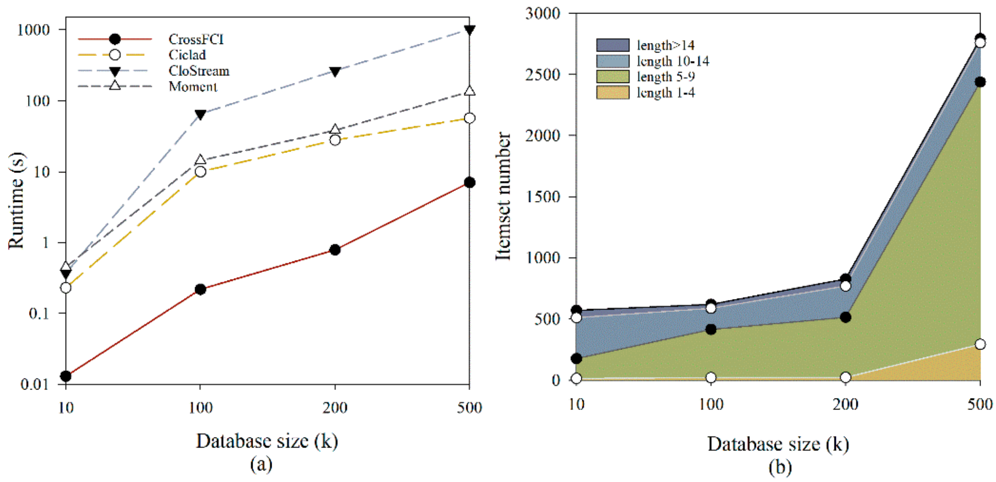

Figure 9. Scalability comparison of different FCIM algorithms. (**a**) Runtime comparison of different FCIM algorithms. (**b**) Itemset number of different FCI lengths.

Figure 9b reveals the reason for the varied time cost from the FCI length perspective, which gives the itemset length distribution of the results on different-sized datasets. It is noteworthy that the percentage of complex FCIs, including the interval 5–9 in green and 10–14 in dark blue, grows rapidly for large datasets of 200 k and 500 k, implying the analysis assignment tends to become harder and leads to high computation time. While *CrossFCI* can obtain the results quickly, benefiting from less time spent on scanning the itemset storage structure, which organizes FCIs in a compressed form.

4.2.3. Pruning Strategy Efficiency Verification

To verify the effectiveness of different pruning strategies in *CrossFCI*, Figure 10 provides the comparison of both efficiency and FCI number for five algorithm configurations: the one with completed pruning strategies, the one without any pruning strategy, and the ones those only adopt the pruning strategy 1, 2, or 3.

The results in Figure 10a denote that pruning strategy 1 can reduce the analysis time from the initial phase of structure construction by the support threshold constraint. However, the itemset comparison in Figure 10b indicates that strategy 1′s effect is limited compared to other strategies. Pruning strategy 2 focuses on the itemnodes in the horizontal linked list Ψ, as shown in Figure 10b, effectively reducing the candidate itemnodes to be analyzed, thus cutting the intersection analysis scope. For pruning strategy 3, the redundant itemnodes of the vertical linked list Θ are pruned by the superset support judgment, which further reduces the analysis scale and yields the shortest running time among the three strategies (see Figure 10a). The comparison of the pruning strategies 2 and 3 reveals that the former toward Ψ is more adept at dropping out storage of itemnodes in the mining process, while the latter toward Θ contributes more to the mining efficiency by shrinking the redundant FCI results.

In conclusion, the performance test illustrates that *CrossFCI* outperforms all the other examined algorithms in terms of time consumption. The scalability test demonstrates that *CrossFCI* is applicable to both dense and relatively sparse datasets even under a harsh support threshold. The pruning strategy efficiency test verifies the effectiveness of the three pruning strategies in the algorithm.

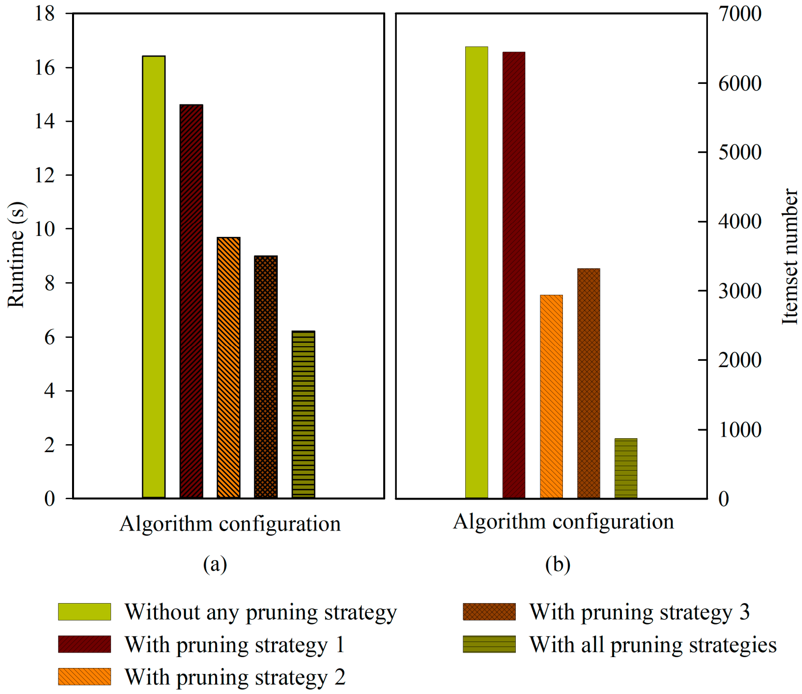

Figure 10. Pruning strategy comparison of different algorithm configurations. (**a**) Runtime comparison. (**b**) Itemset number comparison.

4.3. Results on Wargame Simulation Datasets

This subsection first describes the wargame case to clarify the force composition of both sides, and then states how the experimental data is processed into the FCIs, followed by the knowledge analysis from the perspective of tasks, platforms, and mounts. Finally, the performance of the proposed algorithm is assessed on MPAMS datasets of different sizes.

4.3.1. Case Description

An air-to-sea assault operation was used for the feasibility verification of *CrossFCI*. The attacking side constituted of a flight formation consisting of 17 aircraft deployed in two airports, which are highlighted in blue in Figure 11. The flight formation intended to break through the air defense force and launched a surprise assault against the defender's ship. The defending side included a large destroyer that sailed in the nearby sea, which is highlighted in red in Figure 11, and a formation of six fighters that performed air patrols to prevent the attacking side from striking the ship.

There were two phases in this operation: air combat and sea assault. The task under test included strike, feint, escort, electronic jamming, intelligence, surveillance, and reconnaissance (ISR). The aircraft platform contained fighters, bombers, fighter-bombers, UAVs, electronic warfare aircraft, and air early warning (AEW). The mount comprised ultra-long-range air-to-sea missiles (ASM, whose range is approximately 300 nm), long-range air-to-sea missiles (approximately 150 nm), close-range air-to-sea missiles (approximately 20 nm), air-to-air missiles (AAM), bombs, and electronic warfare weapons. Notably, the total amount of aircraft in the MPAMS architecture was fixed, while the assigned task type, aircraft type, and mount type of the MPAMS could be flexibly changed and paired.

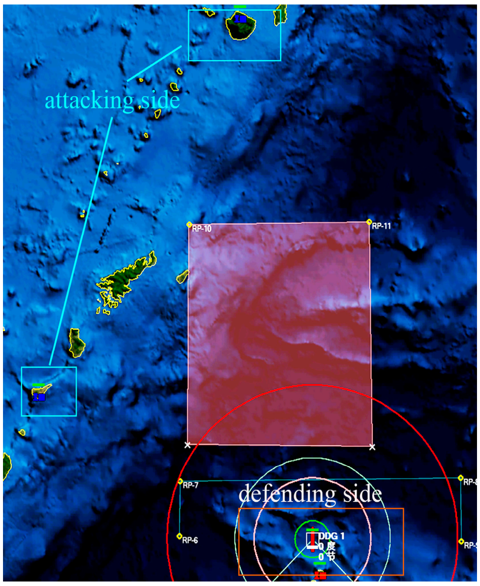

Figure 11. The air-to-sea assault scenario depiction of the attacking side and the defending side.

4.3.2. Data Processing

The experimental data were collected using a wargame software named Origin, developed by Shareetech. The simulations were conducted on 1100 scenarios with 10 MPAMS configurations with different tasks, aircraft platforms, and mount combinations, as explained in detail in Appendix A. In the attacker's view, there were 573 winning samples and 527 losing samples. The criterion of winning was whether the defender's ship was damaged or failed at the end of the sea assault phase, otherwise the operation was lose with the ship remained undamaged or normal. The result was automatically estimated by the game engine, indicating the data's credibility. Then, the feature data were processed as explained in Section 2.3. Subsequently, the MPAMS organization preference knowledge in winning and losing scenarios was analyzed separately by *CrossFCI* under a support threshold of 0.06. The value of the support threshold is determined by the scale of valuable FCIs. Finally, the discovered FCIs, which contained the itemnode of contextual capability code, organization modes, and strike effectiveness at the same time, were filtered to derive meaningful organizational knowledge.

4.3.3. Results Analysis

The discovered knowledge was demonstrated by a set of FCIs, where each FCI represented a mode of itemnode correlation; for instance, itemnodes a, b, and c were correlated in a FCI $\{a, b, c\}$. The itemnode included the MPAMS organization features, as presented in Section 2.3. Then, each FCI, namely each MPAMS organizational knowledge, was further processed into the format of $\{A_1, A_2 \Rightarrow B\}$, as shown in rows in Table 5.

Table 5. Discovered organization preference knowledge of MPAMS.

Contextual Capability(A_1)	Organization Element (A_2)	Index	Firepower Effectiveness (B)	Count
S_{prcp},W_{attk},S_{elec}, S_{defs},S_{mnur},B_{cnc} [1]	ISR(AEW)→Feint(UAV)→Elec(**F(AR)**)→Escort(F[A] + FB[A]) →Strike(B[UL] +F[L] + FB[L]) [2]	1	$S_{F}^{aircraft}, S_{FB}^{aircraft}, L_{ASM}^{mount}, F_{ship}^{enemy}$ [3]	12
B_{prcp},B_{attk},S_{elec}, B_{defs},S_{mnur},W_{cncl}	ISR(AEW)→Escort(F[A] + FB[A]) →Strike(**UAV** + B[UL] + FB[L] + F(L))	2	$S_{FB}^{aircraft}, M_{ASM}^{mount}, L_{AAM}^{mount}, D_{ship}^{enemy}$	13
		3	$L_{F}^{aircraft}, L_{FB}^{aircraft}, M_{ASM}^{mount}, F_{ship}^{enemy}$	10
		4	$L_{B}^{aircraft}, N_{ship}^{enemy}$	1
B_{prcp},B_{attk},S_{elec}, B_{defs},B_{mnur},W_{cncl}	ISR(AEW)→Feint(UAV)→Strike(B[UL] + FB[L] + **F[L]**)	5	$S_{F}^{aircraft}, L_{FB}^{aircraft}, L_{ASM}^{mount}, D_{ship}^{enemy}$	11
		6	$S_{F}^{aircraft}, L_{FB}^{aircraft}, L_{ASM}^{mount}, F_{ship}^{enemy}$	8
B_{prcp},S_{attk},S_{elec}, W_{defs},W_{mnur},S_{cncl}	ISR(AEW)→Feint(UAV)→Escort(F[A])→Strike(**B[UL]**)	7	$L_{B}^{aircraft}, L_{ASM}^{mount}, M_{AAM}^{mount}, N_{ship}^{enemy}$	18
B_{prcp},S_{attk},S_{elec}, W_{defs},W_{mnur},W_{cncl}	ISR(AEW)→Feint(UAV)→ Escort(F[A])→Strike(**B[Bomb]**)	8	$L_{B}^{aircraft}, N_{ship}^{enemy}$	6
B_{prcp},B_{attk},S_{elec}, B_{defs},B_{mnur},S_{cncl}	ISR(AEW)→Feint(UAV)→Escort(F[A])→Strike(B[UL] + **F[C]**)	9	$L_{F}^{aircraft}, M_{ASM}^{mount}, D_{ship}^{enemy}$	4
		10	$L_{F}^{aircraft}, L_{ASM}^{mount}, F_{ship}^{enemy}$	10
		11	$M_{ASM}^{mount}, M_{AAM}^{mount}, N_{ship}^{enemy}$	10
B_{prcp},B_{attk},S_{elec}, B_{defs},W_{mnur},S_{cncl}	ISR(AEW)→Feint(UAV)→Escort(F[A] + **FB[A]**) →Strike(B[UL] + **FB[C]**)	12	$M_{ASM}^{mount}, M_{AAM}^{mount}, N_{ship}^{enemy}$	1
B_{prcp},S_{attk},S_{elec}, S_{defs},B_{mnur},W_{cncl}	ISR(AEW)→Feint(UAV)→Escort(F[A] + FB[A]) →Strike(B[UL] + F[L] + **FB[C]**)	13	$L_{F}^{aircraft}, L_{FB}^{aircraft}, M_{ASM}^{mount}, L_{AAM}^{mount}, D_{ship}^{enemy}$	14
B_{prcp},W_{attk},S_{elec}, S_{defs},B_{mnur},S_{cncl}	ISR(AEW)→Feint(UAV)→Escort(F[A] + FB[A]) →Strike(B[UL] + F[L] + **FB[L]**)	14	$L_{F}^{aircraft}, S_{FB}^{aircraft}, L_{ASM}^{mount}, M_{AAM}^{mount}, F_{ship}^{enemy}$	4
		15	$S_{F}^{aircraft}, S_{FB}^{aircraft}, L_{ASM}^{mount}, M_{AAM}^{mount}, N_{ship}^{enemy}$	12
B_{prcp},B_{attk},S_{elec}, S_{defs},B_{mnur},B_{cncl}	ISR(AEW)→Feint(UAV)→Escort(F[A] + FB[A]) →Strike(B[UL] + F[L] + **FB[C + L]**)	16	$S_{FB}^{aircraft}, M_{ASM}^{mount}, D_{ship}^{enemy}$	14

[1] S: strong, B: balanced, W: weak. The subscript for characters is in accordance with those in Table 2. [2] F: fighter, FB: fighter-bomber, B: bomber, AEW: air early warning, AR: anti-radiation missile, A: air-to-air missile, C: close-range ASM, L: long-range ASM, UL: ultra-long-range ASM. [3] AAM: air-to-air missile, ASM: air-to-sea missile, L: large consumption, M: medium consumption, S: small consumption, D: target ship got damaged, F: target ship got failed, N: target ship remained normal.

- The first column (A_1) denoted the contextual capability configuration code, accompanied by the organization modes, which indicated the advantage and disadvantage of a MPAMS against an enemy. The labelled code of each capability is as follows: W is for weak, B is for balanced, and S is for strong.
- The second column (A_2) included typical MPAMS organization modes extracted from data. For instance, *F1(X1[Y1], X2[Y2])→F2(X3[Y3])* denoted a pattern that task *F1* was first conducted via platform *X1* and *X2*, and *X1* was equipped with mount *Y1* while *X2* was equipped with mount *Y2*. Afterward, task *F2* was conducted via platform *X3* with mount *Y3*. The typical features of each mode are marked with bold and underlined fonts.
- The fourth column (B) represented the firepower effectiveness elements in the pattern, which included the integrity of a defender's key target and the attrition of the attacker. According to the operation objective, the integrity of the defender's key target comprised three integrity levels of the defender's ship: damaged, failed, and normal, which were abbreviated as D, F, and N, respectively. The patterns with different integrity levels are highlighted by separate colors in Table 5. Additionally, the attrition of the attacker, namely the aircraft or mount consumption, was divided into three levels by percentage, large, medium, and small levels, represented by symbols L, M, and S, respectively. The superscript indicates the type of aircraft or mount.
- The last column of Table 5 depicts how frequently a FCI occurs, and "6" means a FCI appears six times in all FCIs. The larger the number is, the more typical the FCI is.

As seen from Table 5, the knowledge includes 16 valuable FCIs. According to MPAMS's multi-layer collaboration modes of tasks, platforms, and mounts, the valuable organizational knowledge will be interpreted in a top-down manner, ranging from tasks to mounts.

(1) Task combinations

Since the MPAMS organization is mission-driven, the knowledge of task-oriented collaboration was derived first. Generally, the integration of pioneering or supporting tasks such as escort, feint, or electronic jamming, apart from the backbone striking task, could enhance operational effectiveness.

As for the escort task, *patterns 5 and 6* without the escort task showed good strike effectiveness of a damaged or failed ship; but these patterns were accompanied by large consumptions of fighter-bomber and ASM, which could be traced to the MPAMS's weak defense capability bias compared to the task arrangement with an escort (see *patterns 14 and 15*). Without an escort, fighters were more focused on attacking the target rather than protecting the bomber, which boosted the firepower advantage against the sea, while posing other aircraft with weaker maneuverability and defense to the enemy in air combat. Therefore, the escort task is recommended to be arranged to maintain MPAMS's overall defense level.

In terms of the feint task, *patterns 2–4* without the feint task, which was opposed to *patterns 14 and 15* with the feint by UAVs, showed fluctuating strike results. On the one hand, the victory was accompanied by massive fighter ($L_F^{aircraft}$) and fighter-bomber ($L_{FB}^{aircraft}$) consumptions. On the other hand, this radical strategy might occasionally lead to unintended results (depicted by *pattern 4*) when a host of bombers were destroyed. In the absence of the feint task, UAVs moved out collectively with the main formation with a stronger attacking capability. However, it was hard to consume enemy missiles with low-cost UAVs, leaving the main force vulnerable to the defender's firepower with reduced defense and concealment capability ($B_{defs}W_{cncl}$). Thus, the feint task is important for MPAMS task organization.

As for the electronic jamming task, when compared to *patterns 14 and 15*, task arrangement of *pattern 1* with the electronic jamming task showed comparable strike performance against the target ship but caused less fighter to be consumed ($S_F^{aircraft}$). The electronic jamming could pose interference with the defender's perception and reinforce the relative maneuverability of MPAMS expressed by code $S_{prcp}S_{mnur}$. Notably, while this task could slightly increase the relative striking effectiveness, it could not completely destroy the defender's ship. So, this task can be set as a complementary task for a MPAMS, but its effectiveness should not be overestimated.

(2) Platform collaboration

The aircraft type that performs the task was critical to ensure the striking effectiveness of a MPAMS. Different MPAMS capability gains in different aircraft collaboration modes made platform pairing knowledge valuable to learn. This type of knowledge was bomber-centric, as bombers took the dominant attacking role in air-to-sea assault operations, which conducted the strike task.

First, all-bomber was not the desired organization preference to conduct the strike task. An all-bomber formation equipped with high-lethality weapons, such as ultra-long-range missiles or bombs, could theoretically increase striking effectiveness. Nonetheless, a large correlation to the undesired striking results (see *patterns 7 and 8*) and heavy losses of bombers and mounts implied that this strategy was not as effective as expected. The reason lies in the limited defensive and maneuverability capabilities of the MPAMS in the two configurations, represented by codes $W_{defs}W_{mnur}$, which hinder the release of the MPAMS's attacking capability.

In addition, pairing bombers with high-performance aircraft, such as fighters or fighter-bombers, to perform escort and strike tasks, as shown by *patterns 9 and 10, 13–16*, was a feasible platform organization strategy of MPAMS, which increased the strike success rate while reducing the attrition of high-value bombers. Namely, this strategy could balance the attacking capability with the defense and maneuverability capabilities of the MPAMS, such as the updated capability code B_{attk},B_{defs},B_{mnur} of *patterns 9 and 10*.

For the modes of pairing the bomber with a pure type of fighter or fighter-bomber, the bomber + fighter pattern provided a better effectiveness ratio (*patterns 9–11*) than the bomber + fighter-bomber pattern (*pattern 12*). As the maneuverability of the fighter was higher than that of the fighter-bomber, it was more appropriate to match the weakly maneuverable bomber, as depicted in the comparison of the updated capability codes of B_{mnur} and W_{mnur}. However, *pattern 11* indicated that this strategy could have negative results, which was attributed to insufficient bomber protection by fighters.

On this basis, a further platform-level collaboration of the bomber and several types of fighter and fighter-bomber to conduct the escort and strike tasks together, as evidenced by *patterns 13–16*, could take advantage of the capabilities of individual platforms to achieve a stronger joint strength. However, these patterns had different effectiveness values under the same platform collaboration configuration. This could be due to the influence of the mount feature on the MPAMS organization, which is explained in the following.

(3) Mount mix

Based on task and platform collaboration, overlaying mount features could further balance the MPAMS capabilities to support task and platform collaboration. The mount organizational knowledge was evaluated considering two factors: range (from close to ultra-long range) and type (such as bombs, missiles, and soft weapons).

Concerning the range factor, *patterns 13–16* with different ASMs for fighter-bombers denoted representative cases for the mount pairing knowledge. In general, the close-range missiles achieved the best firepower effect but confronted with a weak defense capability. The long-range missiles highlighted survivability but failed to defeat the enemy in certain cases. In contrast, the mount combination of long- and close-range balanced the winning probability with troop consumption, which is recommended for the MPAMS organization.

In particular, *pattern 13* with close-range ASM achieved excellent strike performance with dense connections to the ship damage result. Because the highly offensive feature of the close-range missiles was superimposed on the carrier aircraft's inherent maneuverability by code $S_{attk}S_{defs}B_{mnur}$. However, the close-range strike needed a deep burst into the defender's territory, which affected the operation concealment and results under severe losses of fighter-bombers, fighters, and mounts. In summary, this strategy can be one of the best options for a striking effect, but it is expensive for the MPAMS organization in terms of troop attrition.

The key feature of *patterns 14 and 15* with long-range ASM was strong concealment capability (S_{cncl}), which reduced the fighter-bomber consumption (*pattern 14*). However, this strategy caused poor capability coordination of aircraft and missiles in a MPAMS. Because the dependence on the inherent capability of long-range missiles was too severe to penetrate the defender's defenses, it caused moderate strike effects of the ship to fail (F_{ship}^{enemy}). Moreover, in certain cases, this organization mode even had poor results (*pattern 15*) despite low consumption of the attacker's fighter and fighter-bomber (as the aircraft could return to camp after launching long-range firepower). Thus, this insurance strategy is not effective enough for the MPAMS organization.

For *pattern 16* equipped with both close- and long-range ASMs, the capability configuration indicated that the synergy of close- and long-range missiles coordinated attacking capability with concealment of the MPAMS, as given by code $B_{attk}B_{cncl}$. After effectively depleting the enemy's air defense using long-range missiles in the first wave, other aircraft carrying the close-range missiles could exploit the opportunity window to destroy the defender's ship (D_{ship}^{enemy}) successfully. In addition, the overall aircraft attrition was less than that of *pattern 13* when considering fighters and fighter-bombers. Therefore, it is a feasible approach to achieving striking effectiveness for MPAMS.

In terms of the mount type factor, the collaboration of different mount types could boost firepower effectiveness, which was further intertwined with aircraft capabilities. For example, when compared to the pure powerful mount of ultra-long-range missiles or bombs depicted in *patterns 7 and 8*, pairing ultra-long-range missiles and close-range missiles could compensate for each other's capabilities for improving performance, as illustrated by *patterns 9 and 10*. In addition, the FCI of *pattern 1* indicated that the multi-domain pairing of electronic weapons and long-range missiles served as a valuable organization preference to achieve a good effect at a low cost, which could release the unique strengths in different domains for the MPAMS to achieve comprehensive striking gains.

4.3.4. Efficiency Performance Evaluation

To test *CrossFCI*'s effectiveness on different MPAMS datasets, the context factors were extended to increase the simulation data scale, covering more than 20 factors that affected striking effectiveness, such as detailed types of platform and mount. Then, the four methods above were applied to 500, 1000, 2000, 3000, and 5000 records.

The discovered knowledge, namely the FCIs, of each algorithm was identical, which confirmed the correctness of *CrossFCI*. The comparison results of different algorithms regarding the efficiency performance are presented in Figure 12. Figure 12 indicates that the time-cost of *CrossFCI* is not distinguishable when the dataset is small. As the dataset scale increases, the time consumption of *CrossFCI* remains stable and outperformed the other algorithms' time consumption values. In addition, the time cost of *Ciclad* and *Moment* increase smoothly, while *CloStream* has a larger growth rate. For instance, there is a significant difference between *Moment* and *CrossFCI* on the dataset of 50 k, which take more than 100 s and less than 10 s, respectively. The results prove the feasibility of *CrossFCI* for large-scale MPAMS datasets, which shows a certain prospect of extracting more complex strategy knowledge from enormous wargame replay data.

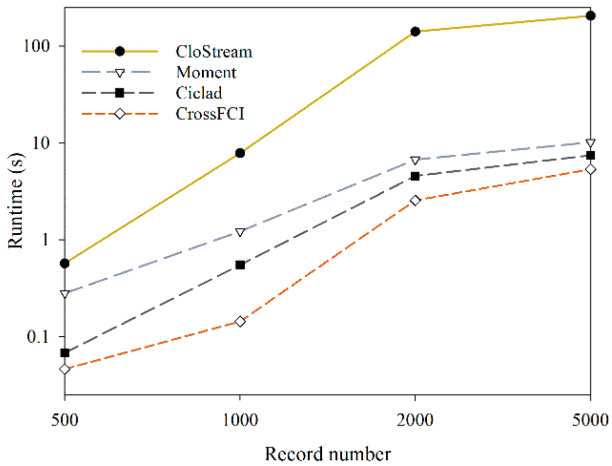

Figure 12. Performance comparison results of different FCIM algorithms on the MPAMS datasets with different sizes.

4.3.5. Discussion

The influence of the MPAMS capability factor on the MPAMS organizational performance has been primarily illustrated in Section 4.3.4. Particularly, there exists a strong correlation between the capability strengths or weaknesses and the collaboration strategy selection. Namely, well-designed decisions based on the capability configuration could increase striking effectiveness while prohibiting unnecessary attrition. In the following, commonalities of the above-mentioned organizational knowledge of tasks, platforms, and mounts are further extracted via capabilities, revealing intrinsic logic that bridges the decision-making process and decision effectiveness.

Knowledge 1: Balance capabilities of tasks, platforms, and mounts

Balancing the overall MPAMS capabilities is critical to the organizational strategy development, which could be reflected in the capability configuration code of a MPAMS. In addition to the attacking capability mainly represented by mounts, the maneuverability and defense capabilities of aircraft platforms to which the mount belongs, as well as task arrangements, could complement the shortcomings of the mount's inherent capability and strengthen the MPAMS's overall capability, thus increasing the victory probability.

For instance, a MPAMS mounted with pure but powerful mounts, such as bombs or ultra-long-range missiles, cannot achieve the desired performance in most circumstances because the mount's poor maneuverability fails to release the deserved level of the MPAMS's attacking capability. From the aircraft perspective, the weakness of the MPAMS's maneuverability can be complemented by the aircraft's capabilities. As a result, bombers carrying bombs or long-range missiles can achieve better outcomes when combined with fighters and fighter-bombers under high maneuverability. In addition, from the task perspective, the original pure-bomb configuration supplemented by an escort task can improve the formation's defense, thus compensating for the MPAMS's weakness regarding the air-to-air attacking capability. Then, the balanced capability distribution between the air and sea can reduce attrition.

Moreover, the cross-platform mix of high-capability mounts with low-capability aircraft in a MPAMS is preferable to achieve a differential advantage for the MPAMS organization. For instance, low-cost UAVs with limited maneuverability and attacking capability could disperse the defending air force and cover other friendly aircraft, which improves the relative defense and concealment capabilities of the aircraft carrying high-quality mounts, thus providing more strike opportunities for the attacking side.

Knowledge 2: Maintain mount type diversity

The mount layout implies a MPAMS's capability distribution. For instance, pairings of close- and long-range missiles can balance a MPAMS's attack and defense, while the ultra-long-range missile emphasizes defense priority.

In general, a MPAMS with multiple platforms carrying a mix of missiles and bombs poses a greater threat to the defender than that carrying a single mount type. Namely, different mount combinations provide more flexible and efficient attacking modes to boost task and platform collaboration. For instance, a diverse missile layout combining close- and long-range missiles can achieve a better striking effect than those carrying a single type of close-range or long-range missile. The first strike phase, which is dominated by the long-range missile, could ensure both concealment and attacking capability. Then the leading striking character is transferred to the close-range missile, which further releases the attacking capability of a MPAMS, improving the comprehensive performance from both attack and defense perspectives.

Despite the mount diversity when considering multiple platforms, a single platform could also benefit from the diversity of mounts when considering its unique capabilities. For instance, a fighter-bomber is an excellent platform for long-range missiles due to its high attacking capability. In contrast, close- and long-range mounts are suggested to be configured together for fighter-bombers when considering the aircraft's inherent maneuverability. Thus, the trade-off between attacking and defense capabilities could improve the firepower effectiveness, indicating its prospect in real-world applications.

Knowledge 3: Create capability asymmetric advantages

Asymmetric capability advantages against the opponent could be created mainly by task arrangements. The motivation is to exacerbate differences in the fragile part of the defender's capabilities. For instance, a MPAMS strikes with a feint task can distract the defender's attacking capability, while an escort task can enhance the MPAMS's air defense capability against the defender, and electronic jamming or suppression can reduce the defender's perception capability. The above task arrangements compensate for the drawbacks in the MPAMS capability configuration (e.g., its lack of outstanding attacking advantage) and magnify the advantage in another dimension, thus increasing the probability of victory.

5. Conclusions

Aiming to the challenge that the existing methods for discovering the organization preference knowledge of MPAMS have a gap in practical application due to weak effectiveness and efficiency, this paper proposes an innovative organizational knowledge mining method based on the FCIM to guide the MPAMS organization optimization.

To ensure knowledge effectiveness, higher systematic and interpretable knowledge is derived, benefiting from designing a hierarchical knowledge discovery framework as well as integrating the contextual capability reflecting decision-making motivation into the knowledge representation, making the knowledge more credible to the MPAMS organization. Further, to ensure knowledge efficiency, the time cost of the knowledge mining process is optimized by designing an itemset storage structure along with three pruning strategies.

In the tests, superior tactical multi-layer MPAMS collaboration knowledge about tasks, platforms, and mounts is extracted under typical air-to-sea assault scenarios at a faster speed. As for tasks, assigning precursor tasks, such as feints, before striking could boost striking advantages. In terms of platforms, collaborating highly attackable platforms with highly maneuverable platforms could better support tasks. Regarding mounts, pairing diverse-range and cross-domain weapons based on platform capabilities can achieve better strike effects.

On this basis, the deep decision logic of balancing capabilities of different layers, maintaining mount type diversity, and creating capability asymmetric advantages is obtained based on capability, which is more generic and understandable to decision-makers. In addition, the proposed algorithm's efficiency superiority over the existing methods is verified regarding time-cost, scalability, and pruning strategy efficiency.

The discovered knowledge is conducive to revealing effective and valuable MPAMS collaboration mechanisms, thus enhancing a MPAMS's comprehensive organization capability to improve striking performance. Moreover, this study provides meaningful guidance and prospects for further MPAMS development. However, the capability comparison of the attacking and defending sides is expressed qualitatively in this study.

In short-term future work, more context features could be explored and concluded in a quantitative description to reinforce knowledge integrity and transferability. For the mid-term plan, the acquired knowledge will be encoded to construct the knowledge database. Moreover, for the long-term plan, this domain knowledge will be integrated with other data-driven methods, such as deep learning and reinforcement learning, to build better autonomous organizing and decision-making capacities in MPAMS.

Author Contributions: Conceptualization, Y.W. and M.W.; methodology, Y.W.; validation, W.C. and G.W.; writing—original draft preparation, Y.W.; writing—review and editing, M.W. and W.C.; supervision, G.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Foundation of Shanghai (No. 20ZR1427800), and New Young Teachers Launch Program of Shanghai Jiao Tong University (No. 20X100040036).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Acknowledgments: The authors would like to thank the editor-in-chief, the associate editor, and the reviewers for their insightful comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAM	Air-to-air missile
AEW	Air early warning
ASM	Air-to-sea missile
CI	Closed itemset
FCI	Frequent closed itemset
FCIM	Frequent closed itemset mining
FI	Frequent itemset
IRS	Intelligence, surveillance, and reconnaissance
MPAMS	Multi-platform aircraft mission system
SoS	System-of-systems
UAV	Unmanned aerial vehicle

Appendix A

Table A1 records the MPAMS configurations in verification scenarios with varying tasks, platforms, and mounts. Specifically, the capability comparison against the defender is encoded based on Section 2.3, including strong (S), balanced (B), and weak (W). Subscript *Xprcp*, *Xattk*, *Xelec*, *Xdefs*, *Xmnur*, and *Xcncl* represent perception, attacking, electronic, defense, maneuverability, and concealment, respectively.

Configuration *Lng-range* is assumed as the basic organization mode of MPAMS. Configuration *Cls-range* and *Cls&Lng-range* vary from *Lng-range* in terms of the mount type for fighter-bombers. Configurations *No-escort* and *No-feint* drop the escort task and feint task from task combinations of configuration *Lng-range*, respectively. Configuration *Elec* is supplemented with the electronic jamming task. Configurations *Ultrlng-range* and *Bomb* are dominated by bombers, which carry pure ultra-long-range air-to-sea missiles or bombs for attacking. Configurations *Bomb&F_cls-range* and *Bomb&Fb_cls-range* differ with configuration *Ultrlng-range* in aircraft and mount types.

Table A1. MPAMS configuration description.

Configuration	Task	Platform	Mount	Capability Code
Lng-range	Surveillance + feint + escort + attack	BM × 4 + FB × 4 + FS × 4 + FA × 2 + UAV × 2 + AEW × 1 *	Long-range ASM	B_{prcp},W_{attk},S_{elec}, S_{defs},B_{mnur},S_{cncl}
Cls-range	surveillance + feint + escort + attack	BM × 4 + FB × 4 + FS × 4 + FA × 2 + UAV × 2 + AEW × 1	Close-range ASM	B_{prcp},S_{attk},S_{elec}, S_{defs},B_{mnur},W_{cncl}
Cls&Lng-range	Surveillance + feint + escort + attack	BM × 4 + FB × 4 + FS × 4 + FA × 2 + UAV × 2 + AEW × 1	Close-range ASM +long-range ASM	B_{prcp},B_{attk},S_{elec}, S_{defs},B_{mnur},B_{cncl}
No-escort	Surveillance + feint + attack	BM × 4 + FB × 4 + FS × 6 + UAV × 2 + AEW × 1	Long-range ASM	B_{prcp},B_{attk},S_{elec}, B_{defs},B_{mnur},W_{cncl}
No-feint	Surveillance + escort + attack	BM × 4 + FB × 4 + FS × 4 + FA × 2 + UAV × 2 + AEW × 1	Long-range ASM	B_{prcp},B_{attk},S_{elec}, B_{defs},S_{mnur},W_{cncl}
Elec	Surveillance + feint + escort + electronic jamming + attack	BM × 4 + FB × 4 + FS × 3 + FA × 2 + UAV × 2 + EF × 1 + AEW × 1	Long-range ASM + electronic warfare weapons	S_{prcp},W_{attk},S_{elec}, S_{defs},S_{mnur},B_{cnc}
Ultrlng-range	Surveillance + escort + attack	BM × 12 + FA × 2 + UAV × 2 + AEW × 1	Ultra-long-range ASM	B_{prcp},S_{attk},S_{elec}, W_{defs},W_{mnur},S_{cncl}
Bomb	Surveillance + escort + attack	BM × 12 + FA × 2 + UAV × 2 + AEW × 1	Bomb	B_{prcp},S_{attk},S_{elec}, W_{defs},W_{mnur},W_{cncl}
Bomb&F_cls-range	Surveillance + escort + attack	BM × 8 + FS × 4 +FA × 2 + UAV × 2 + AEW × 1	Ultra-long-range ASM + close-range ASM	B_{prcp},B_{attk},S_{elec}, B_{defs},B_{mnur},S_{cncl}
Bomb&Fb_cls-range	Surveillance + escort + attack	BM × 8 + FB × 4 + FA × 2 + UAV × 2 + AEW × 1	Ultra-long-range ASM + close-range ASM	B_{prcp},B_{attk},S_{elec}, B_{defs},W_{mnur},S_{cncl}

* AEW: Air early warning, ASM: Air-to-sea missile, BM: bomber, EF: Electronic fighter, FA: Fighter for air strike, FB: Fighter-bomber, FS: Fighter for sea strike, UAV: Unmanned aerial vehicle.

References

1. Sapaty, P. Mosaic Warfare: From philosophy to model to solutions. *Int. Robot. Automat. J.* **2019**, *5*, 157–166. [CrossRef]
2. Lemons, G.T.; Carrington, K. F-35 mission systems design, development & verification. In Proceedings of the 2018 Aviation Technology, Integration, and Operations Conference, Atlanta, GA, USA, 25–29 June 2018; pp. 3519–3535.
3. Bjerke, H.M.; Valaker, S. Command and control in a fifth generation air force: Coordination requirements of air operations with F-35 and the command and control-system of the norwegian armed forces. *Scand. J. Mil. Stud.* **2022**, *5*, 14–30. [CrossRef]
4. Wang, G.; Gu, Q.; Wu, J.; Wang, M. Concept and architecture for new generation aircraft mission system. *Acta. Aeronaut. Astronaut. Sin.* **2015**, *36*, 2735–2752.
5. Harteveld, C.; Kleinman, E.; Rizzo, P.; Schouten, D.; Nguyen, T.H.; Liberty, S.; Kimbrough, W.; Fombelle, P.; El-Nasr, M.S. Teamwork and adaptation in games (TAG) a survey to gauge teamwork. In Proceedings of the 14th International Conference on the Foundations of Digital Games, San Luis Obispo, CA, USA, 26–30 August 2019; pp. 1–12.
6. Cheng, K.; Chen, G.; Yu, X.; Liu, M.; Shao, T. Knowledge traction and data-driven wargame AI design and key technologies. *Syst. Eng. Electron. Technol.* **2021**, *43*, 2911–2917.
7. Huang, C.; Wei, Z.; Yang, Y.; Zhang, H. Knowledge acquisition for the air combat based on gwo. In Proceedings of the 2019 International Conference on Artificial Intelligence Technologies and Applications, Qingdao, China, 5–7 July 2019; pp. 12078–12087.
8. Johansson, T. Tactical Simulation in Air-to-Air Combat: Evolutionary Algorithms and Behavior Tree Framework. Master's Thesis, Lulea University of Technology, Luleå, Sweden, 2018.
9. Masek, M.; Lam, C.P.; Kelly, L.; Benke, L.; Papasimeon, M. A genetic programming framework for novel behaviour discovery in air combat scenarios. In Proceedings of the ASOR/DORS Conference 2018, Melbourne, Australia, 4–6 December 2021; pp. 263–277.

10. Masek, M.; Lam, C.P.; Benke, L.; Kelly, L.; Papasimeon, M. Discovering emergent agent behaviour with evolutionary finite state machines. In Proceedings of the International Conference on Principles and Practice of Multi-Agent Systems, Tokyo, Japan, 29 October–2 November 2018; pp. 19–34.
11. Dong, S.; Xu, W.; Zhang, K. Study on the resolution of multi-aircraft flight conflicts based on an IDQN. *Chin. J. Aeronaut.* **2022**, *35*, 195–213.
12. Lee, D.; Kim, M.J.; Ahn, C.W. Predicting combat outcomes and optimizing armies in StarCraft II by deep learning. *Expert. Syst. Appl.* **2021**, *185*, 115592. [CrossRef]
13. Dong, Y. Deep learning-based opponent aircraft attitude detection in autonomous air combat. *J. Aerosp. Inform. Syst.* **2019**, *16*, 162–167. [CrossRef]
14. Li, B.; Liang, S.; Chen, D.; Li, X. A decision-making method for air combat maneuver based on hybrid deep learning network. *Chin. J. Electron.* **2022**, *31*, 107–115.
15. Hooshyar, D.; Yousefi, M.; Lim, H. Data-driven approaches to game player modeling: A systematic literature review. *ACM Comput. Surv.* **2018**, *50*, 90. [CrossRef]
16. Chee, C.; Jaafar, J.; Aziz, I.A.; Hasan, M.H.; Yeoh, W. Algorithms for frequent itemset mining: A literature review. *Artif. Intell. Rev.* **2019**, *52*, 2603–2621. [CrossRef]
17. Enstad, K. Professional knowledge through wargames and exercises. *Scand. J. Mil. Stud.* **2022**, *5*, 233–243. [CrossRef]
18. Mason, R.C. Wargaming: Its history and future. *Int. J. Intell. Secur. Public. Aff.* **2018**, *20*, 77–101. [CrossRef]
19. López-Felip, M.A.; Davis, T.J.; Frank, T.D.; Dixon, J.A. A cluster phase analysis for collective behavior in team sports. *Hum. Mov. Sci.* **2018**, *59*, 96–111. [CrossRef]
20. Ahmad, S.; Bryant, A.; Kleinman, E.; Teng, Z.; Nguyen, T.H.D.; El-Nasr, M.S. Modeling individual and team behavior through spatio-temporal analysis. In Proceedings of the Annual Symposium on Computer-Human Interaction in Play, Barcelona, Spain, 22–25 October 2019; pp. 601–612.
21. Javvaji, N.; Harteveld, C.; El-Nasr, M.S. Understanding player patterns by combining knowledge-based data abstraction with interactive visualization. In Proceedings of the Annual Symposium on Computer-Human Interaction in Play, Online, 2–4 November 2020; pp. 254–266.
22. Yang, P.; Harrison, B.E.; Roberts, D.L. Identifying patterns in combat that are predictive of success in MOBA games. In Proceedings of the Foundations of Digital Games 2014 Conference, Ft. Lauderdale, FL, USA, 3–7 April 2014; pp. 1–8.
23. Goyal, P.; Sapienza, A.; Ferrara, E. Recommending teammates with deep neural networks. In Proceedings of the 29th on Hypertext and Social Media, Baltimore, MD, USA, 9–12 July 2018; pp. 57–61.
24. Guzmán, J.G.L.; Medina, C.J.B. Relevant independent variables on MOBA video games to train machine learning algorithms. In Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2021, Online, 6–10 September 2021; pp. 171–179.
25. Lin, H.; Chen, J.; Song, Y. Warship combat decision support system based on data mining technology. *Procedia Comput. Sci.* **2022**, *166*, 248–251. [CrossRef]
26. Schwartz, P.J.; O'Neill, D.V.; Bentz, M.E.; Brown, A.; Doyle, B.S.; Liepa, O.C.; Lawrence, R.; Hull, R.D. AI-enabled wargaming in the military decision making process. In Proceedings of the SPIE 11413, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II, Online, 27 April–9 May 2020; pp. 118–134.
27. Cao, G.; Wang, Y.; Zhang, L. Application of association rule mining in operational experiment data analysis. *Command Control Simul.* **2019**, *2*, 70–74.
28. Li, Z.; Zhu, L.; Wei, W. Application of association rule mining to target analysis in electronic countermeasure. *Ship Electron. Countermeas.* **2015**, *38*, 1–4.
29. Xing, S.; Ni, L.; Zhang, H.; Yan, K. Mining of weapon utility based on the replay data of war-game. *J. Command Control* **2020**, *6*, 132–140.
30. Yu, L.; Zhang, D.; Chen, X.; Xie, X. Moba-slice: A time slice based evaluation framework of relative advantage between teams in MOBA games. In Proceedings of the 7th Workshop on Computer Games, Stockholm, Sweden, 13 July 2018; pp. 23–40.
31. Kleinman, E.; Ahmad, S.; Teng, Z.; Bryant, A.; Nguyen, T.H.D.; Harteveld, C.; El-Nasr, M.S. "And then they died": Using action sequences for data driven, context aware gameplay analysis. In Proceedings of the 15th International Conference on the Foundations of Digital Games, Bugibba, Malta, 15–18 September 2020; pp. 1–12.
32. Telikani, A.; Gandomi, A.H.; Shahbahrami, A. A survey of evolutionary computation for association rule mining. *Inform. Sci.* **2020**, *524*, 318–352. [CrossRef]
33. Fasihy, H.; Shahraki, M.H.N. Incremental mining maximal frequent patterns from univariate uncertain data. *Knowl.-Based Syst.* **2018**, *152*, 40–50. [CrossRef]
34. Miholca, D.L.; Czibula, G.; Crivei, L.M. A new incremental relational association rules mining approach. *Procedia Comput. Sci.* **2018**, *126*, 126–135. [CrossRef]
35. Djenouri, Y.; Belhadi, A.; Fournier-Viger, P. Extracting useful knowledge from event logs: A frequent itemset mining approach. *Knowl.-Based Syst.* **2018**, *139*, 132–148. [CrossRef]
36. Dam, T.L.; Ramampiaro, H.; Nørvåg, K.; Duong, Q.H. Towards efficiently mining closed high utility itemsets from incremental databases. *Knowl.-Based Syst.* **2019**, *165*, 13–29. [CrossRef]

37. Martin, T.; Francoeur, G.; Valtchev, P. CICLAD: A fast and memory-efficient closed itemset miner for streams. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Online, 6–10 July 2020; pp. 1810–1818.
38. Yen, S.; Wu, C.; Lee, Y.; Tseng, V.S.; Hsieh, C.H. A fast algorithm for mining frequent closed itemsets over stream sliding window. In Proceedings of the 2011 IEEE International Conference on Fuzzy Systems, Taipei, China, 27–30 June 2011; pp. 996–1002.
39. Chi, Y.; Wang, H.; Yu, P.; Muntz, R.R. Moment: Maintaining closed frequent itemsets over a stream sliding window. In Proceedings of the Fourth IEEE International Conference on Data Mining, Brighton, UK, 1–4 November 2004; pp. 59–66.
40. Škopljanac-Mačina, F.; Blašković, B. Formal concept analysis–overview and applications. *Procedia Eng.* **2014**, *69*, 1258–1267. [CrossRef]
41. Mednikarov, B.; Dereliev, P. Structured description of naval tasks. *Inform. Secur. Int. J.* **2004**, *13*, 25–34. [CrossRef]
42. Chen, Q.; Tian, H.; Wang, P.; Leng, S.; Xiao, Z. A collocation scheme of distributed cooperative operational weapons based on OODA loop. *Acta Armamentarii* **2021**, *42*, 1780–1788.
43. Staff, J. *Universal Joint Task List (UJTL)*; United States Department of Defense: Washington, DC, USA, 2023. Available online: https://www.jcs.mil/Portals/36/Documents/Doctrine/training/ujtl_tasks.pdf?ver=yKLdOVrp5veCK84zYZzIlg%3d%3d (accessed on 13 January 2023).