Expert Systems with Applications 39 (2012) 11642–11656

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A multi-agent data mining system for cartel detection in Brazilian government procurement

Célia Ghedini Ralha [a,*], Carlos Vinícius Sarmento Silva [a,b]

[a] Computer Science Department, University of Brasília, P.O. Box 4466, 70.904-970 Brasília, Brazil
[b] Controladoria-Geral da União, SAS, Qd 01, Bl A, Ed. Darcy Ribeiro, 70.070-905 Brasília, Brazil

ARTICLE INFO

Keywords:
Multi-agent data mining system
Cartel detection
Brazilian government procurement
AGMI
Multi-agent
Distributed data mining
Database knowledge discovery

ABSTRACT

The main focus of this research project is the problem of extracting useful information from the Brazilian federal procurement process databases used by government auditors in the process of corruption detection and prevention to identify cartel formation among applicants. Extracting useful information to enhance cartel detection is a complex problem from many perspectives due to the large volume of data used to correlate information and the dynamic and diversified strategies companies use to hide their fraudulent operations. To attack the problem of data volume, we have used two data mining model functions, clustering and association rules, and a multi-agent approach to address the dynamic strategies of companies that are involved in cartel formation. To integrate both solutions, we have developed AGMI, an agent-mining tool that was validated using real data from the Brazilian Office of the Comptroller General, an institution of government auditing, where several measures are currently used to prevent and fight corruption. Our approach resulted in explicit knowledge discovery because AGMI presented many association rules that provided a 90% correct identification of cartel formation, according to expert assessment. According to auditing specialists, the extracted knowledge could help in the detection, prevention and monitoring of cartels that act in public procurement processes.

1. Introduction

As the massive volume of data stored in distributed infrastructures (e.g., data warehouses) continues to grow, useful analyses of data in connection with government audits have become increasingly difficult. In addition, traditional centralised data mining (DM) methods no longer meet auditors' needs because of security, privacy, data confidentiality and infrastructure limitations (such as network bandwidth). Addressing this problem means utilising different computer science fields, such as distributed data mining (DDM), knowledge discovery in databases (KDD), multi-agent systems (MAS) and other distributed computing technologies (e.g., grid, cloud, etc.). To address this issue, the main goal of this research project is to find a technological solution to the problem of generating and extracting useful knowledge from the vast amounts of information used by auditors in the process of corruption prevention. Thus, this research applies two increasingly interrelated research areas, MAS and DDM/KDD. The motivation for using MAS is that collective intelligence has to be developed through the analysis of DDM because the underlying task is not completely decomposable and/or the computing resources are limited. MAS is particularly useful when applied to DDM/KDD in the context of sharing resources and expertise. Furthermore, because KDD is concerned with extracting hidden knowledge from data, including data that is widely distributed in many different forms and in multiple databases, the integration of MAS offers an even greater advantage.

The integration of agent technology (including individual agents, MAS and societies) into DDM technologies expanded into the agent-mining interaction and integration (AMII) research field, where complementary benefits are sought from both communities (Cao, 2009; Ralha, 2009). In the literature, we find many different approaches to this integration that can be bidirectional: (i) the agent-driven DM approach, where DM with a number of discrete and dependent tasks can use agents to regulate, control and organise potentially distributed activities in the knowledge discovery process, and (ii) the DM-driven agent approach, where the discovered knowledge items can constitute building blocks for agent intelligence.

This paper presents AGMI, an agent-mining tool that focuses the AMII field in two directions, uniting an agent-driven DM approach with a DM-driven agent approach when the discovered knowledge develops agent's intelligence and ability to mine databases through validated rules. As proof of concept, AGMI is utilised in a test case in an audit of the domain of the Brazilian government procurement, where the goal of the audit is to detect cartel formation in government procurement. We believe that the conceptual framework of

* Corresponding author. Tel.: +50 61 3107 0468.
E-mail addresses: ghedini@cic.unb.br (C. Ghedini Ralha), carlos.silva@cgu.gov.br (C.V. Sarmento Silva).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.04.037

AGMI is also useful in other domains, as we discuss further in Section 6. The rest of the paper is divided as follows. Section 2 discusses research motivation, while Section 3 briefly presents the MAS and DM areas of research. Section 4 presents an overview of the related work. Section 5 presents the AGMI tool, including its conceptual framework, implementation process and knowledge evaluation aspects. Section 6 outlines the use of AGMI in the Brazilian government procurement test case. Section 7 discusses important aspects of the research, and Section 8 concludes the paper and identifies directions for future work.

2. Motivation

Currently, enormous volumes of data are being produced and stored in computer systems around the world. The Brazilian Government Information Systems (BGIS) are a good example of this type of computerisation of an enormous volume of data. For example, in 2009, one BGIS, the Integrated Financial Management System of the Federal Government (SIAFI – Sistema Integrado de Administração Financeira do Governo Federal), registered and stored one billion financial transactions.[1] In terms of growth, in 1998, the average number of transactions (data searches and insertions) executed monthly at SIAFI was 39 million. This number reached 63.8 million by 2002 and 87 million in 2009, resulting in a 223% increase over the last 10 years.

All these financial data are used to support the preparation and execution of government auditing. The Office of the Comptroller General (CGU - Controladoria Geral da União)[2] is the agency of the Federal Government of Brazil that assists the president in defending public assets and enhancing management transparency through internal control activities, public audits, corrective and disciplinary measures, corruption prevention and combat and coordinating ombudsman activities. As the internal audit unit and the anti-corruption agency of the Brazilian Federal Government, CGU implements the important action of corruption prevention by applying different technologies to promote transparency and prevent corruption.

Our research has focused on the formation of cartels in the public procurement processes in Brazil. Identifying cartels is difficult because it requires the analysis of several public bidding processes, which usually exceeds the scope of a single government agency. Cartels can operate in various government departments, cities and even states of the Federation, which demands the sophisticated analysis of massive datasets.

In addition, the analysis of data from databases typically proceeds with languages using queries such as those found in the Structured Query Language (SQL),[3] which renders analysis impractical because of the exponential solution space. Therefore, auditing activities involved in detecting cartels are generally limited to confirming suspicions (normally after denunciation), given the difficulties inherent in the process of cartel detection.

Consider the goal of analysing all the possible groups of companies for cartel detection. We have to prepare all the possible combination of companies with at least two companies, as shown in Eq. (1), where n is the total number of companies in the database.

$$\sum_{i=2}^{n} C_i^n = 2^n - n - 1 \tag{1}$$

Thus, the brute-force algorithm runs in $O(2^n)$, and there is no deterministic way to identify cartels effectively because the solution space is exponentially related to the number of companies that participate in the bidding processes being analysed. Thus, the problem consists of creating an efficient process to identify groups of companies that might be suspected of practicing cartels in public procurement processes.

Because the volume of data to be treated is very large and the solution space is exponential, DM techniques are adequate to address the problem of analysing and understanding the massive datasets. However, the DM process demands a substantial amount of work to prepare the data, especially considering the use of different DM algorithms. Furthermore, DM algorithms alone cannot address the problem of workload distribution, considering the runtime execution: for example, through the use of parallel execution, which is important when handling massive datasets. Therefore, together with DM techniques, the MAS approach is important because it speeds up the execution time in a distributed and parallel manner. It is also essential to allow the use of rational agents to prepare data for DM with different algorithms and with the autonomy to analyse and improve the algorithms' results.

Table 1
Database used in the experiment.

Information	Number
Records	26,615
Procurement processes	2701
Companies	3051
Companies with at least 1 victory	1162
Companies with at least 5 victories	121

2.1. Initial problem analysis

When we first considered the problem of cartel detection in public procurement processes, we proposed a solution using association rule mining (ARM) because this technique is useful for finding strong relationships between attributes. The database used in this study was the Brazilian Federal Procurement system ComprasNet.[4] ComprasNet is a large database with information on all the procurement processes of the various types of services that are contracted by the Federal Executive agencies in all twenty-seven states of the Federation (Federation Unit – FU) of Brazil. Table 1 presents information about the dataset used in our experiments. This dataset records all the procurement processes to contract a specific type of service between the years 2005 and 2008. Each record in the dataset represents one bid from one company in the procurement process.

Most procurement processes consist of the following basic attributes: (i) participant companies or suppliers and their bids; (ii) negotiated object or service; (iii) government agency; (iv) city/state; and (v) winner. Table 2 describes the ComprasNet data related to the last procurement of each participant (DF – Federal District and MG – state of Minas Gerais).

To apply the ARM technique to associate only companies/suppliers, we have created a new dataset from the Table 2 data, which is presented in Table 3. Note that each column A, B, C, D, E, F, G represents supplier participation in each procurement process through a Boolean attribute, which registers a company's participation or nonparticipation in the procurement process.

The dataset for the ARM technique must be constructed as the matrix A, consisting of m rows and n columns:

$$m = (\text{total number of procurement processes from the database})$$
$$n = (\text{total number of companies from the database})$$

$$a_{i,j} = \begin{cases} true & \text{if company } j \text{ has participated in procurement } i; \\ false & \text{if company } j \text{ has not participated in procurement } i; \end{cases}$$

[1] The SIAFI official site – http://www.tesouro.fazenda.gov.br/siafi/index_mapa.asp.
[2] The CGU official site -http://www.cgu.gov.br/english/default.asp.
[3] MySQL official site – http://www.mysql.com/.
[4] The ComprasNet official site - http://www.comprasnet.gov.br/.

Table 2
Example of *ComprasNet* procurement data.

Procurement Id	Participant Id	FU	Object	Company	Value	Winner
1	121	DF	X	A	10.00	10.00
1	121	DF	X	B	12.00	–
1	121	DF	X	D	12.50	–
2	133	MG	Y	A	42.00	–
2	133	MG	Y	E	41.50	–
2	133	MG	Y	F	40.00	40.00
2	133	MG	Y	G	43.00	–
3	121	DF	Z	C	21.00	–
3	121	DF	Z	B	18.75	–
3	121	DF	Z	A	18.10	18.10

Table 3
New dataset for procurement process.

Procurement Id	A	B	C	D	E	F	G
1	True	True	False	True	False	False	False
2	True	False	False	False	True	True	True
3	True	True	True	False	False	False	False

where $a_{1,n+1}$ = winner (i), such that $1 \leqslant i \leqslant m$; $1 \leqslant j \leqslant n$.

Thus, we expect to obtain the following type of rule:

$$Company_A = true, \quad Company_B = true \rightarrow Company_C = true$$

To obtain this type of rule, we must suppress rules with attributes that indicate the nonparticipation of companies. This action is necessary to identify cartels, which can be defined as the participation in procurement processes of all the companies of a specific group.

Note that when we include all the companies as attributes in the dataset, it is common to find more false values than true values because not all companies of the dataset take part in most procurement processes. Thus, we have removed the association rules that attest to associations among companies with false participation values, such as the following rule:

$$Company_E = false, \quad Company_F = false \rightarrow Company_G = false$$

Consequently, the dataset preparation for the ARM process was adapted as follows: the dataset of Table 3 was completed with '?' when the company's value is false for participation. In the Weka framework (version 3.6.1 University Waikato, 2009) '?' is used to represent a missing value, and the algorithm ignores it.

In Silva and Ralha (2010), we presented preliminary experiments conducted using DM techniques to detect cartel formation in public procurement processes. These experiments used ARM and clustering to enable the utilisation of MAS and DDM in an integrated approach that then culminated in the conceptual framework of AGMI. The experiments were conducted using the data from the ComprasNet database presented in Table 1 to create the datasets presented in Tables 2 and 3. For these experiments, two datasets were prepared using the Apriori algorithm of the Weka framework, version 3.6.1 (University Waikato, 2009), dataset 1, comprising all the procurements (2,701 instances) and companies (3,048) of Table 1, and dataset 2, including only companies that had won at least two procurements. Those companies were selected to identify only the associations among companies that were likely to participate in a cartel.

The analysis of the preliminary experiments proved the importance of MAS and DM technology interaction and integration under the AMII approach. There are many DM tasks that can be automated using the MAS distributed infrastructure, such as time-consuming dataset preparation (the preprocessing phase) and the work-intensive execution of different DM algorithms (the processing phase) that can be run in parallel. For example, using clustering to divide the database into market regions necessitated the preparation of new datasets to search for cartels, a very time-consuming task.

Additionally, after the dataset preparation, it was necessary to apply rule association algorithms to each of the ten defined clusters, which was computer intensive. Other important agent characteristics, such as autonomy, can be used to parameterise the lower bound (LB) of support and LB of confidence in the use of different DM algorithms. Because reducing the value of the LB of support increases the number of rules and therefore the memory required by the algorithm, the DM algorithm parameterisation work depends on the available infrastructure and must be evaluated to make better use of the distributed environment. In addition, the agents' reasoning mechanism can apply heuristics, using its autonomy to evaluate the aspects of efficiency and the quality of the rules returned by the distributed KDD process. A discussion of the use of agent-mining aspects in cartel detection in public procurement processes is presented in Silva and Ralha (2011b), while AGMI's first architectural draft is presented in a chart at (Silva & Ralha, 2011a).

3. MAS and DM overview

In this section, we present a brief survey of MAS and DM in light of the existence of the AMII area of study. The AMII area benefits from the possibilities that distributed MAS offers to improve overall DM performance. This is important because AMII is not concerned with a specific DM technique, but instead is concerned with the collaborative work of distributed agents in MAS design.

3.1. Agent and MAS

In general, an intelligent software agent (ISA) uses artificial intelligence in the pursuit of its goals (Luger, 2002; Russell & Norvig, 2010). Thus, an ISA is a software module that is capable of performing autonomous actions in certain environments to meet its objectives (Wooldridge, 2009). The features and properties of an ISA highlight the importance of MAS. MAS is a collection of autonomous entities called agents, which interact with each other and their environment in a cooperative or competitive way to achieve individual or group objectives. Contrary to traditional modelling techniques, MAS is not expressed in terms of variables, functions and equations, but in terms of agents, objects and environments. According to Wooldridge (2009), the primary advantages of MAS are decentralised control, robustness, simple extensibility, shared expertise and resources.

According to Wooldridge (2009), an agent is a computing entity that possesses the following four features: reactivity, autonomy, interaction and initiative. Autonomy is the main characteristic of an agent; in other words, agents are capable of acting independently and controlling their internal states, as opposed to traditional event-driven approaches. Discussing the reactivity feature of MAS, (Weiss, 2000) defines a reactive system as a system that maintains continuous interaction with its environment and answers the changes that occur in that environment. Thus, this type of agent perceives and reacts to environmental stimuli; this ability may be improved with a goal-oriented approach.

To this end, the initiative feature of agents can be exploited using goal-directed behavior. In this case, an ISA generates and attempts to achieve goals by taking initiative and recognising opportunities instead of only being driven by external events. Another aspect of the interactive features of an ISA is its social ability, i.e., the capacity to interact with other agents (or humans) through the use of some type of agent-communication language.

Cooperating with others is important because some goals can only be achieved through cooperative work.

Compared to a client–server centralised system, the advantages of MAS include distribution of processing, support for a more flexible peer-to-peer (P2P) model, decentralisation of control, reduction of network bandwidth use and scalability (Meng, Ye, Roy, & Padilla, 2007). In summary, MAS functions well in complex applications that require distributed problem solving.

In many complex applications, the collective behavior of the agents depends on the observed data that is distributed among different sources. The problem of analysing distributed data in MAS is nontrivial, especially considering the limitations of the resources. In this way, the DDM field meets the challenge of analysing distributed data to offer solutions through the use of algorithmic and mining operations in a distributed manner under the resource constraints.

Thus, the idea of integrating MAS into DDM for data-intensive applications is attractive. Recent studies on distributed data analysis using MAS have been developed in different domains because the systems for managing critical infrastructures such as energy, traffic, industry automation, etc., are highly complex, distributed, and increasingly decentralised. Some recent studies using the AMII approach focus on agents for DDM (Bhamra, Verma, & Patel, 2011; Chaimontree, Atkinson, & Coenen, 2011), DM for agent (Chatzidimitriou, Chrysopoulos, Symeonidis, & Mitkas, 2011; Kaur, Goyal, & Lu, 2011), and agent mining applications (Wu, Cao, & Fang, 2011). Regardless of how advanced the agents are, they must perform the underlying data analysis tasks efficiently in a distributed manner from domain knowledge and reasoning perspectives. This task is critical to the success of the multi-agent data mining (MADM) system, and it is complicated because a distributed data collection has properties that are both homogeneous and heterogeneous.

3.2. Data mining techniques

DM has evolved to become a well-established technology field with subfields such as classification, clustering, and rule mining. Research in these fields continues to develop ideas, generate new algorithms and modify/extend existing algorithms. This research has culminated in a diverse body of work and spawned a community prepared to share their expertise, assisted by software freely available for download (Coenen, 2003; University Waikato, 2009).

DM deals with the problem of analysing data scalably, while DDM is a branch of DM concerned with the framework in which the distributed data are mined, paying careful attention to the distributed source of the data and computing resources (Hand, Smyth, & Mannila, 2001; Hastie, Tibshirani, & Friedman, 2009; Witten & Frank, 2005). DM involves fitting models to, or determining patterns from, observed data. Models fitted this way assume the role of inferred knowledge. However, deciding whether these models reflect useful knowledge is part of the overall interactive KDD process, which usually requires subjective human judgment. The literature describes a wide variety and number of DM algorithms. According to Hastie et al. (2009), the common model functions in DM practice include (i) classification; (ii) regression; (iii) clustering; (iv) summarisation; (v) dependency modelling; (vi) sequence analysis; and (vii) link analysis. Because of our research focus, we will describe two of the cited model functions in DM in greater detail and provide a brief overview of DDM.

3.2.1. Clustering

According to Hand et al. (2001), clustering is a descriptive task, in which a finite set of categories or clusters is identified that is useful in describing information. These categories may be mutually exclusive and exhaustive, but they may also be hierarchical or overlapping. The cluster analysis is related to other techniques that divide data objects into groups. For example, clustering can be considered as a form of classification, creating labelled objects with class labels (the clusters).

In contrast, classification is a supervised process in which new, previously unlabelled objects receive a class label using a model developed from objects with known class labels. For this reason, cluster analysis is sometimes referred to as a type of unsupervised classification method (Tan, Steinbach, & Kumar, 2005).

According to Borman (2004), many algorithms perform clustering tasks, which can be classified in the following manner:

- Exclusive clustering – data are grouped in an exclusive way, so that if a certain datum belongs to a definite cluster, it cannot be included in another cluster. An example of this algorithm is the K-means, where the separation of points is achieved by a straight line on a bi-dimensional plane (Wu et al., 2007).
- Overlapping clustering – data are clustered in fuzzy sets, so that each point may belong to multiple clusters with different degrees of membership. Data are associated with the appropriate membership values. Fuzzy C-means is an example of this algorithm (Bezdek, 1981; Dunn, 1973).
- Hierarchical clustering – this is based on the union between the two nearest clusters. The starting condition is realised by setting every datum as a cluster. After a few iterations, the final desired clusters are reached. The hierarchical clustering algorithm is an example of this type of algorithm (Johnson, 1967).
- Probabilistic clustering – each cluster can be mathematically represented by a parametric distribution in a completely probabilistic approach, such as a Gaussian distribution (continuous) or a Poisson distribution (discrete). The algorithm used to find the mixture of Gaussians that can model the data set is called Expectation–Maximization (EM) (Dempster, Laird, & Rubin, 1977).

The EM algorithm computes the probabilities of the members of clusters based on one or more categories of probability distributions, and its goal is to maximise the likelihood of the data in the clusters. According to Han and Kamber (2005), EM is an iterative refinement algorithm that can be used to find estimated parameters. It is similar to an extension of the K-Means paradigm, which connects an object to a similar cluster based on the averages that are found. Instead of assigning each object to a single cluster, objects can be associated with more than one cluster, defining a weight for each association that represents the probability that an object belongs to the cluster. For the EM algorithm and discussions that are more elaborate about EM, see (Borman, 2004; Dempster et al., 1977; Wu et al., 2007).

According to Wu et al. (2007), K-Means and EM are among the top ten algorithms used by the DM community. The EM algorithm allows a richer representation of clusters because each element can belong to more than one cluster (soft classification). Based on its richer representation of clusters, the EM algorithm was selected for this research project because one company can be related to more than one cluster of companies in public procurement processes.

3.2.2. Association rules

ARM consists of a method of finding strong relationships between certain attributes. This technique has the ability to detect patterns in the form of rules that associate attribute values in a given data set. These rules are expressed in the form of conjunctions of the following type:

$$atrib_1 = value_1, \ldots, atrib_m = value_m \rightarrow atrib_{m+1} = value_{m+1}, \ldots, atrib_n = value_n$$

where *atrib* is a data set attribute and *value* is the attribute value identified in the rule.

In Han and Kamber (2005), we find a formal definition of the association rule:

Definition 1. Let $I = \{I_1, I_2, \ldots, I_M\}$ be a set of items, and let D be the data from the database, where T is the set of transactions D such that $T \subseteq I$. A and B are also set items. An association rule is an implication of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, and $A \cap B = \phi$. The rule $A \Rightarrow B$ applies to the set of transactions D with support s, where s is the percentage of transactions in D containing $A \cup B$, i.e., the probability $P(A \cup B)$. The rule $A \Rightarrow B$ has confidence c in the set of transactions D, where c is the percentage of transactions in D that contain A and also contain B: i.e., the conditional probability $P(B|A)$.

According to Tan et al. (2005) and as stated in Section 2, it is prohibitive to calculate all the association rules because the brute-force approach in this case is exponential according to the associative items analysed. The space complexity for the calculation of the rules by brute force has the following function: $R = 3^d - 2^d + 1$, where d is the number of items.

According to Witten and Frank (2005), the difference between the classification and association rules is that these patterns can predict using any attribute and are not limited to predicting with only the selected class. Different association rules express different underlying regularities in the data set, and each one predicts different things. There are a number of algorithms for the association rules. Apriori is a seminal algorithm for ARM, and it is the most popular algorithm for association rules, producing good results (Rakesh & Ramakrishnan, 1994). Apriori is used in this research to mine frequent item set association rules.

According to Han and Kamber (2005), the association rule coverage is measured by the probability that the rule is repeated in the data set. The frequency with which items appear in the database is also called the support. The lower bound of support represents a number that is less than or equal to every number in a given set. In order theory, a lower bound of a subset S of some partially ordered set $(P, \leqslant)$ is an element of P, which is less than or equal to every element of S.

The association rule accuracy is often called the confidence, which can be defined as the number of instances that it predicts correctly expressed as a proportion of all the instances to which it applies. The confidence is used to measure the quality of the rule, i.e., the correctness of the rule inference. In other words, a rule with high confidence means that it infers well in the universe in which it is defined. Therefore, the higher the support and the confidence, the better the quality and the stronger the association rule. For example, consider the following rule:

If temperature = cool then humidity = normal

The coverage is the number of days that are both cool and have normal humidity, and the accuracy is the proportion of cool days that have normal humidity (in this case, 100%).

3.2.3. Distributed data mining: a brief overview

In recent years, the development of DM algorithms that address the constraints of distributed datasets has received significant attention in the DM community, and the DDM field has emerged as an active area of study. In the literature, DDM methods operate over an abstract architecture that includes multiple sites with independent computing resources and storage capabilities. Local computation is performed on each site through a central site that communicates with each distributed site to compute global models or simple distribution methods from the area of grid infrastructure (Congiusta, Talia, & Trunfio, 2008; Luo, Wang, Hu, & Shi, 2007), cloud computing or P2P approaches. The communication process is necessary in all cases, either through resources at a centralised site or with neighboring nodes by passing messages over an asynchronous network.

The communications process is a bottleneck, so one primary goal of many DDM (and MAS) methods is minimising the number of messages sent. As documented by Provost (2000), in many applications, the cost of transferring large blocks of data may be prohibitive and result in inefficient implementations, not to mention security hazards. In this way, DDM algorithms and MAS may offer better solutions because they are designed to work in distributed environments while paying careful attention to the computing and communication resources.

The following is a brief overview of clustering and ARM using DDM methods:

- According to da Silva, Giannella, Bhargava, Kargupta, and Klusch (2005), efficiency-focused algorithms for distributed clustering can be divided into the following two subcategories: (i) methods requiring multiple rounds of message passing that require a significant amount of synchronisation (e.g., parallel K-means Zhao, Ma, & He, 2009, K-harmonic means Forman & Zhang, 2000), and (ii) methods that build local clustering models that are centralised and ensemble-based, transmitting them to a central site asynchronously and requiring only a single round of message passing and modest synchronisation requirements (e.g., DBSCAN (Januzaj, Kriegel, & Pfeifle, 2004), cluster ensembles in a centralised setting (Strehl & Ghosh, 2003)). These two subcategories usually work much better than their centralised counterparts in a distributed environment, which is well documented in the literature (da Silva et al., 2005).
- Survey (Zaki, 1999) discusses parallel and distributed ARM in DDM. According to the survey, researchers expect parallelism to relieve current ARM methods from the sequential bottleneck, providing scalability to massive data sets and improving response time. Achieving good performance on today's multiprocessor systems is no small feat. The main challenges include synchronisation and communication minimisation, workload balancing, finding good data layout and data decomposition, and disk I/O minimisation (which is especially important for ARM). The parallel design space spans three main components: the hardware platform, the type of parallelism, and the load-balancing strategy.
- A parallel mining algorithm called FPM, which is an enhancement of the FDM algorithm, is presented in Cheung, Lee, and Xiao (2002), which we previously proposed for distributed ARM (Cheung, Ng, Fu, & Fu, 1996). FPM requires fewer rounds of message exchanges than FDM and has a better response time in a parallel environment. The efficiency of FPM is attributed to the incorporation of two powerful candidate set-pruning techniques, distributed pruning and global pruning. The two techniques are sensitive to two data distribution characteristics, data skew and workload balance. To increase the efficiency of FPM, we developed methods of partitioning a database so that the partitions have high balance and skew.

The problem of design, implementation, and deployment of parallel and distributed clustering and ARM systems demands further research to achieve better efficiency parameters. For a broad overview of DDM, including clustering, ARM, basic statistics computation, Bayesian network learning, classification, and history, see surveys Kargupta and Sivakumar (2004) and Park et al. (2003). A variety of DDM algorithms, ARM, clustering, classification, preprocessing, systems issues in DDM related to security, architecture,

and topics in parallel data mining is presented in Kargupta and Chan (2000). Finally, survey (Zaki, 2000) discusses a broad spectrum of issues in DDM through a survey of distributed and parallel ARM and clustering.

4. Related work

As discussed in Section 3.1, MAS is designed for complex problem solving in distributed environments though the use of collaborative or competitive intelligent agents. In addition, DDM is a branch of DM concerned with distributed data, which also considers limited computing resources (Section 3.2.3). MAS and DDM involve application environments that deal with empirical analysis and data mining. This section presents some related work focusing on the interaction and integration of DDM and MAS using the AMII bidirectional approach discussed in Section 1.

The authors in Symeonidis, Kehagias, and Mitkas (2003) present an intelligent policy recommendation multi-agent system (IPRA), a MAS that introduces adaptive intelligence as an add-on for enterprise resource planning (ERP) software customisation (a recommendation engine). This MAS (and its add-on) takes advantage of knowledge gained through the use of DM techniques and incorporates it into the resulting policy. IPRA provides the ERP operator with useful customer/inventory/supplier recommendations based on Customer/Supplier Clustering and on Association Rule Extraction in item transactions. The authors also argue that the presented multi-agent architecture can be expanded to fulfill the needs of a distributed network of existing ERP systems. This can be achieved primarily by introducing mobility characteristics to the existing agent types. IPRA methodology relies on pattern recognition and fuzzy theory concepts. An automated evaluation procedure for the DM process was included as a subject for future work.

As discussed above, da Silva et al. (2005) analyses the connection between DDM algorithms (with a focus on distributed clustering) in the context of multi-agent-based problem-solving scenarios with challenges for clustering in sensor-network environments. This article also discusses confidentiality (privacy preservation) related to distributed data clustering systems assuming a P2P model, and it provides a high-level survey of DDM. Finally, the article presents a new algorithm for privacy-preserving density-based clustering, while pointing out that distributed DM algorithms offer a better solution for distributed environments, where data centralisation may be difficult because of limited bandwidth, privacy issues and/or the demand for response time. While offering a survey of DM literature on distributed and privacy-preserving clustering algorithms, the authors conclude that distributed clustering algorithms provide a reasonable and interesting class of choices for the next generation of MAS that may require the analysis of distributed data.

In Ali Albashiri, Coenen, and Leng (2009), the authors propose an extendible multi-agent data mining system (EMADS). EMADS is presented as an anarchic collection of persistent and autonomous KDD agents operating cooperatively across the internet. Individual agents have different functionalities. EMADS comprises many different agents, including data, user, task, mining and a number of 'house-keeping' agents. Users of EMADS may be data providers, DM algorithm contributors or miners of data. The provision of data and mining software is facilitated by a system of wrappers. Users wishing to obtain (for example) classifiers or collections of patterns do not need to have any knowledge of how any particular piece of DM software works or the location of the data to be used. EMADS is a hybrid P2P agent-based system comprising a collection of collaborating agents in a set of containers. It is implemented using the JADE framework (Bellifemine, Caire, Trucco, & Rimassa, 2008; Bellifemine, Caire, & Greenwood, 2007) with an agent management system (AMS) agent and a Directory Facilitator (DF) agent (this terminology is taken from the JADE framework, see Section 5). The authors describe the current functionality of EMADS as the limitation to classification and meta-ARM request styles (the details of this process can be found in Ali Albashiri, Coenen, Sandersonm, & Leng (2007)), and they cite the diversity of mining tasks that EMADS can address, given an appropriate mining software wrapper.

Using an agent-driven DM approach, (Zhou, Rao, & Lv, 2010) proposes a MADM model and implements DDM using multi-agent technology. To improve the performance of communication and reduce the pressure of the network bandwidth, the paper proposes a load-balancing agent and a task-prediction agent. An algorithm-analysis agent is also presented to improve the efficiency of DM agent in the model. Functionally, the model can be divided into three submodels: namely, data warehouse mining (DWM), load balancing and user. The architecture, on the other hand, has three layers: service, task-scheduling and user. The initial experiments were performed with twenty DM testing tasks from a web site, and the performance of the proposed DDM model was compared with traditional DDM models using mature tools (QUEST, MineSet and DBMiner). The results indicate that the efficiency of the traditional model has a uniform distribution (fixed algorithm), while the efficiency of the model increases as time passes. The authors argue, therefore, that the algorithm analysis agent has a self-learning ability, but according to the workflow presented, it needs to test all the available algorithms to set the weight for each algorithm (incur overhead) and replace the current algorithm with the best algorithm determined by testing (we do not consider this an intelligent self-learning feature). Additionally, the efficiency parameter used for the DM agent analyses the precision of the current model to the ideal mined results without citing which model functions in DM are being used with what efficiency attributes. This is problematic because efficiency in algorithms varies according to the DM model function (e.g., association rules use accuracy/confidence and coverage/support; see Section 2.1).

In the literature, there are a number of reports on applications of agent techniques to DM. First, there is a good deal of research on DDM systems, as cited in Section 3.2.3 (da Silva et al., 2005; Kargupta & Chan, 2000; Kargupta & Sivakumar, 2004; Park et al., 2003; Provost, 2000). As for MADM systems, one of the earliest references is the parallel data mining agents (PADMA) proposed by Kargupta and Hamzaoglu (1997). However, the focus of that article is a DDM system, and it only investigates the possibility of using a distributed agent-based architecture for text data mining. There are many references to MADM systems, especially in Ali Albashiri et al. (2009) and Zhou et al. (2010), which present interesting MADM system models. In Ali Albashiri et al. (2009), there is a dense study associated with the conceptual framework, while the current implementation of EMADS functionality is limited to classification with meta-ARM style, which requires an extension towards the diversity of mining tasks proposed. Beyond that, (Zhou et al., 2010) presents a generic architecture that may be useful in several domain applications, but the DM techniques and efficiency attributes require further work detailing performance results to be comparable to other approaches (e.g., what DM algorithms have been used with the efficiency parameters). As for (da Silva et al., 2005), the multi-agent architectures presented are not general enough to address the government auditing context or the cartel detection problem, in particular, because they lack both the structure to support different types of DM techniques and the coordination protocol for agent interactions. Finally, this article did not define an automated evaluation procedure for the DM process to be expanded to fulfill the needs of DDM systems.

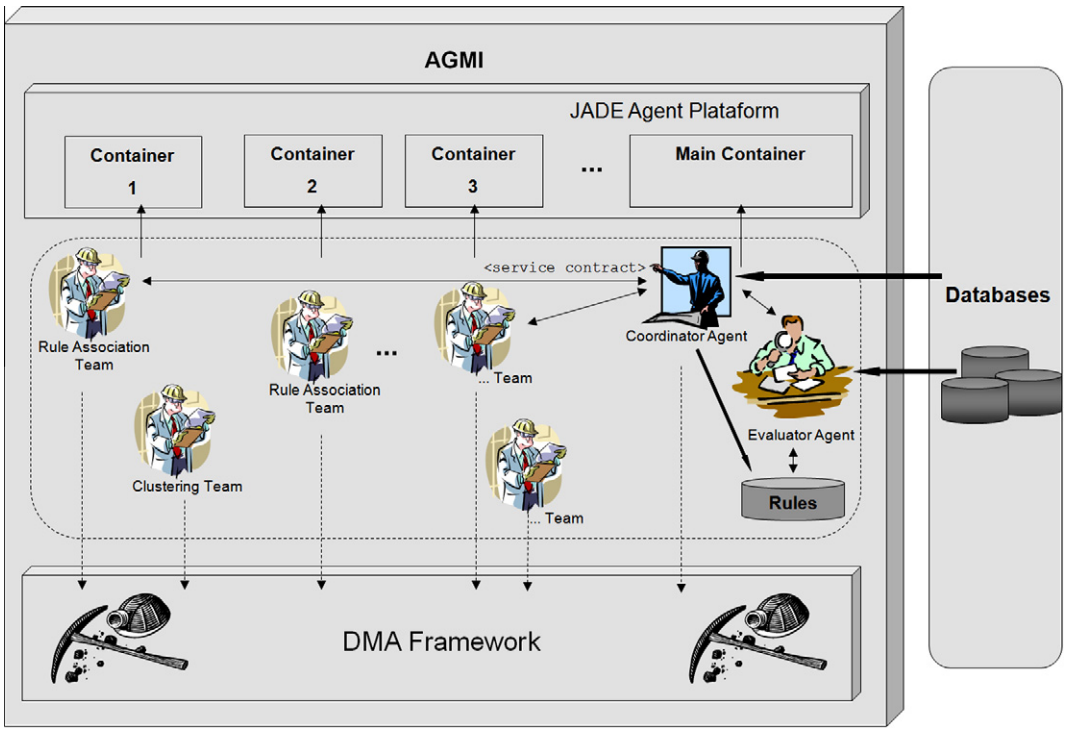

Fig. 1. AGMI architecture implemented in JADE.

In summary, as stated correctly by Gorodetsky, Karsaeyv, and Samoilov (2003), the main problem in MADM is not related to DM algorithms; instead, it remains the issue of finding appropriate mechanisms to allow agents to collaborate. Thus, the study of applied interaction and integration of MAS to DDM/KDD with the intention of solving the problems of dynamic collaboration by intelligent agents is essential and necessary.

5. The AGMI conceptual framework

In this section, AGMI, an agent-mining tool, is presented in its conceptual framework as a MADM system, applying discrete and dependent tasks to regulate, control and organise the algorithms to the KDD process. These functions are followed by the AGMI implementation aspects and operation cycle. Conceptually, AGMI is a MAS oriented to DM services that comprises a collection of collaborating agents grouped in specialised teams. As described in Section 1, AGMI uses an AMII approach that is useful to different domains because the team interaction used in our architecture is general.

All the DM services are provided by specialised agent teams that are able to negotiate their services through the supervisor agent of each team. Agents use a well-defined interaction protocol to announce and negotiate their services, known as the Contract Net Interaction Protocol standardised by the Foundation for Intelligent Physical Agents (FIPA)[5] (Foundation for Intelligent Physical Agents, 2002). To implement the contract net protocol a task list was defined where the tasks are added by the supervisor agent, that uses three different behaviors to control the task announcement, contracting and execution by the mining agent teams. At the beginning, the list is configured with the number of tasks equal to the number of clusters found, where tasks are removed from the list as they are delegated to be executed by the mining agents.

Additionally, there is no difficulty in extending the architecture to include other DM teams that may use specific DM techniques according to the requirements of the domain applications. In order to include a new DM team, the user must initialize a new Supervisor Agent setting its parameters concerning the number of mining agents of the team, maximum amount of memory available to that team and other related parameters.

Agents are distributed in a set of different containers and use terminology taken from the Java Agent Development framework (JADE)[6] (Bellifemine et al., 2007) in which AGMI is implemented (implementation details are discussed further in Section 5.1). The JADE agent platform provides various facilities to maintain the operation of AGMI, and the use of different containers allows the distribution of characteristics using different machines. In this way, starting the agent teams in different containers allows for the distribution of the work load among different hosts.

One of these containers is the main container, which holds an agent management system (AMS) and a directory facilitator (DF). The AMS agent is used to control the life cycles of other agents in the platform, and the DF agent provides an agent lookup service (yellow pages). Both the main container and the remaining containers can hold various MADM agents, allowing the main container to be located in Brasília and the other containers to be located worldwide.

Fig. 1 presents a view of the AGMI architecture implemented in JADE, showing the various categories of the agents and their interactions along with the databases and the DMA framework (detailed further in Section 5.2). The figure shows the main container that holds the coordinator agent, the evaluator agent (strategic), the AMS agent and the DF agent. The remaining containers hold the specialised teams (operational), the rule association team and the clustering team, each one with its supervisor agent (tactical).

[5] FIPA official site – http://www.fipa.org/.

[6] JADE official site – http://jade.tilab.com/.

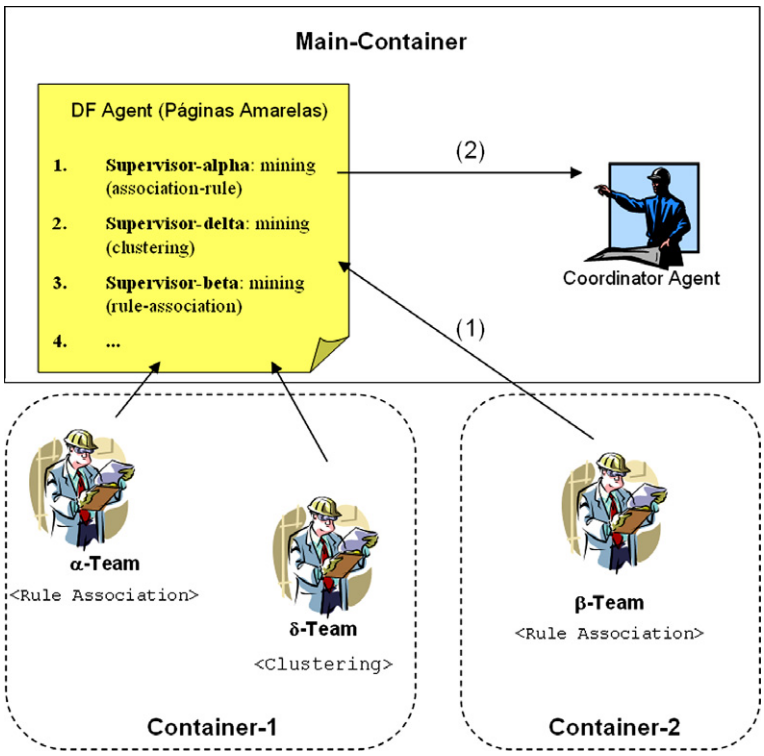

Fig. 2. Yellow page registration service.

As shown in Fig. 1, AGMI is composed of multiple collaborative agents that are grouped into work teams with the following three functional architectural levels:

- Strategic – this level includes the coordinator and evaluator agents because they have a global view of the system related to the interaction and control of the other agents. These agents are goal-directed and pro-active (as discussed in Wooldridge (2009) in Section 3.1). The coordinator agent is responsible for distributing tasks among the supervised teams using FIPA Contract Net Protocol because it has access to the databases, the results already produced, the work teams and the pending jobs. The evaluator agent has permission to read and save rules on the rule base, while the coordinator agent only saves rules on the base. The evaluator agent recovers the rules saved on the rule base to evaluate the quality of the rules produced by the KDD process, using a diverse LB of support and LB of confidence parameters.
- Tactical – this level includes the local mining supervisor of the mining agent teams. The supervisors use specific DM techniques, which are the DM services that are provided by the MADM system. They use FIPA Contract Net Protocol to distribute mining tasks to specialised mining agent teams. For example, the supervisor agent in a negotiation schematic publishes a rule association task and receives bids for work from ARM agents that can execute the service, enabling it to choose the agent team that has the best skills and resources to execute the task.
- Operational – this level includes the mining agent teams. Each mining agent executes one specific algorithm available in the DMA framework. According to the conceptual framework, one mining agent team may have different algorithms for the same mining technique, such as, for example, the clustering team having EM, K-means and Fuzzy C-means agents, as described in Section 3.2.1.

As noted in Fig. 1, the AGMI environment is composed of several databases. There is a rule base (knowledge base), as well as the datasets produced during the task executions. Agents in the strategic layer have a wider view of the environment and may access databases and the rule base, while team supervisors have a narrower view of the environment and with access only to the data that they will work with and the DM methods. These data and algorithms are visible to agents at the service contract with the coordinator agent. At the operational layer, each mining agent has access only to the dataset to which the DM algorithm will be applied.

5.1. The AGMI agents

As described in Section 5, AGMI is composed by a collection of collaborating agents grouped in specialised teams. All agents have the same basic lifecycle as defined in JADE framework: they are unknow until they are created, initiated, invoked to be active, and once active they can be suspended, waiting, on transit (move) or destroyed. AGMI has four types of agents and each agent has the following associated and specific tasks defined in its behavior model:

1. Coordinator agent – this agent controls the KDD process with established goals. It knows the basic tasks to be developed and can negotiate with local mining supervisor agents on the execution of the tasks. It can find agent teams in the JADE platform and contract them to execute tasks, preparing datasets in specific formats to pass to the supervisor agent. AGMI implemented prototype does not support rules-based goals, but it can be extended to support such feature.
2. Evaluator agent – this agent is responsible for evaluating the knowledge that is produced by the KDD process, using the LB of support and LB of confidence parameters and rule quality (Eq. 3). This agent keeps better rules and deletes poorly

performing rules at the rule base. Using heuristics, this agent can also create new rules to enhance the quality of the rules stored in the rule base. AGMI has only one evaluator agent which is activated by the coordinator agent when it receives the first result of a mining task executed by a DM agent team. The evaluator agent is kept alive to execute all the tasks planned by the coordinator agent.

3. Local mining supervisor agent – this agent coordinates the work of a DM agent team, which can use association rules, clustering, classification, and attribute selection. Each local supervisor has at least one agent to coordinate and negotiate tasks, and it sends work offers and receives bids for service. If all the agents on a team are busy, this agent refuses new service requests. It also passes the dataset that is received from the Coordinator Agent to the mining agents whenever the contract is accepted. The supervisor agents are responsible to create and organize DM agent teams, and they are kept alive during the entire lifecycle of the mining teams. They are the first agents created in AGMI, based on the user settings, and they die only when the coordinator agent finishes all the programed tasks.
4. Mining agent – these operational agents are responsible for executing DM algorithms for the prepared datasets and attempting to extract useful rules from them. These agents are normally computing intensive, and they consume most of the computational resources that are available on the platform. The mining agents are created to execute a mining algorithm and once they return the result they die.

5.2. The AGMI implementation

AGMI is implemented using the JADE framework (Bellifemine et al., 2007). JADE is FIPA-compliant middleware that enables the development of distributed applications based on the agent paradigm and is adequate to process large amounts of data with a DDM approach. JADE allows portability, which is assured by the use of Java, and defines an agent platform comprising a set of containers that may be distributed across a network. In the JADE platform, the main container holds a number of mandatory agent services, such as the AMS and DF agents, whose functionalities have been described in Section 5. JADE includes two main products, a FIPA-compliant agent platform and a package to develop Java agents. JADE also provides the implementation of FIPA agent communication language (ACL), a message-based protocol defined by FIPA.

AGMI allows agents to offer a DM service with autonomy and in an extensible way, making it possible to extend a prototype to insert different agents into the operational level and integrate the work of mining agent teams. Fig. 2 depicts the AGMI execution cycle that begins with the registration of the mining team services by the local mining supervisor (1) and the mining agents offering services through the local mining supervisor to the coordinator agent (2). For service registration, we used a DF agent that is responsible for the yellow pages service at the main container of the JADE framework.

Because AGMI is a MADM, one of the first implementation measures was to integrate the DM algorithms into the JADE platform. As a DM tool, we chose the Weka 3.6.1 Java Library (University Waikato, 2009), which had to be extended to allow the manipulation of the data structure by JADE agents. The extended Weka is called the Data Mining for Agents (DMA) framework, as presented in Fig. 1. Besides the Weka algorithms, the DMA framework offers other features making it possible to automate important tasks such as the creation of datasets to be used by the algorithms. DMA also uses an ontology associated with the data structure that results from the algorithms (ontology details are discussed further in Section 5.2.1). The ontology allows communication among the agents in a conceptual manner to evaluate the rule result from the KDD process, according to the defined RQ (Eq. (3)).

The current implementation of AGMI does not set the environmental resources automatically, so the user must define the agent requirements, e.g., memory capacity, number of agents on each team, DM algorithms, hosts and other attributes. Additionally, the current implementation of AGMI does not have a graphical user interface. Thus, in the test case presented in Section 6, a configuration file was used to specify the main tasks of the coordinator agent (as cited in Section 5.1 it is not a rule-based scheme). In this file, we defined the database tables of procurements, the attributes used for discovering the association rules and the clustering algorithm (EM). The algorithm settings, such as the minimum threshold of support and confidence, can be changed in the configuration file, although the prototype contains default values. As we explained in Section 2.1, the test case for the discovery of cartel formation during the procurement process began with the coordinator agent contracting a team to apply the Apriori algorithm to the dataset of companies. Additionally, a clustering team is contracted to apply the EM algorithm to cluster the dataset into regions using the DMA framework.

5.2.1. The AGMI interactive features

The interaction protocol used in the AGMI to contract service among the coordinator agent, the local mining supervisor agent and the mining agent is the FIPA Contract Net Protocol, which is available in the JADE framework. The contract process generally begins when the coordinator agent finds possible service providers that are registered in the yellow pages (DF agent of the JADE main container) and notifies them of the available services to be contracted with a Call for Proposals (CFP). The local mining supervisor agents that are available to work send bids to the coordinator agent, and the coordinator agent, using a "greedy" strategy, analyses the offers and contracts the best bids according to each team's resources. The accepted bidder is notified by the coordinator agent.

Fig. 3 illustrates the negotiation among a coordinator agent and two association rule teams, Alpha and Beta, which are represented by their local mining supervisors. Note that the messages in lines 30 and 31 are the CFP sent from the coordinator agent to both Alpha and Beta teams (the first two messages from top to bottom). The messages in lines 32 and 33 represent the bids send from the Alpha team and Beta team to the coordinator agent. The message in line 34 is the proposal acceptance to the bid sent from the Alpha team, while line 35 represents the rejection of the proposal sent by the Beta team. The message in line 36 is the information that the Alpha team will accept the execution of the job. Also note that the Alpha team is in Container 1 and uses the Apriori algorithm. Thus, the bid sent by each team has different resources; the coordinator agent is able to select the more competitive bid based on the available resources and performance rate.

Note that DM techniques such as the association rule are memory intensive. As in our test case, ARM is the most frequently used activity in the process. Thus, the terms of negotiation of the Contract Net Protocol was based on the amount of available memory per each agent of the team. The local mining supervisor uses Eq. (2) to calculate the value of the memory available to perform the required service. Based on that value, the coordinator agent selects the best bid from all the bids that were sent to it (using a greedy strategy).

$$P = Mem_{max} - \frac{Mem_{max}}{|AgM|} \times |AgM_{buzy}| \tag{2}$$

where

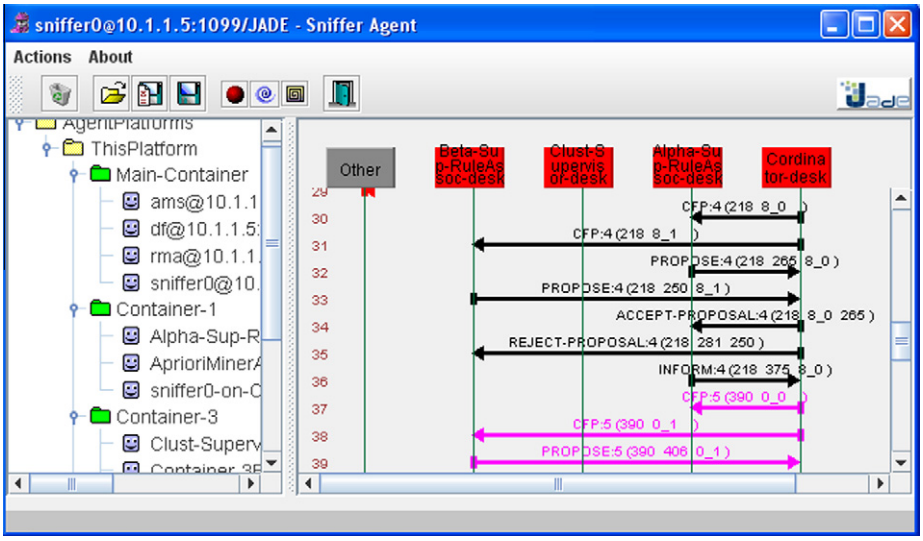

Fig. 3. Contract net example using the JADE *Sniffer* agent.

Mem_{max}	maximum amount of memory available in JVM;
AgM	data-mining agent set for a specific team;
AgM_{buzy}	data-mining agent set that is current buzy in the team.

Because agents are autonomous entities, the communication protocol is the instrument that achieves synchrony between the agents to reach the ultimate goal of knowledge discovery. AGMI uses the FIPA ACL communication language available in the JADE framework (Bellifemine et al., 2008; Bellifemine et al., 2007).

According to Bellifemine et al. (2007), the JADE platform uses the following three important element types, derived from the FIPA ACL language, to construct common terminology in the agent communication protocol: concepts, predicates and agent actions. These three elements must be defined in an ontology. In AGMI, we have defined an ontology with these elements using the plugin OntologyBeanGenerator, available at Protégé for integration into the JADE framework.[7] Fig. 4 presents the defined ontology used in AGMI, which is used in the form of a structured vocabulary to allow communication among different types of agents.

The ontology can be extended to different domain applications. The concepts related to action requisitions for DM techniques should be inherited from the main mining task *DoMiningTask*. DM actions should be inherited from the principal action *AgentAction*, and the different types of DM results should be inherited from the principal result *MiningResult*. The knowledge structures for representing knowledge rules, DM models, and clusters should be inherited from the principal concept *StructuredKnowledge*.

5.3. The evaluator agent: discovered knowledge improvement

Because the total number of rules returned from the Apriori algorithm varied greatly, we defined a rule evaluation function to select the best rules for cartel practices in the procurement process using the auditing experts' knowledge to preview and detect cartel-like scenarios. The rule quality (RQ) equation is presented in Silva and Ralha (2010) and recalled here in Eq. (3).

$$RQ = 100 \cdot \frac{V(C)}{Sup. \times Inst.} \tag{3}$$

where

$Sup.$	rule support value;
$Inst.$	total number of instances in the database;
C	set of companies in the rule;
$V(C)$	number of victories that a company of set C wins in the procurement process when the entire group has participated.

In Silva and Ralha (2010), more explanations are provided for the RQ equation, but the RQ broadly represents the probability of the suspicious group winning a procurement process. The higher the RQ, the more strongly the group is suspected of cartel practice. The top ten rules that were evaluated using Eq. (3), i.e., the rules obtained from the association of clustering and ARM, presented an increase of 100% in the medium value of RQ when compared to the experiments that only used ARM (for more details on the rules and the analysis see (Silva & Ralha, 2010)).

To improve the KDD results, the evaluator agent defined in AGMI has several important tasks. Every time new rules are saved in the rule database (rules in Fig. 1), the evaluator agent is notified by the coordinator agent. The evaluator agent applies the quality function RQ (Eq. (3)) to the rule whenever possible because it does not make sense to apply the rule quality function to a cluster model. The evaluator agent also executes the removal from the rule database of rules that are semantically equal: for example, the rule A, B, C → D with 100% confidence has the same effect of rule A, B → C, D and A → B, C, D, if they have 100% confidence. In our experiments, we worked with a minimum confidence value of 90%.

Although our approach is quite generic the evaluation phase requires domain-specific knowledge from the auditor experts. This knowledge is going to be used by the evaluator agent. In this way, the evaluator agent has the autonomy to decide whether a rule can be improved or not through the use of heuristics. For the specific test case of cartel formation in government procurement, we have defined heuristics by consulting auditor experts. These heuristics are used by the AGMI evaluator agent to improve the association rule results. When evaluating whether a rule can be improved through the use of heuristics, the Evaluator Agent will determine whether each rule can be applied to a specific region cluster formed by some FUs or to a region formed by all the FUs. However, it is possible that a group of companies that form cartels have more success in a specific region using techniques such as, for example, grouping the larger number of contracts into one FU.

[7] http://protege.cim3.net/cgi-bin/wiki.pl?OntologyBeanGenerator.

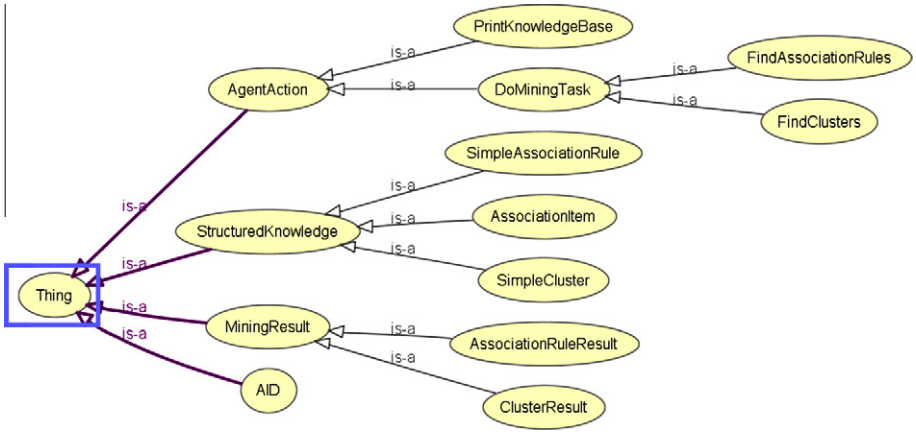

Fig. 4. The AGMI ontology.

During the rule evaluation, the evaluator agent will count how many times the group referred by that rule has won in each FU in which that rule was applied. Using this information, the evaluator agent classifies the selected number of FUs in which the greater number of wins has occurred. If this number is larger than the minimum value of the RQ (Eq. (3)), the evaluator agent applies the rule to the FU that is selected, computing local support and the new value of the rule quality and comparing whether the rule, when applied locally, is better or worse than the original rule. If the rule is better, the original is replaced by the new rule. When changing the rule, the support and confidence of the new rule must be equal to or greater than the limits established by the system.

6. Experimental evaluation

In this section, we illustrate AGMI's capacity to discover knowledge and detect cartel formation with a test case in the domain of the Brazilian Government Procurement. We conducted many experiments to test the AGMI prototype, but we will focus on the results of the performance evaluation compared to traditional DM techniques using the same ARM Apriori and EM clustering algorithms in the Weka framework, version 3.6.1 (University Waikato, 2009) (presented in Section 2.1). We will also evaluate the quality of the rules returned from the KDD process using the RQ (Eq. (3)) and the heuristic strategy implemented by the evaluator agent.

To test the DDM aspects of AGMI in a distributed infrastructure, the test case involved heterogeneous computers distributed in two hosts:

1. Host A – Intel Core 2, 2.40 GHz, 2.00 GB RAM; and
2. Host B – Intel Pentium Dual, 1.86 GHz, 1.99 GB RAM.

The type and number of agents varied according to the experiment, but the following four types of agents presented in Section 5.1 were used with their specific associated roles: coordinator, evaluator, local mining supervisor and mining agents, divided into different teams. The mining agent teams were called Alpha, Beta and Gamma and were usually composed of one or two agents and a supervisor. The algorithms used in this test case were Apriori for the ARM technique and EM for the clustering technique, which are both available in the DMA framework.

Table 1 of Section 2.1 shows the dataset used in the test case experiments. We used the same basic attributes and datasets presented in Tables 2 and 3 of Section 2.1 to compare AGMI performance with the results using the traditional DM approach; however, the entire process of AGMI, including all the data preparations, the mining algorithm execution and the rule evaluation, was automatically performed by the agents and the algorithm in parallel. For this performance evaluation, the LB of support for the main dataset was defined at 0.9% and the LB of confidence was 90%. For the datasets that were created from the cluster model, we configured the association rule agents to execute the algorithm to obtain rules with at least nine occurrences in the dataset. The EM clustering algorithm used in this experiment resulted in seven clusters (for more details about the clustering used see (Silva, 2011)).

Table 4 presents the agent distribution and Table 5 presents the execution results of the DM approach compared to the AGMI approach. As discussed in Section 5.2.1, the ARM algorithm is memory intensive; therefore, we have increased the number of agents in the team in Test 2 (Table 5), which has certainly affected the execution time. It is necessary to note that the percentage improvement in the evaluation (44%) is merely illustrative to some extent because we did not consider aspects like the operational system, the number of live processes during the experiments' execution time (multitask system) or the communication among agents, all of which should be considered in distributed systems performance analyses.

6.1. Autonomous evaluator agent

When evaluating the quality of the rules returned from the KDD process, an autonomous mechanism was introduced in the evaluator agent behavior, which is presented in Section 5.3. This mechanism analyses all the rules that are produced and searches for the element of the cluster that has the most victories in the procurement processes of the group. We defined a constant value as the lower threshold of the group victories in the procurement processes ($c = 4$ in the tests) to demonstrate this. If any cluster element (Brazilian states or FU) holds a number of group victories greater than the constant number defined (c), the evaluator agent

Table 4
Agent distribution among hosts.

	Host A	Host B
Coordinator agent	X	
Evaluator agent	X	
Assoc. rules team		
α-Team	X	
Assoc. rules team		
β-Team		X
Clustering team	X	

Table 5
Traditional DM approach and AGMI performance comparison.

	Coordinator agent (number)	Evaluator agent (number)	Assoc. rule team no. agents	Clustering team no. agents	Time	Improvement (%)
Traditional	–	–	1	1	01:15:00	–
Test 1	1	1	1	1	01:01:07	18.7
Test 2	1	1	2	1	00:42:19	44.0

calculates the quality of the rule that is applied locally to select a better scope in which to apply the rule.

For example, let us consider a rule discovered in a cluster composed of three Brazilian states, MG, RJ and SP. If the state of SP concentrates the largest number of victories of a group pointed to by a rule, then the evaluator agent will analyse whether the rule would be better for that state, considering its frequency (local support). This construct means that a cartel that has acted in three states might be more successful in a specific state. Furthermore, it would be interesting to discover in which states a cartel was successful. The purpose of this heuristic is to attempt an improvement in rule quality values in local applications. To avoid the need for multiple database accesses, we configured the agent to analyse only rules that would have local support greater than or equal to the lower threshold of the global support value (c) defined in the system. In our relevant tests, $c = 4$.

Table 4 presents the agent distribution for this experiment. The mining teams for this experiment are made up of one mining agent and its supervisor, and Table 6 presents the results of applying this autonomous mechanism. Despite our use of the same settings in Test 2 of Table 5, the execution time increased (00:42:19 to 01:01:23) because of the database accesses performed by the evaluator agents when applying their heuristics. Table 6 presents 341 rules selected by the evaluator agent when applying heuristics. Among these rules, 191 could be improved. The rules that have a better improvement (the top ten rules) had an average improvement of 56.1%, proving the importance of the heuristics application.

To validate the use of the evaluator agent in relation to the increase in agents in the teams of the AGMI, we performed three tests that change the number of mining agents of the rule association team. The agent distribution is presented in Table 12. In our architecture, each team is composed of the Supervisor Agent and a certain number of mining agents.

Table 7 presents the three test results with their execution times. Note that there was a significant gain with respect to the execution time when the two mining agents were in the ?-Team (ARM Team). This action saved approximately 30% of the execution time. When we increased the ?-Team (ARM Team), only a few seconds were saved, an insignificant gain. This result can be explained by considering the following two tasks that were performed by the agents, which take several minutes:

- The ARM technique over the complete database. This task is the first task negotiated by the coordinator. The agent that performs this task takes almost all the process execution time. When the agent finishes its work, the other agents have usually already finished the clustering task and the other association rule tasks.
- The database access performed by the evaluator agent. When there is no rule in the database, the evaluator agent must wait for the initial results of the mining agents to begin its analysis and is, therefore, always the last to finish its processing.

Fig. 5 presents the runtime improvements applying the agent integration approach with AGMI. Note that Test 0 used only a simple automation of the DM tasks in a distributed method without the evaluator agent's service contracting and rule improvement (00:75:00). Test 1, with AGMI in the experiments, tested the parallel execution with the evaluator agent (01:01:23, Table 7). In Test 2 and 3, we used more mining agents in the association rule teams to accelerate the processing of the datasets that were generated by the clusters found through the clustering mining team (00:42:53 and 00:42:19, Table 7). Therefore, considering the runtime execution, our solution set performed well using AGMI on the problem of cartel formation in the procurement process.

Table 6
Evaluator agent's rule improving mechanism.

Informations	Results
Execution time	01:01:23
Selected rules	341
Improved rules	191
Average rate of improvement	11.5%
Average rate of improvement (top 10)	56.1%

Table 7
Tests including other association rule teams.

	α-Team (n. of mining agents)	β-Team (n. of mining agents)	Execution time
Test 1	1	1	01:01:23
Test 2	2	1	00:42:53
Test 3	2	2	00:42:19

We conducted many experiments to test the AGMI prototype and varied the number of agents in the mining teams. To test the quality of the rules returned from the KDD process in the specific domain of government procurement, we applied the EM clustering algorithms of a DMA framework in two different clustering teams. Although our approach originally introduced the clustering technique only to divide the solution space for data processing, the clusters discovered revealed the trends of company participation in public procurements in Brazil. Because the clustering algorithm is sensitive to data order, the EM algorithm provided two models that showed that almost all of the clusters were formed by states that shared geographical borders. This result proves companies' preference for acting in regional markets, usually next to their headquarters. The first cluster model presented ten clusters (as described in Section 2.1) and the second one, seven clusters. Therefore, we tested the parallel execution of three ARM teams together with two clustering teams to determine whether the quality of the rules using RQ (Eq. (3)) would increase. The results were not encouraging because the RQ raised only 4.24% (58.48% to 56.10%) when using the second cluster model (seven clusters) and 7.11% (58.48% to 54.60%) using the first cluster model (ten clusters). In (Silva, 2011), all the experiments are described.

6.2. Knowledge discovery

The AGMI experiments presented many rules, but some of these rules could be improved by the autonomy of the evaluator agent's

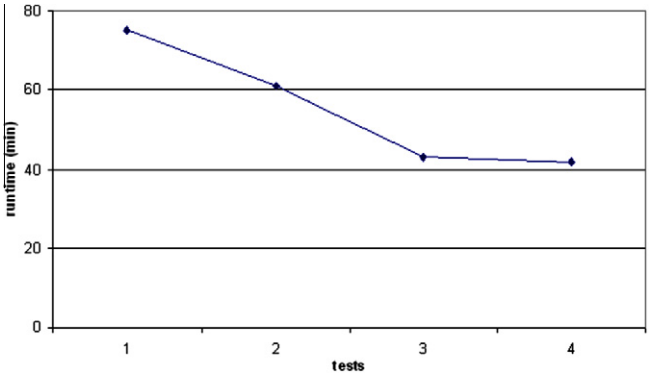

Fig. 5. Evolution of runtime improvement using agents.

performance. As one example of a rule improved by the evaluator agent, we offer the following:

- The rule pointed to two companies with 133 procurement processes in common. However, the RQ of this rule was not very good because it considered the number of victories together. When the Evaluator Agent applied this rule in the state of Mato Grosso (MT), the RQ improved 2.71 times because all the group's victories were in that state. Therefore, the heuristic of the evaluator agent proved to be a good mechanism to improve the rules for the problem of cartel formation.

We present three of the top ten rules that were indicated by AGMI to illustrate the knowledge discovery process:

- In nine different public procurements performed in the same state for only one government agency, one rule has pointed to the participation of two specific companies. In spite of the fact that both companies participated in all the procurement processes, only one of the companies won all the procurements. When we analysed the history of the losing company, we found that it had participated only in those nine procurement processes which were won by the suspected cartel, evidence of a possible simulation of competition to disguise cartel formation. The company was probably created only to simulate an artificial competition in public procurements, in which competition is mandatory.
- In 2006, a group made up of three companies participated in nine procurement processes. Seven of these processes were defined by one government agency in the state of Rio Grande do Norte (RN), and only one company of the group won all of these processes. When we analysed the history of participation in procurements of the other companies of the group, we realised that they were not used to win the processes. One of them won only once, and the other never won. The total number of contracts that were won by the winning company of the group was 13. In the expert's analysis, this scenario could represent a simulation of competition.
- Another group made up of four companies was suggested by a rule that indicated nine procurement processes in common. This rule was applied to the region that was formed by the states of Paraná (PR) and Rio Grande do Sul (RS). In this group, two companies stood out by winning eight processes. One of the companies won six processes and the other company won two processes. Analysing their historical victories in procurement processes, we found that the first company had won only three other processes in the entire database, and the other company had won two other processes. The other two companies indicated by the rule are large companies, which often participate in the procurement processes. According to the experts, these companies cannot yet be considered a cartel; however, it can be taken as an alert to watch the forthcoming actions of those two companies.

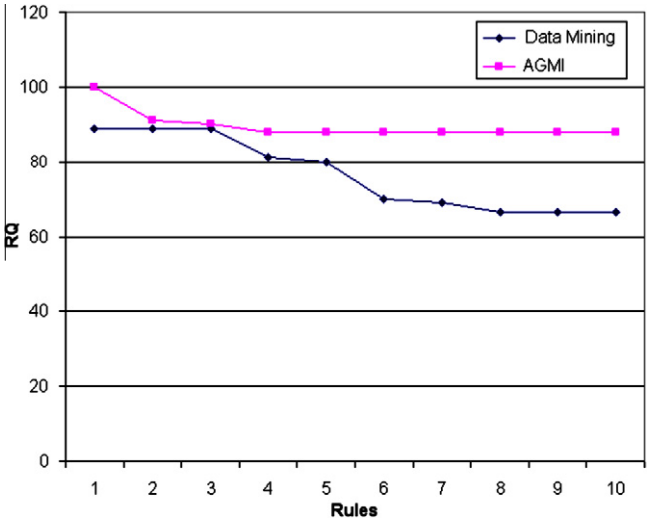

Fig. 6. *RQ* comparisons between AGMI and DM.

These presented rules demonstrate the potential of our approach to assist in the process of cartel formation in procurement processes. Fig. 6 presents the improvement of RQ, comparing AGMI to the previous DM techniques approach. Note that the comparison is made within the top ten rules in both approaches, in which AGMI presents RQ as always higher than 80%, with its best rule at 100%, and the RQ average at approximately 90%, while the previous DM approach varies from 70 to 80%.

7. Discussion

As described in Section 4, the main problem in MADM is not related to DM algorithms but to finding appropriate mechanisms to allow agents to collaborate. Therefore, study of the application of interaction and integration of MAS with DDM/KDD to solve the problems of dynamic collaboration by intelligent agents is essential and requires improvement. Additionally, the specific domain covered by this research work, cartel detection in government procurement processes, is an important corruption issue that is challenging for researchers to fight. We were unable to find any work that can be adequately applied to this domain using an AMII approach.

While MAS has great potential abilities in DDM because of its distribution, openness, and adaptability and the collaboration, reactivity, communication, and self-learning between agents, defining an appropriate mechanism for an agent collaborative problem while making the best use of distributed and intelligent resources is not a trivial task. Although DM became a well-established technology, DDM is a process of discovering unanticipated knowledge from logically or physically distributed databases using distributed computing technology, which faces many complex issues, such as large data resources, complex data elements, data transmission and so on.

Thus, this paper presents AGMI, an agent-mining tool that combines MAS and DDM fields of study using an AMII approach. AGMI is presented with a test case in the domain of the Brazilian government procurement to detect cartel formation, which tackles an

important real-world problem in what we believe is an interesting way. Considering the possibility to apply AGMI to different sets of entities other than sets of FUs, the developed prototype is ready to work with any set of entities to search for clusters. In this paper we presented a test case with clusters of FUs, but we could have worked with any other entities such as public agencies, cities, government branches, etc. We have chosen FUs because the biding processes of our databases were related to purchases of only one kind of service. In this specific case, companies tend to focus in all government agencies in one region. We have tested the prototype to cluster other entities with the same database, but the results were not significant.

Many improvements are necessary related to the actual implementation, considering the best usage of distributed environments and paying careful attention to the computing and communication resources, but we believe the definition of the AGMI conceptual framework and its elements can be useful in approaching complex problems in different domains.

Other distributed computing infrastructures, such as grid and cloud computing, which enables coordinated resource sharing within dynamic organisations consisting of individuals, institutions, and resources, can be helpful to DDM, in which DM algorithms and knowledge discovery processes are both computation- and data-intensive. However, MAS' major advantages, such as the decentralised control, robustness, simple extensibility, and sharing of expertise and resources are very important characteristics for use in this scenario. Therefore, the distributed infrastructure fields of study can be used together with AGMI to extend the distributed and parallel computing paradigms, allowing resource negotiation and dynamical allocation, heterogeneity, open protocols, and services, and making the best use of the agents' four features: reactivity, autonomy, interaction and initiative. The agent's autonomy characteristic can form the core of solutions for larger problems in the real world.

8. Conclusions

Apart from being in charge of inspecting and detecting fraud in the use of federal public funds, the CGU is also responsible for developing mechanisms to prevent corruption. In this direction, the CGU must act proactively to develop a means of preventing the occurrence of corruption. In this context, cartel formation in procurement processes is an important issue of focus. Therefore, in this article we present AGMI, an agent-mining tool that integrates MAS into DDM techniques, and applied it to the Brazilian government procurement domain.

AGMI includes the DMA (Weka-based) and JADE frameworks with three architectural levels (strategic, tactical and operational) and four different types of agents (coordinator, evaluator, local mining supervisor and mining). AGMI was defined to apply different DM techniques using a collaborative approach of interaction among agents to work over a distributed environment, with an integrated, intelligent perspective that primarily intends to improve the knowledge discovery process.

We tested AGMI as a possible solution to the problem of detecting cartels in public procurements. Many experiments were performed, and the results are presented in Section 6, showing that AGMI is potentially applicable to this problem. The experimental results presented in Figs. 5 and 6 proved that AGMI performed well with respect to run-time performance and the quality of the rules. The evaluator agent also proved its autonomous aspects by improving the quality of 191 rules through the use of heuristics, where the average rate of improvement of the top ten rules was 56.1% (Table 6). A comparison of the AGMI results with the previous DM techniques approach using rule quality shows that AGMI presents RQ that is always higher than 80%, with the best rule at 100% and the average RQ at approximately 90% (Fig. 6).

Several association rules that could indicate evidence for cartel actions in public procurements were discovered, as well as some rules that suggest fraudulent simulations of competition. The findings related to cartel detection were presented to the government official and some of the groups found by the prototype were considered potential cartel in its first analysis. However, in order to proceed with the lawsuit they must gather more consistent evidences, like overcharges in the contracted services, familiar or social bonds between the owners of the companies which make up the suspicious groups. In summary, the knowledge discovery presented in Section 6.2 may assist auditors who work in the Brazilian government to detect and prevent the occurrence of cartels during procurement processes, a very important and prosecutable corruption issue.

In future work, we will study other ways of automating the attribute selection and maintaining reduced runtime processing. Toward this goal, we are studying other mechanisms that could improve agent behavior, specifically assigning different degrees of autonomy to them that are associated with other distributed computing infrastructures (e.g., grid, cloud). We also intend to enrich the knowledge discovery process using different heuristics and applying AGMI to other auditing databases.

The data mining-driven agent approach described in Section 1 can be used because using the best rules, the knowledge discovered can provide the agents with the intelligence to mine databases using the experts' validated rules. In this way, fraud detection can be automated. Since 2002, the Brazilian government has adopted an Internet-based reverse auction and all companies that participate of these e-procurements have their bids recorded in ComprasNet databases. As a future work, AGMI could be adapted to analyze on the fly bids and companies in running procurements during the purchase processes. Therewith, any suspicious bid could be detected by AGMI which could send messages to the procurement managers and also to anti-corruption agencies such as the Office of the Comptroller General and the Council for Economic Defense (CADE). Additionally, new AI techniques could be used to enrich the integrated intelligence of our agents, as the use of machine learning algorithms, advancing towards the next-generation MADM systems.

Acknowledgements

C.V.S. Silva thanks the Dean of Research and Postgraduate Studies of the Brasília University of Brazil for grant support to present (Silva & Ralha, 2010) at the II Workshop of Applied Computing in Electronic Government (Computação Aplicada em Governo Eletrônico – WCGE), Data Mining Section of the XXX Congress of the Brazilian Computer Society (Congresso da Sociedade Brasileira de Computação – SBC), organised by Viviane Malheiros (Serpro) and Wagner Meira (UFMG).

References

Ali Albashiri, K., Coenen, F., & Leng, P. (2009). Emads: An extendible multi-agent data miner. *Knowledge-Based Systems*, 0950-7051, *22*, 523–528<http://dl.acm.org/citation.cfm?id=1613329.1613367>.

Ali Albashiri, K., Coenen, F., Sandersonm, R., & Leng, Paul H. (2007). Frequent set meta mining: Towards multi-agent data mining. In *research and development in intelligent systems XXIV, Proceedings of AI-2007, the twenty-seventh SGAI international conference on innovative techniques and applications of artificial intelligence, Cambridge, UK* (pp. 139–151).

Bellifemine, L. F., Caire, G., & Greenwood, D. (2007). *Developing multi-agent systems with JADE*. Wiley. ASIN 0470057475, <http://www.bibsonomy.org/bibtex/20dd9082032ee3c7fada23a14ff0f61e0/neilernst>.

Bellifemine, F., Caire, G., Trucco, T., & Rimassa, G. (2008). Jade programmers guide.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Norwell, MA, USA: Kluwer Academic Publishers. ISBN 0306406713.

Bhamra, G. S., Verma, A. K., & Patel, R. B. (2011). Agent enriched distributed association rules mining: A review. In *Agents and data mining interaction – 7th international workshop on agents and data mining interation, ADMI 2011, Taipei, Taiwan, May 2–6, 2011*. Revised selected papers (pp. 30–45).

Borman, S. (2004). The expectation maximization algorithm: A short tutorial. <http://www.seanborman.com/publications/EM_algorithm.pdf>. Accessed September, 2011.

Cao, L. (2009). Introduction to agent mining interaction and integration. In L. Cao (Ed.), *Data mining and multi-agent integration*. US: Springer.

Chaimontree, S., Atkinson, K., & Coenen, F. (2011). A multi-agent based approach to clustering: Harnessing the power of agents. In *Agents and data mining interaction – 7th international workshop on agents and data mining interation, ADMI 2011, Taipei, Taiwan, May 2–6, 2011* (pp. 16–29). Revised selected papers.

Chatzidimitriou, K., Chrysopoulos, A., Symeonidis, A. & Mitkas, P. (2011). Enhancing agent intelligence through evolving reservoir networks for predictions in power stock markets. In *Agents and data mining interaction – 7th international workshop on agents and data mining interation, ADMI 2011, Taipei, Taiwan, May 2–6, 2011* (pp. 228–247). Revised selected papers.

Cheung, D. W., Lee, S. D., & Xiao, Yongqiao (2002). Effect of data skewness and workload balance in parallel data mining. *IEEE Transactions on Knowledge and Data Engineering*, 1041-4347, *14*(3), 498–514.

Cheung, D. W., Ng, V. T., Fu, A. W., & Fu, Yongjian (1996). Efficient mining of association rules in distributed databases. *IEEE Transactions on Knowledge and Data Engineering*, 1041-4347, *8*(6), 911–922.

Coenen, F. (2003). Lucs-kdd (Liverpool University computer science – knowledge discovery in data) dn (discretization/normalisation) software. Address = Department of Computer Science, The University of Liverpool, UK. <http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS_KDD_DN/>.

Congiusta, A., Talia, D., & Trunfio, P. (2008). Service-oriented middleware for distributed data mining on the grid. *Journal of Parallel Distributed Computing, 68*(1), 3–15.

da Silva, J. C., Giannella, C., Bhargava, R., Kargupta, H., & Klusch, M. (2005). Distributed data mining and agents. *Engineering Applications of Artificial Intelligence*, 0952-1976, *18*(7), 791–807<http://www.sciencedirect.com/science/article/pii/S095219760500076X> .

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B, 39*(1), 1–38.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics, 3*(3), 32–57. http://dx.doi.org/10.1080/01969727308546046.

Forman, G., & Zhang, B. (2000). Distributed data clustering can be efficient and exact. *SIGKDD Explorations, 2*(2), 34–38.

Foundation for Intelligent Physical Agents. FIPA contract net interaction protocol specification, December 2002. <http://www.fipa.org/specs/fipa00029/SC00029H.html>.

Gorodetsky, V., Karsaeyv, O., & Samoilov, V. (2003). Multi-agent technology for distributed data mining and classification. *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology, IAT '03*. 0-7695-1931-8 (pp. 438). Washington, DC, USA: IEEE Computer Society<http://dl.acm.org/citation.cfm?id=946245.946383> .

Hand, D. J., Smyth, P., & Mannila, H. (2001). *Principles of data mining*. Cambridge, MA, USA: MIT Press. ISBN 0-262-08290-X.

Han, J., & Kamber, M. (2005). *Data mining: Concepts and techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.. ISBN 1558609016.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction* (2nd ed.). Springer. ISBN 978-0-387-84857-0.

Januzaj, E., Kriegel, H. -P. & Pfeifle, M. (2004). Dbdc: Density based distributed clustering. In *advances in database technology – EDBT 2004, 9th International Conference on Extending Database Technology, Greece* (pp. 88–105).

Johnson, S. (1967). Hierarchical clustering schemes. *Psychometrika, 32*(3), 241–254<http://ideas.repec.org/a/spr/psycho/v32y1967i3p241-254.htm> .

Kargupta, H., & Chan, P. (Eds.). (2000). *Advances in distributed and parallel knowledge discovery*. Cambridge, MA, USA: MIT Press. ISBN 0262611554.

Kargupta, H., & Hamzaoglu, I. (1997). Scalable, distributed data mining using an agent based architecture. *Proceedings of Knowledge Discovery And Data Mining* (1), 211–214<http://www.osti.gov/energycitations/product.biblio.jsp?osti_id=50149> .

Kargupta, H., & Sivakumar, K. (2004). Existential pleasures of distributed data mining. *Data Mining: Next Generation Challenges and Future Directions*, 1–25.

Kaur, P., Goyal, M. L., & Lu, J. (2011). Pricing analysis in online auctions using clustering and regression tree approach. In *Agents and Data Mining Interaction – 7th International Workshop on Agents and Data Mining Interation, ADMI 2011, Taipei, Taiwan, May 2–6, 2011* (pp. 248–257). Revised selected papers.

Luger, G. F. (2002). *Artificial intelligence: Structures and strategies for complex problem solving* (4th ed.). USA: Addison-Wesley. ISBN 0-201-64866-0.

Luo, J., Wang, M., Hu, J., & Shi, Z. (2007). Distributed data mining on agent grid: Issues, platform and development toolkit. *Future Generation Computer Systems, 23*, 61–68. ISSN 0167-739X. URL <http://dl.acm.org/citation.cfm?id=1276047.1276055>.

Meng, A., Ye, L., Roy, D., & Padilla, P. (2007). Genetic algorithm based multi-agent system applied to test generation. *Computers and Education, 49*(4), 1205–1223. ISSN 0360-1315.

Park, B.-H., & Kargupta, H. (2003). Distributed data mining: Algorithms, systems, and applications. In N. Ye (Ed.), *The handbook of data mining* (pp. 341–358). Lawrence Erlbaum Associates.

Provost, F. (2000). Distributed data mining: Scaling up and beyond. In K. Sivakumar, H. Kargupta, A. Joshi, & Y. Yesha (Eds.), *In advances in distributed and parallel knowledge discovery* (pp. 3–27). MIT/AAAI Press.

Rakesh, A., & Ramakrishnan, S. (1994). Fast algorithms for mining association rules in large databases. In *VLDB '94: Proceedings of the 20th international conference on very large data bases* (pp. 487–499). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.. ISBN 1-55860-153-8.

Ralha, C. G. (2009). Towards the integration of multiagent applications and data mining. In Longbing Cao (Ed.), *Data mining and multi-agent integration* (pp. 37–46). US: Springer. http://dx.doi.org/10.1007/978-1-4419-0522-2_2.

Russell, S. J., & Norvig, P. (2010). *Artificial intelligence – a modern approach* (3rd international ed.). 978-0-13-207148-2. Pearson Education.

Silva, C. V. S. (2011). Agentes de mineração e sua aplicação no domínio de auditoria governamental. Digital Repository site: <http://monografias.cic.unb.br/dspace/bitstream/123456789/318/1/DISSERTACAO_CARLOS_VINICIUS.pdf>.

Silva, C. V. S., & Ralha, C. G. (2010). Utilização de técnicas de mineração de dados como auxílio na detecção de cartéis em licitações. In *XXX Congresso da Sociedade Brasileira de Computação (SBC), II Workshop de Computação Aplicada em Governo Eletrônico (WCGE)*. ISSN 2175-2761, Retrieved July 23, 2010 from <http://www.inf.pucminas.br/sbc2010/anais/wcge/index_arquivos/artigos_1.htm>.

Silva, C. V. S., & Ralha, C. G. (2011b). Detecção de cartéis em licitações públicas com agentes de mineração de dados, Revista Eletronica de Sistemas de Informação (RESI), 10(1) (pp. 1–19). ISSN 1677-3071.

Silva, C. V. S., & Ralha, C. G. (2011a). AGMI: An AGent-MIning tool and its application to brazilian government auditing. In J. Cordeiro & J. Filipe (Eds.). *WEBIST 2011, Proceedings of the 7th International Conference on Web Information Systems and Technologies, Noordwijkerhout, The Netherlands, 6–9 May, 2011*. ISBN 978-989-8425-51-5 (pp. 535–538). SciTePress.

Strehl, A., & Ghosh, J. (2003). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 1532-4435, *3*, 583–617<http://dx.doi.org/10.1162/153244303321897735> .

Symeonidis, A. L., Kehagias, D. D., & Mitkas, P. A. (2003). Intelligent policy recommendations on enterprise resource planning by the use of agent technology and data mining techniques. *Expert Systems with Applications*, 0957-4174, *25*(4), 589–602<http://www.sciencedirect.com/science/article/pii/S095741740300099X> .

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining* (1st ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.. ISBN 0321321367.

University Waikato. Weka machine learning project, Setember 2009. <http://www.cs.waikato.ac.nz/~ml/index.html>.

Weiss, G. (Ed.). (2000). *Multiagent systems: A modern approach to distributed artificial intelligence*. Cambridge, MA, USA: MIT Press. ISBN 0-262-23203-0.

Witten, I. H., & Frank, E. (2005). Data mining: Practical machine learning tools and techniques. In *Morgan Kaufmann series in data management systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.. ISBN 0120884070.

Wooldridge, M. (2009). *An introduction to multiagent systems* (2nd ed.). Wiley Publishing. ISBN 0470519460, 9780470519462.

Wu, Z., Cao, J., & Fang, C. (2011). Data cloud for distributed data mining via pipelined mapreduce. In *Agents and Data Mining Interaction – 7th International Workshop on Agents and Data Mining Interation, ADMI 2011, Taipei, Taiwan, May 2–6, 2011* (pp. 316–330), Revised selected papers.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., et al. (2007). Top 10 algorithms in data mining. *Knowledge and Information Systems, 14*, 1–37. ISSN 0219-1377.

Zaki, M. J. (1999). Parallel and distributed association mining: A survey. *IEEE Concurrency, 7*, 14–25. ISSN 1092-3063.

Zaki, M. J. (2000). Parallel and distributed data mining: An introduction. *Revised Papers from Large-Scale Parallel Data Mining, Workshop on Large-Scale Parallel KDD Systems, SIGKDD*. 3-540-67194-3 (pp. 1–23). London, UK: Springer-Verlag<http://dl.acm.org/citation.cfm?id=648035.744383> .

Zhao, W., Ma, H., & He, Q. (2009). Parallel k-means clustering based on mapreduce. In M. Jaatun, G. Zhao, & C. Rong (Eds.), *Cloud computing. Lecture notes in computer science* (Vol. 5931, pp. 674–679). Springer. ISBN 978-3-642-10664-4. <http://dx.doi.org/10.1007/978-3-642-10665-1_71>.

Zhou, D., Rao, W., & Lv, F. (2010). A multi-agent distributed data mining model based on algorithm analysis and task prediction. In *2nd International Conference on Information Engineering and Computer Science (ICIECS), December 2010* (pp. 1–4) http://dx.doi.org/10.1109/ICIECS.2010.5678352.