Automatic Optimization for a Clustering Based Approach to Support IT Management

Can BOZDOGAN
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, Canada
bozdogan@cs.dal.ca

A. Nur ZINCIR-HEYWOOD
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, Canada
zincir@cs.dal.ca

Yasemin GOKCEN
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, Canada
gokcen@cs.dal.ca

Abstract- **In this paper, we propose a system for automatically optimizing a decision support system to assist IT helpdesk operators for IT management tasks. The proposed system aims to leverage the publicly available experience that can be accessed on the Internet (such as blogs, discussion list etc.) in order to manage problems in IT environments. To achieve this, we propose a data driven system that is based on a combination of clustering and optimization techniques. Our results show that the proposed system can work on real life data sets and is a promising model for supporting decision making processes in IT environments.**

Keywords: **Decision support, clustering, optimization, data mining**

I. INTRODUCTION

Fixing failures and problems of computer systems in organizations and companies, requires Information Technology (IT) management teams who have knowledge and experience about the system failures. There are already software and hardware systems such as trouble ticket systems, incident management systems and so on to give support to such teams for solving IT problems [1][2]. However, these systems are more to track and manage occurring problems than manage knowledge and experience of IT teams.

As shown in previous work [3], using previous problem cases, which we call as experience, is an effective way of performing the required corrective actions quickly when a problem occurs. This is also effective for, documenting or sharing knowledge and experience of an IT management team. In big companies or organizations, it is more probable to find an IT team member with the right skills to solve a problem. In such organizations, receiving hundreds of questions or failure reports from many users, and the process of resolving them accumulates huge amounts of experience as such problems are solved. We believe that such experience data can be used for any decision support action in the future.

On the other hand, for a small to medium sized enterprises (SMEs), hiring experienced employees for IT management team can be really costly. Experience data, which is collected from previous problem/solution cases can be helpful to reduce the time to solve a new case. However, collecting experience data to extract such useful information is a problem that an organization might face. In this case, two main issues that can be faced are: (i) How to collect/generate experience data in an organization without any accumulated or stored experience? (ii) How to extract structured information and knowledge from the experience data?

To solve the first problem, we propose to employ publicly available experience data from Internet sources such as FAQs (Frequently asked questions) to generate a database of such experience. To solve the second problem, we employ CES+ clustering algorithm [3] together with MOGA (Multi objective genetic algorithm), which is an optimization technique. In this case, MOGA is employed to automate the process of choosing clustering parameters. MOGA type of algorithms is necessary in such automations because they aid the clustering to adapt from one data set to another.

II. BACKGROUND

Even though this a relatively new research area, there are some successful works providing support for fault management, and IT problem solving tasks. Marcu et al. presented an information model for inter-organizational fault management [4]. They present a model to cope with providing better service quality problems by analyzing phases of fault life cycle to define requirements.

Li et al. proposes a solution to automate the change management process by using past experiences to make more appropriate assessments [5]. The authors aim to reduce the time and cost to prevent service interruption in the change management process. They use a business-driven approach for automation of the process by guiding the change by using the historical experience information.

Bartolini et al. presents a prototype system to discuss development challenges of decision support tools [6]. Their aim is to autonomously identify optimal configurations for IT support where the inputs are provided by a user.

Moreover, Bartolini et al. presents a decision support tool, which is named as Symian-Web, in [7]. It is a tool that uses what-if scenario analysis to improve IT support and uses cloud systems for computing capabilities. Symian is used to create and simulate a model of IT organizations for real life. These models are used to see the effect of IT strategies and their processes.

In our previous work [3], we presented how utilization of

978-3-901882-50-0

data mining, specifically clustering algorithms, and information retrieval approaches can give assistance for a decision support system in SMEs. In that work, two different similarity measures and five different clustering algorithms were employed on three different datasets to compare their efficiency. In those evaluations, CES+ clustering algorithm have given the best performance. However, in that work we have not studied the automatic adaptation of the CES+ algorithm to different data sets. In this paper, we study is problem and propose the use of MOGA.

III. METHODOLOGY

Our aim is to provide a system that gives support for IT management. Minimizing experience acquisition efforts and extracting information from experience data are the two major goals of our proposed system. In the proposed system, there are four major components as following:

A. Web Crawling Engine(WCE)

The Web Crawling Engine, WCE, is used to collect experience data from the web pages. Different publicly available crawlers can be used for this purpose. We used WebSphinx [10], which has customizable parameters such as depth, thread number, and page size. To collect the experience data, we used FAQs of three well known internet web sites, namely (i) Princeton University IT Help Desk Web Site [9]; (ii) Parallels Free Support Resources Web Site [10]; and (iii) Go Daddy Help Center Web Site [11].

B. Experience Database

To move to this step of the experience data transformation, which is named as Experience Database, we developed a parser in C#. Before storing the data collected from the web sources, the parser collects the experience information in folders and then stores the experience data in the form of Problem-Solution structures in the Experience Database. The Problem-Solution structure means that every experience instance in the database is represented by a problem (which is created by the question part of a FAQ) and a solution (which is created by the answer part of a FAQ). To evaluate the proposed adaptation technique, MOGA supported CES+, we used the same datasets in [3] for the Experience Knowledgebase in our experiments.

C. Text Clustering Engine (TCE)and Experience KnowledgeBase

The Text Clustering Engine, TCE, consists of two components. These are the CES+ clustering algorithm and the MOGA optimization algorithm. The engine uses the MOGA to automate the selection of useful parameters for the CES+ algorithm.

1) CES+

The CES+ algorithm is a clustering algorithm that is specifically designed for short text clustering [3]. It has three main steps as the following.

In the first step, small clusters are created from problem-solution threads in the text. Each cluster has a dictionary of words, which are used to create the cluster and to represent the text. The core clusters are constructed based on the similarity between the problem-solution threads in a given dataset.

$$Similarity(Doc_1, Doc_2) = (1-\alpha) \times CosSimQ + \alpha \times CosSimA \quad \text{Eq.(1)}$$

where the *Content Value* (α)specifies the rate that defines the effect of the question(*CosSimQ()*) and the answer similarity(*CosSimQ()*).

In the second step , Expanding Clusters, the core clusters are expanded by combining different clusters or adding a thread to a cluster. The aim of the second step is merging similar clusters. During this step, the dictionaries of the clusters are updated. The similarity between two clusters is calculated as Eq. (3) by using their feature vectors:

$$Similarity(Cluster_1, Cluster_2) = \vec{tfidf}_1 \bullet \vec{tfidf}_2 \quad \text{Eq.(3)}$$

where TF/IDF represents the feature vector of the cluster. The similarity between a cluster and a thread is defined as Eq. (4):

$$Similarity(Cluster_C, Thread_1) = (1-\alpha) \times CosSimQ + \alpha \times CosSimA \quad \text{Eq. (4)}$$

Finally, in the third step, Fine Tuning Clusters, the threads, which belong to another topic, are extracted from a cluster. Sometimes, such unrelated threads can be found in a cluster because of the low threshold value used in the second step. Moreover, because of creating a new dictionary for each new cluster, it is possible to have threads that are not related to that new cluster anymore after merging. Thus, the extraction of the unrelated threads is performed by using the function, *C(i)* as Eq. (5):

$$C(i) = \sum_{W_i \in Dictionary(Cluster_C)} f(w_i) \quad \text{Eq.(5)}$$

where *Dictionary(ClusterC)* is the number of words of ClusterC in the dictionary, and

$$f(w_i) = \frac{Word(w_i)(tfidf)}{Number\_of\_words}$$

where $Word(w_i)(tfidf)$ means *tfidf* value of $W_i \epsilon Dictionary(Cluster_C)$ of the relative thread. If the *C(i)* is lower than the specified *fine-tuning Threshold* then the thread

is extracted. For more details of CES+ algorithm, please refer to [3].

2) Multi Objective Genetic Algorithm (MOGA)

Optimization aims to find the best solution for a given problem with a given set of limitations [12]. If an optimization problem involves only one objective, the searching for an optimal solution space is called as single – objective optimization [13]. MOGA has the basic idea of using evolutionary concepts from the nature for solving engineering problems, which has more than one objective. Genetic Algorithms (GAs) has basic elements as chromosomes that is used to represent candidate solutions, a fitness function that is used to define a set of parameter values to define better chromosomes, and crossover/mutation operators to provide diversity and generate new chromosomes. In this work, we followed the MOGA framework presented by Bacquet et al. [14], but we modified the reorientation of candidate solutions and the fitness function.

Using MOGA, we aim to optimize the parameter selection process of CES+ algorithm to automate the proposed system that we explained earlier. So each chromosome is represented by five fields as Content Value, Dictionary Threshold, Creating Core Clusters Threshold, Expanding Threshold and fine-tuning threshold. As for calculating the fitness function, we used Fwithin and Fbetween performance metrics, which are explained in Section 4.

Finally, Experience Knowledgebase is used to store the support structure, which is created by the Text Clustering Engine. The support structure consists of cluster information and components of clustered threads. We used MS SQL Server 2008 to develop Experience Knowledgebase for the proposed system.

IV. Experiments and Results

In this paper, MOGA supported CES+ algorithm is implemented in C# programming language using Microsoft Visual Studio 2010 environment. F**ive** metrics are used to compare the effectiveness of CES+MOGA for the evaluation phase**:**

Precision is a standard Information Retrieval (IR) metric [15] that represents the rate of retrieved documents that are relevant to the actual category.

Recall is a standard IR metric [16] that represents the rate of the retrieved documents to the relevant documents that are grouped in the same category.

Fwithin is estimated by calculating the average standard deviation per cluster. *Fwithin* shows us how much deviation exists in a cluster on average so it measures cluster cohesiveness [14]. So we look for lower Fwithin.

Fbetween is a metric that shows how separate the clusters are from each other [14]. So we look for higher Fbetween.

Standard Deviation is used to show how much variation from the average exists between attributes (features) of instances.We look for lower standard deviation.

Figure 1 shows the precision, recall and standard deviation for clusters, which are created by CES+ and CES+MOGA algorithms on PrincetonDS. In Figure 2, x-axis represents the cluster numbers (C1: the first cluster, C2: the second cluster, and so on). As we can see from the results, CES+MOGA generates clusters with higher precision/recall values and the results obtained by CES+ only and CES+MOGA are very close.

On the other hand, CES+ algorithm generates more number of clusters with some higher and some lower precision and recall values on ParallelsDS, Figure 2. Overall, there does not seem to be huge differences between results of CES+MOGA and CES+ only.

In Figure 3, we see that CES+ and CES+MOGA give close results with the same number of clusters on GoDaddyDS. Both have high precision/ recall values with low standard deviation.

In Figure 4, Fwithin and Fbetween results are represented where Fbetween is divided by 1000 to enable comparing both values. CES+MOGA gives results with higher Fbetween and with lower Fwithin on PrincetonDS whereas the results of both algorithms on ParallelsDS and GoDaddyDS have close Fwithin and Fbetween values.

Figure 5 shows the average precision, recall and standard deviation values of the results on each dataset. We see that, in average CES+MOGA and CES+ results have close precision, recall and standard deviation values on three different datasets.

These results show that our technique to automatically cluster different data sets, also automating the selection of parameters from one data set to another works as good as (if

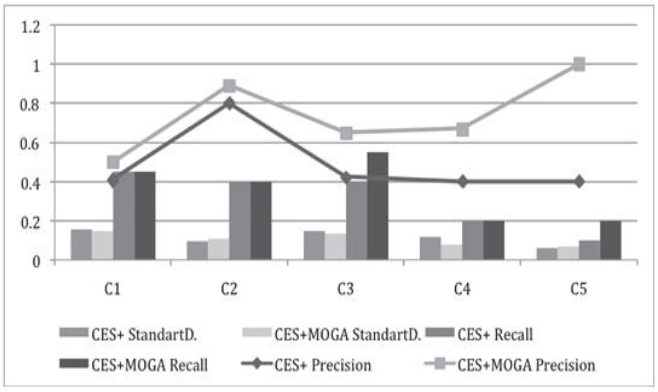

Figure 1: Results on PrincetonDS

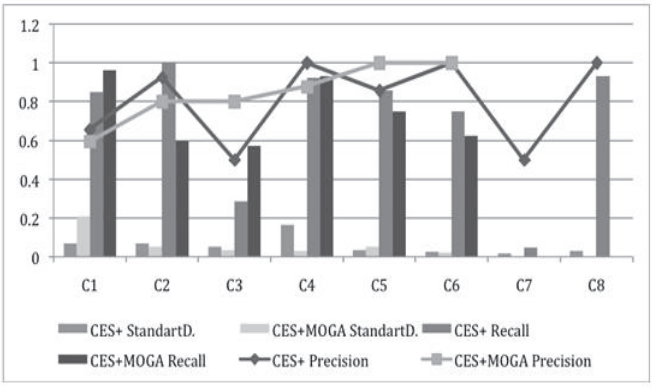

Figure 2: Results on ParallelsDS

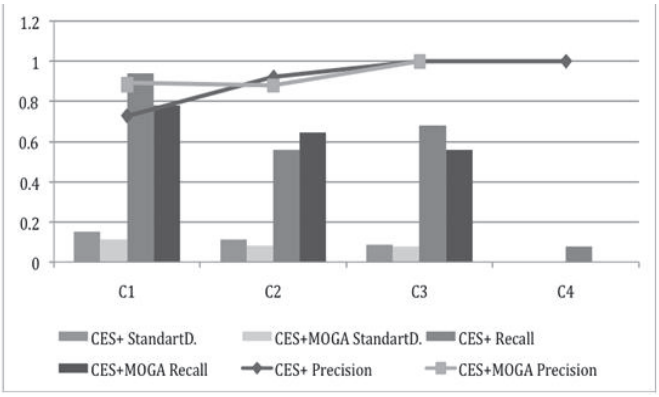

Figure 3: Results on GoDaddyDS

not better than) the manually chosen parameters for CES+ on these three real life data sets. This makes it possible to crawl publicly available data sets to find useful experience information (reported problems and their solutions). Then, they are clustered automatically according to their similarities (without manual parameter selection). By receiving close results between the CES+MOGA and manually configured CES+ algorithm, we are able to say that CES+MOGA can be used to automate the proposed system for real life usage.

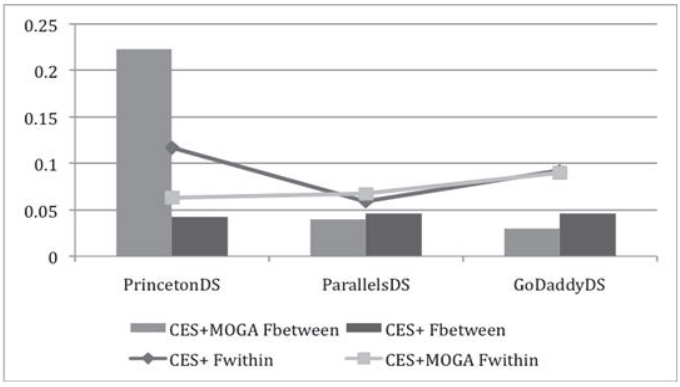

Figure 4: Fwithin and Fbetween results

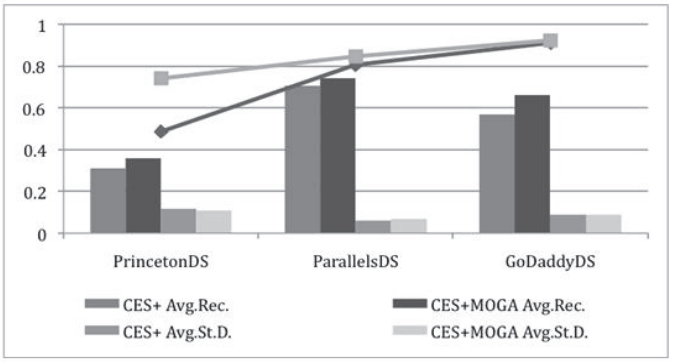

Figure 5: Average Precision, Recall and Standard Deviation values on each dataset

V. CONCLUSION

The objective of this paper is to present a system to provide decision support for IT management team members in small to medium sized companies and organizations by using a clustering approach that can be automatically optimized based on the experience data used. In this proposed system, we aim to explore if it is possible to make the system run automatically on real life datasets for IT management teams. To this end, by using the Multi Objective Genetic Algorithm (MOGA), we automated the parameter selection of CES+ clustering algorithm and showed that CES+ can be used automatically on different real life experience data sets without necessitating any manual intervention from the user.

REFERENCES

[1] Yixin Diao; Jamjoom, H.; Loewenstern, D.; , "Rule-Based Problem Classification in IT Service Management," *Cloud Computing, 2009. CLOUD '09. IEEE International Conference on* , vol., no., pp.221-228, 21-25 Sept. 2009

[2] Bartolini, C.; Stefanelli, C.; Tortonesi, M.; "On decision making in business-driven IT management," IFIP/IEEE International Symposium on Integrated Network Management, pp.1082-1088, 2011.

[3] Bozdogan C., Zincir-Heywood A. N., "Data Mining for Supporting IT Management", the 7th IFIP/IEEE International Workshop on Business-Driven IT Management (BDIM) in conjunction with IFIP/IEEE NOMS,2012

[4] Marcu, P.; Schaaf, T.; , "An information model for inter-organizational fault management," *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on* , vol., no., pp.1043-1049, 23-27 May 2011

[5] Li H., Zhan Z., "Bussiness-Driven Automatic IT Change Management Based on Machine Learning", the 7th IFIP/IEEE International Workshop on Business-Driven IT Management (BDIM) in conjunction with IFIP/IEEE NOMS,2012 - Maui,Hawaii,USA

[6] Bartolini C., Stefanelli C., Tortonesi M., "Potential Benefits and Challenges of Closed-Loop Optimization Processes for IT Support Organizations", the 7th IFIP/IEEE International Workshop on Business-Driven IT Management (BDIM) in conjunction with IFIP/IEEE NOMS,2012

[7] Bartolini C., Stefanelli C., Targa D., Tortonesi M., "A Cloud-based Solution for the Performance Improvement of IT Support Organizations", the 7th IFIP/IEEE Mini Conference

[8] WebSPHINX : A Personal, Customizable Web Crawler. Available : http://www.cs.cmu.edu/~rcm/websphinx/

[9] Princeton University Office of Information Technology Knowledgebase. Available: http://helpdesk.princeton.edu/

[10] Parallels. Virtualization and automation Software Parallels Knowledgebase. Available: http://kb.parallels.com/

[11] Go Daddy, Go Daddy Help Center. Available: http://help.godaddy.com/

[12] Coello Coello, C.A.; , "Evolutionary multi-objective optimization: a historical view of the field," *Computational Intelligence Magazine, IEEE* , vol.1, no.1, pp. 28- 36, Feb. 2006 doi: 10.1109/MCI.2006.1597059

[13] Kalyanmoy D., Multi-Objective Optimization using Evolutionary Algorithms, England : John Wiley&Sons Ltd, 2002, 1-5.

[14] Bacquet C., Zincir-Heywood A.N., Heywood M. I., "Genetic Optimization and Hierarchical Clustering applied to Encrypted Traffic Identification", IEEE Symposium on Computational Intelligence on Cyber Security, pp. 194-201, 2011.

[15] Baldi P., Frasconi P., Smyth P., Modeling the Internet and the Web,2003,ISBN :0-470-84906-1