Expert Systems with Applications 36 (2009) 5549–5556

Contents lists available at ScienceDirect

Expert Systems with Applications

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

The generic genetic algorithm incorporates with rough set theory – An application of the web services composition

Wen-Yau Liang [a], Chun-Che Huang [b,*]

[a] Department of Information Management, National Changhua University of Education, Changhua, Taiwan, ROC
[b] Department of Information Management, National Chi Nan University, Pu-Li, Na-Tau, Taiwan, ROC

ARTICLE INFO

Keywords:
Generic
Genetic algorithm
Rough set
Web services
Web services composition

ABSTRACT

Evolutionary computing (EC) techniques have been used traditionally used for solving challenging optimization problems. But the increase in data and information has reduced the performance capacity of the GA, but highlighted the cost of finding a solution by GA. In addition, the genetic algorithm employed in previous literature is modeled to solve one problem exactly. The GA needs to be redesigned, at a cost, for it to be applied to another problem. For these two reasons, this paper proposes a method for incorporating the GA and rough set theory. The superiority of the proposed GA in this paper lies in its ability to model problems and explore solutions generically. The advantages of the proposed solution approach include: (i) solving problems that can be decomposed into functional requirements, and (ii) improving the performance of the GA by reducing the domain range of the initial population and constrained crossover using rough set theory. The solution approach is exemplified by solving the problem of web services composition, where currently the general analysis and selection of services can be excessively complex and un-systemic. Based on our experimental results, this approach has shown great promise and operates effectively.

1. Introduction

Evolutionary computing (EC) is the collective name for problem-solving techniques based on principles of biological evolution, such as natural selection and genetic inheritance (Eiben & Smith, 2007). Evolutionary computing techniques include genetic algorithms, particle swarm optimization and ants system (Kwok, Liu, & Dissanayake, 2006). Further, it has been used traditionally for solving challenging optimization problems (Eberbach, 2005), such as, optimization problems of searching (Bautista & Pereira, 2006; Bergey & Ragsdale, 2005; Hwang & He, 2006), scheduling (Chang, Hsieh, & Wang, 2007; Ruiz, Maroto, & Alcaraz, 2006; Yoo & Gen, 2007), production and distribution (Aliev, Fazlollahi, Guirimov, & Aliev, 2007; Chan, Chung, & Wadhwa, 2005; Chen, Pan, & Lin, 2008), knowledge discovery problems (Chen & Hsu, 2006; Wachla & Moczulski, 2007), and machine learning (Chi, Ersoy, Moskowitz, & Ward, 2007; Shon & Moon, 2007).

GA algorithms are efficient search methods based on the principles of natural selection and population genetics in which random operators in a population of candidate solutions are employed to generate new points in the search space (Chen & Hsu, 2006). But the increase in data and information has reduced the performance capacity of the GA, but highlighted the cost of finding a solution by using it. To solve the problem, Passone, Chung, and Nassehi (2006) combined the GA with the guidance provided by domain-specific knowledge. The GA consists of a modified version of the classical genetic operations of initialization, selection, crossover and mutation designed to incorporate practical but general principles of model without reference to any specific problems. The incorporated knowledge helps the GA to improve performance and quickly converge without affecting the result.

In this paper, the rough set theory introduced by Pawlak as a mathematical method (Beaubouef & Lang, 1998; Questier, Arnaut-Rollier, Walczak, & Massart, 2002) is used to improve the GA performance. Rough set theory is a powerful mathematical tool for dealing with uncertainty of data. It relies only on the available data and attributes to work on the analysis of important features as well as generation of classification rules (Wang, 2005). To cope with the inconsistency, lower and upper approximations of decision classes are defined. In the rough set method, we can extract the minimal attribute sets without deterioration of quality of approximation, and minimal length decision rules corresponding to lower or upper approximation (Inuiguchi & Miyajima, 2007). The applications of the rough set model in various problems have demonstrated its usefulness (Pattaraintakorn, Cercone, & Naruedomkul, 2006; Zhao, Yao, & Luo, 2007). It has been applied in many fields, such as pattern recognition (Wang, Tsang, Zhao, Chen, & Yeung, 2007a; Wang, Yang, Teng, Xia, & Jensen, 2007b), machine learning (Hu, Yu, & Xie, 2008; Wang, 2005), knowledge discovery

* Corresponding author.
E-mail address: cchuang@ncnu.edu.tw (C.-C. Huang).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.06.084

(Li, Ruan, Geert, Song, & Xu, 2007; Wang et al., 2007a; 2007b), decision analysis (Fan, Liu, & Tzeng, 2007; Inuiguchi & Miyajima, 2007), data mining (Li et al., 2007; Tsai, Cheng, & Chang, 2006; Yang, Liu, & Lin, 2007) and so on (Su & Hsu, 2006; Zhu, 2007).

This study proposes a method for incorporating the GA and rough set theory. The superiority of the proposed GA in this paper lies in its ability to model problems and explore solutions generically. A problem can be solved if it can be decomposed into functional requirements which can be satisfied by the associated databases of basic units. However, more data in each database makes the GA less efficient. To solve this problem, the knowledge discovery approach, and in particular the rough set theory, is used to reduce the range of basic database units. The proposed GA incorporating the rough set theory can: (i) solve problems that can be decomposed into functional requirements, and (ii) improve the performance of the GA by reducing the domain range of the initial population and constrained crossover using rough set theory. Incorporating the RST rules improves GA performance.

This paper is organized as follows: In Section 2, the rough set theory and the literature review of web services composition is illustrated. Section 3 proposes a solution approach of generic genetic algorithm incorporating rough set theory. In Section 4, an illustrative example proves the superiority of the proposed approach. Section 5 concludes the paper.

2. Literature review

2.1. Rough set theory

The starting point of rough set theory is the assumption with every object of interest we associate some knowledge (for example, if the objects are personal computers (PC), the attributes of PC correspond to the data about the PC's functionality). Objects are indiscernible if they are characterized by the same information. The generated indiscernibility relation is the mathematical basis of rough set theory. In rough set theory, a reduct is defined as a minimal sufficient subset of a set of attributes, which has the same ability to discern concepts as when the full set of attributes is used (Ziarko, 1999). Basically, the reducts represent necessary condition attributes in decision making. Further, a subset of the attributes has more than one reduct, so simplifying the decision rules will not yield unique results. To implement the rough set theory, a procedure for determining the reducts is necessary: including generate reducts and identifying the decision rule.

Reducts generation can be found in many works, for example, (Huang & Tseng, 2004; Pawlak, 1991) and is not the focus of this paper. To identify or compose the candidate reduct rules, a rule identification algorithm is developed based on (Huang & Tseng, 2004). Each object (reduct) is denoted as rt_i. The rt_i is defined by the frequency of objects which are associated with the reduct rule i. Using the rt_i, an auxiliary matrix $[e_{ij}]$ will be generated from the original reduct – the input feature matrix. Suppose the generated reducts are transformed into a matrix form and are denoted as $[a_{ij}]$.

The entries of the transformed reduct-input feature entries are defined as follows:

$e_{ij} = rt_i$ if a_{ij} < > "x" ("x" corresponding to "not care to feature j")
e_{ij} = "x" if a_{ij} = "x"

The rule identification algorithm

Step 0: (i) List the auxiliary matrix.
(ii) Set iteration number $k = 1$.
Step 1: Compare those reducts (rows of matrix $[a_{ij}](k)$) for one specific case at a time. If there is more than one solution for the reduct selection, then select the reduct which can be merged with most of the objects; otherwise, select the reducts whose entries received a higher *eij*. Draw a horizontal line hi through each of the rows of the matrix $[a_{ij}](k)$ corresponding to these reducts.
Step 2: For each column in $[a_{ij}](k)$ corresponding to an entry of the feature which is not "x", and is crossed once by any of the horizontal lines h*i*, draw a vertical line v*j*.
Step 3: Repeat Step 1 and Step 2 until one reduct has been selected for each object in the current outcome. All double-crossed entries of the features of the matrix will form the rules.
Step 4: If all the objects have been dealt with in the current outcome, transform the incidence matrix $[a_{ij}](k)$ into $[a_{ij}](k+1)$ by removing all the rows and columns included in the current outcome.
Step 5: If matrix $[a_{ij}](k+1)$ = " " (where " " denotes a matrix with all elements equal to blank, stop and output the results; otherwise set $k = k + 1$ and go back to Step 1.

The applications of the rough set model in various problems have demonstrated its usefulness. For example, the researchers proposed the approach to illustrate formulating more meaningful rules using the notion of ordinal prediction. It proved to be an improvement for rule learning both in computing performance and the usefulness of the rules derived from a case study on melanoma data (Pattaraintakorn et al., 2006). There is other research, Zhao et al. (2007) that made an empirical experiment for letter recognition for demonstrating the usefulness of the discussed relations and reducts. Rough set theory has been applied in many fields, such as pattern recognition, the researchers proposed a new feature selection strategy based on rough sets and particle swarm optimization (Wang et al., 2007a; 2007b); machine learning, Wang (2005) adopted a generalization of rough set model based on fuzzy lower approximation for information granules; knowledge discovery, the researchers proposed a rough sets based characteristic relation approach for dynamic attribute generalization in data mining (Li et al., 2007); decision analysis, Inuiguchi and Miyajima (2007) studied rule induction from two decision tables as a basis of rough set analysis of more than one decision table, and Fan et al. (2007) proposed some decision logic languages for rule representation in rough set-based multi-criteria analysis, and so on (Su & Hsu, 2006; Zhu, 2007).

2.2. Web services composition

A web service, as defined by the W3C web services architecture working group, is a software application identified by a URI, whose interfaces and bindings can be defined, described, and discovered as XML artifacts (Park, 2006). The benefits of Web services include the decoupling of services interfaces from implementations and platform considerations, enabling dynamic service binding, and an increase in cross-language, cross-platform interoperability (Ferris & Farrell, 2003). One of the strengths of web services is their capacity to be composed. Composing services rather than accessing a single service is essential and offers much better benefits to users (Ghedira & Mezni, 2006). It allows a high-level of interoperability. Web services are quickly becoming the prominent criterion for distributed computing and electronic business. This has given service providers and application developers the opportunity to develop value-added services, by combining existing web services (Yang & Papazoglou, 2004). Web services are independent of specific platforms and computing paradigms, and have the capacity to form composed processes, referred to as composite web services (Maamar, Benslimane, Thiran, Ghedira, & Dustdar, 2007). Elementary web services do not rely on other web services to fulfill external re-