Neurocomputing 150 (2015) 50–57

Contents lists available at ScienceDirect

Neurocomputing

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

Evolutionary algorithm and decisional DNA for multiple travelling salesman problem

Peng Wang [a], Cesar Sanin [a], Edward Szczerbicki [b,*]

[a] School of Engineering, Faculty of Engineering and Built Environment, The University of Newcastle, Newcastle, NSW, Australia
[b] Gdansk University of Technology, Gdansk, Poland

ARTICLE INFO

Article history:
Received 20 October 2013
Received in revised form
31 December 2013
Accepted 29 January 2014
Communicated by "Chennai Guest Editor"
Available online 6 October 2014

Keywords:
Decisional DNA (DDNA)
Set of experience knowledge structure (SOEKS)
Heuristics
Evolutionary algorithm
Genetic algorithm
Optimization problem

ABSTRACT

In the real world, it is common to face optimization problems that have two or more objectives that must be optimized at the same time, that are typically explained in different units, and are in conflict with one another. This paper presents a hybrid structure that combines set of experience knowledge structures (SOEKS) and evolutionary algorithms, NSGA-II (Non-dominated Sorting Genetic Algorithm II), to solve multiple optimization problems. The proposed structure uses experience that is derived from a former decision event to improve the evolutionary algorithm's ability to find optimal solutions rapidly and efficiently. It is embedded in a smart experience-based data analysis system (SEDAS) introduced in the paper. Experimental illustrative results of SEDAS application to solve a travelling salesman problem show that our new proposed hybrid model can find optimal or close to true Pareto-optimal solutions in a fast and efficient way.

1. Introduction

Thanks to the advent of the information age, with its rapid development of science and technology, a substantial amount of information has been created and has quickly spread throughout global networks and media. Organizations today are becoming increasingly knowledge-intensive and collaborative [1]. However, large part of useful knowledge is hidden and is not readily available. A growing number of enterprises have realized that a tool that effectively enables the capture, representation, retrieval, and reuse of knowledge is the key to supporting various organizational decisions [2,3]. Another issue that is having increased new pressure on decision-makers results from intensive competition and increased spectrum of choices being available for customers. Although a considerable number of multi-objective evolutionary algorithms have the ability to address processing optimization problems in such areas as engineering and technology, there is a lack of comprehensive and generalized structures for the optimization of business processes [4].

Set of experience knowledge structure (SOEKS or SOE for short) is an experience-based knowledge representation that can store uncertain and incomplete data and make qualitative and quantitative extractions of knowledge from the available information, which can often be unstructured, semi-structured, fuzzy, and vague [5–8]. Additionally, SOEKS can be shaped in an extensive understandable and transportable language such as Extensible Markup Language (XML) or Ontology Web Language (OWL) [6]. XML and OWL representation of SOE allows knowledge to be exchanged quickly and securely between applications and systems [9]. The research presented in this article introduces a novel adaptable knowledge structure, which has been designed to work with meta-heuristics or evolutionary algorithms (EA) to guarantee simplicity and improvement in decision-making processes. SOEKS offers solutions for complex problems by means of integrating knowledge into the computer-based system. It uses many computer science domains, such as knowledge representation, artificial intelligence, data mining and evolutionary algorithms, to make powerful, efficient and effective systems for learning, reasoning, and forecasting from current knowledge and past experience. The proposed overall architecture that we call smart experience-based data analysis system (SEDAS) combines and integrates advantages of both techniques (SOE and EA) in an enhanced effort to find optimal solutions from past optimization problems and to store the gathered results as formal decision events to be reused in the future.

The next section introduces relevant underlying techniques, i.e. SOEKS, Decisional DNA, and multi-objective evolutionary algorithms.

* Corresponding author.
E-mail addresses: Peng.Wang@uon.edu.au (P. Wang), Cesar.Sanin@newcastle.edu.au (C. Sanin), Edward.Szczerbicki@newcastle.edu.au (E. Szczerbicki).

http://dx.doi.org/10.1016/j.neucom.2014.01.075
0925-2312/

Next, a novel structure of smart experience-based data analysis system (SEDAS) that embeds an EA and SOEKS is presented. The remainder of this paper presents experimental and illustrative case study application of the proposed model to evaluate its performance.

2. Background

2.1. Set of experience knowledge structure and decisional DNA

The SOEKS has been designed to collect experiences and knowledge from multiple applications that are assembled as formal decision events [5]. This collected knowledge assists organizations in making precise decisions, predictions, and recommendations, and it is a dynamic structure that is dependent on the information and data that it has received. Decisional DNA (DDNA) is a knowledge repository that is designed to capture decisional fingerprints inside organizations through the use of SOEKS [9]. Furthermore, it can be expressed in Extensible Markup Language (XML) or Ontology Web Language (OWL) to make it shareable and transportable [6].

SOEKS is composed of variables, functions, constraints, and rules [5]. Each formal decision event is stored in a combined structure of those four components of the SOEKS. Variables are the centre root of the SOEKS. They are used to express the states of the formal decision event. Functions are formed as equations that are intended to depict the relationships among a dependent variable and a set of input variables. Functions depict our experience in decision-making process by providing relationships between variables. One decision could differ from another solely by the addition or subtraction of a new function. Constraints are another method of representing associations between variables. Although a constraint is a form of function, it has a different purpose. It restricts the performance and configuration of a system and the feasible solutions in a decision-making problem. Finally, rules are another way of illustrating links between variables. They condition possible relationships that operate on the universe of variables. Essentially, rules use the statements IF–THEN–ELSE to connect conditions with their consequences.

In addition, the structure of the SOEKS is similar to some important features of natural DNA [7]. This structure imitates a gene in combining four nucleotides of DNA by integrating four components of experience to adapt to different needs. In the same way that a gene produces a phenotype, a set of experience yields a value for a decision with its elements. Each SOEKS can be categorized and stores data as a gene would in DNA [5,7]. A set of experiences in the same category makes up a decisional chromosome, which stores decisional strategies for that category. Collection of chromosomes establishes an entire inference tool to offer a blueprint of knowledge inside a system, machine, or organization [9].

2.2. Heuristic

The word ‘heuristic’ originally derives from the Greek word “find” or “discover” [10]. Scientists apply this notion to the meaning of experience-based techniques for problem solving, learning, and discovering. Processing a heuristic method is to rapidly approach a solution that is expected to be close to the optimal solution. In real-world problems, heuristics are often used to address optimization problems that must be solved by people or by machines [11].

2.3. Multi-objective evolutionary algorithm (MOEA)

Evolutionary algorithms are meta-heuristics that are inspired on the “survival of the fittest” concept from Darwin’s evolutionary theory; evolutionary algorithms have been used as search and optimization techniques since the 1960s in a wide variety of disciplines [12]. Evolutionary algorithm represents a type of stochastic optimization method that mimics the process of natural evolution [13]. Multi-objective optimization (MO), which is considered in this paper, involves multiple-criteria decision making with more than one objective that are to be optimized simultaneously. Unlike single objective optimization, the goal of MOEA is to find as many different Pareto-optimal solutions as possible [14]. The Pareto Optimal solutions are optimal in the sense that no other solutions in the search space are better than them when all of the objectives are accounted for [15]. There are two tasks that a MOEA should complete when solving a multi-objective optimization problem: (1) evaluate the fitness of population members and sort them based on this evaluation, and (2) maintain population diversity.

The NSGA-II algorithm is a well-known EA that uses a fast non-dominated sorting approach with $O(MN^2)$ arithmetic operations (steps) (where M is the number of objectives and N is the population size). In comparison with other EAs, the NSGA-II is more efficient and effective for converging solutions at the true Pareto-optimal set and maintains diversity among the solutions [16]. The goal of the NSGA-II algorithm is to find a Pareto set by using a set of objective functions to rank the fitness of a population of candidate solutions. This process includes many evolutionary operators, such as selection, genetic crossover, and genetic mutation. The NSGA-II algorithm assumes that potential solutions of a problem are individuals or phenotypes. Each individual has a set of variables, much as the genes of a chromosome, which can be mutated and altered. Traditionally, these variables are organized by a string of values in binary form. A fitness value, which is always positive, is used to reflect the degree of suitability of the chromosome for survival. Throughout the genetic evolution, the chromosomes of the fittest individuals are stochastically selected from the current population. The genes of the chromosomes are combined and are mixed within the population of offspring in the next generation. A superior chromosome is expected to have a higher chance of producing better offspring in the subsequent generation in nature. This cycle of evolution is repeated until a desired termination criterion is reached. The algorithm can also be terminated by the number of evolutionary cycles.

To facilitate the genetic evolution cycle, the operators of a Simulated Binary Crossover (SBX) and polynomial mutation are required to produce offspring [17]. The crossover selects the genes of a parent chromosome and an offspring chromosome from one generation to the next. This technique is also known as fitness proportionate selection, where the individual is selected on the basis of its fitness. A mixing ratio with a typical value of 0.9 is usually used as the probability of SBX crossover. There are many crossover methods for ordered chromosomes. The mutation operator is applied to each offspring individually after the crossover exercise. This operator alters one or more genes in a chromosome with a small probability (typically a value of less than 0.1). The aim of the mutation operator is to obtain faster convergence. Therefore, NSGA-II employs two performance metrics in evaluating each of the above two tasks in a solution set [16]. The NSGA-II algorithm is among the most extensively used algorithms, and it offers a population of selected solutions if it is applied to a multiple-objective problem. Such a population or set of solutions is mathematically known as the Pareto-optimal set, and it is a suite from which the user can choose an optimal solution [16].

2.4. Drawbacks of using EAs

EAs are used for calculations, data processing, and many other applications. By using an EA, the calculation time can be reduced when solving an optimization problem. However, there is no