Received 3 June 2015
Revised 28 August 2015
24 October 2015
Accepted 26 November 2015

Big data analytics with swarm intelligence

Shi Cheng
Division of Computer Science, The University of Nottingham, Ningbo, China
Qingyu Zhang
Department of Management Science, Shenzhen University, Shenzhen, China and Research Institute of Business Analytics and Supply Chain Management, Shenzhen, China, and
Quande Qin
Department of Management Science, Shenzhen University, Shenzhen, China; Research Institute of Business Analytics and Supply Chain Management, Shenzhen, China and Center for Energy and Environmental Policy Research, Beijing Institute of Technology, Beijing, China

Abstract

Purpose – The quality and quantity of data are vital for the effectiveness of problem solving. Nowadays, big data analytics, which require managing an immense amount of data rapidly, has attracted more and more attention. It is a new research area in the field of information processing techniques. It faces the big challenges and difficulties of a large amount of data, high dimensionality, and dynamical change of data. However, such issues might be addressed with the help from other research fields, e.g., swarm intelligence (SI), which is a collection of nature-inspired searching techniques. The paper aims to discuss these issues.

Design/methodology/approach – In this paper, the potential application of SI in big data analytics is analyzed. The correspondence and association between big data analytics and SI techniques are discussed. As an example of the application of the SI algorithms in the big data processing, a commodity routing system in a port in China is introduced. Another example is the economic load dispatch problem in the planning of a modern power system.

Findings – The characteristics of big data include volume, variety, velocity, veracity, and value. In the SI algorithms, these features can be, respectively, represented as large scale, high dimensions, dynamical, noise/surrogates, and fitness/objective problems, which have been effectively solved.

Research limitations/implications – In current research, the example problem of the port is formulated but not solved yet given the ongoing nature of the project. The example could be understood as advanced IT or data processing technology, however, its underlying mechanism could be the SI algorithms. This paper is the first step in the research to utilize the SI algorithm to a big data analytics problem. The future research will compare the performance of the method and fit it in a dynamic real system.

Originality/value – Based on the combination of SI and data mining techniques, the authors can have a better understanding of the big data analytics problems, and design more effective algorithms to solve real-world big data analytical problems.

Keywords Evolutionary computation, Optimization, Data mining, Big data, Swarm intelligence, Big data analytics

Paper type Research paper

This work is partially supported by National Natural Science Foundation of China under Grant Nos. 71402103, 71572115, 60975080; National Science Foundation of SZU under Grant No. 836; Foundation for Distinguished Young Talents in Higher Education of Guangdong under Grant No. 2012WYM_0116; MOE Youth Foundation Project of Humanities and Social Sciences at Universities in China under Grant No. 13YJC630123; Ningbo Science and Technology Bureau (No. 2012B10055).

Industrial Management & Data Systems
Vol. 116 No. 4, 2016
pp. 646-666

0263-5577
DOI 10.1108/IMDS-06-2015-0222

1. Introduction

In the traditional decision making or statistical analytics, the results are obtained based on the analysis of the samples from a small data set. The analysis bias or ridiculous coincidence may occur due to the choices of samples. In other words, the partial and insufficient information from a small data set may produce wrong or biased analytical results. The quality and quantity of data are vital for the effectiveness of problem solving. The more data we obtain, the more clear structure of a problem can be observed and thus the more accurate analysis can be made to solve the problem.

Nowadays, the big data analytics has attracted more and more researchers' attentions (Alexander *et al.*, 2011). The big data is defined as the data set whose size is beyond the processing ability of typical database or computers. Four elements are emphasized in the definition, which are capture, storage, management, and analysis (Manyika *et al.*, 2011). The focus of the four elements is the last stage, the big data analytics, which is automatically extracting knowledge from a large amount of data. It can be seen as the mining or processing of the massive data, and thus useful information could be retrieved from the large data set (Rajaraman *et al.*, 2012). The traditional methods for data analysis are based on the mathematical models of the problems first and then see if the data fit the models. With the growth of the variety of timely data, these mathematical models may be ineffective in solving problems. The paradigm should shift from the model-driven to the data-driven approach. The data-driven approach not only focusses on predicting what is going to happen, but also concentrates on what is happening right now and further getting ready for the future events.

The data-driven-analytics problem is hard to solve since there are several obstacles to overcome. The first obstacle is how to handle the massive amount of high-dimensional data rapidly. The dimensionality of data affects the performance of algorithms. Many algorithms suffer from the curse of dimensionality, which implies that their performance deteriorates quickly as the dimension of the search space increases (Hastie *et al.*, 2009; Domingos, 2012; Donoho, 2000). The big data analytics also suffers from this problem where the large scale data will be processed in a limited time with a reasonably good performance.

The second obstacle is the velocity of data, which means the rapid change of the data content. Since the content of the big data keeps increasing over time, the targets of big data analytics also need to change with time. The variety of data, coming from different sources with different types, is the third obstacle. Sometimes, the different types of unstructured data need to be pre-processed to semi-structured and/or structured data before the analytics. In many cases, multiple objectives need to be achieved simultaneously in these large data sets. The majority of traditional methods can only work with continuous and differentiable functions, and have to perform a series of separate runs to satisfy different objectives (Coello *et al.*, 2007). Thus, there is a need for further study to crack the multiple objective problems with less restriction on the objective functions.

The big data analytics is a relatively new research field in information processing; however, the problems faced by the big data analytics might be addressed with the help from other research fields, such as swarm intelligence (SI). SI is a set of search and optimization techniques (Kennedy *et al.*, 2001; Dorigo and Stutzle, 2004; Eberhart and Shi, 2007). There is no complex mathematical model in SI. The algorithm is updated based on few iterative rules and the evaluation of solution samples. Different from traditional single-point-based algorithms such as hill-climbing algorithms, each SI

algorithm is a population-based algorithm, which consists of a set of points (population of individuals). Each individual represents a potential solution to the problem being optimized. The population of individuals is expected to have high tendency to move toward better and better solution areas with iterations over iterations through cooperation and/or competition among themselves.

In this paper, the association between big data analytics and SI techniques is discussed. The potential application of the SI in the big data analytics is analyzed and vice versa. The big data analytics problems are divided into four elements: handling a large amount of data, handling high-dimensional data, handling dynamical data, and multiobjective optimization. Most real-world big data problems can be modeled as a large scale, dynamical, and multiobjective problem where SI has demonstrated great success. With the SI, more effective methods can be designed and applied in the big data analytical problems.

This paper is organized as follows. Section 2 reviews the basic concepts of big data analytics, which include the data classification and data clustering. Section 3 introduces the SI methods. The potential use of SI in the big data analytics problems is discussed in Section 4. An application of big data analytics – the commodity routing system is briefly introduced in Section 5, followed by conclusions in Section 6.

2. Big data analytics

The data set of the big data cannot be handled by typical relational or object-oriented database, normal computers, or traditional desktop application software. It needs huge parallel processing power of computer clusters. Mostly the big data processing is based on a nonlinear system whose behavior sometimes appears to be unpredictable or counterintuitive. For a linear system, it satisfies the superposition principle – the additivity and homogeneity properties as below:

$$f(x_1 + x_2 + \ldots) = f(x_1) + f(x_2) + \ldots$$

$$f(ax) = af(x) \text{ for scalar } a$$

Unlike a linear system, the output of a nonlinear system is not directly proportional to the input. It needs to use the data from multiple sources with multiple dimensions, so it looks like chaotic and random, but actually it is not always random and some hidden knowledge can be discovered.

Five important aspects of big data, which begin with character "V," are emphasized in big data analytics. These aspects include volume, variety, velocity, veracity, and value. These five aspects represent the different difficulties in analyzing the big data. The details of five aspects are as follow:

(1) volume: a large amount of data;

(2) variety: the range of data types and sources;

(3) velocity: the speed of data changes;

(4) veracity: the uncertainty due to data inconsistency, incompleteness, and/or model approximations; and

(5) value: the value of the insights and benefits.

The big data researches aim to get the insights or benefits from the massive amount of dynamically changing data. The large amount and rapidly changing data increases the hardness of problem. Even for a simple sort or search operation, the problem with the big data are much more difficult than a problem with relatively small data.

The big data analytics is also to automatically extract knowledge from a large amount of data. It can be seen as the mining or processing of the massive data in repositories for useful information or patterns (Rajaraman *et al.*, 2012). Data mining is a part of the bigger process of the knowledge discovery in databases (KDD). KDD is the process of converting raw data into useful information (Fayyad *et al.*, 1996; Tan *et al.*, 2005). Figure 1 shows the general process of KDD. Data mining is the critical analysis step of KDD. The techniques in data mining field could be utilized in big data analytics, for example, data classification, data clustering, prediction, and descriptive statistic, just to name a few.

2.1 Data classification

Data classification, or termed as data categorization, is a problem that finds correct category (or categories) for objects (i.e. data) by giving a set of categories (subject, topics) and a collection of data set. Data categorization can be considered as a mapping f: $\mathcal{D} \rightarrow \mathcal{C}$, which is from the object space $\mathcal{D}$ onto the set of classes $\mathcal{C}$. The objective of a classifier is to obtain an accurate categorization results or predictions with high confidence.

The big data may contain many kinds of unstructured or semi-structured data; In some cases, these data need to be transformed into structured data. Each data record with many attributes or features is transformed as a vector with many dimensions. The dimension of the feature space is equal to the number of different attributes that appear in the data set. Different weight can be assigned to each feature. The methods of assigning weights to the features may vary. The simplest is the binary method in which the feature weight is either one – if the corresponding feature is present in the data – or zero otherwise (Cervantes *et al.*, 2009; Cheng *et al.*, 2012b).

2.2 Data clustering

The data clustering analysis is a technique that divides data into several groups (clusters). The goal of clustering is to classify objects being similar (or related) to one another into the same cluster, and put objects being distant from each other in different clusters (Tan *et al.*, 2005).

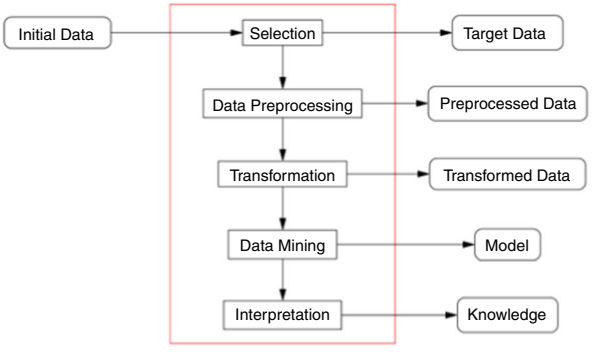


Figure 1.
The process of knowledge discovery in databases (KDD)

Clustering is the process of grouping similar objects together. From the perspective of machine learning, the clustering analysis is sometimes termed as unsupervised learning. There are N points in the given input, $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$, the "interesting and/or useful pattern" can be obtained through the similarity calculation among points (Murphy, 2012).

The data clustering methods can also be applied to the SI field (Shi, 2011a, b). In the brain storm optimization algorithm, every solution is spread in the search space. The distribution of solutions can be utilized to reveal the landscapes of a problem. From the clustering analysis, the search results can be obtained.

3. SI

Many real-world applications can be represented as an optimization problem of which algorithms are required to have the capability to search for the optimum. Most traditional methods can only be applied to continuous and differentiable functions (Shi, 2011a). For problems with non-continuous or differentiable functions, the traditional methods cannot solve or at least are difficult to solve. So the meta-heuristic algorithms are proposed to solve such problems. Recently, the SI, as one type of the meta-heuristic algorithms, is attracting more and more attentions.

The SI, which is based on a population of individuals, is a collection of nature-inspired searching techniques (Kennedy *et al.*, 2001). To search a problem domain, a SI algorithm processes a population. A population is a collection of individuals. Each individual represents a potential solution to the problem being optimized. In the SI, an algorithm maintains and successively improves a collection of potential solutions until some stopping condition is met. The solutions are initialized randomly in the search space, and are guided toward the better and better areas through the interactions among solutions.

Mathematically, the updating process of population of individuals over iterations can be looked as a mapping process from one population of individuals to another population of individuals from one iteration to the next iteration, which can be represented as $P_{t+1} = f(P_t)$, where P_t is the population of individuals at the iteration t, $f()$ is the mapping function.

As a general principle, the expected fitness of a solution returned should improve as the search method is given more computational resources in time and/or space. More desirable, in any single run, the quality of the solution returned by the method over iterations should improve monotonically – that is, the quality of the solution at time $t+1$ should be no worse than the quality at time t, i.e., $fitness(t+1) \leqslant fitness(t)$ for minimum problems (Ficici, 2005). There exist many SI algorithms; among them most common ones are the particle swarm optimization (PSO) algorithm (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995), which was originally designed for solving continuous optimization problems, and the ant colony optimization (ACO) algorithm, which was originally designed for discrete optimization problems (Dorigo *et al.*, 1996).

3.1 PSO

PSO, which is one of the SI techniques, was invented by Eberhart and Kennedy in 1995 (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995). It is a population-based stochastic algorithm modeled on the social behaviors observed in flocking birds. Each particle, which represents a solution, flies through the search space with a velocity that is dynamically adjusted according to its own and its companion's historical behaviors. The particles tend to fly toward better and better search areas over the course of the search process (Eberhart and Shi, 2001, 2007; Clerc and Kennedy, 2002; Hu *et al.*, 2004).

In the PSO problem, a particle not only learns from its own experience, it also learns from its companions. It indicates that a particle's "moving position" is determined by its own experience and the neighbors' experience (Cheng *et al.*, 2011).

3.2 ACO

ACO is another type of SI, which takes inspiration from the foraging behavior of some ant species (Dorigo and Stutzle, 2004; Dorigo *et al.*, 1996; Dorigo and Gambardella, 1997). These ants deposit a chemical called pheromone on the ground, and other ants tend to choose routes with strong pheromone concentration. When an ant finds a short route, the signal of pheromone can mark some favorable path that should be followed by other members of the colony. ACO exploits a similar mechanism for solving optimization problems.

In the ACO problem, a group of artificial ants will build many solutions to an optimization problem at the same time, and the search information is exchanged on their quality (fitness) via a communication scheme.

The most important factor affecting a SI algorithm's performance may be its ability to explore and exploit the search areas. Exploration means the ability of a search algorithm to explore different areas of the search space in order to have high probability to find good promising solutions. Exploitation, on the other hand, means the ability to concentrate the search around a promising region in order to refine a candidate solution. A good optimization algorithm should optimally balance the two conflicted objectives.

4. SI in big data analytics

The big data analytics is a new research area of information processing, however, the problems of big data analytics have been studied in other research fields for decades under a different title. The rough association between big data analytics and SI (or more generally, computational intelligence (CI)) can be established and shown in Table I.

The characteristics of the big data analytics are summarized into several words with initial "V," which are volume, variety, velocity, veracity, and value. These complexities are a collection of different research problems that are existed for decades. Corresponding to the SI, the volume and the variety mean large scale and high-dimensional data; the velocity means data are rapidly changing, like an optimization problem in dynamic environment; the veracity means data are inconsistent and/or incomplete, like an optimization problem with noise or approximation; and the value is the objective of the big data analytics, like the fitness or objective function in an optimization problem.

The big data analytics is an extension of data mining techniques on a large amount of data. Data mining has been a popular academic topic in computer science and statistics for decades. The SI is a relatively new subfield of CI which studies the collective

Table I. The rough association between big data analytics and swarm intelligence

Big data analytics	Swarm intelligence
Volume	Large scale/high dimension
Variety	
Velocity	Dynamic environment
Veracity	Noise/uncertain/surrogates
Value	Fitness/objective

intelligence in a group of simple individuals. Like data mining, in the SI, useful information can be obtained from the competition and cooperation of individuals.

Generally, there are two types of approaches that apply the SI as data mining techniques (Martens *et al.*, 2011). The first category consists of techniques where individuals of a swarm move through a solution space and search for solution(s) for the data mining task, e.g., the parameter tuning. This is a search approach. In the second category, swarms help move and place data instances on a low-dimensional feature space in order to come up with a suitable clustering or low-dimensional mapping solution of the data, e.g., dimensionality reduction of the data. This is a data organizing approach.

The SI, especially PSO or ACO algorithms, can be used in data mining to solve single objective (Abraham *et al.*, 2006) and multiobjective problems (Coello *et al.*, 2009). Based on the two characteristics of the particle swarm (i.e. self-cognitive and social learning), the particle swarm has been utilized in data clustering techniques (Cohen and de Castro, 2006; Ahmadi *et al.*, 2007; Tsai and Kao, 2011; Xu *et al.*, 2012), document clustering (Cui *et al.*, 2005; Abraham *et al.*, 2006), variable weighting in clustering high-dimensional data (Lu *et al.*, 2011), semi-supervised learning-based text categorization (Cheng *et al.*, 2012b), and the web data mining (Pal *et al.*, 2002).

In a SI algorithm, there are several solutions exist at the same time. The premature convergence may happen due to the solution getting clustered together too fast. However, the solution clustering is not always harmful for the optimization. In a brain storm optimization algorithm, the clustering analysis is used to reveal the landscapes of problems and to guide the individuals to move toward the better and better areas (Shi, 2011a, b). Every individual in the brain storm optimization algorithm is not only a solution to the problem to be optimized, but also a data point to reveal the landscapes of the problem.

The data analysis techniques can also be used in the estimation of distribution algorithms (EDAs). In an EDAs, the distribution is estimated from a set of selected solutions, and then the estimated distribution model is used to generate new solutions (Zhang and Mühlenbein, 2004), i.e., the potential solutions are explored by establishing and sampling probabilistic models of promising candidate solutions (Hauschild and Pelikan, 2011). The SI and data mining techniques can be combined to produce benefits above and/or beyond what either method could achieve alone (Zhang *et al.*, 2011).

The big data analytics is required to manage an immense amount of data quickly (Rajaraman *et al.*, 2012); however, the dimension of data and the number of objective of problems also increase the "hardness" of problems. Four types of difficulties should be overcome to solve big data problems.

4.1 Handling large amount of data

The big data analysis requires a fast mining on a large scale data set, i.e., the immense amount of data should be processed in a limited time to reveal useful information. As the computing power improves, the more volume of data can be processed. The more data are retrieved and processed, the better understanding of problems can be obtained.

The analytic problem can be modeled as an optimization problem. The SI algorithms, – or more broadly, the evolutionary computation algorithms – are a search process based on the previous experiences. To reveal knowledge from a large volume of data within the big data context, the search ranges of the solved problem have to be widened and even extended to the extreme.

A quick scan is critical to solve the problem with massive data sets. The SI algorithms are also techniques based on the sampling of the search space. Through the meta-heuristics rules, data samples are chosen from the massive data space. From these representative data samples, the problem structure could be obtained. Based on the SI, we could find a good-enough solution with a high search speed to solve the problem with a large volume of data.

A large amount of data does not necessarily mean high-dimensional data, and a high volume of data can accumulate in single dimension such as high-frequency data sampled by sensors with higher resolutions.

4.2 Handling high-dimensional data

In general, the optimization problem concerns with finding the best available solution (s) for a given problem within allowable time, and the problem may have several or numerous optimal solutions, of which many are local optimal solutions. Normally, the problem will become more difficult with the growth of the number of variables and objectives. Specially, problems with a large number of variables, e.g., more than a thousand variables, are termed as large scale problems.

Many optimization methods suffer from the curse of dimensionality, which implies that their performance deteriorates quickly as the dimension of the search space increases (Hastie *et al.*, 2009; Domingos, 2012; Bellman, 1961; Lee and Verleysen, 2007; Cheng *et al.*, 2012a). There are several reasons that cause this phenomenon.

The solution space of a problem often increases exponentially with the problem dimension and thus more efficient search strategies are required to explore all promising regions within a given time budget. The evolutionary computation or SI is based on the interaction of a group of solutions. The promising regions or the landscape of problems are very difficult to reveal by small solution samples (compared with the number of all feasible solutions).

The "empty space phenomenon" gives an example that problems get harder when the dimension increases (Lee and Verleysen, 2007; Scott and Thompson, 1983; Verleysen, 2003). The number of possible solutions is increased exponentially when the dimension increases. There are m^n possible solutions in total for a problem with m possible solutions in each dimension (assume that each dimension has the same number of possible solutions). For example, when the m equals to 1,000, 100 samples cover 10 percent solutions for one dimensional problem. However, 100 samples only cover 0.01 percent solutions for two dimensional problems. For continuous problems, even we consider the computational accuracy, the number of possible solutions in one dimension is larger than 1,000. The percentage of data points in the solution space will decrease rapidly. The search performances of most algorithms are based on the previous search experiences. Considering the limitation of computational resources, the percentage of data points retrieved will be close to zero when the dimension increases to a large number. The performance of the algorithms is affected by the escalating dimensions of the problems.

The characteristics of a problem may also change with the scale. The problem will become more difficult and complex when the dimension increases. Rosenbrock's function, for instance, is unimodal for two dimensional problems but becomes multimodal for higher dimensional problems. Because of such a worsening of the features of an optimization problem resulting from an increase in scale, a previously successful search strategy may no longer be capable of finding an optimal solution. Fortunately, an approximate result with a high speed may be better than an accurate

result with a tardy speed. The SI algorithms can find a good-enough solution rapidly, which is the strength of the SI in solving the big data analytics problems.

The curse of dimensionality also happens on the high-dimensional data mining problems (Hastie *et al.*, 2009; Domingos, 2012; Donoho, 2000). Many algorithms' performance deteriorates quickly as the dimension of the data space increases. For example, the nearest neighbor approaches are very effective in the categorization problem. However, for high-dimensional data, it is very difficult to solve the similarity search problem due to the computational complexity, which was caused by the increase of the dimensionality.

Many methods are proposed on the high dimension data mining problems (Kriegel *et al.*, 2009). Transforming the high-dimensional mining problems into low-dimensional space via a projection operation is an effective way. The locality sensitive hashing algorithm is also proposed to find the nearest neighbors in the high-dimensional space (Datar *et al.*, 2004; Slaney and Casey, 2008). This algorithm is based on hashing functions with strong "local-sensitivity" in order to retrieve the nearest neighbors in a Euclidean space with a complexity sublinear with the amount of data.

A data mining problem can be modeled as an optimization problem, and thus the research results of the large scale optimization problems can also be transferred to data mining problems. In the SI or more generally the CI, many effective strategies are proposed for high-dimensional optimization problems, such as problem decomposition and subcomponents cooperation (Yang *et al.*, 2007), parameter adaptation (2011), and surrogate-based fitness evaluations (Jin, 2005; Jin and Sendhoff, 2009). Especially, the PSO or ACO algorithms can be used in the data mining to solve single objective (Abraham *et al.*, 2006) and multiobjective problems (MOPs) (Coello *et al.*, 2009).

In the SI, the problem of handing a large amount of data and/or high-dimensional data can be represented as large scale problems, i.e., problems with massive variables to be optimized. Based on the SI, an effective method could find good solutions for large scale problems, in terms of both the time complexity and the result accuracy.

4.3 Handling dynamical data

The big data, such as the web usage data of the internet and real-time traffic information, rapidly changes over time. The analytical algorithms need to process these data swiftly. The dynamic problems are sometimes also termed as non-stationary environment (Morrison and De Jong, 1999) or uncertain environment (Jin and Branke, 2005) problems. The SI has been widely applied to solve both stationary and dynamical optimization problems (Yang and Li, 2010; Li and Yang, 2012).

The SI often has to deal with the optimization problems in the presence of a wide range of uncertainties. Generally, uncertainties in the problems can be divided into the following categories:

- The fitness function or the processed data are noisy.
- The design variables and/or the environmental parameters may change after the optimization, and the quality of the obtained optimal solution should be robust against environmental changes or deviations from the optimal point.
- The fitness function is approximated (Jin, 2005), such as surrogate-based fitness evaluations. The fitness function suffers from the approximation errors.
- The optimum in the problem space may change over time. The algorithm should be able to track the optimum continuously.

- The optimization target may change over time. The computing demands need adjust to the dynamical environment. For example, there should be a balance between the computing efficiency and the power consumption for different computing loads.

In all these cases, additional measures must be taken so that the SI algorithms are still able to solve the dynamic problems satisfactorily (Jin and Branke, 2005; Bui *et al.*, 2012).

4.4 Handling multiobjective problems

Different sources of data are integrated in the big data research, and in most of the big data analytics problems, more than one objective need to be satisfied at the same time. According to the number of objectives, the optimization problems can be divided as single objective and MOPs. For the multiobjective problems, the traditional mathematical programming techniques have to perform a series of separate runs to satisfy different objectives (Coello *et al.*, 2007).

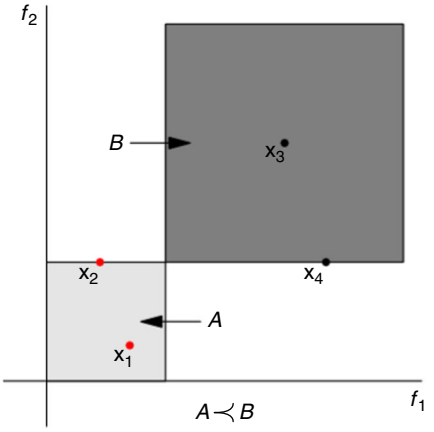
Multiobjective optimization refers to optimization problems that involve two or more objectives, and a set of solutions is sought instead of one (Coello *et al.*, 2007). A general multiobjective optimization problem can be described as a vector function **f** that maps a tuple of n parameters (decision variables) to a tuple of k objectives. Without loss of generality, minimization is assumed throughout this paper:

$$\text{minimize} \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))$$

$$\text{subject to} \quad \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbf{X}$$

$$\mathbf{y} = (y_1, y_2, \ldots, y_k) \in \mathbf{Y}$$

where $\mathbf{x}$ is called the decision vector, $\mathbf{X}$ is the decision space, $\mathbf{y}$ is the objective vector, and $\mathbf{Y}$ is the objective space, and $\mathbf{f}$: $\mathbf{X} \rightarrow \mathbf{Y}$ consists of k real-valued objective functions (Figure 2).



Note: Set A dominates the set B, and the points x_1 and x_2 dominate the point x_3 and x_4

Figure 2.
The example of domination

A decision vector, $\mathbf{x}_1$ dominates a decision vector, $\mathbf{x}_2$ (denoted by $\mathbf{x}_1 \prec \mathbf{x}_2$), if and only if:

- $\mathbf{x}_1$ is not worse than $\mathbf{x}_2$ in all objectives, i.e. $f_k(\mathbf{x}_1) \leqslant f_k(\mathbf{x}_2)$, $\forall k = 1, \ldots, n_k$; and
- $\mathbf{x}_1$ is strictly better than $\mathbf{x}_2$ in at least one objective, i.e. $\exists k = 1, \ldots, n_k$: $f_k(\mathbf{x}_1) < f_k(\mathbf{x}_2)$.

A point $\mathbf{x}^* \in \mathbf{X}$ is called Pareto optimal if there is no $\mathbf{x} \in \mathbf{X}$ such that $\mathbf{x}$ dominates $\mathbf{x}^*$. The set of all the Pareto optimal points is called the Pareto set (denoted as *PS*). The set of all the Pareto objective vectors, $PF = \{f(x) \in X | x \in PS\}$, is called the Pareto front (denoted as *PF*).

Unlike the single objective optimization, the MOPs have many or infinite solutions (Bosman and Thierens, 2003). The optimization goal of an MOP consists of three objectives:

- the distance of the resulting nondominated solutions to the true optimal Pareto front should be minimized;
- a good (in most cases uniform) distribution of the obtained solutions is desirable; and
- the spread of the obtained nondominated solutions should be maximized, i.e., for each objective a wide range of values should be covered by the nondominated solutions.

In a multiobjective optimization problem, we aim to find the set of optimal trade-off solutions known as the Pareto optimal set. Pareto optimality is defined with respect to the concept of nondominated points in the objective space. The MOPs can be effectively solved by the SI methods. Several new techniques are combined to solve MOPs with more than ten objectives, in which almost every solution is Pareto nondominated in the problems (Ishibuchi *et al.*, 2008). These techniques include objective decomposition (Zhang *et al.*, 2010), objective reduction (Brockhoff and Zitzler, 2009; Saxena *et al.*, 2013), and clustering in the objective space (Shi, 2011a, b).

5. An application

The big data are created in many areas in our everyday life. With the big data analytical techniques and SI methods, more effective applications or systems can be designed to solve real-world problems. The big data analytics problem not only occurs in web data mining and business intelligence, but also in complex engineering or design problems. The following example gives an introduction to the applications of the big data analytics in a real-world commodity routing problem in a real port in China (we call it XYZ port).

The XYZ port is one of the most important and busiest ports in China. The XYZ port comprises nine different ports. Containers are transported by trucks among these ports. The objective of the routing system is to transport the containers to the destination port more efficiently while completing each task according to the schedule. Many constraints have made this problem much harder to solve, for example, containers may delay in unpredictable times; drivers are shifted after at most 12 hours; the number of trucks is not fixed; each truck has the fixed capacity; and the truck needs repairing and maintenance every day. For each task record, it has several attributes, which are the source port, destination port, container quantity, container size, and a time window (a, b) indicating its available time and deadline (Chen *et al.*, 2013). Table II below gives several examples of commodity routing tasks. A task must be started after its import

time, and be completed before the finish time. The time item "20120207075715" means that the year 2012, date February 7, and time 07:57:15. The "small" and "large" items mean the different type of containers.

It is easy to solve a routing problem with few examples. However, the current problem needs to handle thousands of samples per day, which means thousands of tasks should be assigned at the beginning of a shift. The arrivals of new and emerging tasks also increase the complexities of this problem. Some trucks need to be re-arranged to the first emerging tasks. The current routing system is not efficient. All tasks are operated by the manual assignment. The historical data shows that the average number of trucks with the heavy loads is below 70 percent (Chen *et al.*, 2013). Besides that, the current routing system has several weaknesses. The weaknesses are as follows:

- The new task is not predictable. All tasks are assigned after it is noticed.
- The waiting time for a task is not predictable. When the freighter delays, a truck needs to wait until the container arrives.
- All assignments are assigned at the beginning of a shift. It is not easy to change each truck's assignment. The emerging tasks may be delayed when it is noticed after the task assignments.
- It is difficult to monitor each driver's work. To obtain more tasks, a driver may lie about the progress status of the current task.

To overcome the problems in current routing system, the new routing system based on the SI and big data analytics are designed. Monitoring all trucks' positions and assigning tasks dynamically to trucks is a straightforward method to improve the performance of all trucks. The Figure 3 shows the architecture of new vehicle routing system. Through the global positioning system devices in the smart phone, the trucks' positions are sent to vehicle routing system per 30 seconds. The trucks' real-time positions are displayed on the map for monitoring and management. The computing server will optimize the truck routes when the new containers arrive or the unexpected delays occur.

Index	Source	Destination	Import time	Finish time	Small	Large
T99028406	BLCTZS	BLCT2	20120207075715	20120209090000	13	2
T99028419	BLCTZS	BLCT3	20120207165615	20120209090000	0	1
T12028447	BLCTZS	BLCT	20120208160110	20120209120000	2	0
T12028448	BLCTZS	BLCT2	20120208160110	20120209120000	1	3
T12020069	BLCTYD	BLCTMS	20120207075715	20120209180000	1	0
T12020115	BLCTYD	BLCT2	20120208160110	20120209180000	0	7

Table II. The examples of commodity routing tasks

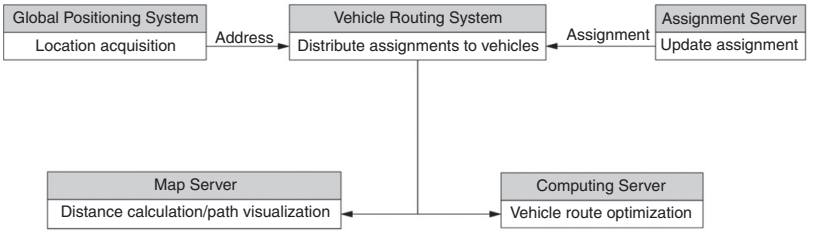


Figure 3. The architecture of vehicle routing system

The position information of every truck is stored in a database and analyzed by vehicle routing system. The trucks will generate massive position information every day, and the position information is dynamically changing in every minute. This large volume and dynamic data are difficult to handle for the big data analytics. Also, the data has multiple dimensions including port information, time information, container types, position information, and other unpredictable delays. There are at least two objectives: first, shortening the time and distance of the transportation; and second, meeting the schedule. The vehicle routing problem (VRP) can be modeled as a large scale, dynamical and constrained optimization problem. The objective function is to reduce the empty loading time for trucks and the waiting time, i.e., to enhance the efficiency of the vehicle routing. This could be formulated as a multiobjective problem as follows.

Let $a_{i,j}^t$ be the arrival time of vehicle i for jth task, $l_{i,j}^t$ be the leaving time. The import time for jth task is s_j^t, and f_j^t is finish time for jth task. The number of vehicles is N, the number of total tasks is M.

The travel time and travel distance is symmetric.

Objective 1: min empty load distance rate (ELDR) = empty loaded distance/total travel distance:

$$f_1(\mathbf{x}) = \sum_{i=1}^{N} Dist_i^e / \sum_{i=1}^{N} Dist_i$$

Objective 2: min total travel distance:

$$f_2(\mathbf{x}) = \sum_{i=1}^{N} Dist_i$$

Objective 3: min cost of fuels:

$$f_3(\mathbf{x}) = \sum_{i=1}^{N} fuel_i$$

Objective 4: min total waiting times:

$$f_4(\mathbf{x}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \left(s_j^t - a_{i,j}^t \right)$$

Objective 5: min environmental cost:

$$f_5(\mathbf{x}) = \sum_{i=1}^{N} envi_i$$

S.T.

(1) Truck constraint: the maximum routes←the number of available trucks; the number of trucks with assignment $n \leftarrow N$

(2) Task constraint: the time window of each task is satisfied, the start time is a soft constraint, and the finish time is a hard constraint:

$$a_{i,j}^t < s_j^t, \ \forall i \in \{1, \ldots, N\}, j \in \{1, \ldots, M\}$$

(3) Container constraint: the weight of containers should not exceed the capacity of a truck:

$$w_j \leqslant c_i,\ \forall i \in \{1,\ \ldots,\ N\}, j \in \{1,\ \ldots,\ M\}$$

(4) Service constraint: each task is serviced once only:

$$Task_j = 0, 1\ \forall j \in \{1,\ \ldots,\ M\}.$$

(5) Travel time constraint: vehicles should return to depot before the shift. Each shift is 12 hours. This is a soft constraint, but the over time should not be too long, such as less than 4 hrs. The difference of depot arrival time and depot leave time should not exceed the predefined shift time:

$$a^t_{i,depot} - l^t_{i,depot} \leftarrow S^t\ \forall i \in \{1,\ \ldots,\ N\}$$

(6) Dynamic constraint: the container may be delayed, and the delay times are unexpected:

$$s^t_{j,real} = s^t_j + delay$$

where delay is a random number that in a range of [0, 2] hours. The percentage of the containers that are delayed is 5 percent.

As shown in Table III, the SI techniques can be used to solve this problem. Specifically, as shown in Algorithm 1 with PSO and Algorithm 2 with artificial ACO (Karaboga, 2005; Karaboga and Basturk, 2008), starting with the most important objective (i.e. ELDR), these two algorithms can be used to solve the single objective problem. Then with the result of the first objective as an added constraint and the second priority (i.e. total travel distance) as the objective, a separate problem can be run with these two algorithms. We continue in this manner until the last run is finished. Based on the SI techniques, the commodity routing system will be more efficient and effective:

Algorithm 1. Procedure of PSO algorithm:

1: Initialize velocity and position randomly for each particle;
2: **While** the stopping criteria is not satisfied do
3: Calculate each particle's fitness value;
4: Determine each particle's best position, and the best position of entire swarm;
5: **For** each particle do
6: Update particle's velocity;
7: Update particle's position;
8: **End For**
9: **End while**

Algorithm 2. Procedure of artificial bee colony algorithm:

1: Initialize the set of food sources X_i, $i = 1,2, \ldots, SN$
2: Evaluate each X_i, $i = 1,2, \ldots, SN$
3: **While** termination condition is not met do
4: **For** $i = 1$ to SN
5: Generate U_i with X_i
6: Evaluate U_i

7: **If** $fit(U_i) \geqslant fit(X_i)$
8: $X_i = U_i$
9: **End If**
10: **End For**
11: **For** $i = 1$ to SN
12: Select an employed bee
13: Try to improve food source quality according to Step 5-Step 9
14: **End For**
15: Generate a new randomly food source for those does not improve with successive *limit* iterations
16: Memorize the best food source achieved so far
17: **End While**

In a summary, the big data and optimization are combined together in this case. Based on the data analytics, the optimization model could help the XYZ port obtain the largest profits. The position information of vehicles is a foundation of vehicle routing system – more generally, intelligent transportation system. From big data analytics on the vehicles' positions and other information, more rapid, safe, and more efficient transportation systems can be constructed.

Besides the above example, another case is the economic load dispatch problems in the optimal planning of modern power system. Due to large amount of data and the non-convex/non-smooth characteristics of objective functions and/or constraints, SI algorithms can be used to solve such problems effectively.

Since our case study is still in the modeling and data collection stage, we use SI approaches on benchmark problems to demonstrate the effectiveness and efficiency of the SI for solving big data problems. Usually big data problems have a huge number of decision variables; multiple, conflicting, non-convex/non-smooth objectives and constraints; and a huge volume of data (Sanders and Ganeshan, 2015; Chai *et al.*, 2013). The strength of SI algorithms could be illustrated by the results for solving benchmark problems with a huge number of decision variables. For simplicity, we take the experimental results of SIs algorithms, PSO algorithm in particular, on large scale problems as an illustration.

The big data problem is normally represented as a large scale optimization problem. Complexity, nonlinearity, and a large number of variables are the key factors that pose significant challenges in solving such problems (Cheng *et al.*, 2014). The experimental results of the three variants of PSO algorithms, which include cooperative coevolving particle swarm optimization (CCPSO2) (Li and Yao, 2012), competitive swarm optimizer

1	Generate random solutions for route problem, repair solutions if solutions not obey the constraints.
2	x←initialize_solution( )
3	Evaluate the initialize solution with archive updating
4	While running time ⩽ maximum computation time do
5	For all individuals in the swarm:
6	Update the solutions
7	Evaluate the fitness of each solutions
8	Select solutions with better fitness values
9	Update non-dominate solutions in the archive
10	End for
11	End while

Table III.
General procedure of swarm intelligence algorithms

(CSO) (Cheng and Jin, 2015), and dynamic multi-swarm particle swarm optimizer (DMS-PSO), solving seven benchmark problems with 1,000 decision variables, are given in Table IV. All algorithms were run 50 times and 5×10^6 fitness evaluations at each run (Li and Yao, 2012). f_1, f_4, and f_6 are separable functions, and f_2, f_3, f_5, and f_7 are non-separable functions (Li and Yao, 2012). These benchmark functions represent different levels of complexity and nonlinearity of the problems.

The PSO variants perform well in solving different kinds of problems, i.e., CCPSO2 performs best on f_4 and f_6, DMS-PSO performs best on f_1 and f_5, and CSO perform best on the other three problems. In Table IV, the global optimum of f_1 and f_5 are found via DMS-PSO algorithm, while for the other problems, the optima found are very close to the real optima. The solutions found by the PSO variants are good enough for these problems, which indicate that the SI algorithms are feasible methods on the problems with a huge number of decision variables.

The SI algorithms also perform well on the real-world applications, such as the Traveling Salesman Problem (TSP), VRP, and Arc Routing Problem, and so on. Taking TSP problem as an example, there is a well-known library TSPLIB, which contains 110 test instances of symmetric TSPs, and the number of nodes n ranging from 14 to 85,900 (Reinelt, 1991). In total, 193 kinds of evolutionary computation/SI algorithms are tested on these 110 instances. After one week computation on an eight core machine, 20 GB log files are generated. The experimental results show that the pure global optimization algorithms are outperformed by local search, but the hybrid algorithms performed obtain the best results among the tested algorithms (Weise *et al.*, 2014). In this real-world application, SI algorithms have shown good search performance on large scale global optimization problems.

From the above experimental results, the conclusions could be made that the SI algorithms could solve large scale problems effectively and efficiently. Based on the combination of SI and data mining techniques, we can have a better understanding of the big data analytics problems, and design more effective algorithms to solve real-world big data analytical problems.

6. Conclusion

The big data analytics problem is a hot and new topic. It has attracted more and more attentions currently. Most of the big data researches focus on the huge amount of data, however, handling the high-dimensional data and the multiple objectives are also important in solving big data problems. The big data analytics problem has many

Function	Name	CCPSO2 (Hastie *et al.*, 2009)	CSO (Domingos, 2012)	DMS-PSO (Domingos, 2012)
f_1	Shifted Sphere	5.18E−13 (9.61E−14)	1.09E−21 (4.20E−23)	0.00E+00 (0.00E+00)
f_2	Schwefel Problem	7.82E+01 (4.25E+01)	4.15E+01 (9.74E−01)	9.15E+01 (7.14E−01)
f_3	Shifted Rosenbrock	1.33E+03 (2.63E+02)	1.01E+03 (3.02E+01)	8.98E+09 (4.39E+08)
f_4	Shifted Rastrigin	1.99E−01 (4.06E−01)	6.89E+02 (3.10E+01)	3.84E+03 (1.71E+02)
f_5	Shifted Griewank	1.18E−03 (3.27E−03)	2.26E−16 (2.18E−17)	0.00E+00 (0.00E+00)
f_6	Shifted Ackley	1.02E−12 (1.68E−13)	1.21E−12 (2.64E−14)	7.76E+00 (8.92E−02)
f_7	Fast Fractal	−1.43E+04 (8.27E+01)	−3.83E+06 (4.82E+04)	−7.50E+03 (1.63E+01)

Table IV. Results (mean and standard deviations) of three PSO variants solving problems with 1,000 decision variables

difficulties, which have been researched separately for several years with different names, such as the high-dimensional problems, problems with massive data, dynamic problems, just to name a few. Due to the properties of big data problems, it is difficult to use some "divide-and-conquer" strategies to solve these problems. The SI algorithm is a new kind of computing and information techniques, which have obtained good performance on the search and optimization problems, especially for the problems that the traditional method cannot solve or is very difficult to solve.

In this paper, the association between big data analytics and SI techniques is discussed. The potential applications of the SI in the big data analytics and the big data analytics techniques in SI are analyzed. The big data analytics problems are divided into four elements: handling a large amount of data, handling high-dimensional data, handling dynamical data, and multiobjective optimization. Most real-world big data problems can be modeled as a large scale, dynamical, and multiobjective problems.

With the possible cross-fertilization of the two fields of big data analytics and the SI, we discussed an example of a real-world commodity routing problem in the XYZ port. The algorithm has been used to show the feasibility of the SI techniques. This paper is the first step in our research to utilize the SI algorithm to a big data analytics problem. Due to the complexity of the real commodity routing systems in the XYZ port, the project is an ongoing one. We only got a large static data set to model the real system and to verify the search ability of the SI algorithm. The initial results have shown that the empty loading rate is significantly reduced compared with the existing algorithm. However, it is difficult to replace the current sub-system with our algorithm. The first obstacle is that the test data set is a static one but the data are more dynamic and stochastic in real-world system (e.g. the vehicles may have some accidents). The second obstacle is that the current port system is very large and complex. The port system has many tasks and different technological processes. To avoid chaos in the port system, it may need many testing work to utilize a new method on the real system. In this research, we presented a comparison of different PSO methods on benchmark problems. Our future research will compare the performance of our method and fit it in a dynamic real system.

Another interesting instance is the economic load dispatch problem in the planning and design of modern power system. With large amount of data and the non-convex/non-smooth nature of objective functions and/or constraints, SI algorithms can solve such problems effectively. These examples could be understood as advanced IT or data processing technologies, however, their underlying mechanism could be the SI algorithms. With the applications of the SI, more rapid and effective methods can be designed to solve big data problems.

References

Abraham, A., Das, S. and Konar, A. (2006), "Document clustering using differential evolution", *Proceedings of the 2006 IEEE Congress on Evolutionary Computations (CEC 2006), July*, pp. 1784-1791.

Abraham, A., Grosan, C. and Ramos, V. (Eds) (2006), *Swarm Intelligence in Data Mining*, Studies in Computational Intelligence, Vol. 34, Springer, Berlin and Heidelberg.

Ahmadi, A., Karray, F. and Kamel, M. (2007), "Multiple cooperating swarms for data clustering", *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007), April*, pp. 206-212.

Alexander, F.J., Hoisie, A. and Szalay, A. (2011), "Big data", *Computing in Science & Engineering*, Vol. 13 No. 6, pp. 10-13.

Bellman, R. (1961), *Adaptive Control Processes: A Guided Tour*, Princeton University Press, Princeton, NJ.

Bosman, P.A.N. and Thierens, D. (2003), "The balance between proximity and diversity in multiobjective evolutionary algorithms", *IEEE Transactions on Evolutionary Computation*, Vol. 7 No. 2, pp. 174-188.

Brockhoff, D. and Zitzler, E. (2009), "Objective reduction in evolutionary multiobjective optimization: theory and applications", *Evolutionary Computation*, Vol. 17 No. 2, pp. 135-166.

Bui, L.T., Michalewicz, Z., Parkinson, E. and Abello, M.B. (2012), "Adaptation in dynamic environments: a case study in mission planning", *IEEE Transactions on Evolutionary Computation*, Vol. 16 No. 2, pp. 190-209.

Cervantes, A., Galván, I.M. and Isasi, P. (2009), "AMPSO: a new particle swarm method for nearest neighborhood classification", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 39 No. 5, pp. 1082-1091.

Chai, T., Jin, Y. and Sendhoff, B. (2013), "Evolutionary complex engineering optimization: opportunities and challenges", *IEEE Computational Intelligence Magazine*, Vol. 8 No. 3, pp. 12-15.

Chen, J., Bai, R., Qu, R. and Kendall, G. (2013), "A task based approach for a real-world commodity routing problem", *Proceedings of 2013 IEEE Symposium on Computational Intelligence in Production and Logistics Systems (CIPLS 2013), IEEE*, pp. 1-8.

Cheng, R. and Jin, Y. (2015), "A competitive swarm optimizer for large scale optimization", *IEEE Transactions on Cybernetics*, Vol. 45 No. 2, pp. 191-204.

Cheng, S., Shi, Y. and Qin, Q. (2011), "Experimental study on boundary constraints handling in particle swarm optimization: from population diversity perspective", *International Journal of Swarm Intelligence Research (IJSIR)*, Vol. 2 No. 3, pp. 43-69.

Cheng, S., Shi, Y. and Qin, Q. (2012a), "Dynamical exploitation space reduction in particle swarm optimization for solving large scale problems", *Proceedings of 2012 IEEE Congress on Evolutionary Computation, (CEC 2012), IEEE, Brisbane*, pp. 3030-3037.

Cheng, S., Shi, Y. and Qin, Q. (2012b), "Particle swarm optimization based semi-supervised learning on Chinese text categorization", *Proceedings of 2012 IEEE Congress on Evolutionary Computation (CEC 2012), IEEE, Brisbane*, pp. 3131-3198.

Cheng, S., Ting, T.O. and Yang, X.-S. (2014), "Large-scale global optimization via swarm intelligence", in Koziel, S., Leifsson, L. and Yang, X.-S. (Eds), *Solving Computationally Extensive Engineering Problems: Methods and Applications*, Springer Proceedings in Mathematics and Statistics, Springer International Publishing, Vol. 97, pp. 241-253.

Clerc, M. and Kennedy, J. (2002), "The particle swarm-explosion, stability, and convergence in a multidimensional complex space", *IEEE Transactions on Evolutionary Computation*, Vol. 6 No. 1, pp. 58-73.

Coello, C.A.C., Dehuri, S. and Ghosh, S. (Eds) (2009), *Swarm Intelligence for Multi-Objective Problems in Data Mining*, Studies in Computational Intelligence, Vol. 242, Springer, Berlin and Heidelberg.

Coello, C.A.C., Lamont, G.B. and Veldhuizen, D.A.V. (2007), *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed., Genetic and Evolutionary Computation Series, Springer, Science+Business Media, New York, NY.

Cohen, S.C.M. and de Castro, L.N. (2006), "Data clustering with particle swarms", *Proceedings of the 2006 IEEE Congress on Evolutionary Computations (CEC 2006), July*, pp. 1792-1798.

Cui, X., Potok, T.E. and Palathingal, P. (2005), "Document clustering using particle swarm optimization", *Proceedings of 2005 IEEE Swarm Intelligence Symposium (SIS 2005), June*, pp. 185-191.

Datar, M., Immorlica, N., Indyk, P. and Mirrokni, V.S. (2004), "Locality-sensitive hashing scheme based on p -stable distributions", in Snoeyink, J. and Boissonnat, J.-D. (Eds), *Proceedings of the 20th ACM Symposium on Computational Geometry Brooklyn*, ACM, New York, NY, pp. 253-262.

Domingos, P. (2012), "A few useful things to know about machine learning", *Communications of the ACM*, Vol. 55 No. 10, pp. 78-87.

Donoho, D.L. (2000), "Aide-memoire. High-dimensional data analysis: the curses and blessings of dimensionality", technical report, Stanford University, Stanford, CA, August.

Dorigo, M. and Gambardella, L.M. (1997), "Ant colony system: a cooperative learning approach to the traveling salesman problem", *IEEE Transactions on Evolutionary Computation*, Vol. 1 No. 1, pp. 53-66.

Dorigo, M. and Stutzle, T. (2004), *Ant Colony Optimization*, MIT Press, Cambridge, MA.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996), "Ant system: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 26 No. 1, pp. 29-41.

Eberhart, R. and Kennedy, J. (1995), "A new optimizer using particle swarm theory", *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39-43.

Eberhart, R. and Shi, Y. (2001), "Particle swarm optimization: developments, applications and resources", *Proceedings of the 2001 Congress on Evolutionary Computation (CEC2001)*, pp. 81-86.

Eberhart, R. and Shi, Y. (2007), *Computational Intelligence: Concepts to Implementations*, Morgan Kaufmann Publishers, San Francisco.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996), "From data mining to knowledge discovery in databases", *AI Magazine*, Vol. 17 No. 3, pp. 37-54.

Ficici, S.G. (2005), "Monotonic solution concepts in coevolution", *Genetic and Evolutionary Computation Conference (GECCO 2005), June*, pp. 499-506.

Hastie, T., Tibshirani, R. and Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer Series in Statistics, Springer, Science +Business Media, New York, NY.

Hauschild, M. and Pelikan, M. (2011), "An introduction and survey of estimation of distribution algorithms", *Swarm and Evolutionary Computation*, Vol. 1 No. 3, pp. 111-128.

Hu, X., Shi, Y. and Eberhart, R. (2004), "Recent advances in particle swarm", *Proceedings of the 2004 Congress on Evolutionary Computation (CEC2004)*, pp. 90-97.

Ishibuchi, H., Tsukamoto, N. and Nojima, Y. (2008), "Evolutionary many-objective optimization: a short review", *Proceedings of 2008 IEEE Congress on Evolutionary Computation (CEC2008), Hong Kong, June*, pp. 2424-2431.

Jin, Y. (2005), "A comprehensive survey of fitness approximation in evolutionary computation", *Soft Computing*, Vol. 9 No. 1, pp. 3-12.

Jin, Y. and Branke, J. (2005), "Evolutionary optimization in uncertain environments – a survey", *IEEE Transactions on Evolutionary Computation*, Vol. 9 No. 3, pp. 303-317.

Jin, Y. and Sendhoff, B. (2009), "A systems approach to evolutionary multiobjective structural optimization and beyond", *IEEE Computational Intelligence Magazine*, Vol. 4 No. 3, pp. 62-76.

Karaboga, D. (2005), "An idea based on honey bee swarm for numerical optimization", Technical Report No. TR-06, Engineering Faculty, Computer Engineering Department, Erciyes University, Kayseri.

Karaboga, D. and Basturk, B. (2008), "On the performance of artificial bee colony (ABC) algorithm", *Applied Soft Computing*, Vol. 8 No. 1, pp. 687-697.

Kennedy, J. and Eberhart, R. (1995), "Particle swarm optimization", *Proceedings of IEEE International Conference on Neural Networks (ICNN)*, pp. 1942-1948.

Kennedy, J., Eberhart, R. and Shi, Y. (2001), *Swarm Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA.

Kriegel, H.-P., Kroger, P. and Zimek, A. (2009), "Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering", *ACM Transactions on Knowledge Discovery from Data*, Vol. 13 No. 1, pp. 1-58.

Lee, J.A. and Verleysen, M. (2007), "Nonlinear dimensionality reduction", in Jordan, M., Kleinberg, J. and Scholkopf, B. (Eds), *Information Science and Statistics*, Springer, Science + Business Media, New York, NY, pp. 243-246.

Li, C. and Yang, S. (2012), "A general framework of multipopulation methods with clustering in undetectable dynamic environments", *IEEE Transaction on Evolutionary Computation*, Vol. 16 No. 4, pp. 556-577.

Li, X. and Yao, X. (2012), "Cooperatively coevolving particle swarms for large scale optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 16 No. 2, pp. 210-224.

Lu, Y., Wang, S., Li, S. and Zhou, C. (2011), "Particle swarm optimizer for variable weighting in clustering high-dimensional data", *Machine Learning*, Vol. 82 No. 1, pp. 43-70.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. and Byers, A.H. (2011), "Big data: the next frontier for innovation, competition, and productivity", technical report, McKinsey Global Institute, San Francisco, CA, May.

Martens, D., Baesens, B. and Fawcett, T. (2011), "Editorial survey: swarm intelligence for data mining", *Machine Learning*, Vol. 82 No. 1, pp. 1-42.

Morrison, R.W. and De Jong, K.A. (1999), "A test problem generator for non-stationary environments", *Proceedings of the 1999 Congress on Evolutionary Computation (CEC 1999), Vol. 3, July*, pp. 2047-2053.

Murphy, K.P. (2012), *Machine Learning: A Probabilistic Perspective*, Adaptive Computation and Machine Learning Series, The MIT Press, Cambridge, MA.

Pal, S.K., Talwar, V. and Mitra, P. (2002), "Web mining in soft computing framework: relevance, state of the art and future directions", *IEEE Transactions on Neural Networks*, Vol. 13 No. 5, pp. 1163-1177.

Rajaraman, A., Leskovec, J. and Ullman, J.D. (2012), *Mining of Massive Datasets*, Cambridge University Press, Cambridge, MA.

Reinelt, G. (1991), "TSPLIB – a traveling salesman problem library", *ORSA Journal on Computing*, Vol. 3 No. 4, pp. 376-384.

Sanders, N.R. and Ganeshan, R. (2015), "Special issue of production and operations management on 'big data in supply chain management'", *Production and Operations Management*, Vol. 24 No. 3, pp. 519-520.

Saxena, D.K., Duro, J.A., Tiwari, A., Deb, K. and Zhang, Q. (2013), "Objective reduction in many-objective optimization: linear and nonlinear algorithms", *IEEE Transactions on Evolutionary Computation*, Vol. 17 No. 1, pp. 77-99.

Scott, D.W. and Thompson, J.R. (1983), "Probability density estimation in higher dimensions", in Gentle, J.E. (Ed.), *Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface*, Elsevier Science Publishers, North-Holland, pp. 173-179.

Shi, Y. (2011a), "An optimization algorithm based on brainstorming process", *International Journal of Swarm Intelligence Research (IJSIR)*, Vol. 2 No. 4, pp. 35-62.

Shi, Y. (2011b), "Brain storm optimization algorithm", in Tan, Y., Shi, Y., Chai, Y. and Wang, G. (Eds), *Advances in Swarm Intelligence*, Vol. 6728, lecture notes in computer science, Springer, Berlin and Heidelberg, pp. 303-309.

Slaney, M. and Casey, M. (2008), "Locality-sensitive hashing for finding nearest neighbors", *IEEE Signal Processing Magazine*, Vol. 25 No. 2, pp. 128-131.

Tan, P.N., Steinbach, M. and Kumar, V. (2005), *Introduction to Data Mining*, Addison Wesley, Boston, MA.

Tsai, C.-Y. and Kao, I.-W. (2011), "Particle swarm optimization with selective particle regeneration for data clustering", *Expert Systems with Applications*, Vol. 38 No. 6, pp. 6565-6576.

Verleysen, M. (2003), "Learning high-dimensional data", in Ablameyko, S., Gori, M., Goras, L. and Piuri, V. (Eds), *Limitations and Future Trends in Neural Computation*, Vol. 186, NATO Science Series, III, Computer and Systems Sciences, IOS Press, Amsterdam, pp. 141-162.

Weise, T., Chiong, R., Lässig, J., Tang, K., Tsutsui, S., Chen, W., Michalewicz, Z. and Yao, X. (2014), "Benchmarking optimization algorithms: an open source framework for the traveling salesman problem", *IEEE Computational Intelligence Magazine*, Vol. 9 No. 3, pp. 40-52.

Xu, R., Xu, J. and Wunsch, D.C. II (2012), "A comparison study of validity indices on swarm-intelligence-based clustering", *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol. 42 No. 4, pp. 1243-1256.

Yang, S. and Li, C. (2010), "A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments", *IEEE Transaction on Evolutionary Computation*, Vol. 14 No. 6, pp. 959-974.

Yang, Z., Tang, K. and Yao, X. (2007), "Differential evolution for high-dimensional function optimization", *Proceedings of 2007 IEEE Congress on Evolutionary Computation (CEC 2007), IEEE*, pp. 3523-3530.

Zhang, J., Zhan, Z., Lin, Y., Chen, N., Gong, Y., Zhong, J., Chung, H., Li, Y. and Shi, Y. (2011), "Evolutionary computation meets machine learning: a survey", *IEEE Computational Intelligence Magazine*, Vol. 6 No. 4, pp. 68-75.

Zhang, Q. and Mühlenbein, H. (2004), "On the convergence of a class of estimation of distribution algorithms", *IEEE Transactions on Evolutionary Computation*, Vol. 8 No. 2, pp. 127-136.

Zhang, Q., Liu, W., Tsang, E. and Virginas, B. (2010), "Expensive multiobjective optimization by MOEA/D with Gaussian process model", *IEEE Transactions on Evolutionary Computation*, Vol. 3 No. 14, pp. 456-474.

Further Reading

Yang, Z., Tang, K. and Yao, X. (2011), "Scalability of generalized adaptive differential evolution for large-scale continuous optimization", *Soft Computing*, Vol. 15 No. 11, pp. 2141-2155.

Corresponding author
Qingyu Zhang can be contacted at: q.yu.zhang@gmail.com