Robotics and Computer-Integrated Manufacturing 38 (2016) 42–51

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

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

Review

Estimating simulation workload in cloud manufacturing using a classifying artificial neural network ensemble approach

Toly Chen [a], Yu-Cheng Wang [b,*]

[a] Department of Industrial Engineering and Systems Management, Feng Chia University, Taiwan
[b] Department of Aviation Mechanical Engineering, China University of Science and Technology, Taiwan

ARTICLE INFO

Article history:
Received 9 June 2015
Received in revised form
17 August 2015
Accepted 9 September 2015
Available online 23 October 2015

Keywords:
Cloud manufacturing
Simulation
k-means
Artificial neural network
Ensemble
Workload estimation

ABSTRACT

Cloud manufacturing (CMfg) is an extension of cloud computing in the manufacturing sector. The CMfg concept of simulating a factory online by using Web services is a topic of interest. To distribute a simulation workload evenly among simulation clouds, a simulation task is typically decomposed into small parts that are simultaneously processed. Therefore, the time required to complete a simulation task must be estimated in advance. However, this topic is seldom discussed. In this paper, a classifying artificial neural network (ANN) ensemble approach is proposed for estimating the required time for a simulation task. In the proposed methodology, simulation tasks are classified using *k*-means before their simulation times are estimated. Subsequently, for each task category, an ANN is constructed to estimate the required task time in the category. However, to reduce the impact of ANN overfitting, the required time for each simulation task is estimated using the ANNs of all categories, and the estimation results are then weighted and summed. Thus, the ANNs form an ensemble. In addition to the proposed methodology, six statistical and soft computing methods were applied in real tasks. According to the experimental results, compared with the six existing methods, the proposed methodology reduced the estimation time considerably. In addition, this advantage was statistically significant according to the results of the paired *t* test.

Contents

1. Introduction

In the manufacturing sector, cloud manufacturing (CMfg) is an extension of cloud computing that provides factories worldwide with ubiquitous access to manufacturing resources, such as online management information and decision support systems, virtual capacity networks, online factory simulation services, toolkits for converting manufacturing execution systems, and online collaborative design systems [25,40]. Chen and Wang [15] listed concerns regarding CMfg, namely interoperability, scalability, data security, business models, and innovative applications [19,39]. An innovative application of CMfg is to simulate a factory online by using Web services, that is, a cloud-based factory simulation (CFS)

* Corresponding author.

http://dx.doi.org/10.1016/j.rcim.2015.09.011
0736-5845/

system. A CFS system is a type of cloud-based cyber-physical system [17] that decomposes a simulation task (including the simulation model and options) into parts that can be simultaneously processed using multiple simulation clouds. A CFS system can be implemented at the following levels [12]: replicating the same simulation on several clouds, considering different possible values for uncertain or stochastic parameters, evaluating the performance of different scheduling methods, and partitioning the factory simulation model. This study examined simulation workload estimation, which is critical to operations at the first level.

To distribute a simulation workload evenly among simulation clouds, a simulation task is typically decomposed into small parts that are simultaneously processed. Therefore, the time required for fulfilling a simulation task must be estimated in advance. In addition, estimating the simulation time of a task facilitates determining the number of clouds that are required for meeting the deadline set by a client.

In the limited relevant literature, Fujimoto [22] compared the effectiveness of various distributed simulation strategies. He concluded that the hybrid use of deadlock avoidance, deadlock detection, and recovery techniques can provide significant speedups relative to sequential event list implementation. Ferscha and Tripathi [20] described five levels of parallel and distributed simulation: application, subroutine, component, centralized event, and decentralized event levels. Amin and Vinnakota [3] established a distribution of the fault simulation time. They concluded that the number of inputs and the time required for evaluating a gate are dominant factors for establishing the simulation time. In addition, the relationship between these factors and the simulation time can be approximated using an exponential function. Majumdar and Ochieng [29] performed a multivariate analysis to determine factors that influence the workload on an air traffic controller by using simulations. They concluded that the hybrid use of principal component and factor analyses yields high performance. Meng and Zhao [30] performed a simulation experiment involving 3D coupled thermomechanical finite element analyses and identified three factors that influence the simulation time: mass scaling, time scaling, and remeshing sweeps per increment. Chen and Lin [13] proposed a fuzzy collaborative forecasting method for estimating the required time for factory simulation, which was particularly useful for establishing an upper bound on the simulation time. However, this method was linear and might not be sufficiently accurate when applied to complex tasks.

To further enhance the effectiveness of estimating the time required for simulating a factory, a artificial neural network (ANN) classification ensemble was used in this study. In the proposed methodology, simulation tasks are classified using *k*-means (*k*M) before their simulation times are estimated. Subsequently, an ANN is constructed for each category to estimate the required time for tasks in the category. Amin and Vinnakota [3] noted that nonlinear methods such as ANNs provide a close approximation of the relationship between the decisive factors and the simulation time. However, ANN-based approaches are typically affected by overfitting [35]. To reduce the impact of ANN overfitting on the estimation performance, the required time for each simulation task is estimated using the ANNs of all categories. The estimation results are then weighted and summed. Thus, the ANNs form an ensemble.

The remainder of this paper is organized as follows. Section 2 describes the three steps of the proposed methodology: classifying tasks by using *k*M, estimating the simulation time by using ANNs, and forming an ANN ensemble. In Section 3, data on real tasks are used for evaluating the estimation performance of the proposed methodology. In addition, six statistical and soft computing approaches are applied to these tasks for comparison. Finally, Section 4 presents the conclusion and provides directions for future research.

2. Methodology

The proposed methodology is a classifying ANN ensemble approach that estimates the required execution time for a simulation task according to the attributes of the simulation model. The objective of this study was to optimize the estimation accuracy; in other words, the estimation result should be as close to the actual value as possible. The estimation accuracy can be measured using the mean absolute error (MAE), mean absolute percentage error (MAPE), or root mean squared error (RMSE). All the performance measures are smaller-the-better indices.

2.1. Task classification by using kM

First, the rationale for combining *k*M and ANN to estimate the simulation workload is explained as follows. Theoretically, a well-trained ANN (an ANN that does not converge on local minima) with a satisfactory selected topology can successfully map any complex relationships [31]. However, estimating the simulation workload is a complex problem, and the results of some previous studies (e.g., [15]) have shown the incapability of an ANN in solving such a problem. By contrast, classification-based methods, such as classification and regression tree (CART) and case-based reasoning (CBR), achieved fair performance in some aspects [15]. This is because multiple production environments are available for simulation. These production environments might vary (even for manufacturing similar products), but can roughly be classified into some common types, such as job shops, assembly lines, re-entrance production systems, and warehouse operations. Therefore, classifying a simulation task before estimating the simulation time seems to be a reasonable treatment. Hence, *k*M is applied in the proposed methodology [28] to minimize the within-group variability and maximize the between-group differences. Compared with other classifiers, such as self-organization map (SOM) and FCM, *k*M is relatively easier to implement and can be easily integrated with other modules, making it particularly suitable for online applications [23]. In addition, *k*M has been applied to classify manufacturing data such as adopted advanced manufacturing technologies [16] and job attributes [10].

2.2. Steps of kM

Recently, Jing et al. [26] proposed an entropy weighting *k*M that minimizes the within-cluster compactness and maximizes the negative weight entropy for stimulating more features that contribute to the identification of a cluster. *k*M can be implemented in the following steps:

(1) Normalize the data

$$N(x_{jq}) = \frac{x_{jq} - \min\limits_{j} x_{jq}}{\max\limits_{j} x_{jq} - \min\limits_{j} x_{jq}} \tag{1}$$

where x_{jq} is the qth attribute of task j; $j=1-n$, $q=1-Q$.

(2) Produce a preliminary classification result.

(3) (Iterations) Calculate the centroid of each category as follows:

$$c_k = \{c_{kq}\},\ k = 1 - K \tag{2}$$

$$c_{kq} = \sum_{j \in k} x_{jq} / \sum_{j \in k} 1, \quad k = 1 - K, \quad q = 1 - Q \tag{3}$$

(4) Remeasure the Euclidean distance from each task to the centroid of each category.