Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/070545

A Hybrid GA-TS Algorithm for Optimizing Networked Manufacturing Resources Configuration

Yan Zhan**, Jiansha Lu and Shiyun Li*

College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou, Zhejiang, 310014, China

Received: 1 Nov. 2012, Revised: 3 Mar. 2013, Accepted: 8 Mar. 2013
Published online: 1 Sep. 2013

Abstract: In recent years, more and more researchers have focused on various significant issues of resource configuration, such as maximum computing performance and minimum response time, especially taking existing resource utilization into account. The goal of this research is to design a resource configuration optimization system under networked manufacturing environment. A prediction mechanism is realized by using support vector regression (SVR) to estimate resource utilization according to network protocol of each manufacturing process, while redistributing resources based on the current status of existing resources pool. A resource configuration mechanism applying genetic algorithm working along with the tabu search process (GA-TS) is proposed in this study to determine the redistribution of resources. The experimental results show that the proposed scheme achieves an effective configuration via reaching the balance between the utilization of resources within existing resources and network protocol of each manufacturing process between potential resource requirements and the network resource providers.

Keywords: Networked manufacturing resources configuration, support vector regression, genetic algorithm, tabu search, resources requirement prediction module

1 Introduction

The world-wide vertical and horizontal mergers wave clearly shows that global networked manufacturing has become the development trend in the 21st century recently [1, 2]. Supply chains are often modeled as a multi-stage production and inventory network under a periodically reviewed base-stock policy. Deciding what option should be used at each node and deciding where inventory (resources) should be placed among these nodes is what Graves and Willems refer to as supply chain configuration (SCC) [3], especially for networked manufacturing resources configuration (NMRC). In fact, a network representation provides such a powerful visual and conceptual aid for portraying the relationships between the components of systems that it is used in virtually every field of scientific, social and economic endeavor. One of the most significant developments in operational research in recent years has been the rapid advance in both the methodology and application of network optimization models. A number of algorithms have had a major impact and are being used to solve huge problems on a routine basis that would have been completely intractable two or three decades ago.

A structured NMRC scheme, handling changes systematically so that a system maintains its integrity over time, ensures that resources (e.g., requirements, design, and test) for manufacturing are accurate and consistent with the actual physical operations, which can be divided into five fundamental components as follows:(i) resource pool;(ii) resource leveling;(iii) tracking system;(iv) configuration scheme;(v) resource requirement. The goal of scheme planning is to achieve adequate utilization at the lowest reasonable cost, when weighted by important metrics and subject to constraints, for example: meeting a minimum service level. The main objective of resource leveling is to smooth the stock of resources on hand, reducing both excess inventories and shortages. The required data are: the demands for various resources, forecast by time period for the planning horizon, as well as the resources' configurations required in those demands, and the supply of the resources, again forecast by time period for the planning horizon. Tracking planning is an important part of NMRC scheme planning, which track requirements throughout the life cycle through acceptance and operations and maintenance to determine an optimal network configuration according to

* Corresponding author e-mail: yzhan@zjut.edu.cn changes growth, the scheduling of activities and the resources required by those activities while taking into consideration both the resource availability and system economy.

In general, tracking network planning is generally divided into two stages: (i) scheme formation; (ii) scheme evaluation. At the former stage, the topology and capacity of the tracking lines are determined, whereas at the latter stage, an analysis is performed of the network's characteristics, such as change flow, inadequate tracking capacity, stability analysis, reliability, etc. Many well-established methods of analysis have been applied at the scheme evaluation stage and new ones are under constant development [4–6]. Scheme formation can be a complex task with many constraints. Optimal network designs are important because they can result in large cost savings. Clearly, further work is needed to develop methods for the design of networks that are as near to optimal as is possible.

Networked manufacturing resources configuration is a complex mathematical optimization problem that is mostly applied in the form of some rearrangement of manufacturing resources, which have their own different usages in problem solving and decision making [2]. Different resources configuration strategies for making selections will result in different costs and levels of capacity utilization. Therefore, it is a demanding task to find a sub-optimal resources configuration strategy, especially for limited resources aiming at each type of goal. This research work focused on the application of Evolutionary Algorithms (EA) under network environment. Numerous researchers have proposed genetic algorithm (GA) and tabu search (TS) to deal with optimization problems, and the schema theorem for GA proposed by Holland illustrated that GA is a robust searching approach. And TS proposed by Glover and Laguna is a well-known meta-heuristic that guides a local heuristic search procedure to explore the solution space beyond local optimality.

GA is not only an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetic, but also represents an intelligent exploitation of a random search in a vase search space. Compared with other heuristic methods, its main advantage is that it only needs a fitness function to evaluate the quality of different solutions and there is no necessary to offer a particular algorithm to solve a given problem. Topcuoglu et al. presented a GA-based method for the uncapacitated single allocation hub location problem [7]. Randaccio et al. applied a GA-based solution to find the appropriate combination of the trees to comply with the bandwidth needs of the group of multicast sessions simultaneously, while allowing the operator to find the desired balance between quality of service and network resource utilization [8]. Feng et al. presented a method to select members from different departments to resolve a manpower distribution problem by using an improved non-dominated sorting genetic algorithm (INSGA) [9]. In recent years, TS has emerged as a highly efficient, search paradigm for finding high quality solutions quickly to combinative problems [10].

However, GAs often takes a lot of time in order to find the global optimum [11–15], and some studies have utilized a hybrid genetic algorithm to improve its performance. The drawback of TS is that its effectiveness depends very much on the strategy for manipulation of the tabu list. Clearly, how to specify the size of the tabu list in the searching process plays an important role in the search for good solutions. Smirnov et al. used genetic algorithms (GA) to find sub-optimal solution applying the theory of games with fuzzy coalitions, and constraint satisfaction problem solving for resource allocation task [16]. Cantarella et al. specified some metaheuristics (Hill Climbing, Simulated Annealing, Tabu Search, Genetic Algorithms and Path Relining) for urban road network topology design singularly or jointly. More tests are required for more accurate calibration of the parameters by using other transportation systems [17].Lin et al. used a design point's relative efficiency score from data envelopment analysis(DEA) as its fitness value in the selection operation of genetic algorithm(GA) to determine optimal resource levels in surgical services [18]. Schneider et al. also developed a Flexibility Design Genetic Algorithm (FGA) that exploits qualitative insights into the structure of good solutions, such as the well-established chaining principle, to enhance its performance [19].

A support vector machine (SVM) is a supervised learning technique from the field of machine learning applicable to both classification and regression. Rooted in the Statistical Learning Theory, SVMs are based on the principle of structural risk minimization (SRM), which attempts to minimize an upper threshold on the generalization rather than minimize the training error, and is expected to perform better than the traditional empirical risk minimization (ERM) approach. Support vector machines (SVMs) have a novel neural network algorithm based on statistical learning theory. Due to the SVMs structure risk minimization principles, they have great generalization ability and provide superior performance in practical applications. With the introduction of Vapnik's ε-insensitivity loss function, the regression model for SVMs, called SVR, has also received increasing attention in the solution of nonlinear estimation problems. It has been successfully applied to problems of time series prediction, such as production value forecast of the machinery industry [20, 21]. Yang et al. proposed the localized support vector regression (LSVR) model to improve the performance of the standard SVR model for time series prediction, by offering a systematic and automatic scheme to adapt the margin locally and flexibly. The experimental results on synthetic data and real financial data demonstrate its advantages over the standard SVR [22]. In addition, Zou et al. compared three regression approaches, including SVR, Artificial Neural Networks (ANNs), and Partial

Least Squares (PLS), in quantitative analysis of components of solid pharmaceutical samples on near-infrared spectroscopy, and the results showed that SVR obtained better performance in terms of parameter-selecting process than those obtained with the traditional methods [23]. Although utilizing SVR in time series prediction has yielded many successful results, there has been far less research on the application of SVR to demand forecasting problems. Pai et al.'s work on a support vector regression model with scaling preprocessing and marriage in honey-bee optimization, however, suggests that SVR can be used to accurately forecast developments in the IC industry [21].

This paper aims to consider the efficiency and optimization of resource configuration under networked manufacturing environment. A resources requirement prediction module built with SVR is used to evaluate resource utilization according to network protocol of each manufacturing process. When potential resources are demanded, the system will consider the individual loading to precede the prediction with SVR. If the result needs to change the configuration strategies, the scheme will utilize GA-TS at this stage to try its best to achieve the global deployment of the resources effectively, including creating or collecting resources, to make sure that the proposed scheme can satisfy the network protocol requested by the customers. The remainder of this paper is organized as follows. Section 2 formulates the algorithm for the proposed resource configuration mechanism. Section 3 gives comparative results between the non-optimized approach and the optimized approach proposed in this paper. Conclusions are given in Section 4.

2 Resource Management Pattern

Fig.2.1 illustrates the architecture of the resource configuration scheme under networked manufacturing environment proposed in this paper. An operation resources pool is used to collect all operation resources provided by the networked manufacturing, recording the overall utilization of system resources. In addition, an existing resources pool is used to provide resources, material, information, capital, or knowledge, for the network nodes; and two query tables, including the remaining resource table and the resource utilization rate table, are used to assist in determining the strategies of increasing or decreasing potential resources required by each manufacturing process from network nodes. Here, a resources requirement prediction module built with Support Vector Regressions (SVRs) is used to estimate resource utilization according to network protocol of each manufacturing process. Meanwhile, a networked resources configuration module applied with the hybrid GA-TS algorithm is utilized to redistribute the aggregated resources to route tasks to the most suitable execution nodes, including creating or collecting resources, allocating tasks to idle nodes, to make sure that the proposed scheme can minimize the global makespan and fulfill deadline requirements from the customers.

2.1 Resources Requirement Prediction Module

This module mainly estimates the resource requirements of manufacturing process to remind the system to create or collect resources in accordance with the reality of networked manufacturing. Furthermore, a well-known time series predictor, namely SVR, is embedded in this module to evaluate resource utilization according to network protocol of each manufacturing process with the assistance of two query tables, that is, remaining resource table, which records all related usable resources along the manufacturing processes, and resource utilization rate table, which stores the utilization rate of each type of resources in the manufacturing system. With the above-mentioned methods, this module will determine whether the resources should be increased or decreased for the requests from the network protocol.

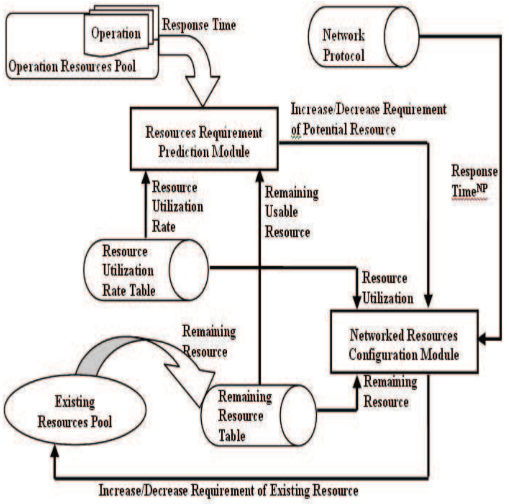
Operation
Operation Resources Pool
Response Time
Network Protocol
Resources Requirement Prediction Module
Increase/Decrease Requirement of Potential Resource
Response Time^NP
Resource Utilization Rate
Remaining Usable Resource
Resource Utilization Rate Table
Resource Utilization
Networked Resources Configuration Module
Remaining Resource
Existing Resources Pool
Remaining Resource Table
Remaining Resource
Increase/Decrease Requirement of Existing Resource

Figure 2.1: The structure of the resource configuration scheme

The model produced by SVR depends only on a subset of the training data, because the cost function for building the model ignores any training data close to the model prediction (within a threshold). To solve a nonlinear regression or functional approximation problem using SVR, the inputs are first nonlinearly mapped into a high dimensional feature space (F) wherein they are correlated linearly with the outputs. The parameters can then be derived by solving a quadratic programming problem with linear equality and inequality constraints.

Consider a training data set $G = \{(x_i, q_i), i = 1, ..., l\}$, where $x_i \in \Re^l$,and $q_i \in \Re$.The input x_i terms are l-dimensional vectors, and the system response q_i terms are continuous values. The SVR attempts to approximate the following function using data set G:

$$f(x,w)=\sum_{i=1}^{l} w_i\times\varphi_i(x)+b \tag{1}$$

where b denotes the bias term, and the w_i terms represent the subjects of learning. Furthermore, a mapping $z=\Phi(x)$ is selected in advance to map input vectors x into a higher-dimensional feature space F, which is spanned by a set of fixed functions $\varphi_i(x)$.

By defining a linear loss function with the following ε-insensitivity zone as illustrated in Fig.2.2:

$$|q_i-f(x_i,w)|_\varepsilon=\begin{cases}0; \quad if \quad |q_i-f(x_i,w)|\le\varepsilon\\ |q_i-f(x_i,w)|-\varepsilon; otherwise\end{cases} \tag{2}$$

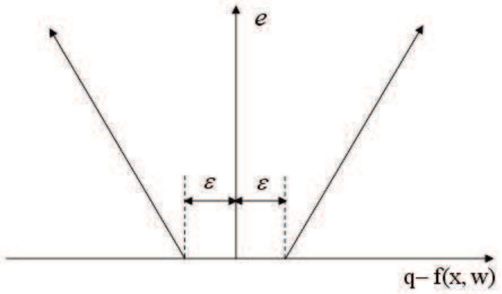

Figure 2.2: A plot of ε-insensitive loss function

The w_i terms in Eq.(1)can be estimated by minimizing the risk:

$$R=\frac{1}{2}||w||^2+\frac{C}{l}(\sum_{i=1}^{l}|q_i-f(x_i,w)|_\varepsilon) \tag{3}$$

where C denotes a user-chosen penalty parameter that determines the trade-off between the training error and VC-dimension of the SVR model. Significantly, the VC-dimension is a scalar value that measures the capacity of a set of functions [24]. Eq.(3)can be further derived as the following constrained optimization problem:

$$R(w,\zeta,\zeta^*)=\frac{1}{2}||w||^2+\frac{C}{l}(\sum_{i=1}^{l}\zeta_i+\sum_{i=1}^{l}\zeta_i^*) \tag{4}$$

subject to constraints:

$$\begin{cases}q_i-w^Tx_i-b\le\varepsilon+\zeta_i^*\\ w^Tx_i+b-q_i\le\varepsilon+\zeta_i\\ \zeta_i,\zeta_i^*\ge 0\end{cases} \tag{5}$$

where ζ_i and ζ_i^* denote the respective measurements above and below the zone with the radius ε in Vapnik's loss function as given in Eq.(2).

Schölkopf et al. [25] developed a modification of Vapnik's original SVR algorithm, called v-SVR, and claimed that it can automatically minimize the radius ε . Lagrange multiplier methods can be employed to demonstrate that the constrained optimization problem in Eqs.(4) and (5) maximizes the solution of the following equation:

$$W(\alpha,\alpha^*)=\sum_{i=1}^{l}(\alpha_i^*-\alpha_i)q_i-\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)k(x_i,x_j) \tag{6}$$

subject to

$$\begin{cases}\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0\\ 0\le\alpha_i,\alpha_i^*\le\frac{C}{l}, i=1,...,l\\ \sum_{i=1}^{l}(\alpha_i^*+\alpha_i)\le C\times v\end{cases} \tag{7}$$

where (α_i,α_i^*) denotes one of l Lagrange multiplier pairs; C represents a regularization constant specified a priori; v is a constant greater than or equal to zero, and $k(x_i,x_j)$ denotes normally a Gaussian kernel or polynomial kernel. The regression estimate can be shown to take the form

$$f(x)=\sum_{i=1}^{l}(\alpha_i^*-\alpha_i)\times k(x_i,x_j)+b \tag{8}$$

where b denotes the optimal bias.

2.2 Networked Resources Configuration Module

As shown in Fig. 2.1, this module collects the information from the resources requirement prediction module and two query tables, along with network protocol requested by the different network nodes, to achieve optimal resource utilization and fairness among competing end-to-end flows. A hybrid GA-TS algorithm is applied in this paper to deal with the NMRC optimization problem, and the fitness function is designed in accordance with the reality of networked manufacturing.

Fig.2.3 shows the procedures of the SVR model with GA-TS algorithm. We first transfer the potential resources which are established in the existing resource into a binary code as the initial population, called chromosomes. Each element in the chromosome is either 0 or 1, and the higher fitness value will be kept to generate the next generation during the procedure of recombination and mutation. The new generation will run the same steps as their parents did until the stop criteria are satisfied. In addition, we do not need to set fixed time interval to activate GA because the system will proceed with the adjustment strategies according to the real-time demand of potential resources. Once a new request of potential resources arrives, the system will run GA to adjust the overall configuration of the resources.

2.2.1 Conventional genetic algorithm (CGA)

The binary coded Conventional GA (CGA) with one-point crossover is described below.

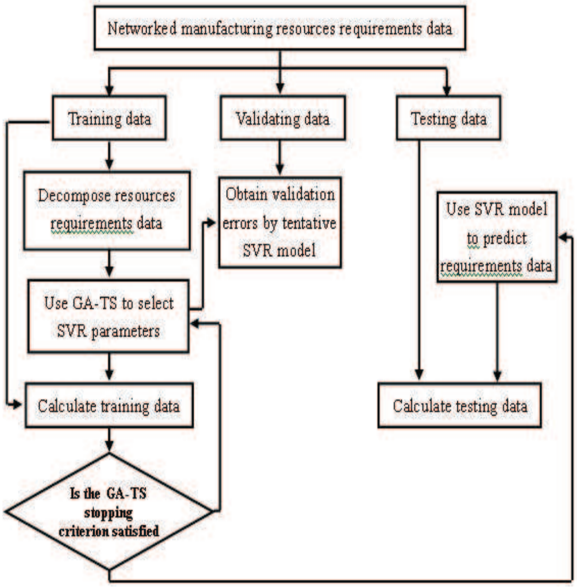
Networked manufacturing resources requirements data
Training data
Validating data
Testing data
Decompose resources requirements data
Obtain validation errors by tentative SVR model
Use SVR model to predict requirements data
Use GA-TS to select SVR parameters
Calculate training data
Calculate testing data
Is the GA-TS stopping criterion satisfied

Figure 2.3: A flowchart of the SVR model with GA-TS algorithm

Step 1.Generate randomly the initial population of N individuals, each one is a bit string chosen from $\{0,1\}$, and let current generation $k=1$ and K_{max} is the maximum number of generation/iteration. Initialize p_c and p_m as crossover probability and mutation probability, respectively.

Step 2.Evaluate the fitness score for each individual $x_i, i \in \{1,\ldots,N\}$ of the population based on the objective function $f(x_i)$.

Step 3.Select a pair of individuals x_α and x_β at random, depending on their fitness values (using roulette wheel method) from the population of N individuals. Here, selection refers to choosing parents for recombination; a new population was formed, usually with a probabilistic mechanism.

Step 4.Conduct crossover between the chosen individuals x_α and x_β with $p_c$and mutate each of their bits with mutation probability p_m. Each pair of parents (x_α, x_β) thus creates a pair of new individuals called offspring (x'_α, x'_β) to generate a pool of individuals, $x'_j, j \in \{1,\ldots,N\}$ as a population of next generation.

Step 5.Terminate the process if the stopping criterion $(k > K_{max})$ is satisfied. Otherwise, $k = k+1$ and go to Step 2.

2.2.2 Tabu Search (TS)

Tabu search is a metaheuristic local search algorithm that can be used for solving combinatorial optimization problems. Tabu search enhances the performance of a local search method by using memory structures; i.e., once a potential solution has been determined, it is marked as "taboo" ("tabu" being a different spelling of the same word) so that the algorithm does not visit that possibility repeatedly. Generally, the tabu list size should grow with the size of the given problem.

The iterative search procedure starts with a set of probable or feasible solutions. Each solution is a string of bits, chosen from $\{0,1\}$. Let AR_{tr} , AR_{cu} and AR_{be} denote the trial, current, and best array(s),OB_{tr} ,OB_{cu} and OB_{be} denote the corresponding trial, current, and best objective function value(s), respectively. The process assigns the current solution AR_{cu} for starting its operation. Then the trial solutions $AR_{tr}s$ are generated through some moves. At each iteration step, a best solution AR_{be} is found. With the progress of the search process if it is found that the best solution is in the forbidden or tabu list but satisfies the *aspiration criteria*, then it is considered to be the new current solution. An aspiration criterion is a rule that overrides tabu restrictions; i.e., if a certain move is forbidden by a tabu restriction, the aspiration criterion, when satisfied, can make this move allowable. The aspiration criterion used in the present method is the overriding of the tabu status of a move if this move yields a solution which has a better objective function than the one obtained earlier with the same move. The stopping criteria are the conditions under which the search process will terminate. In the present method, the search will terminate if one of the following criteria is satisfied: the number of iterations since the last change of the best solution is greater than a prespecified number; or the number of iterations reaches the maximum allowable number. The TS algorithm is given by the following steps.

Step 1: Initialize the parameters S_{mtl} (Maximum Tabu List Size), the number of trial solutions μ , and the maximum number of iterations I_{max}. Let AR_{cu} be an arbitrary solution and OB_{cu} be the corresponding objective function value. Initially, let $AR_{be} = AR_{cu}, OB_{be} = OB_{cu}, L_{tl}$ (Tabu List Length)$= 0$ and iteration $I = 1$.

Step 2: Using AR_{cu}, generate μ trial solutions $AR^1_{tr}, AR^2_{tr}, \ldots, AR^\mu_{tr}$ and evaluate their corresponding objective function values $OB^1_{tr}, OB^2_{tr}, \ldots, OB^\mu_{tr}$. Given a current solution AR_{cu}, one can generate a trial solution using several strategies. In our case, given AR_{cu} , we have flipped a bit of AR_{cu} if the probability threshold is higher than a randomly generated value between 0 and 1. Otherwise, the corresponding bit is kept unchanged.

Step 3: Arrange the objective function values $OB^1_{tr}, OB^2_{tr}, \ldots, OB^\mu_{tr}$ in ascending order and denote them as $OB^{1'}_{tr}, OB^{2'}_{tr}, \ldots, OB^{\mu'}_{tr}$. If $OB^{1'}_{tr}$ is not tabu or if it is tabu but $OB^{1'}_{tr} < OB_{be}$ (in case of minimization) then make $AR_{cu} = AR^{1'}_{tr}$ and $OB_{cu} = OB^{1'}_{tr}$. Next, go to step 4. Otherwise, let $AR_{cu} = AR^L_{tr}$ and $OB_{cu} = OB^L_{tr}$, where OB^L_{tr} is the best objective function values of $OB^{1'}_{tr}, OB^{2'}_{tr}, \ldots, OB^{\mu'}_{tr}$, that is not tabu and go to step 4. If $OB^{1'}_{tr}, OB^{2'}_{tr}, \ldots, OB^{\mu'}_{tr}$ are all tabu, then go to step 2.

Step 4: Insert AR_{cu} at the bottom of the tabu list and increment L_{tl} by 1. If $L_{tl} = S_{mtl} + 1$ then delete the first element in the list and $L_{tl} = L_{tl} - 1$. If $OB_{be} > OB_{cu}$ then $AR_{be} = AR_{cu}$ and $OB_{be} = OB_{cu}$. Terminate the process if $I = I_{max}$ with AR_{be} as the best solution and OB_{be} as the corresponding best objective function. Otherwise, make $I = I + 1$ and go to step 2.

2.2.3 Hybridization of GA with TS process (GS-TS)

The genetic methods are hybridized with the local TS process described in Section 2.2.2. The hybridization of each genetic method eventually enhances the performance of the respective hybrid genetic approach.

The genetic process starts with a population $p^t(g)$ (where $1 \leq t \leq T_{max}$ and $1 \leq g \leq G_{max}$) of N (population size) randomly created individuals. The new GA (NGA) adopted in our paper differs from CGA in two ways: the search is transferred from a larger hypercube to a smaller one for a specified number of times T_{max} and the search space resolution is redefined with the transfer of the search process to the newly defined hypercube. Here, we define the parameters of the function $f(x)$ in R^n where $x = (x_1, ..., x_n)$ and R^n represents the n-dimensional space. The searching process starts with low resolution and the algorithm finds a best solution $x^t_{BE} = (x^t_{BE,1}, x^t_{BE,2}, ..., x^t_{BE,n})$ which has not changed for several consecutive K_g generations. The search is then transferred to a smaller hypercube around x^t_{BE}. At the t th stage the hypercube on n-dimensional space can be represented as $R^{t,n} = \{r^{t,1}, r^{t,2}, ..., r^{t,n}\}$, where $1 \leq t \leq T_{max}$ and $r^{t,j} > 0, j = 1, 2, ..., n$.

Once the search process converges at $(t-1)$th step, the GA is reinvoked in the new hypercube $R^{t,n}$ with same mutation rate and higher space resolution. The population $p^t(g)$ at t th step is reproduced after transfer of the search process to $R^{t,n}$ hypercube. The individuals that are located both in $R^{t-1,n}$ and in $R^{t,n}$ except the elitist individual in $R^{t-1,n}$ are removed from the population $p^{t-1}(g)$(where $1 < t \leq T_{max}$ and $1 < g \leq G_{max}$) . A new population $p^t(g)$ of N individuals is regenerated with $N-1$ randomly created new individuals and the elite one of $p^{t-1}(g)$. Thus, the diversity of the population is restored at each stage.

We propose a hybrid genetic algorithm where the NGA works along with the TS process. From the discussions above, it is known that the NGA advances its search process hierarchically in multiple stages. At tth stage (initially $t = 1$), it starts with a population $p^t(g)$ (where $1 \leq t \leq T_{max}$ and $1 \leq g \leq G_{max}$) of N individuals and finds the best solution x^t_{BE} in the initial hypercube in the multi-dimensional space. In the hybrid process the TS technique is invoked at this point at the same hypercube without the change of resolution. The TS algorithm starts processing with the best solution x^t_{BE} found as its current solution AR_{cu} and generates randomly a pool of trial solutions consisting of μ individuals where $\mu < N$. The trial solutions are denoted by $AR^j_{tr}, j \in \{1, ..., \mu\}$ Each AR^j_{tr} consists of the same number of bits of x^t_{BE} . The fitness values of μ trial solutions are calculated and the best solution AR_{be} is picked, depending on the fitness values. If AR_{be} satisfies the conditions of TS algorithm to be considered as the new current solution for the next generation, then $AR_{cu} = AR_{be}$. The process is continued I_{max} times ($I_{max} \ll G_{max}$) to achieve the final best solution AR^t_{be} (at tth stage) of the TS process. At the end,AR^t_{be} may or may not be better (but not worse) than x^t_{BE} . The GA is then reinvoked where the search domain jumps from the hypercube $R^{t-1,n}$ to the new smaller hypercube $R^{t,n}$ with AR^t_{be} as its best solution. The GA generates a new population of $p^t(g)$ of N individuals including AR^t_{be} while redefining its search space resolution. The GA restarts with a new population and it may again converge to a solution. As before, TS is again initiated at that point. Thus, to find the global optimal solution the hybrid genetic process is continued with the transfer of the search location from one hypercube to the neighboring hypercube for at most T_{max} times. The changeover from GA to TS is continued T_{max} times. The algorithmic steps of the hybrid process are briefly discussed below where the steps of the GA and the TS process remain as before. Both algorithms are executed equal number of times for a problem.

Step 1: Set $t = 1$. Initialize the GA as well as the TS parameters.

Step 2: Start the GA with a pool of N individuals and continue the process.

Step 3: If the best solution of GA has not changed during K_g iterations, invoke TS.

Step 4: Consider the best solution of the GA as the current solution of TS at the respective stage.

Step 5: Continue the TS process for I_{max} times.

Step 6: If $t > T_{max}$, terminate the hybrid algorithm. Otherwise,$t = t + 1$ and consider the best solution of TS as the best solution of the GA for the next generation.

Step 7: Redefine the new hypercubic search space with higher resolution. Generate a pool of N individuals including the best solution found in Step 6 and go to Step 2.

3 Simulation results

We first observed the difference between the non-optimized approach and the optimized approach proposed in this paper. Six existing groups which were equipped with 12 lathes and 8 lathemen, and a simulation software installed in the Windows XP platform with Intel Core 2 Quad 3.2 GHz and 4 GB RAM, was adopted in this work as the simulator of networked manufacturing infrastructures. The maximum number of potential resources was set to 100. To reflect the characteristics of the real world, different complexities of service applications were designed to verify the feasibility and effectiveness of our proposed work.

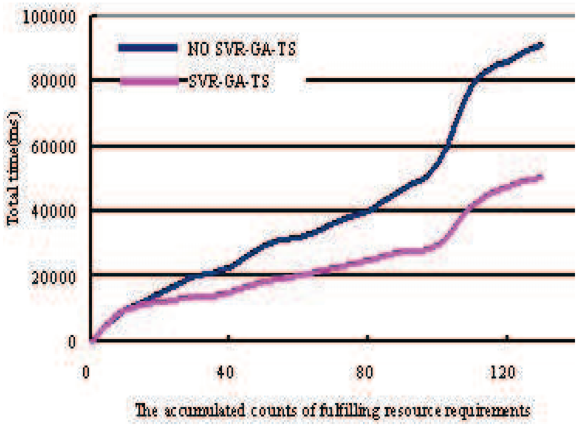

Figure 3.1: The comparison of the accumulated counts of fulfilling resource requirements

Fig.3.1 shows the comparison of the accumulated counts of fulfilling resource requirements between optimized and non-optimized approaches. Our proposed SVR-GA-TS mechanism adopted the SVR technique as a predictor that can moderately assign potential resources according to current network utilizations. Additionally, the hybridization of GA with TS process is applied to find out an optimized set of resource allocation strategies. The system that utilized GA without executing the prediction module via SVR is denoted by NO SVR-GA-TS. NO SVR-GA-TS used random allocation in the initialization step of GA without allocating potential resources in advance. The difference between SVR-GA-TS and NO SVR-GA-TS is that the latter cannot determine if the distribution strategies of potential resources need adjusting, and also cannot undergo local preallocation for individual demands in the initialization. Therefore, our approach can fulfill more resource requirements compared with the non-optimized method.

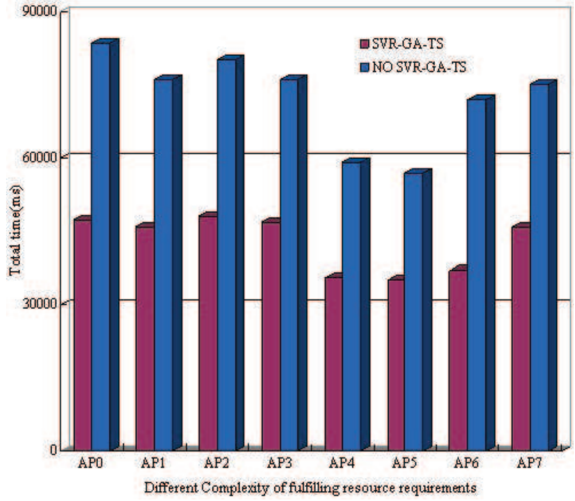

Figure 3.2: The comparison of time consumption for each resource requirement.

Fig.3.2 shows the comparison of time consumption for processing each resource requirement between the optimized and non-optimized approaches. Obviously, with the help of the application of resources requirement prediction module, the proposed approach can dynamically modify the requirement of the potential resources and the networked resources configuration module can effectively meet the requirements of existing resources by using the GA-TS algorithm. Because the system utilized SVR as a prediction mechanism before applying GA-TS, the total waiting time of executing applications can be thus dramatically decreased through allocating potential resources in appropriate existing resources pool in advance. Besides, our proposed SVR-GA-TS can undergo the operation of local allocation in advance for individual demands in the initialization step of GA. Therefore, we can decrease the computation complexity of GA-TS and increase the overall performance. We realize that GA-TS can obtain the optimal solution accurately, but the problem is that the traditional GA needs much time to converge, and this will be the main problem for real-time applications. That is the reason we combined SVR as a prediction mechanism to decease the computational times of GA-TS, and the advantage can be thus shown without affecting the overall performance in real-time systems.

4 Conclusions

In this paper, a resource configuration optimization scheme under networked manufacturing environment is proposed. According to our experimental findings, the proposed approach, which employs a resources requirement prediction module built with SVR, can estimate the number of resource utilization according to network protocol of each manufacturing process more accurately and efficiently than those non-optimized approaches found in the recent literature.

This paper mainly examines the performance of NGA working along with the TS process, applied in networked manufacturing. At present stage, we only evaluated two resources, including lathe and latheman, and the experimental results showed the effectiveness of applying GA-TS in the resource configuration of networked manufacturing. In future work, we will focus on more resources, such as the access of networked information to make our work more practical in real manufacturing processes.

In addition, the networked resources configuration module applied with GA-TS can effectively adjust the resource configuration strategies and accomplish more applications in a limited time, compared with the non-optimized approach. In future work, we plan to choose other Evolutionary Algorithms, such as the hybrid meta-heuristic algorithm, called evolution strategy, as compared methods.

Acknowledgement

The authors wish to sincerely thank the anonymous referees for their constructive comments on the previous version of this paper. The authors are grateful for the financial support from the National Natural Science Foundation of China (No. 71201144), the Scientific Research Fund of Zhejiang Provincial Education Department (No. Y200803886), the Scientific Research Project of Science and Technology Department of Zhejiang Province (No. 2013C33054), and the Scientific Research Startup Foundation of Zhejiang University of Technology (No. 102006429).

References

[1] D. Z. Zhao, Y. J. Shi and M. Gregory, Joint Optimal Output of Members in Global Manufacturing Virtual Network System, The Proceedings of The 6th Annual International Manufacturing Symposium, Cambridge, UK (2001).

[2] Y. Zhan, J. S. Lu and X. H. Ji, Research on Strategies of Networked Manufacturing Resources Configuration Based on Evolutionary Game, Advanced Materials Research. 430-432, 1330-1334 (2012).

[3] S. C. Graves and S. P. Willems, Optimizing the supply chain configuration for new products. Working Paper, Leaders for Management Program and A. P. Sloan School of Management, MIT, (2001).

[4] F. Li, B. Golden and E. Wasil, The open vehicle routing problem: Algorithms, large-scale test problems, and computational results, Computers & Operations Research. **38**, 2918-2930 (2007).

[5] E. Zachariadis and T. Kiranoudis, An open vehicle routing problem metaheuristic for examining wide solution neighborhoods, Computers & Operations Research, **37**, 712-723 (2010).

[6] S. M. Abbas, I. Saleem, B. Ahmed, H. Khurshid, UWB Antenna with Parasitic Patch and Asymmetric Feed, Information Sciences Letters, **2**, 27-33 (2013).

[7] H. Topcuoglu, F. Corut, M. Ermis and G. Yilmaz, Solving the uncapacitated hub location problem using genetic algorithms, Computers & Operations Research, **32**, 967-984 (2005).

[8] L. S. Randaccio and L. Atzori, Group multicast routing problem: A genetic algorithms based approach, Computer Networks, **51**, 3989-4004 (2007).

[9] B. Feng, Z. Jiang, Z. P. Fan and N. Fu, A method for member selection of cross-functional teams using the individual and collaborative performances, European Journal of Operational Research, **203**, 3652-3661 (2010).

[10] F. S. Wen and C. S. Chang, Transmission network optimal planning using the tabu search method, Electric Power Systems Research, **42**,153-163 (1997).

[11] M. R. Girgis, T. M. Mahmoud, H. F. A. El-Hameed, Z. M. El-Saghier, Routing and Capacity Assignment Problem in Computer Networks Using Genetic Algorithm, Information Sciences Letters, **2**, 13-25 (2013).

[12] A. Kumar, R. M. Pathak, Y. P. Gupta and H. R. Parsaei, A genetic algorithm for distributed system topology design, Computers & Industrial Engineering, **28**, 3659-3670 (1995).

[13] L. H. Zhang, X. Zou and Z. X. Su, GA Optimization Model for Time/cost Trade-off Problem in Repetitive Projects Considering Resource Continuity, Applied Mathematics & Information Sciences, **7**, 611-617 (2013).

[14] H. X. Yang, W. Zhou and C. Z. Lou. Optimal design and techno-economic analysis of a hybrid solar-wind power generation system, Applied Energy, **86**, 2163-2169 (2009).

[15] O. masood, A. Akram, M. N. Majeed, Performance Evaluation of ADV with AODV for Real-time and Multimedia Applications in Vehicular Ad-hoc Networks (VANETs), International Journal of Computing and Network Technology, **1**, 119-127 (2013).

[16] A. V. Smirnov, L. B. Sheremetov, N. Chilov, and J. R. Cortes. Soft-computing technologies for configuration of cooperative supply chain, Applied Soft Computing, **4**, 187-207 (2004)

[17] G. E. Cantarella, G. Pavone and A. Vitetta. Heuristics for urban road network design: Lane layout and signal settings, European Journal of Operational Research, **175**, 1682-1695 (2006).

[18] R. C. Lin, M. Y. Sir and K. S. Pasupathy. Multi-objective simulation optimization using data envelopment analysis and genetic algorithm: Specific application to determining optimal resource levels in surgical services, Omega, **41**, 881-892 (2013)

[19] M.Schneider, J. Grahl, D. Francas and D. Vigo. A problem-adjusted genetic algorithm for flexibility design, International Journal of Production Economics, **141**,56-65 (2013).

[20] Y. Yi, F. L. Rong, H. Y. Chang and Z. J. Xiao, SVR mathematical model and methods for sale prediction, Journal of Systems Engineering and Electronics, **18**, 769-773 (2007).

[21] P. F. Pai, S. L. Yang and P. T. Chang, Forecasting output of integrated circuit industry by support vector regression models with marriage honey-bees optimization algorithms, Expert Systems with Applications, **36**, 10746-10751 (2009).

[22] H. Q. Yang, K. Z. Huang, I. King and M. R. Lyu, Localized support vector regression for time series prediction, Neurocomputing, **72**, 2659-2669 (2009).

[23] T. T. Zou, Y. Dou, H. Mi, J. Y. Zou, and Y. L. Ren. Support vector regression for determination of component of compound oxytetracycline powder on near-infrared spectroscopy, Analytical Biochemistry, **355**, 1-7 (2006).

[24] C. H. Wu, J. M. Ho and D. T. Lee, Travel-time prediction with support vector regression, IEEE Transactions on Intelligent Transportation Systems, **5**, 276-281 (2004).

[25] B. Schölkopf, A. Smola, R. Williamson and P. L. Bartlett, New support vector algorithms, Neural Comput., **12**, 1207-1245 (2000).

Yan Zhan received the MS degree in Management Science & Engineering from Wuhan University of Technology in 2005 and the PhD degree in Industrial Engineering from College of Management and Economics, Tianjin University in 2008. She is currently a lecturer in Zhejiang University of Technology. Her research interests are in the areas of applied mathematics including the mathematical methods and models for complex systems, network optimization and modeling and simulation of logistics. She has published research articles in reputed international journals of mathematical and engineering sciences.

Jiansha Lu is a professor and PhD candidate advisor at Zhejiang University of Technology, founder and director of Department of Industrial & Logistics Engineering. He is among China’s first group of experts with outstanding contributions to Industrial Engineering. He is referee of several international journals in the IE field. His current research areas include lean production, system layout planning and design.

Shiyun Li received the PhD degree in Mechanical Engineering and Automation from Beijing Institute of Technology in 2006. He is currently a lecturer in Zhejiang University of Technology. His current research areas include digital integrated manufacturing, product data management, and collaborative design research.