Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 34 (2008) 2417–2423

Expert Systems with Applications

www.elsevier.com/locate/eswa

Design of BOM configuration for reducing spare parts logistic costs

Muh-Cherng Wu *, Yang-Kang Hsu

Department of Industrial Engineering and Management, National Chiao Tung University, Hsin-Chu, Taiwan, ROC

Abstract

This paper proposes an approach to reduce the total operational cost of a spare part logistic system by appropriately designing the BOM (bill of material) configuration. A spare part may have several vendors. Parts supplied by different vendors may vary in failure rates and prices – the higher the failure rate, the lower the price. Selecting vendors for spare parts is therefore a trade-off decision. Consider a machine where the BOM is composed of *s* critical parts and each part has *k* vendors. The number of possible BOM configurations for the machine is then k^s. For each BOM configuration, we can use OPUS10 (proprietary software) to calculate an optimum inventory policy and its associated total logistic cost. Exhaustively searching the solution space by OPUS10 can yield an optimal BOM configuration; however, it may be formidably time-consuming. To remedy the time-consuming problem, this research proposes a GA-neural network approach to solve the BOM configuration design problem. A neural network is developed to efficiently emulate the function of OPUS10 and a GA (genetic algorithm) is developed to quickly find a near-optimal BOM configuration. Experiment results indicate that the approach can obtain an effective BOM configuration efficiently.

Keywords: Bill of material; Spare parts; Stocking policy; Genetic algorithm; Neural network

1. Introduction

Machine availability is very important in capitally intensive industries. The higher the machine availability, the higher is the capacity. The level of machine availability partly depends on the inventory level of its spare parts. At a lower inventory level, the time required to repair a machine would be longer due to having higher possibility of lacking spare parts. A higher inventory level by contrast would increase machine availability at the expense of paying more inventory cost. Since spare parts in capitally intensive industries are quite expensive, much research investigated the stocking policies for spare parts to resolve the trade-off decision (Sherbrooke, 2004).

Spare parts are typically replenished through a *multi-echelon* supply chain system, which is a hierarchical structure comprising multiple layers of facilities (Fig. 1). A facility has two main functions: *storing* and *repairing* spare parts. Facilities in the lowest layer directly supply parts to machines, whereas those at a high layer supply parts to its succeeding lower layer facilities. Facilities at a higher layer are generally equipped with higher repairing capability. That is, a part that cannot be repaired by a particular facility would be sent upward to its parent facility. In this paper, the information for characterizing such a supply chain system is called BOS (bill of stations).

A machine is typically composed of several modules; each module comprises several assemblies that are assembled by subassemblies/parts (Fig. 2). The hierarchical representation for modeling the materials of a machine is called BOM (bill of materials), where a layer in the BOM structure is usually called an *indenture* in literature. In the BOM, each part is defined with several BOS-independent attributes such as cost and failure rate, and some BOS-dependent attributes such as eligible stations for repairing the part.

At a globalization era, a part tends to have multiple vendors who may provide parts that are functionally identical but with various failure rates and costs. The lower the

* Corresponding author. Tel.: +886 35 731 913; fax: +886 35 720 610.
E-mail address: mcwu@cc.nctu.edu.tw (M.-C. Wu).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.04.001

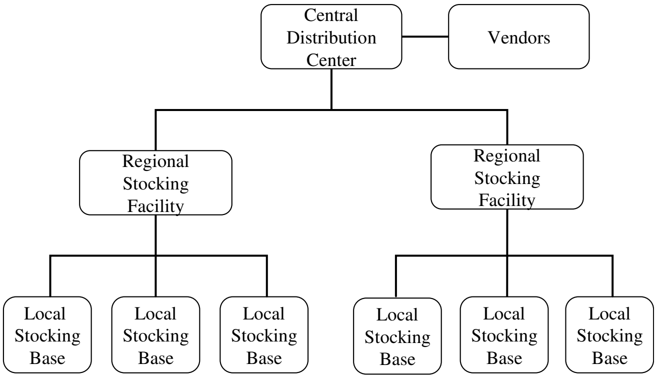

Fig. 1. BOS of a multi-echelon logistic system.

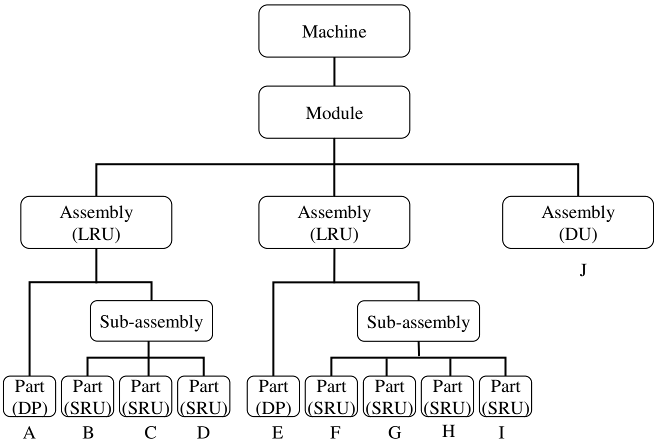

Fig. 2. BOM of a multi-indenture machine.

failure rate, the higher is the price of a part. Purchasing more reliable parts (i.e. with lower failure rates) would reduce the inventory level of spare parts required to maintain the target machine availability. As a result, this would reduce the holding cost of part inventory at the expense of increasing its purchasing cost. Choosing an appropriate configuration of BOM is therefore a very important way to reduce the total operational cost of a spare part supply chain system. Yet, this idea has been rarely noticed in literature.

This paper proposes the idea of choosing appropriate BOM configurations, formulates the decision problem, and develops an efficient approach to solve the problem. Suppose a machine has s critical parts, each of which has k vendors. The possible number of BOM configurations may be quite huge (k^s). It might take a formidable computation time if we exhaustively evaluate the performance of each BOM configuration by using the evaluation methods developed in literature. To efficiently solve the problem, we propose a GA-NN approach. The NN (neural network) technique is used to efficiently emulate the function of an existing method for evaluating BOM configurations, whereas the GA technique is used to efficiently identify a near-optimal BOM configuration from the huge solution space.

The remainder of this paper is organized as follows. Section 2 reviews the literature on stocking spare parts for a supply chain system. Section 3 introduces OPUS10 (proprietary software), which can be used to evaluate the performance of a BOM configuration. That is, OPUS10 can yield the optimal stocking policy and its associated total operational cost for a BOM configuration. Section 4 describes the procedure for establishing the neural network that could emulate the function of OPUS10. Section 5 presents the genetic algorithm. Section 6 illustrates the experimental results and concluding remarks are placed in Section 7.

2. Related literature

The inventory policy for a multi-echelon repairable item (spare part) system has been a research issue for several decades. Much literature has been published and some of them have included a comprehensive survey (Diaz & Fu, 2005; Guide & Srivastava, 1997; Kennedy, Patterson, & Fredendall, 2002; Rustenburg, 2000; Sleptchenko, 2002). These previous studies can be categorized into two main streams.

The first stream addressed a scenario equipped with an *infinite repair capacity*. An early and representative study is the METRIC (Multi-echelon Technique for Recoverable Item Control) model developed by Sherbrooke (1968). Many studies that extend the METRIC model were subsequently developed (Graves, 1985; Muckstadt, 1973; Sherbrooke, 1986, 2004; Simon, 1971; Slay, 1984). With the ample-server assumption, the queue-time for repair is negligible; therefore, these METRIC-variant models have been able to deal with large and complex systems. However, a real-world problem is typically equipped with limited repair capacity. The METRIC-variant models thus tend to underestimate the stocking levels in some real applications.

The various versions of the METRIC-variant models can be characterized from three perspectives: the demand pattern of spare part, BOM, and the complexity of supply chain. In the original METRIC model (Sherbrooke, 1968), the BOM is a single-indenture system involving multi-repairable-items, the demand is a compound Poisson process, and a replenishment (S, $S-1$) policy is used throughout a two-echelon supply chain. Simon (1971) developed a METRIC-variant model that uses (s, S) policy in the second-echelon facilities. Muckstadt (1973) enhanced the METRIC model by including two-indenture BOM systems. Slay (1984), Graves (1985), and Sherbrooke (1986) further developed methods in order to estimate the variances of service levels. Hausman and Erkip (1994) broadened the METRIC model by including a scenario where an emergency-ordering policy is allowed.

The second stream addressed a scenario with *finite repair capacity*, which leads to the need of modeling the machine-repair queueing behavior. The models in this stream, more realistic than the METRIC-variant models, are certainly more difficult to solve. Due to the inherent complexity, by the possible inclusion of enumeration techniques, most of these studies are computationally extensive (Gross, Miller, & Soland, 1983). Approximations to reduce the complexity have therefore been proposed in order to develop efficient algorithms for determining the capacities of repair facilities as well as spares level (Albright, 1989; Albright & Soni, 1988a, 1988b; Diaz & Fu, 1997; Gross & Miller, 1984; Gross, Kioussis, & Miller, 1987; Kim, Shin, & Park, 2000; Rustenburg, van Houtum, & Zum, 2001; Sleptchenko, van der Heijden, & van Harten, 2002).

In summary, most previous research focused on how to appropriately determine stocking level and repair capacities to reduce the total logistic costs of spare parts. Our problem of interest—how to choose a BOM configuration to reduce the total logistic cost of spare parts has been scarcely addressed.

3. OPUS10

OPUS10, proprietary software of a Swedish company, was developed based on the techniques of the METRIC-variant models. The input to OPUS10 involves the BOM/BOS of a spare part supply chain system. The output of OPUS10 yields optimal stocking levels for achieving target machine availability, and computes the associated total operational cost of the supply chain. The BOS, BOM, and the input/output relationships of OPUS10 are described below.

3.1. BOS

As shown in Fig. 1, the BOS of a spare part supply chain in OPUS10 is a hierarchical structure (also called multi-echelon in literature) consisting of multiple facilities. The attributes for describing a BOS includes the number of layers in the hierarchy, the number of facilities at each layer, the replenishment/repair logistic relationships between any two facilities, the lead time between any two facilities, and the number of machines logistically supported by each facility at the lowest layer. The facility at the lowest layer, which directly supports machines, hereafter is called a *terminal facility*.

A replenishment logistic relationship defines the eligibility of a particular facility to supply quality parts to other facilities. A repair logistic relationship defines the eligibility of a particular facility to send a failed part to other facilities for repair. Two facilities, if not eligible for shipping parts between each other, do not have a logistic relationship. The more number of logistic relationships denotes the more flexibility is embedded in the hierarchy of BOS. Lateral transshipment (logistic relationships among facilities of the same layer are eligible), which has been investigated by some literature (Axsäter, 2003; Lee, 1987; Wong, Cattrysse, & van Oudheusden, 2005), can also be modeled in the BOS of OPUS10.

3.2. BOM

As shown in Fig. 2, the BOM is a hierarchical structure that involves two main types of information: the hierarchical relationship and the attributes of each part/module. The typical attributes of a part/module involve the failure rate, the unit cost, and the facilities that are eligible for repairing the part/module.

The modules/parts of a BOM are classified into two types: *replaceable* and *discardable*. A *replaceable* part/module denotes that it is *repairable*, which involves two types: LRU and SRU. An LRU (line replacement unit) is a part/module that can be directly replaced by a terminal facility. Any LRU if failed can be *individually* taken out from the machine, replaced by a quality unit, and sent upward for repairing. Referring to Fig. 2, an SRU (shop replacement unit), a member of a particular LRU assembly, cannot be *individually* taken out from the machine. When an SRU fails, we cannot replace the SRU on line. Rather, we have to replace its parent LRU on line and sent the LRU upward to a shop-facility, where the LRU can be disassembled so that the SRU can be taken out from the LRU for repairing, and a quality SRU can be filled in the LRU.

A *discardable* part/module denotes that it is *non-repairable*, which also involves two types: DU and DP. A DU (discardable unit) can be individually taken out from the machine. When a DU fails, we can take it out on line, discard it, and replace it by a new quality DU. A DP (discardable part), like SRU, cannot be individually taken out from the machine and always has a parent LRU. When a DP fails, we cannot discard the DP on line. Rather, we have to replace its parent LRU on line and sent the LRU upward to a shop-facility, where the LRU can be disassembled so that the failed DP can be taken out and discarded for filling in a new quality DP in the LRU.

In summary, each part/module in the BOM has a *replacement attribute* to clarify where it is of LRU, SRU, DP, or DU. The distribution of the replacement attributes would significantly affect the total operational cost of a spare part logistic system and have to be modeled in the BOM.

3.3. Input/output relationship of OPUS10

The input/output relationships of OPUS10 in dealing with a multi-vendor spare part supply chain system that adopts $(S-1,\ S)$ stocking policy to achieve a specified machine availability can be formulated as follows:

$$(S_{ij}, \mathrm{TSC}) = f(A, r_j, Q_j, T_j^{\mathrm{pr}}, P_{jk}, F_{jk}, T_{io}^{\mathrm{tr}}, C_{io}^{\mathrm{tr}}, T_{ij}^{\mathrm{re}}, C_{ij}^{\mathrm{re}}) \qquad (1)$$

where

i index of facility, $1 \leqslant i \leqslant I$

j index of part, $1 \leqslant j \leqslant J$

k index of vendor, $1 \leqslant k \leqslant K$

S_{ij} stocking level of part j at facility i

TSC total operation costs of the supply chain

A target average machine availability specified by users

Q_j quantity of part j in the BOM

r_j replacement attribute of part j, which denotes that part j is LRU, SRU, DP or DU

T_j^{pr}	production/purchasing lead time of part *j*
P_{jk}	unit price of part *j* provided by vendor *k*
F_{jk}	failure rate of part *j* provided by vendor *k*
T_{io}^{tr}	transportation time between facilities *i* and *o*
C_{io}^{tr}	transportation cost between facilities *i* and *o*
T_{ij}^{re}	repair time of part *j* at facility *i*
C_{ij}^{re}	repair cost of part *j* at facility *i*

In Eq. (1), *f* denotes the function of OPUS10. The right-hand side represents the input parameters of OPUS10, which characterize the BOM/BOS of the supply chain and the user-specified machine availability (A). The left-hand side represents the output of OPUS10 – an optimal setting for stocking levels (S_{ij}) and the total operational cost at this setting (TSC).

In dealing with the decision of selecting BOM configurations, the function of OPUS10 can be further formulated as follows:

$$(S_{ij}, \mathrm{TSC}) = f(V|A, r_j, Q_j, T_j^{\mathrm{pr}}, P_{jk}, F_{jk}, T_{io}^{\mathrm{tr}}, C_{io}^{\mathrm{tr}}, T_{ij}^{\mathrm{re}}, C_{ij}^{\mathrm{re}}) \tag{2}$$

where $V = [v_j]_{1\times J}$, $1 \leqslant v_j \leqslant K$, v_j denotes the vendor that supplies part *j*.

That is, V is used to model a BOM configuration, which represents the decision variables of this problem. S_{ij} ($1 \leqslant i \leqslant I$ and $1 \leqslant j \leqslant J$) are intermediate output variables denoting optimal stocking policy, which can be used to compute TSC – the final output variable.

The combination of these decision variables would yield K^J possible BOM configurations, where $K^J = 59{,}049$ if $K = 3$, $J = 10$. Using a particular personal computer, our experiment indicates that it takes OPUS10 about 5 s to evaluate the performance (TSC) of a BOM configuration. For a decision problem with $K^J = 59{,}049$, the required computation time is about 3.4 days if we search these configurations exhaustively. In this paper, we propose a GA-NN technique to reduce the computation time. The NN (neural network) technique is used to emulate the function of OPUS10 in a more efficient manner. The (GA) genetic algorithm technique is used to reduce the number of BOM configurations that are searched during the process of finding a near-optimal one.

4. Neural networks (NN)

The procedure for establishing an NN for emulating the function of OPUS10 involves two major steps. First, we randomly sample *n* number of BOM configurations from the solution space and evaluate their performance at the specified machine availability (A) by using OPUS10. Referring to Eq. (2), this sampling would yield *n* pairs of input/output vectors, where V (BOM configuration) is the input and TSC (total operation cost) is the output.

Second, from the *n* pairs of input/output vectors, n_1 pairs are randomly sampled and used to train or develop a back-propagation neural network (Fausett, 1994). The remaining $n - n_1$ pairs are used to test the effectiveness of the network. The trained network if effective can be used to emulate the function of OPUS10.

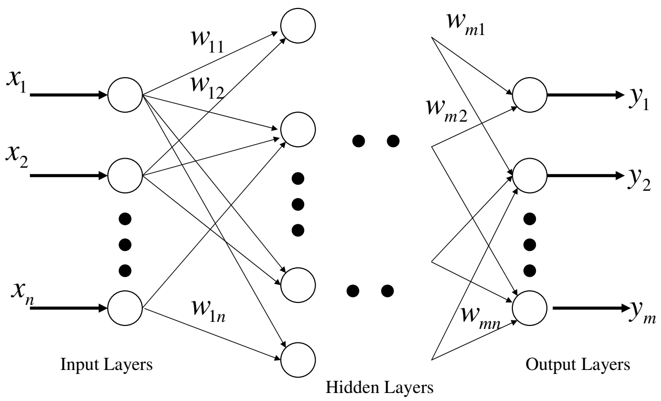

Fig. 3. Architecture of a back propagation neural network.

The architecture of the back-propagation neural network involves three layers of neurons (Fig. 3). The first layer represents the input, the third layer represents the output, and the second layer (called hidden layer) tends to model the transformation mechanism from input to output. Each neuron in a layer and that in its subsequent layer is connected by a link on which a weight is to be found. Training or developing the back-propagation neural network is to determine appropriate weights of each link.

The algorithm for training the NN iteratively changes the network weights by the following formula:

$$w_{ijk}(t+1) = w_{ijk}(t) + \eta \cdot w_{ijk}(t) + \alpha \cdot w_{ijk}(t-1) \tag{3}$$

where *t* denotes the index of change, *i* denotes a neuron in layer *k*, *j* denotes a neuron in the preceding layer ($k - 1$), and w_{ijk} represents the weight between the two neurons. Parameters η (learning constant) and α (momentum constant) are intended to adjust the speed of convergence. The detailed procedures for training the NN can be referred to Fausett (1994).

The validity of the trained NN is evaluated by measuring the deviation between the predicted output of the NN and the actual output of OPUS10, in terms of the root-mean-square error (RMSE). During the network development process, the network architecture (number of neurons at each layer) and the training parameters (η and α) are iteratively selected in order that the RMSE is minimized.

5. Using GA to find a near-optimal BOM configuration

The solution space of BOM configuration may be quite huge. Though the proposed NN could efficiently evaluate the performance of a particular BOM configuration. However, exhaustively searching the space may still be computationally extensive. In order to reduce the computation time, we proposed a GA (genetic algorithm) for efficiently finding a near-optimal BOM configuration from the solution space. GAs have been widely used in various applications (Bäck, Hammel, & Schwefel, 1997; Stockton, Quinn, & Khalil, 2004a, Stockton, Quinn, & Khalil, 2004b) and found to be efficient and effective in solving a complex space-searching problem.

Referring to Eq. (2), a BOM configuration that involves J parts and each part has K vendors on can be represented by $V = (v_1, v_2, \ldots, v_J)$, where $1 \leqslant v_i \leqslant K$ represents the vendor that supplies part i. In the GA, a BOM configuration V is called a *chromosome* and v_i is a called a *gene*. The performance of a BOM configuration refers to the total operation cost (TSC) subject that the stocking policy is optimally determined by using the developed NN that emulates the function of OPUS10. The TSC of a chromosome V, computed by the NN rather than OPUS10, is represented by $F(V)$, which is called the *fitness* of chromosome V.

The procedure of the GA involves the following major steps.

Step 1: Set $t = 0$. Randomly sample N chromosomes to form a population $P(0)$.
Step 2: Reproduction operation
- $S = \phi$
- Reproduce N chromosomes in $P(t)$ and place them in set S.

Step 3: Crossover operation
- Create $N \times P_{\mathrm{cr}}$ new chromosomes and place them in set S.

Step 4: Mutation operation
- Create $N \times P_{\mathrm{mu}}$ new chromosomes and place them in set S.

Step 5: Termination check
- If a terminating condition is met, output the best chromosome from S and stop.
- Otherwise, $t = t + 1$, $P(t) \leftarrow S$, go to Step 2.

In the aforementioned GA procedure, the reproduction, crossover, and mutation operators as well as the terminating conditions are further explained below. The reproduction operator in Step 2 is intended to select "good" chromosomes from $P(t)$ to form a new population; that is, a chromosome with higher fitness value has higher probability of being reproduced. We use the *tournament selection method* (Blickle, 1997, chap. C2.3: C2.3:1–C2.3:4; Goldberg, Korb, & Deb, 1989) in the reproduction process. This method randomly samples two chromosomes, from which the one with higher fitness is selected and placed in set S. This selection procedure is repeated until N chromosomes have been chosen.

The crossover operator in Step 3 is intended to create "new" chromosomes by using the single-point crossover technique (Booker, Fogel, Whitley, & Angeline, 1997, chap. C3.3:C3.3:1–C3.3:27; Goldberg, 1989; Spears, 1997, chap. E1.3:E1.3:1–E1.3:11). This technique firstly samples a pair of chromosomes randomly. Secondly, a *cut-point* is randomly chosen so that each chromosome is interpreted as having two segments. Thirdly, the left-hand segments of the two chromosomes are swapped to form a new chromosome, so does the right-hand segments. This as a result yields a new pair of chromosomes. The crossover operation is repeated until $N \times P_{\mathrm{cr}}$ new chromosomes have been created, where P_{cr} is termed crossover rate whose value is manually given.

The mutation operator in Step 4 is intended to create "new" chromosomes by changing the value of a particular gene. This operator firstly samples a chromosome randomly. Secondly, from the chromosome, one of its genes is randomly selected. Thirdly, the value of the gene (denoting a particular vendor) is replaced by randomly selecting a new vendor. The mutation operator is repeatedly performed until $N \times P_{\mathrm{mu}}$ new chromosomes have been created, where P_{mu} is termed mutation rate.

The GA procedure terminates either when the population has been updated T_{f} times (i.e., $t = T_{\mathrm{f}}$) or when the *best solution* in $P(t)$ keeps unchanged for B_{f} generations.

6. Numerical experiments

The proposed GA-NN method for solving the decision problem of BOM configuration has been tested by numerical experiments.

The tested scenario assumes the BOS/BOM structures as shown in Figs. 1 and 2 respectively. In the BOM, there are 10 critical parts (A–J), each of which has three vendors to choose. Table 1 shows the quantity of each part per BOM,

Table 1
Failure rates and prices of critical parts provided by different vendors

Critical part	Quantity of each part per BOM	Failure rates provided by each vendor (number of failures per 10^6 h)			Unit price provided by each vendor		
		FRT_1	FRT_2	FRT_3	C_1	C_2	C_3
A	3	54.75	33.46	30.42	$19,086	$31,231	$34,355
B	3	59.59	31.21	28.38	$20,385	$32,277	$38,733
C	3	102.61	98.43	60.36	$48,776	$75,040	$97,552
D	20	116.81	55.86	50.79	$34,672	$39,625	$55,476
E	3	114.68	97.69	67.46	$26,006	$29,474	$44,211
F	3	95.31	88.48	56.06	$35,026	$40,030	$56,042
G	3	52.75	38.36	23.98	$38,951	$50,936	$66,218
H	24	37.82	33.37	22.25	$38,465	$57,698	$69,238
I	6	127.58	77.31	60.28	$40,866	$70,588	$97,647
J	6	104.25	91.08	37.92	$14,261	$19,015	$28,523

Table 2
Comparing performance between BOM configurations

	A	B	C	D	E	F	G	H	I	J	TSC	% of cost reduction
GA	1	2	1	2	1	3	3	1	1	3	$9,304,599	
Vendor 1	1	1	1	1	1	1	1	1	1	1	$11,243,894	16.13%
Vendor 2	2	2	2	2	2	2	2	2	2	2	$10,906,230	14.58%
Vendor 3	3	3	3	3	3	3	3	3	3	3	$10,553,935	11.82%

the failure rates and the prices of each part offered by each vendor. As indicated in the table, the higher the price, the lower the failure rate. Assume that the desired machine availability (A) is 85%.

Out of these 3^{10} (59,049) BOM configurations, 1500 configurations are randomly sampled to train the neural network. Of these samples, 1200 samples are used to train the NN and the remaining 300 are used for testing. The NN is a 10-10-1 back-propagation neural network architecture trained by setting $\eta = 0.10$ and $\alpha = 0.80$, where 10-10-1 denotes that 10 neurons in the input layer, 10 in the hidden layer, and 1 in the output layer. Experiment results indicate that the accuracy of the trained NN is RMSE = 0.00795.

A personal computer equipped with 3.0 G CPU and 512 MB is used in the experiments. It takes about 5 s for OPUS10 to evaluate the performance of a BOM configuration. It takes about 1330 s to train the neural network. That is, the trained NN needs only about 10^{-4} s to evaluate the performance of a BOM configuration, about 50,000 times faster than OPUS10.

Parameters of the GA are so defined: $N = 100$, $P_{\text{cr}} = 0.80$, $P_{\text{mu}} = 0.05$, $T_{\text{f}} = 99{,}999$, and $B_{\text{f}} = 1000$. The GA was executed with 50 replicates. Experiment results showed that the solutions of the 50 replicates are all the same. The computation time for executing the GA with one replicate takes about 1 s.

Table 2 compares the performance of four BOM configurations. The first row in the table is the solution proposed by the GA. Each of the remaining three rows denotes a BOM configuration whose parts are exclusively supplied by a single vendor, rather than multiple ones. The table shows that the BOM configuration proposed by the GA outperforms the other three alternatives, reducing the cost up to 16.13%.

7. Concluding remarks

This paper proposes a *new perspective* to reduce the total operational cost of a spare part logistic system through appropriately selecting the BOM configuration. Most previous studies on spare part logistics assumed that the BOM configuration is fixed and aimed to determine the optimal inventory levels. However, each part in a BOM may have several options in terms of failure rates. A part option with lower failure rate is generally more expensive. Therefore, a BOM configuration with lower failure rates tends to require less quantity of inventory, at the price of incurring higher unit inventory cost. This research enhances previous works by relaxing the assumption of fixed BOM configuration in order to further reduce the total operational cost of a spare part logistic system.

The proposed solution method involves the use of OPUS10 as well as the development of a back-propagation neural network and a genetic algorithm. Given a BOM configuration, OPUS10, (proprietary software) can be used determine the optimal inventory levels for a spare part logistic system. However, using OPUS10 to exhaustively search a huge solution space of BOM configurations may be quite time-consuming. The time-consuming issue is resolved in twofold. First, through the development of an artificial neural network, we can reduce the computation time for estimating the performance of a BOM configuration. Second, through the development of a GA, we can greatly reduce the number of BOM configurations to be evaluated. Numerical experiments indicate that selecting an appropriate BOM configuration could significantly reduce the total operational cost.

Other than redesigning BOM configuration, we may further reduce the total operational cost of a spare part logistic system by redesigning the BOS (bill of stations) and the associated transportation configurations. Possible extensions of this research involve how to develop an integrated method to comprehensively design a spare part logistic system; that is, such a design should consider simultaneously the effects of BOM, BOS, and transportation configurations.

Acknowledgement

This research was financially supported by National Science Council, Taiwan, under a research contract NSC94-2623-7009-004.

References

Albright, S. C. (1989). An approximation to the stationary distribution of a multiechelon repairable-item inventory system with finite sources and repair channels. *Naval Research Logistics Quarterly, 36*, 179–195.

Albright, S. C., & Soni, A. (1988a). Markovian multiechelon repairable inventory system. *Naval Research Logistics Quarterly, 35*, 49–61.

Albright, S. C., & Soni, A. (1988b). An approximation to the stationary distribution of a multidimensional Markov process. *IIE Transactions, 20*, 111–118.

Axsäter, S. (2003). Evaluation unidirectional lateral transshipments and substitutions in inventory systems. *European Journal of Operational Research, 149*, 438–447.

Bäck, T., Hammel, U., & Schwefel, H. P. (1997). Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation, 1*, 3–17.

Blickle, T. (1997). Tournament selection. In T. Bäck, D. B. Fogel, & Z. Michalewicz (Eds.), *The handbook of evolutionary computation*. Philadelphia, PA: IOP Publishing Ltd. and Oxford University Press.

Booker, L. B., Fogel, D. B., Whitley, D., & Angeline, P. J. (1997). Recombination. In T. Bäck, D. B. Fogel, & Z. Michalewicz (Eds.), *The handbook of evolutionary computation*. Philadelphia, PA: IOP Publishing Ltd. and Oxford University Press.

Diaz, A., & Fu, M. C. (1997). Models for multi-echelon repairable item inventory systems with limited repair capacity. *European Journal of Operational Research, 97*, 480–492.

Diaz, A., & Fu, M. C. (2005). Multi-echelon models for repairable items: A review. *Decision & Information Technologies Research Works Collection*, Digital Repository at the University of Maryland.

Fausett, L. (1994). *Fundamental of neural networks: Architectures, algorithms, and applications*. New Jersey: Prentice-Hall.

Goldberg, D. E. (1989). *Genetic algorithms in search optimization and machine learning*. Reading, MA: Addison-Wesley.

Goldberg, D. E., Korb, G., & Deb, K. (1989). Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems, 3*, 493–530.

Graves, S. C. (1985). A multi-echelon inventory model for a repairable item with one-for-one replenishment. *Management Science, 31*(10), 1247–1256.

Gross, D., Kioussis, L. C., & Miller, D. R. (1987). A network decomposition approach for approximating the steady-state behavior of Markovian multi-echelon repairable item inventory systems. *Management Science, 33*, 1453–1468.

Gross, D., & Miller, D. R. (1984). Multiechelon repairable-item provisioning in a time-varying environment using the randomization technique. *Naval Research Logistics Quarterly, 31*, 347–361.

Gross, D., Miller, D. R., & Soland, R. M. (1983). A closed queueing network model for multi-echelon repairable item provisioning. *IIE Transaction, 15*, 344–352.

Guide, V. D. R., Jr., & Srivastava, R. (1997). Reparable inventory theory: Models and applications. *European Journal of Operational Research, 102*, 1–20.

Hausman, W. H., & Erkip, N. K. (1994). Multi-echelon vs. single-echelon inventory control policies for low-demand items. *Management Science, 40*(5), 597–602.

Kennedy, W. J., Patterson, J. W., & Fredendall, L. D. (2002). An overview of recent literature on spare parts inventories. *International Journal of Production Economics, 76*, 201–215.

Kim, J. S., Shin, K. C., & Park, S. K. (2000). An optimal algorithm for repairable-item inventory system with depot spares. *Journal of the Operational Research Society, 51*, 350–357.

Lee, H. L. (1987). A multi-echelon inventory model for repairable items with emergency lateral transshipments. *Management Science, 33*(10), 1302–1316.

Muckstadt, J. A. (1973). A model for a multi-item, multi-echelon, multi-indenture inventory system. *Management Science, 20*(4), 472–481.

Rustenburg, W. D. (2000). A system approach to budget-constrained spare parts management. Ph.D. thesis. Enschede, Netherlands: Eindhoven University of Technology.

Rustenburg, W. D., van Houtum, G. J., & Zum, W. H. M. (2001). Spare parts management at complex technology-based organizations: An agenda for research. *International Journal of Production Economics, 71*, 177–193.

Sherbrooke, C. C. (1968). METRIC: A multi-echelon technique for recoverable item control. *Operational Research, 16*, 122–141.

Sherbrooke, C. C. (1986). VARI-METRIC: Improved approximation for multi-indenture, multi-echelon availability models. *Operations Research, 34*(2), 311–319.

Sherbrooke, C. C. (2004). *Optimal inventory modeling of systems: Multi-echelon techniques*. MA: Kluwer Academic Publishers.

Simon, R. M. (1971). Stationary properties of a two-echelon inventory model for low demand items. *Operations Research, 19*(3), 761–773.

Slay, F. M. (1984). VARI-METRIC: An Approach to modeling multi-echelon resupply when the demand process is poisson with gamma prior, Technical Report AF301-3, Logistic Management Institute, Washington, DC.

Sleptchenko, A. (2002). Integral inventory control in spare parts networks with capacity restrictions. Ph.D. thesis. Enschede, Netherlands: University of Twente.

Sleptchenko, A., van der Heijden, M. C., & van Harten, A. (2002). Effects of finite repair capacity in multi-echelon, multi-indenture service part supply systems. *International Journal of Production Economics, 79*, 209–230.

Spears, W. (1997). Recombination parameters. In T. Bäck, D. B. Fogel, & Z. Michalewicz (Eds.), *The handbook of evolutionary computation*. Philadelphia, PA: IOP Publishing Ltd. and Oxford University Press.

Stockton, D. J., Quinn, L., & Khalil, R. A. (2004a). Use of genetic algorithms in operations management. Part 1: Applications. *ProQuest Science Journals, 3*, 315–327.

Stockton, D. J., Quinn, L., & Khalil, R. A. (2004b). Use of genetic algorithms in operations management. Part 2: Results. *ProQuest Science Journals, 3*, 329–343.

Wong, H., Cattrysse, D., & van Oudheusden, D. (2005). Inventory pooling of repairable spare parts with non-zero lateral transshipment time and delayed lateral transshipments. *European Journal of Operational Research, 165*, 207–218.