International Journal of
Operations Research

International Journal of Operations Research Vol. 7, No. 3, 15–22 (2010)

Optimal Staffing for a SOA Company

Shin-Guang Chen*

Department of Industrial Engineering and Management, Tungnan University,

Taipei County 222, Taiwan, ROC

Received October 2010; Revised November 2010; Accepted November 2010

***Abstract*—** Due to the IT advances in the Enterprise Resource Planning Systems (ERPS), more and more companies adapted the Service-Oriented Architecture (SOA) as the main infrastructure of their core business operations. So, process is the key identity of business activity to be monitored. However, how to evaluate the performance of as well as to staff such process, especially in a volatile environment, so that the business goal can be fulfilled is still unknown to most of the mangers. This paper proposes an analytic method to do optimal staffing for the companies facing people floating while keeping the capability to predict the performance of such process. So, the staffing decision can be verified. A numerical example is illustrated for the proposed method.
Keywords —ERP net, optimal staffing, people floating, process performance, reliability.

1. INTRODUCTION

Service-Oriented Architecture (SOA) was known to achieve agility, efficiency and flexibility of core processes in companies (Lawler, *et al*, 2009). More and more firms adapted the SOA as the main infrastructure of their core business operations. So, process is the key identity of business activity to be monitored. The usual Human Resource Management (HRM) practices may not be well-fitted for such a process-based staffing, because they are based on a static environment (Armstrong, 2006). Anderson (2001) proposed an elegant staffing method based on random walk, but his approach only applied for the solution of a long-term business cycle. For a daily operation, this may not be appropriate. Hence, how to evaluate the performance of as well as to staff such process, especially in a volatile environment, so that the business goal can be fulfilled is still unknown to most of the mangers. Here, the term "performance" means the probability to fulfill the desired throughput of the underlined process. Chen and Lin (2008b) first proposed an analytic method to evaluate the Bayesian performance of a business process. Later, they further introduced a method to give the linguistic performance for a business process (Chen and Lin, 2009). Chen (2009) extended it to cover the performance of a business process in case of system failures. Chen and Lin's model gave the excellent solution to evaluate the short-term performance for a daily process even in a volatile environment.

In this paper, we extend the Chen and Lin's approach (Chen and Lin, 2008a) to generate the optimal staff plan in a SOA company with absentees such that the required total cost for staffing is minimum and the performance of the process kept acceptable. So, the approach not only searches for the optimal staff plan but also calculates the derived performance simultaneously. A process network (or ERP net) is defined as that the nodes of the net are the persons responsible for the operations of the processes. The arcs are the precedence relationships (or the systems) between processes. So, the node has multi-states or -capacity and may fail (i.e., person absence). The faults in such a network usually depict a decrease in capacity (i.e., people's throughput) of the node and that stops the network due to an insufficient supply of document flow (i.e., the low performance). A conventional way to work out such issues is to increase the number of persons for each node. That is, for demand d (i.e., the expected throughput), a trivial plan is to let the maximal capacity of each node equal d. Such a scheme is not optimal and considered less benefit in cost. This paper is based on the structural analysis for the network and further assessed by the critical analysis. The structural analysis is used to identify the structurally important nodes which can not fail during the system operation. The critical analysis identifies the critical nodes of the underlined network. A node is critical if and only if its failure causes the system performance dropped to zero. Thus, a network full of critical nodes is very fragile. Any node's failure may stop the functionality of the process immediately. So, a network is robust if and only if any failure in non-critical nodes would not stop the network. In this paper, the calculation of system performance is based on the Minimal Path (MP) technique (Chen and Lin, 2008b). An MP is a sequence of nodes and arcs from source to sink without cycles. This

* Corresponding author's email: bobchen@mail.tnu.edu.tw

paper addresses the optimal conditions for such a network and illustrates the efficiency of the proposed algorithm by a numerical example. The remainder of the work is described as follows: The mathematical preliminaries and assumptions for the approach are presented in Section 2. Section 3 describes the algorithm of searching for an optimal plan. Then, the proposed method is illustrated by a numerical example in Section 4. Section 5 concludes this paper.

2. PRELIMINARIES

Let $G=(A, B, M, C, W)$ be a process network where A is the set of arcs, $B=\{b_i | 1\le i \le n\}$ is the set of nodes, and $M=(m_1, m_2, \ldots, m_n)$ is a vector with m_i (an integer) being the maximal capacity of node b_i. M is normally allocated by experience and strongly affects the cost. It is treated as a constant vector here; however, it will be treated as a variable vector later to be solved in this paper. $C=(c_1, c_2, \ldots, c_n)$ is the cost vector for nodes. W is the penalty when process failed. Such a G is assumed to satisfy the following assumptions.

(1) The capacity of each node b_i is an integer-valued random variable which takes values from the set $\{0, 1, 2, \ldots, m_i\}$ according to a given distribution. Note that 0 often denotes a failure or being unavailable.

(2) The arcs are perfect.

(3) The persons in each operation of the process have the same work capability and characteristics.

(4) Flow in G must satisfy the flow-conservation law (Ford and Fulkerson, 1962).

(5) The nodes are statistically independent from each other.

2.1 The Process Network Model

Suppose $mp_1, mp_2, \ldots, mp_z$ are totally the MPs from the source to the sink. Thus, the network model can be described in terms of two vectors: the capacity vector $X=(x_1, x_2, \ldots, x_n)$ and the flow vector $F=(f_1, f_2, \ldots, f_z)$ where x_i denotes the current capacity on b_i and f_j denotes the current flow on mp_j. Then such a vector F is feasible if and only if

$$\sum_{j=1}^{z}\{f_j \mid b_i \in mp_j\} \le m_i \quad \text{for each } i = 1, 2, \ldots, n. \tag{1}$$

Equation (1) describes that the total flow through b_i can not exceed the maximal capacity on b_i. We denote such set of F as $U_M \equiv \{F | F \text{ is feasible under } M\}$. Similarly, F is feasible under $X=(x_1, x_2, \ldots, x_n)$ if and only if

$$\sum_{j=1}^{z}\{f_j \mid b_i \in mp_j\} \le x_i \quad \text{for each } i = 1, 2, \ldots, n. \tag{2}$$

For clarity, let $U_X = \{F | F \text{ is feasible under } X\}$. The maximal flow under X is defined as $V(X) \equiv \max\{\sum_{j=1}^{z} f_j \mid F \in U_X\}$.

2.2 System Performance Evaluation

Given a demand d, the system performance R_d is the probability that the maximal flow is no less than d, i.e., $R_d \equiv \Pr\{X | V(X) \ge d\}$. To calculate R_d, it is advantageously to find the minimal vector in the set $\{X | V(X) \ge d\}$. A minimal vector X is said to be a lower boundary point (LBP) for d if and only if (i) $V(X) \ge d$ and (ii) $V(Y) < d$ for any other vector Y such that $Y < X$, in which $Y \le X$ if and only if $y_j \le x_j$ for each $j=1, 2, \ldots, n$ and $Y < X$ if and only if $Y \le X$ and $y_j < x_j$ for at least one j. Suppose there are totally t LBPs for d: $X_1, X_2, \ldots, X_t$, the system performance is equal to

$$\Pr\{\bigcup_{i=1}^{t}\{X \mid X \ge X_i\}\}. \tag{3}$$

2.3 Probability Calculation Scheme

To calculate R_d, the probability for $\Pr\{X_i\}$ of node b_i should be defined in advance. This can be done by assuming that there are c_i persons for node b_i to produce the corresponding capacity. Each person has the availability of r_i. Then the probability for the current capacity X_i is denoted as a binomial distribution:

$\Pr\{X_i = k\} = \binom{c_i}{k} r_i^k (1 - r_i)^{c_i - k}$.

2.4 Generation of all LBPs for *d*

At first, we find the flow vector $F \in U_M$ such that the total flow of F equals d. It is defined as in the following demand constraint.

$$\sum_{j=1}^{z} f_j = d \tag{4}$$

Then, let $\mathbf{F}=\{F \mid F \in U_M$ and satisfies Equation (4)$\}$. We show that a lower boundary point X for d existed then there exists an $F \in \mathbf{F}$ by the following lemma (Chen, 2009) without proof.

Lemma 1. Let X be a lower boundary point for d, then there exists an $F \in \mathbf{F}$ such that

$$x_i = \sum_{j=1}^{z} \{f_j \mid b_i \in mp_j\} \quad \text{for each } i = 1, 2, \ldots, n. \tag{5}$$

Given any $F \in \mathbf{F}$, we generate a capacity vector $X_F = (x_1, x_2, \ldots, x_n)$ via Equation (5). Then the set $\Omega = \{X_F \mid F \in \mathbf{F}\}$ is built. Let $\Omega_{\min} = \{X \mid X$ is a minimal vector in $\Omega\}$. Lemma 1 implies that the set Ω includes all lower boundary points for d. The following lemma (Chen, 2009) further proves that $\Omega_{\min}$ is the set of lower boundary points for d.

Lemma 2. $\Omega_{\min}$ is the set of lower boundary points for d.

3. STAFF PLANNING

Our problem is to find a proper M such that the network is survived and the required total cost is minimal and the performance can be derived. Given a network, the MPs are determined by the topology of the network. One can analyze the flow of a node via the binding MPs. Let $P_i= \{mp_j \mid b_i \in mp_j\}$ denote the subset of MPs binding with b_i. We define the coverage of b_i by the following definition.

Definition 1. (**Coverage**): Let $b_i, b_j \in B$. b_j is covered by b_i if and only if $P_j \subseteq P_i$.

Definition 1 implies that there is no flow through b_j if b_i totally failed. A structural impact (SI) Si for b_i is then defined as:

Definition 2. (**SI**): $S_i= |\{b_j \mid P_j \subseteq P_i\}|/n$.

The symbol $|\bullet|$ denotes the total number of elements in the set. 'n' is the total number of nodes in the network. If S_i=1.0, it means that b_i covers all nodes in the network and has the strongest structural impact upon the network. The smaller S_i is, the less impact b_i has. S_i can not be zero, since it must cover itself.

3.1 Critical Analysis

If the capacity of a node is decreased to zero (i.e., totally failed) while keeping the other nodes unchanged, we can analyze the derived impact of the network via performance. The calculated performance is thought as a ``survivability" for d of the network when the specific node totally failed. It is defined as:

Definition 3. (**Survivability**): $R_{d,i}$ is the derived performance when b_i totally failed.

This concept can be extended to the identification of critical nodes.

Definition 4. (**Critical node**): b_i is critical if and only if $R_{d,i}$=0.

From Definition 3 and 4, we have the following lemma.

Lemma 3. b_i is critical if S_i = 1.0.

This implies that b_i may be non-critical if S_i < 1.0.

3.2 Robustness

Given M, a network G is robust if it satisfied the following definition:

Definition 4. (**Robustness**): M is robust for d if and only if $R_{d,i}$>0 for all i such that S_i <1.0.

That is, if the vector M is robust for d, it should provide sufficient capacity to support such failure except those are structurally important. It can be shown that $m_i \geq d\ \forall\ i$ is a sufficient condition for M to be robust.

Lemma 4. M is robust for d if $m_i \geq d\ \forall\ i$.

However, when M is robust, it is not necessary for all m_i to be greater than d. The combination of flows may fulfill the demand d. Lemma 4 only describes the fact that d is a feasible upper bound for m_i. Our goal is then restated as to find a feasible lower bound of m_i to support robustness for d such that the total cost is minimal and the performance is predictable. A novel way is to inspect the capacity vector in LBPs generated by initially setting all $m_i = d$. By Equations 4 and 5, an LBP is a minimal capacity vector such that the total flow in G equals d. Lemma 4 shows that the set of LBPs when initially setting all $m_i = d$ can not be empty. Although an LBP is generated after given M, the column value in the vector is less than or equal to the corresponding m_i. If we reduce m_i of each node to the corresponding column value of LBP, G still survived for d. However, no capacity of any column can be further decreased, since an LBP is a minimal vector to support G being survived for d. Similarly, if G is robust when b_i failed, a feasible vector for the remaining nodes can be derived from the LBPs generated by $G\backslash\{b_i\}$. The newly derived capacity can cover the lost flow of b_i. Let $\Phi = \{i \mid S_i < 1.0\}$ be the index set of the structurally unimportant nodes, and $\Omega_{\min,i}$ denote the set of LBPs generated after setting $m_i = 0$ and $m_j = d$ for any $j \neq i$. The LBPs generated from $\Omega_{\min,i}$ can be used as a guide to select the minimal M. However, an LBP may consist of numerous zeros in the vector, which are undesirable for applications (i.e., they are corresponding to the faulty nodes). Consequently, a vector with least zeros is preferable for choice. To filter out the LBPs with undesirable zeros, an efficient strategy based on SI value can be applied. We firstly show that if the flow of the self-covering node b_i (i.e., S_i=1/n) is greater than 0, then the flow of other nodes covering b_i would not be zero. That is, the flow of such b_i will dominate the non-zero flow to other nodes.

Lemma 5. For b_i with S_i=1/n, if $\sum_{k=1}^{z}\{f_k \mid b_i \in mp_k\} > 0$, then

$$\sum_{k=1}^{z}\{f_k \mid b_j \in mp_k\} > 0\ \forall\ b_j \text{ covering } b_i, j \neq i\}. \tag{6}$$

Let $\varphi = \{i \mid S_i = 1/n\}$ be the index set of the self-covering nodes. Further relations can be derived as in the following lemma.

Lemma 6. If $\varphi_\cap = \cap_{\{i\in\varphi\}} P_i$ for acyclic networks, then $\varphi_\cap$ is empty.

The above lemma describes that all the self-covering nodes have no common intersection of MPs in acyclic networks. The following lemma further shows that the total number of MPs' binding with the self-covering nodes should be greater than or equal to the number of self-covering nodes.

Lemma 7. If $\varphi_\cup = \cup_{\{i\in\varphi\}} P_i$, then $|\varphi_\cup| \geq |\varphi|$.

Finally, the number of MPs can be shown to have the following relations.

Lemma 8. Suppose $mp_1, mp_2, \ldots, mp_z$ are totally the MPs from source to sink, then $z \geq |\varphi_\cup| \geq |\varphi|$.

When the network is acyclic, each self-covering node should only belong to one MP exclusively.

Lemma 9. If G is acyclic and $i \in \varphi$, then $|P_i|$=1.

Then, we can inspect those columns corresponding to the self-covering nodes in each LBP generated from $\Omega_{min,i}$. There are two conditions for inspection. One is $|\varphi|$\leq d, and the other is $|\varphi| > d$. The former denotes the supply of flow d is sufficient for all self-covering nodes and so are the other nodes. If any self-covering node is not zero, then every other node should cover at least one self-covering node and its flow would not be zero. In this case, one can directly delete those LBPs generated from $\Omega_{min,i}$ with zero columns other than column i and its coverage columns (which should be zero). The latter condition states the insufficient flow situation. In this case, one even dispatch flow d to each self-covering node by only 1 unit of flow, there are still some other self-covering nodes with zero flow. However, such vector consists of the least number of zeros among all other LBPs. That is, one can delete those LBPs with columns corresponding to the self-covering nodes having the value greater than 1. Such filtering process can keep the vectors with the least number of zeros in hand and decrease the search space tremendously for M. It can be shown that the filtered $\Omega_{min,i}$ has the possible range of lower bound for M. Let $X=(x_1, x_2, \ldots, x_n)$ be an LBP in $\Omega_{min,i}$, then

Theorem 1. $M= (m_1, m_2, \ldots,m_n)$ is robust for d if and only if

$$m_j = \max\{x_j \mid X\in \Omega_{min,i}\ \forall\ i\in \Phi\} \text{ for } j=1, 2, \ldots,n. \tag{7}$$

Theorem 1 shows that such M exists. We denote such set as $\Gamma\equiv\{M \mid M \text{ is robust for } d\}$. The performance under M is defined as $R_d(M)$. The optimal solution for M would be the one such that

Minimize $\sum_{j=1}^{n}\{m_j c_j + Wd(1-R_d(M))\}$ subject to $M \in \Gamma$.

Let $M_{i,j}=\{x_j \mid x_j> 0\ \forall\ X \in \Omega_{min,i}\}$ denote the set of possible values for column j. We further define $M_{min,j}=\max\{x_j \mid x_j\in M_{i,j}$ and x_j is minimal $\forall\ i\in\Phi\}$ as the largest minimal value among i for column j and $M_{max,j}=\max\{x_j \mid x_j\in M_{i,j}$ and x_j is maximal $\forall\ i\in\Phi\}$ as the largest maximal value among i for column j. One can show that for any $M\in\Gamma$, m_j exists in the interval $[M_{min,j}, M_{max,j}]$.

Theorem 2. If $M= (m_1, m_2, ..., m_n) \in \Gamma$, then

$$M_{min,j}\leq m_j \leq M_{max,j} \text{ for } j=1, 2, ..., n. \tag{8}$$

For clarity, we define $\mathbf{m}_j=\{ m_j \mid M_{min,j}\leq m_j \leq M_{max,j}\}$ as the interval set for column j. One can show that $\Gamma \subseteq \mathbf{m}_1\times \mathbf{m}_2\times ... \times \mathbf{m}_n$, where the symbol '×' denotes the Cartesian product among sets and is defined as $\mathbf{m}_1\times \mathbf{m}_2=\{(x,y) \mid x\in \mathbf{m}_1$ and $y\in \mathbf{m}_2\}$.

Theorem 3. $\Gamma \subseteq \mathbf{m}_1\times \mathbf{m}_2\times ... \times \mathbf{m}_n$.

Theorem 3 denotes that the set $\mathbf{m}_1\times \mathbf{m}_2\times ... \times \mathbf{m}_n$ includes all the feasible M in Γ. This implies that Γ can be searched from the Cartesian product.

4. ALGORITHM

Let $\Omega_i = \{X_F \mid F\in \mathbf{F}$ under $m_i=0\}$. The following algorithm is proposed to find the optimal M for d.

Algorithm 1: The optimal staff plan of G for d.

Step 1. For $i \in \Phi$ do // for structurally unimportant nodes.

a. Initially set $m_i=0$, $m_k=d$ for any $k\neq i$. Generates $\Omega_{min,i}$ as in the following:

1) Compute $\mathbf{F}$ for satisfying $\sum_{k=1}^{z}\{f_k \mid b_l \in mp_k\} \leq m_l$ and $\sum_{k=1}^{z} f_k = d$.

2) Construct Ω_i via X_F, which is formed by $x_l=\sum_{k=1}^{z}\{f_k \mid b_l \in mp_k\}$.

3) Generate $\Omega_{min,i}$ via simple comparison. //i.e., pairwise comparison.

b. For $X \in \Omega_{min,i}$ do // The filtering process.

If $|\varphi|\leq d$, then

For $1 \leq l\leq n$ do

If $x_l=0$ and $l\neq i$ and $b_l\notin b_i$'s coverage set, then $\Omega_{min,i} = \Omega_{min,i} \setminus \{X\}$.

End for.

Else,

For $1 \le l \le n$ do
If $x_l>1$ and $S_l=1/n$, then $\Omega_{\min,i} = \Omega_{\min,i} \setminus \{X\}$.
End for.
End for.
End for.
Step 2. For $i\in\Phi$ do // Construct $M_{i,j}$
For $X\in \Omega_{\min,i}$ do
For $1 \le j \le n$ do
If $x_j \notin M_{i,j}$ and $x_j>0$, then $M_{i,j}=M_{i,j}\cup\{x_j\}$.
End for.
End for.
End for.
Step 3. For $1 \le j \le n$ do // Construct $\mathbf{m}_j$.
$M_{\min,j}=\max\{x_j \mid x_j\in M_{i,j}$ and x_j is minimal $\forall\ i\in\Phi\}$.
$M_{\max,j}=\max\{x_j \mid x_j\in M_{i,j}$ and x_j is maximal $\forall\ i\in\Phi\}$.
$\mathbf{m}_j=\{ m_j \mid M_{\min,j}\le m_j \le M_{\max,j}\}$.
End for.
Step 4. For $M\in \mathbf{m}_1\times \mathbf{m}_2\times ... \times \mathbf{m}_n$ do // Construct Γ.
Set $CNT=0$ // CNT is a counter for Φ.
For $i\in\Phi$ do
For $X\in \Omega_{\min,i}$ do
If $M \ge X$, then $CNT=CNT+1$ and break.
End for.
End for.
If $CNT=|\Phi|$, then $\Gamma=\Gamma\cup\{M\}$.
End for.
Step 5. Let $sum_C = ndW$. // ndW is a feasible large number.
Step 6. For $M\in\Gamma$ do // Search for the optimal M.
If $\sum_{j=1}^{n}\{m_j c_j + Wd(1-R_d(M))\} < sum_C$, then $M_{best}=M$ and $sum_C = \sum_{j=1}^{n}\{m_j c_j + Wd(1-R_d(M))\}$.
End for.
Step 7. Output: The optimal plan is M_{best}, the lowest cost is sum_C and the performance is $R_d(M_{best})$.

For $i\in\Phi$, Step (1) generates $\Omega_{\min,i}$ for d. Step (1b) filters out the vectors with undesired zeros. Step (4) constructs the candidate set Γ. Step (6) finds the best choice of M, where R_d is calculated via $\Pr\{\bigcup_{k=1}^{t}\{X \mid X \ge X_k\}\}$ and X_k is the LBP in $\Omega_{\min}$ under M. Since the computation of R_d is not emphasized in this paper, the following complexity will exclude the complexity derived from the computation of R_d.

The number of solutions for $\mathbf{F}$ is bounded by $\binom{z+d-1}{z-1}$. The number of X_F generated is then bounded by $\binom{z+d-1}{z-1}$. The storage space needed for all the final $\Omega_{\min,i}$ (i.e., the filtered set) is then bounded by $O(n\binom{z+d-1}{z-1})$ in the worst case. Let $n_i=|\Omega_{\min,i}|$, then the number of $M_{i,j}$ is bounded by $O(\sum_{i\in\Phi} dn n_i)$. The number of Γ is then bounded by $O(\binom{|\varphi|}{d}\prod_{k=1}^{n-|\varphi|}|\mathrm{m}_k|)$ in the worst case. In sum, the total storage space needed is $O(\binom{|\varphi|}{d}\prod_{k=1}^{n-|\varphi|}|\mathrm{m}_k|)$ in the worst case. A pairwise comparison is required for generating $\Omega_{\min,i}$ between $\binom{z+d-1}{z-1}$ solutions. This takes $O(n^2\binom{z+d-1}{z-1})$ time to generate all the final $\Omega_{\min,i}$. To generate Γ, it spends $O(n\prod_{i=1}^{n}|\mathrm{m}_i|\sum_{k=1}^{n}n_k)$ time in the worst case. Then, it consumes $O(|\Gamma|)$ time to test the best M. In short, the total computational time required is $O(n\prod_{i=1}^{n}|\mathrm{m}_i|\sum_{k=1}^{n}n_k)$ in the worst case.

5. NUMERICAL EXAMPLE

Suppose an order entry operation existed to accept orders. Two alternative factories can be employed to support these orders. Four production lines operate in the two factories. Two delivery departments existed to deliver all of these manufactured products. Finally, the payment department closes all these orders. Figure 1 denotes such process network. There are totally 7 MPs existed: $mp_1=\{b_1, b_2, b_4, b_8, b_{10}\}$, $mp_2=\{b_1, b_2, b_5, b_8, b_{10}\}$, $mp_3=\{b_1, b_2, b_6, b_8, b_{10}\}$, $mp_4=\{b_1, b_2, b_6, b_9, b_{10}\}$, $mp_5=\{b_1, b_3, b_6, b_8, b_{10}\}$, $mp_6=\{b_1, b_3, b_6, b_9, b_{10}\}$, and $mp_7=\{b_1, b_3, b_7, b_9, b_{10}\}$. Assuming the historical availability, the cost, the coverage set, and SI for each node are in Table 1. Each person contributes 10 capacity in average. The penalty is 120 (in \$$10^3$/unit). The demand for the example is 40.

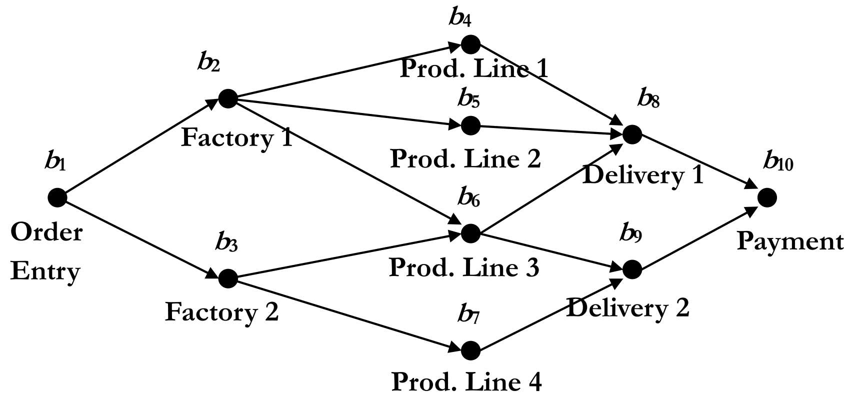

Figure 1 An order fulfillment process

Table 1 The data for Figure 1

Nodes	Cost (\$$10^3$/Month)	Availability	P_i	Coverage sets	SI
b_1	0.6	0.99	$P_1=\{mp_1, mp_2, mp_3, mp_4, mp_5, mp_6, mp_7\}$	$\{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}\}$	$S_1=1.0$
b_2	0.8	0.98	$P_2=\{mp_1, mp_2, mp_3, mp_4\}$	$\{b_2, b_4, b_5\}$	$S_2=0.3$
b_3	0.7	0.97	$P_3=\{mp_5, mp_6, mp_7\}$	$\{b_3, b_7\}$	$S_3=0.2$
b_4	1.2	0.98	$P_4=\{mp_1\}$	$\{b_4\}$	$S_4=0.1$
b_5	1.1	0.98	$P_5=\{mp_2\}$	$\{b_5\}$	$S_5=0.1$
b_6	1.3	0.99	$P_6=\{mp_3, mp_4, mp_5, mp_6\}$	$\{b_6\}$	$S_6=0.1$
b_7	1.1	0.97	$P_7=\{mp_7\}$	$\{b_7\}$	$S_7=0.1$
b_8	0.8	0.98	$P_8=\{mp_1, mp_2, mp_3, mp_5\}$	$\{b_4, b_5, b_8\}$	$S_8=0.3$
b_9	0.7	0.97	$P_9=\{mp_4, mp_6, mp_7\}$	$\{b_7, b_9\}$	$S_9=0.2$
b_{10}	0.8	0.99	$P_{10}=\{mp_1, mp_2, mp_3, mp_4, mp_5, mp_6, mp_7\}$	$\{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}\}$	$S_{10}=1.0$

According to the above algorithm, the optimal staff plan for this example is M= (4, 4, 4, 1, 1, 2, 2, 4, 4, 4) and the minimal cost is 618.44 (in 10^3). The performance is predicted as R_4=0.922616, a pretty good level. A critical analysis is conducted to test the above result comparing with a random plan in Table 2.
In this table, there are eight nodes critical in the random plan. This means that the major operations can not fail; otherwise the order fulfillment process will be stopped. Because of the high penalty during the service down time, the random plan incurred higher total cost. It is apparent that the proposed plan is superior to the random plan.

6. CONCLUSION

This paper proposes an algorithm to find the optimal staff plan in a SOA company such that the process is robust and the required staffing cost is minimal while the performance is predictable. Since the globalized economics changes very fast, the people float for each firm's operation is also versatile. Our approach can provide companies a helpful toolkit to manage such situation while keeping the ability to predict the performance of process. At first, we do the structural analysis as a basis for the staff plan strategy. The structural analysis is to determine the nodes which can not fail despite how many resources engaged. Then, such plan is evaluated by the critical analysis which identifies the critical nodes of the underlined process. A critical node is a node where only its failure causes the system performance dropped to zero. This paper also demonstrates a numerical example to show the efficiency of the proposed approach.

Table 2. A critical analysis for the proposed plan vs. an empirical plan.

Nodes	Proposed Plan (M)	$R_{4,i}$	Critical?	Random Plan (M)	$R_{4,i}$	Critical?
b_1	4	0.0000	Y	6	0.0000	Y
b_2	4	0.7532	N	1	0.0000	Y
b_3	4	0.8510	N	3	0.0000	Y
b_4	1	0.9226	N	1	0.7977	N
b_5	1	0.9226	N	2	0.7820	N
b_6	2	0.9227	N	1	0.0000	Y
b_7	2	0.9227	N	3	0.0000	Y
b_8	4	0.7532	N	2	0.0000	Y
b_9	4	0.8510	N	2	0.0000	Y
b_{10}	4	0.0000	Y	6	0.0000	Y
Performance $R_4(M)$		0.922616				0.797996
Cost (in $\$10^3$)		618.4				1192.6
Robustness		Yes				No

ACKNOWLEDGEMENT

This work was supported in part by the National Science Council, Taiwan, Republic of China, under Grant No. NSC 99-2221-E-236-009.

REFERENCES

1. Anderson, Jr. E. G. (2007). The Nonstationary Staff Planning Problem with Business Cycle and Learning Effects. *Management Science*, 47: 817-832.
2. Armstrong, M. (2006). *A Handbook of Human Resource Management Practice* (10th ed.). Kogan Page, London.
3. Chen, S. G. (2009). Performance Evaluation for an ERP System in Case of System Failures . In proceeding: *16th ISPE International Conference on Concurrent Engineering,* Taipei Taiwan.
4. Chen, S. G. and Lin, Y. K. (2008a). Capacity Assignment for a Stochastic-Flow Network based on Structural Importance. In proceeding: *Asian International Workshop on Advanced Reliability Modeling,* Taichung Taiwan.
5. Chen, S. G. and Lin, Y. K. (2008b). An Evaluation Method for Enterprise Resource Planning Systems. *Journal of the Operations Research Society of Japan,* 51: 299-309.
6. Chen, S. G. and Lin, Y. K. (2009). On Performance Evaluation of ERP Systems with Fuzzy Mathematics. *Expert Systems with Applications,* 36: 6362-6367.
7. Ford, L. R. and Fulkerson, D. R. (1962). *Flows in Networks*. Princeton University Press, NJ.
8. Lawler, J. P., Benedict, V., Howell-Barber, H. and Joseph, A. (2009). Critical Success Factors in the Planning of a Service-Oriented Architecture (SOA) Strategy for Educators and Managers. *Information Systems Education Journal*, 7: 1-30.