On a Dynamic Optimization Technique for Resource Allocation Problems in a Production Company

Chuma R. Nwozo¹, Charles I. Nkeki²

¹Department of Mathematics, University of Ibadan, Ibadan, Nigeria
²Department of Mathematics, University of Benin, Benin City, Nigeria
Email: drcrnwozo@gmail.com, nkekicharles2003@yahoo.com

Received May 7, 2012; revised June 10, 2012; accepted June 24, 2012

ABSTRACT

This paper examines the allocation of resource to different tasks in a production company. The company produces the same kinds of goods and want to allocate $m$ number of tasks to 50 number of machines. These machines are subject to breakdown. It is expected that the breakdown machines will be repaired and put into operation. From past records, the company estimated the profit the machines will generate from the various tasks at the first stage of the operation. Also, the company estimated the probability of breakdown of the machines for performing each of the tasks. The aim of this paper is to determine the expected maximize profit that will accrue to the company over $T$ horizon. The profit that will accrued to the company was obtained as $\frac{4,571,100,000}{48}$ after 48 weeks of operation. At the infinity horizon, the profit was obtained to be $\frac{20,491,000,000}{\omega}$. It was found that adequate planning, prompt and effective maintainance can enhance the profitability of the company.

Keywords: Dynamic Optimization; Resource Allocation; Company; Machines; Tasks

1. Introduction

We consider the allocation of tasks to different machines in a production company. A certain number of machines is proposed to be purchased at the beginning of a planning horizon. From statistics, the company has an estimate of the profit each tasks is to yield at the first stage of operation. Also, the company estimates the probability of breakdown of the machines allocated to each tasks. When a machine breaks down, it goes in for repairs after which it returns to the factory for re-allocation at the beginning of the next period.

In this paper, we formulate the problem as a dynamic optimization (DO). Our approach builds on previous research. [1] used the stochastic programming technique of dynamic Programming in financial asset allocation problems for designing low-risk portfolios. [2] proposed the idea of using a parsimonious sufficient static in an application of approximate dynamic programming to inventory management. [3] described an algorithm for computing parameter values to fit linear and separable concave approximations to the value function for large-scale problems in transportation and logistics. [4] described a more complicated variation of the algorithm that implores execution time and memory requirements. The improvement is critical for practical applications to realistic large-scale problems. [5] used DO for large-scale asset management problems for both single and multiple assets. [6] extended an approximate DO method to optimize the distribution operations of a company manufacturing certain products at multiple production plants and shipping to different customer locations for sales. [7] considered the allocation of buses from a single station to different routes in a transportation company in Nigeria.

In this section, we consider the methodology adopted in this paper. We start with the problem formulation.

2. Problem Formulation

In this section, we consider the methodology adopted in this paper. We start with the problem formulation. Given a certain number of tasks that are to be allocated to different machines at the beginning of each time period, we expect some machine(s) breakdown at the end of each period. Due to the uncertainty in the number of breakdown machine(s), we assume that the states of the machines are random. The company must know the number of machines available for the next period before decision will be made on how to allocate the tasks to the remaining machines. The number of machines to be put into operation in the next period depends on the number of breakdown at the end of the previous period. Our aim is to maximize the total expected profit over a timehorizon. We define the following notations which presented
in Table 1.
In the next subsection, we define the one-period expected profit function and formulate the problem as a dynamic program.

3. The Objective and One-Period Expected Profit Function

If the profit for allocating the $k$ task to the machines at period $t$ is $\Psi^k_t$, the state of the machines is $s_t$, number of machines allocated to operate on task $k$ at period $t$ under policy $\pi$, is $x^k_t$ and the number of breakdown machines is $b_t$, then the profit that will accrue to the company over $T$-horizon is given by

$$\sum_{t=0}^{T} \sum_{k=1}^{m} \Psi^k_t \left( x^k_t \left( s_{t-1} ( b_t ) \right) \right).$$

The expected maximum profit that will accrue to the company under policy $\pi$ is given by

$$Y^\pi_t ( s_t ) = E \left[ \max_{x^k_t \in X_t} \sum_{t=0}^{T} \sum_{k=1}^{m} \beta^t \Psi^k_t \left( x^k_t \left( s_{t-1} ( b_t ) \right) \right) \right]$$

subject to

$$\sum_{k=1}^{m} x^k_t \left( s_{t-1} ( b_t ) \right) \leq s_t, \ t = 0, 1, \ldots, T$$

$$x^k_t \geq 0, t = 1, \ldots, T; k = 1, \ldots, m$$

Note: $X(s_t)$ is the set of possible solution of problem (1). Conditioning (1) on $s_t \in S$. We now have the following optimization problem (3)

$$Y^\pi_t ( s_t ) = E \left[ \max_{x^k_t \in X_t} \sum_{t=0}^{T} \sum_{k=1}^{m} \beta^t \Psi^k_t \left( x^k_t \left( s_{t-1} ( b_t ) \right) \right) \right] / s_t \in S_t.$$ (3)

Problem (1.3) maximizes the expected profit over $X(s_t)$ subject to

$$\sum_{k=1}^{m} x^k_t \left( s_{t-1} ( b_t ) \right) \leq s_t, t = 0, 1, \ldots, T$$

For the profit function $\Psi: S \rightarrow R$, if we accumulate the profit of the first $T$-stage and add to it the terminal profit

$$\Psi^* ( s_T ) = \sum_{k=1}^{m} \Psi^k_t \left( s_T \right),$$

then (3) becomes

$$Y^\pi_t ( s_t ) = E \left[ \max_{x^k_t \in X_t} \sum_{t=0}^{T} \sum_{k=1}^{m} \beta^t \Psi^k_t \left( x^k_t \left( s_{t-1} ( b_t ) \right) \right) \right]$$

subject to

$$\sum_{k=1}^{m} x^k_t \left( s_{t-1} ( b_t ) \right) \leq s_t, t = 0, 1, \ldots, T.$$ (4)

Note: $x^k_t \geq 0, t = 1, \ldots, T; k = 1, \ldots, m.$

4. Dynamic Programming Formulation and Optimality

Using $s_t$ as the state variable at period $t$ and $S$ as the state space, we can formulate the problem as a dynamic program. The number of breakdown machines for task $k$ at period $t$ is given by $P_k x^k_t$, where $P_k$ is the probability of breakdown machines for task $k$. Hence, total number of breakdown machines for all the tasks is given by

$$\sum_{k=1}^{m} P_k x^k_t, t = 1, 2, \ldots, T.$$ (5)

We therefore have that

<table>
<thead>
<tr>
<th>Table 1. Notations and their Definitions.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Notation</strong></td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>$\beta$</td>
</tr>
<tr>
<td>$S$</td>
</tr>
<tr>
<td>$T$</td>
</tr>
<tr>
<td>$\pi$</td>
</tr>
<tr>
<td>$x^k_t$</td>
</tr>
<tr>
<td>$S_t$</td>
</tr>
<tr>
<td>$b_t$</td>
</tr>
<tr>
<td>$x^k_t \left( s_{t-1} ( b_t ) \right)$</td>
</tr>
<tr>
<td>$\Psi^k_t$</td>
</tr>
<tr>
<td>$\Pi$</td>
</tr>
<tr>
<td>$s_0$</td>
</tr>
<tr>
<td>$Y(s_0)$</td>
</tr>
</tbody>
</table>
Let $s_t$ be the number of machines to be allocated in period $t$ and let $\alpha$ be the percentage of break down (but repaired) machines that are expected to join the functional ones in period $t$, then the transformation equation for the system is given by

$$s_t = s_{t-1} - (1 - \alpha) \sum_{k=1}^{m} P_k x^k_t, \quad t = 1, \ldots, T. \quad (5)$$

Observe that the transformation equation is a random variable.

We now set $1 - \alpha = \beta$ in (5) to have,

$$s_t = s_{t-1} - \beta \sum_{k=1}^{m} P_k x^k_t, \quad t = 1, \ldots, T. \quad (6)$$

In this case, the optimal policy can be found by computing the value functions through the optimization problem

$$F^*_t (s_t) = \max_{x^k_t \in X(s_t)} \sum_{k=1}^{m} \Psi^k_t \left( x^k_t (s_{t-1} (b)) \right)$$

subject to

$$\sum_{k=1}^{m} x^k_t (s_{t-1} (b)) \leq s_t, \quad t = 1, 2, \ldots, T.$$  

Equivalently,

$$\xi_t + \sum_{k=1}^{m} x^k_t (s_{t-1} (b)) = s_t, \quad t = 1, 2, \ldots, T; \quad \xi_t \geq 0. \quad (8)$$

Since all the available functional machines must be allocated in the next period, we have that $\xi_t = 0$, for all $t \in T$, which is the slack variable.

We show that (4) is equivalent to (7), and then use (4) and (7) interchangeably. The theorem below establish this claim.

**Lemma 1.1:** Let $s_t$ be a state variable that captures the relevant history up to time $t$, and let $Y_t (s_{t+1})$ be some function measured at $t' \geq t + 1$ conditional on the random variable $s_t$. Then,

$$E \left[ E \left[ Y_{t'} | x_{t+1} \right] | s_t \right] = E \left[ Y_{t'} | s_t \right].$$

For the proof, see [3].

**Theorem 1.1:** Suppose that $Y^*_t (s_t)$ satisfies (4) and $F^*_t (s_t)$ satisfies (7), then $Y^*_t (s_t) = F^*_t (s_t)$.

**Proof:** We are to show that $Y^*_t (s_t) = F^*_t (s_t)$. We first use a standard method of DP. Obviously

$$Y^*_t (s_t) = F^*_t (s_t) = \Psi^*_t (s_t).$$

Suppose that it hold for $t + 1, t + 2, \ldots, T$, then we show that it is true for $t$.

We now write

$$F^*_t (s_t) = \max_{x^k_t \in X(s_t)} \sum_{k=1}^{m} \Psi^k_t \left( x^k_t (s_{t-1} (b)) \right)$$

$$+ E \left[ E \left[ F_{t+1} | s_t \right] \right]$$

Applying Lemma 1.1, we have

$$F^*_t (s_t) = \max_{x^k_t \in X(s_t)} \sum_{k=1}^{m} \Psi^k_t \left( x^k_t (s_{t-1} (b)) \right)$$

$$+ E \left[ E \left[ F_{t+1} | s_t \right] \right]$$

When we condition on $x^k_t (s_{t-1} (b))$, we obtain

$$F^*_t (s_t) = E \left[ \max_{x^k_t \in X(s_t)} \sum_{k=1}^{m} \Psi^k_t \left( x^k_t (s_{t-1} (b)) \right) \right]$$

$$= Y^*_t (s_t).$$

For any given objective function, we desire to find the best possible policy, $\pi$, that optimizes it, that is, we search for

$$Y^*_t (s_t) = \max_{s_{t+1}} Y^*_t (s_t).$$

This is obtained by solving the optimality equation

$$F_t (s_t) = \max_{x^k_t \in X(s_t)} \sum_{k=1}^{m} \Psi^k_t \left( x^k_t (s_{t-1} (b)) \right)$$

$$+ E \left[ F_{t+1} | s_t \right]$$

$$= \max_{x^k_t \in X(s_t)} \sum_{k=1}^{m} \Psi^k_t \left( x^k_t (s_{t-1} (b)) \right)$$

$$+ F_{t+1} (s_t - \beta \sum_{k=1}^{m} P_k x^k_{t+1}).$$

If we find the set of $F$'s that solves (9), then we have found the policy that optimizes $Y^*_t (s_t)$. The result below establishes this claim.

**Theorem 1.2:** The expression $F^*_t (s_t)$ is a solution to equation (1.9) if and only if

$$Y^*_t (s_t) = F_t (s_t) = \max_{s_{t+1}} Y^*_t (s_t).$$

The if part: This is shows by induction that $F_t (s_t) \geq Y^*_t (s_t)$, for all $s_t \in S$ and $t = 0, 1, \ldots, T - 1$.

Since $F_t (s_t) = \Psi^*_t (s_t)$, for all $s_t \in S$, we have that $F_t (s_t) = Y^*_t (s_t)$.

Suppose that $F_t (s_t) \geq Y^*_t (s_t) \geq F^*_t (s_t)$, for $t = n + 1, n + 2, \ldots, T$, and let $\pi \in \Pi$ be an arbitrary policy. For $t = n$, we obtain the optimality equation as follows
By induction hypothesis, \( F_{n+1}(s_{n+1}) \geq Y^*_n(s_{n+1}) \). So we have
\[
F_n(s_n) = \max_{x^d \in X(s_n)} \left[ \sum_{i=1}^{m} \Psi^d_i \left( x^d_i \left( s_{n-i}(b) \right) \right) + F_{n+1} \left( s_n - \beta \sum_{k=1}^{m} P_k x^d_{n+1} \right) \right].
\]

Also, we have that \( Y^*_n(s_{n+1}) \leq Y^*_{n+1}(s_{n+1}) \) for an arbitrary policy, \( \pi \epsilon \Pi \), hence
\[
F_n(s_n) \geq \max_{x^d \in X(s_n)} \left[ \sum_{i=1}^{m} \Psi^d_i \left( x^d_i \left( s_{n-i}(b) \right) \right) + Y^*_n \left( s_n - \beta \sum_{k=1}^{m} P_k x^d_{n+1} \right), s' = s_{n+1} \right].
\]

This result shows that solving the optimality equation also gives the optimal value function.

**Theorem 1.3:** Let \( B(s) \) be the set of all bounded real-valued functions \( F: S \rightarrow R \). The mapping \( \Gamma \): \( B(s) \rightarrow B(s) \) is a contraction.

**Theorem 1.4:** For any bounded return or scoring functions \( F_1: S \rightarrow R \) and \( F_2: S \rightarrow R \), and for all \( t = 0, 1, 2, \ldots \), the inequality below holds
\[
\max_{s \epsilon S} \left\{ \left| \Gamma^t F_1(s) - \Gamma^t F_2(s) \right| \right\} \leq \beta^t \max_{s \epsilon S} \left| F_1(s) - F_2(s) \right|
\]

\[ \text{max}_{s \epsilon S} \left| F_1(s) - F_2(s) \right| \leq \beta^t \max_{s \epsilon S} \left| F_1(s) - F_2(s) \right| \]

We now state this claim formally as follows:

**Theorem 1.5:** For any bounded return or scoring function \( F: S \rightarrow R \), \( F(0) \rightarrow F(\infty), T \rightarrow \infty \) that is,
\[ F(s_0) = \lim_{T \to \infty} \Gamma^T(F)(s_0), \quad \nexists s_0 \in S. \]

**Proof:** Let \( H \) be a positive integer, \( s_0 \in S \) and policy \( \pi = \{ \pi_0, \pi_1, \cdots \} \), we can decompose the return

\[ F^\pi(s_0) = \lim_{T \to \infty} E \left\{ \sum_{t=0}^{m} \beta^t \Psi_t^}\left(x_t^\pi(s_t(b))\right) \right\} \]

into the portion received over the first \( H \) stages and over the remaining stages.

\[ F^\pi(s_0) = \lim_{T \to \infty} E \left\{ \sum_{t=0}^{H-1} \beta^t \Psi_t^\left(x_t^\pi(s_t(b))\right) \right\} \]

\[ = E \left\{ \sum_{t=0}^{H-1} \beta^t \Psi_t^\left(x_t^\pi(s_t(b))\right) \right\} + \lim_{T \to \infty} E \left\{ \sum_{t=H}^{\infty} \beta^t \Psi_t^\left(x_t^\pi(s_t(b))\right) \right\} \]

But

\[ \lim_{T \to \infty} E \left\{ \sum_{t=H}^{\infty} \beta^t \Psi_t^\left(x_t^\pi(s_t(b))\right) \right\} \leq N \sum_{t=H}^{\infty} \beta^t = \beta^H \frac{N}{1-\beta} \]

Since

\[ \sum_{t=H}^{\infty} \beta^t \]

is a geometric progression and \( 0 < \beta < 1 \).

Now

\[ F^\pi(s_0) \leq E \left\{ \sum_{t=0}^{H-1} \beta^t \Psi_t^\left(x_t^\pi(s_t(b))\right) + \beta^H \frac{N}{1-\beta} \right\} \]

using this relations, it follows that

\[ F^\pi(s_0) - \beta^H \frac{N}{1-\beta} \max_{s_t \in S} \left| \Psi(s) \right| \]

\[ \leq E \left[ \beta^H \Psi(s) + \sum_{t=H}^{\infty} \beta^t \Psi_t^\left(x_t^\pi(s_t(b))\right) \right] \]

\[ \leq F^\pi(s_0) + \beta^H \frac{N}{1-\beta} + \beta^H \max_{s_t \in S} \left| \Psi(s) \right|. \]

By taking the maximum over \( \pi \), we obtain for all \( s_0 \) and \( H \).

\[ F^\pi(s_0) - \beta^H \frac{N}{1-\beta} - \beta^H \max_{s_t \in S} \left| \Psi(s) \right| \]

\[ \leq (\Gamma^H F)(s_0) \leq F^\pi(s_0) + \beta^H \frac{N}{1-\beta} + \beta^H \max_{s_t \in S} \left| \Psi(s) \right|. \]

and by taking the unit as \( H \to \infty \), we have

\[ F^\pi(s_0) \leq \lim_{H \to \infty} (\Gamma^H F)(s_0) \leq F^\pi(s_0) \]

Hence, \( F^\pi(s_0) = \lim_{H \to \infty} (\Gamma^H F)(s_0) \), \( \nexists s_0 \in S \).

We use value iteration algorithm for finite and infinite stage to solve our problem, The algorithm converges to an optimal policy.

**Step 1:** Initialization

- Set \( F_0(s_0) = 0 \), \( \forall s_0 \in S \).
- Set \( n = 0 \)
- Set \( 0 < \beta < 1 \)

Fix a tolerance parameter, \( \varepsilon > 0 \).

**Step 2:** For each \( s_j \in S \), calculate

\[ F_{n+1}(s_j) = \max_{x_j \in X} \sum_k \Psi_k^\left(x_j^\pi(s_j(b))\right) \]

\[ + F_n(s_j - \beta \sum_k P_{jk}x_k^\pi) \]

Let \( x^{\pi} \) be the decision vector that solve (13).

**Step 3:** For \( \beta = 1 \);

If \( \|F_{n+1} - F_n\| \leq \varepsilon \), \( \text{set} \quad x = x^{\pi} \), \( F^{\pi} = F_{n+1} \) and stop;
else set \( n = n + 1 \) and return to step 2.

**Step 4:** For \( 0 < \beta < 1 \);

If \( \|F_{n+1} - F_n\| \leq \frac{1-\beta}{2\beta} \), \( \text{set} \quad x = x^{\pi} \), \( F^{\pi} = F_{n+1} \)

and result stop; else set \( n = n + 1 \) and return to step 2.

The theorem below guarantees the convergent of the algorithm.

**Theorem 1.6:** If the algorithm with stopping parameter \( \varepsilon \), terminates at iteration \( n \) with value function \( F_{n+1} \), then

\[ \|F_{n+1} - F^\pi\| \leq \varepsilon/2 \]

In addition, if \( x^* \) is the optimal decision rule and \( F^{\pi} \) is the value of this policy, then

\[ \|F^{\pi} - F^\pi\| \leq \varepsilon. \]

This theorem implies that \( F^{\pi} \) is the fixed point of the equation \( F = \Gamma(x^*)F \). Since \( x^* \) is the decision that solves \( \Gamma x^* \) \( = \Gamma F \), it implies that \( \Gamma(x^*)F_{n+1} = \Gamma F_{n+1} \). Since

\[ \|F^\pi - F_{n+1}\| = \|\Gamma F^\pi - \Gamma F_{n+1} + \Gamma F_{n+1} - F_{n+1}\| \]

\[ F_{n+1} = \Gamma F_n \] and \( \Gamma \) is contraction, we have that

\[ \|F^\pi - F_{n+1}\| \leq \frac{\beta}{1-\beta} \|F_{n+1} - F_n\| \]

and

\[ \|F_{n+1} - F_n\| \leq \frac{\beta}{1-\beta} \|F_{n+1} - F_n\| \]

But the value iteration algorithm stops when

\[ \|F_{n+1} - F_n\| \leq \varepsilon \]

From (16) and (17), we have
\[
\left\| F_{n+1} - F^* \right\| \leq \frac{\beta}{1 - \beta} \frac{e(1 - \beta)}{2} = \epsilon/2 \quad (18)
\]

Similarly,
\[
\left\| F_{n+1} - F^{x*} \right\| \leq \frac{\epsilon}{2}
\]

Therefore,
\[
\left\| F^* - F^{x*} \right\| \leq \left\| F_{n+1} - F^* \right\| + \left\| F^* - F^{x*} \right\| \leq \epsilon/2 + \epsilon/2 = \epsilon.
\]

5. Computational Result

A production company in Nigeria proposed to purchase 50 machines that can perform nine different tasks. These machines are subject to breakdown. The Table 2 gives information of their decisions.

The company further estimated that out of the number of breakdown machines per week, 95% will join the functional ones for the next period. The aim of the company is to maximize profit over T horizon.

Let \( s_i \) represents the number of machines to be allocated in the next period, so that \( s_i = x_i^* \left( s_{i-1} \right) \). Since \( s_0 \) is the number of machines at the beginning of the planning horizon,
\[
\sum_{i=1}^{m} P_i x_i^k
\]
is the total number of breakdown machines, and
\[
\frac{19}{20} \sum_{i=1}^{m} P_i x_i^k
\]
is expected to join the functional machines in the next period of operation, we have that
\[
s_i = s_{i-1} - \frac{1}{20} \sum_{i=1}^{m} P_i x_i^k, \quad t = T, T-1, \cdots, 1,
\]
which is our transformation equation and is a random variable.

We can now express our one-period expected return function as follows:
\[
Y^k_i \left( s_i \right) = E \left[ \max \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n} \beta^j r_i^k \left( x_i^k \left( s_{i-1} \right) \right) \right\} \right] \quad (19)
\]
subject to
\[
\sum_{i=1}^{m} x_i^k \left( s_{i-1} \left( s_{i-1} \right) \right) = s_{i-1}, \quad t = T, T-1, \cdots, 1.
\]

(The feasible region for stage \( t \)).

Since the company cannot allocate negative resources to any one task, we write \( x_i^k \left( s_{i-1} \left( s_{i-1} \right) \right) \geq 0, \quad t = T, T-1, \cdots, 1; k = 1, \cdots, m. \)

Of course, \( (19) \) is the same as
\[
F_T \left( s_{i-1} \right) = \max \left\{ \sum_{i=1}^{m} \Psi_i^k x_i^k \left( s_{i-1} \left( s_{i-1} \right) \right) + EF_{T-1} \left( s_{i-1} \right) \right\}, \quad t = T, T-1, \cdots, 1,
\]
\[
= \max \left\{ 180000x_1^k + 150000x_2^k + 192000x_3^k + 240000x_4^k + 228000x_5^k + 120000x_6^k + 168000x_7^k + 222000x_8^k + 300000x_9^k \right\},
\]
\[
+ \sum_{i=1}^{m} P_i x_i^k
\]
\[
+ EF_{T-1} \left( s_{i-1} - \frac{1}{20} \sum_{i=1}^{m} P_i x_i^k \right), \quad t = T, T-1, \cdots, 1, k = 1, \cdots, m.
\]

Note: Our problem has 9 tasks and 48 periods.

Hence, \( m = 9 \) and \( T = 48 \). Set \( \epsilon = \frac{1}{9} \).

Therefore,
\[
F_T \left( s_{i-1} \right) = \max \left\{ \begin{array}{l}
180000x_1^k + 150000x_2^k + 192000x_3^k + 240000x_4^k + 228000x_5^k + 120000x_6^k + 168000x_7^k + 222000x_8^k + 300000x_9^k
\end{array} \right\},
\]
\[
+ \sum_{i=1}^{m} P_i x_i^k
\]
\[
+ EF_{T-1} \left( s_{i-1} - \frac{1}{20} \sum_{i=1}^{m} P_i x_i^k \right), \quad t = T, T-1, \cdots, 1, \quad k = 1, \cdots, m.
\]

subject to:
\[
\sum_{i=1}^{m} x_i^k = S_{i-1}, \quad t = 48, 47, \cdots, 1,
\]
\[
x_i^k \geq 0, \quad t = 48, 47, \cdots, 1; \quad k = 1, \cdots, 9 \]
which is a parametric linear programming problem with 9 variables.

A program using MatLab was used for (20). At the end, the following results were obtained.

The profit over 48 weeks is given by

<table>
<thead>
<tr>
<th>Machines</th>
<th>Task x^1</th>
<th>Task x^2</th>
<th>Task x^3</th>
<th>Task x^4</th>
<th>Task x^5</th>
<th>Task x^6</th>
<th>Task x^7</th>
<th>Task x^8</th>
<th>Task x^9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Expected Profit (in Naira)</td>
<td>180,000</td>
<td>150,000</td>
<td>192,000</td>
<td>240,000</td>
<td>228,000</td>
<td>120,000</td>
<td>168,000</td>
<td>222,000</td>
<td>300,000</td>
</tr>
<tr>
<td>Probability of Breakdown (P_T)</td>
<td>2/11</td>
<td>1/10</td>
<td>2/13</td>
<td>5/21</td>
<td>3/19</td>
<td>1/12</td>
<td>1/9</td>
<td>4/15</td>
<td>5/18</td>
</tr>
</tbody>
</table>

Table 2. The Expected Initial Profit and Probability of Breakdown Machines.
\[ F_{T_0}(s_0) = \mathcal{N}91422000s_0 \]
\[ = \mathcal{N}91422000 \times 50 \]
\[ = \mathcal{N}4,571,100,000. \]

By theorem 1.5, as \( T \) approaches infinity, \( F_T(s_0) \) approaches \( F(s_0) \),
\[ F(s_0) = \lim_{T \to \infty} (\Gamma^T F)(s_0) \]
\[ = \lim_{T \to \infty} F_T(s_0) = \mathcal{N}409820000s_0 \]
\[ = \mathcal{N}20,491,000,000. \]

and the optimal policy \( \pi^*_t(s_0), (t = 1, 2, \cdots) \), approaches the limit zero.

**Discussion:** Figure 1 shows the expected profit that will accrue to the company over a period of 48 weeks. We found that at 48 weeks, the maximum profit that accrued to the company to be \( \mathcal{N}4,571,100,000. \) **Figure 2** shows the expected profit that will accrue to the company over an infinite weeks. It was found that at infinity, the maximum profit that will accrue to the company to be \( \mathcal{N}20,491,000,000. \)

**6. Conclusion**

Many production companies have for long been allocating resources to different tasks without putting into consideration certain factors that may hinder the realization of their objectives. This paper dealt with allocation of machines to tasks in order to maximize profit over finite and infinite horizon. Careful analysis of the situation reveals that adequate planning, prompt and effective maintenance can enhances the profitability of the company.

**REFERENCES**


