International Journal of Management Science
Vol 12, No 1, May 2006

A Converging Exact Algorithm for Determining an Optimal 3-Class-Based Dedicated Linear Storage System*

Moonhee Yang**
Department of Industrial Engineering, Dankook University,
San 29, Anseodong, Cheonan, Choongnam, 330-714, Korea

(Received Oct. 2005; Accepted Nov. 2005)

ABSTRACT

In this paper, we readdress a layout design problem, PTL[3], for determining an optimal 3-class-based dedicated linear storage layout in a class of unit load storage systems. Based on some fundamental properties derived, we provide a converging exact algorithm with $O(n\lceil \log n \rceil)$, which is more efficient than that of Yang and Kim [8] and can be applied to PTL[K] with $K \geq 4$ in order to reduce computational execution time. In addition, we prove that the necessary condition suggested by them is also a sufficient condition to PTL[3].

Keywords: Storage System, Class-based Dedicated Storage Layout, AS/RS

1. INTRODUCTION

K-class-based dedicated storage policy or simply K-class-based storage policy employs K zones in which lots from a class of products are stored randomly. Tompkins and White pointed out that class-based storage with randomized storage within each class can yield both the throughput benefits of dedicated storage and the space benefits of randomized storage. Also they suggested that in order to achieve both benefits, three to five classes may be defined [6].

There have appeared many papers such as Cho [3], Lee [5], Bozer [1], Chang [2], and Hausman [4] so on, which focused on both benefits or either the throughput benefits or the space benefits based on simulation techniques under some operating policies. Yang suggested a deterministic 2-class-based nonlinear storage

---

* The present research was conducted by the research fund of Dankook University in 2004.
** Email: myfriend@dankook.ac.kr

problem, PTN[2], and provided a heuristic algorithm with $O(n^2)$ [7]. He conjectured strongly that PTN[2] is NP-complete. However, by restricting one-way travel time to a linear form, he provided a greedy exact algorithm. Using the same linearity assumption for the travel time, Yang and Kim addressed a deterministic 3-class-based linear storage problem, PTL[3], and provided an exact algorithm with $O(n \log n)$ [8].

In this paper, we readdress PTL[3] suggested by Yang and Kim, and provide more efficient converging exact algorithm than their algorithm, in addition to some useful techniques for reducing the computational execution time of PTL[3]. For convenience to readers, in section 2, we summaries PTL[3] and some previous results which are cited in this paper. In section 3, we provide a converging exact algorithm with $O(n \log n)$ for PTL[3] and prove that our algorithm gives an optimal solution. In addition, we provide useful properties required for the proof. In section 4, we give an example and additional comments. Throughout this paper, in order to denote optimality for a decision variable, a superscript (*) will be used at the upper right side of each symbol.

2. DESCRIPTION AND SOME PREVIOUS RESULTS

2.1 Description of PTL[3]

We summarize PTL[3] described in Yang and Kim [8] as follows. The storage system consists of R storage locations each of which accommodates only one unit load The storage/retrieval operation is based on the 3-zone-based storage policy and within each zone, random assignment rule (RAN rule) is used.

The expected one-way travel time from a Pick-up/Deposit (P/D) station to storage location j is given as t_j for j =1, 2, ..., R. Without loss of generality, it is assumed that $t_1 \le t_2 \le \cdots \le t_R$. Let A_k be a set of storage locations assigned to zone k for k = 1, 2, 3. Given the t_j-nondecreasing ordering, we assign the first $|A_1|$ storage locations to A_1, and assign the next $|A_2|$ storage locations to A_2, and the remaining storage locations to A_3, where $|A_k|$ denotes the cardinality of set A_k.

An arriving replenishment lot of a product i, the size of which is r_i in unit load, contains a single product and are assigned randomly to open storage locations in one of three separate zones by using an storage/retrieval (S/R) machine or operator which or who can carry only one unit load at a time. Let C_k be the set (or class) of products assigned to zone k. Then space requirement or the number of storage locations required for class k, R_k, can be expressed as $R_k = |A_k| = \sum_{i \in C_k} r_i$.

The average demand rate (d_i unit loads/unit time) for a product i, which is defined as the average number of retrievals per unit time, is given as a real constant in advance. Retrievals are performed on first-in first-out basis. The average demand rate(D_k) from zone k is obtained as $\sum_{i \in C_k} d_i$. Since a class contains at least one product, it can be assumed that $|C_k| \geq 1$.

Their objective is to minimize the expected single command (SC) travel time as follows. The expected SC travel time (T_k) to zone k can be expressed as $T_k = \frac{2}{|A_k|} \sum_{j \in A_k} t_j$. Since the probability of visiting zone k is $\frac{D_k}{D}$, where $D = \sum_{k=1}^{3} D_k$, $E(SC_3)$ can be expressed as

$$E(SC_3) = \sum_{k=1}^{3} \frac{D_k}{D} T_k \tag{1}$$

$$= \frac{2}{D} \sum_{k=1}^{3} \frac{D_k}{R_k} \sum_{j \in A_k} t_j \tag{2}$$

By replacing t_j with $p_j + q$, Eq.(2) can be further reduced as

$$E(SC_3) = \frac{p}{D} \{D + DR + (R_1 D_2 - D_1 R_2) + (R_2 D_3 - D_2 R_3) + (R_1 D_3 - D_1 R_3)\} + 2q \tag{3}$$

where $R = \sum_{k=1}^{3} R_k$. Since D, R, p and q are constant and each class must contain at least one product, PTL[3] can be stated as

PTL[3]: Given n products with $\{(r_i, d_i), i = 1, 2, \cdots, n\}$, find an optimal partition, $P^*(3) = \{C_1^*, C_2^*, C_3^*\}$ such that the following function is minimized :

Minimize $Z = (R_1 D_2 - D_1 R_2) + (R_2 D_3 - D_2 R_3) + (R_1 D_3 - D_1 R_3)$

subject to $|C_k| \geq 1$ for k=1, 2, 3

$D_k = \sum_{i \in C_k} d_i$ for k=1, 2, 3

$R_k = \sum_{i \in C_k} r_i$ for k=1, 2, 3

2.2 Some Previous Results of PTL[2] and PTL[3]

Define PTL[K] and $P^*(K)$ (or $\{C_1^*, C_2^*, \cdots, C_K^*\}$) to be the K-class-based linear storage problem and an optimal solution to PTL[K] respectively. For integers a and b, define a class activity index, $CAI(a,b)$, as $\sum_{i=a}^{b} d_i / \sum_{i=a}^{b} r_i$. The following two theorems are due to Yang [7] and Yang and Kim [8].

Theorem 1. $P^*(2)$ is optimal to PTL[2] if and only if $P^*(2)$ satisfies

$$\mathrm{PAI}(i_1) \geq \mathrm{CAI}(1,n) > \mathrm{PAI}(i_2) \quad \text{for} \quad i_k \in C_k^*,\ k=1,\ 2,$$

and $P^*(2)$ is one of the partitions based on a PAI-nonincreasing ordering.

Theorem 2. If $P^*(3)$ is optimal to PTL[3], then $P^*(3)$ satisfies

$$\mathrm{PAI}(i_1) \geq \mathrm{CAI}(1,N_2^*) > \mathrm{PAI}(i_2) \geq \mathrm{CAI}(N_1^*+1,n) > \mathrm{PAI}(i_3) \quad \text{for} \quad i_k \in C_k^*,\ k=1,\ 2,\ 3 \qquad (4)$$

And $P^*(3)$ is one of the partitions based on a PAI-nonincreasing ordering.

Since $CAI(1,n)$ is a constant as $\frac{D}{R}$, the greedy algorithm with $O(n)$ was given based on Theorem 1; assign each product to C_1^* if and only if $\mathrm{PAI}(i) \geq \frac{D}{R}$. However, since both $\mathrm{CAI}(1,N_2^*)$ and $\mathrm{CAI}(N_1^*+1,n)$ in Theorem 2 are unknown constants until $P^*(3)$ is obtained, based on a PAI-nonincreasing ordering and the partial enumeration from a set of solutions, Yang and Kim [8] proposed an exact algorithm with $O(n \log n)$.

3. A CONVERGING ALGORITHM AND PROPERTIES

3.1 A Converging Algorithm

From theorem 1, an important property which contributes to reducing computational complexity of PTL[3] can be obtained.

Corollary 1. Given a data set, $\{(r_i, d_i), i = 1,2,\cdots,n\}$, if there are m products with the same PAI value, say d/r, and if the number of different PAI values is greater than or equal to three, then replacing those m products with a single product having space requirement $m \cdot r$ and average retrieval rate $m \cdot d$ does not affect the optimality to PTL[3].

Proof: From Theorem 2, those products with the same PAI value can be assigned to the same class. In addition, replacing those m products with a single product having PAI=$m \cdot d / m \cdot r$ does not affect the optimal CAI values. □

Applying Corollary 1 to an original input data set, we can obtain a modified input data set where PAI's are unique for each product. For this reason, we use the term "PAI-decreasing" instead of "PAI-nonincreasing". In real-life warehouse, some products, especially slow-moving items, are likely to have almost same PAI values. In this case, the products can be treated as a single product for a near optimal solution. Note that the idea of Corollary 1 happens to be the same as that of Page and Paul's algorithm (1976) for minimizing total inventory cost.

Without loss of generality, assume that a PAI-decreasing ordering is given as O=$\{1,2,\cdots,n\}$. Then any candidate solution (X) to PTL[3] can be expressed as (N_1, N_2), where N_k is the number of products assigned to classes 1 through k for k=1, 2.

It can be observed from Eq.(1) that the minimization of $E(SC_3)$ given N_1 products of an O assigned to class 1 is equivalent to solving a 2-class-based linear storage problem with the last $(n - N_1)$ products of an O. Let this "conditional optimal" solution given N_1 be expressed as $X_c = (N_1, N_2^c(N_1))$, where $N_2^c(N_1)$ can be determined from Theorem 1 such that

$$\text{PAI}(N_2^c(N_1)) \geq \text{CAI}(N_1 + 1, n) > \text{PAI}(N_2^c(N_1) + 1) \tag{5}$$

Clearly, we have

$$E(SC_3 \mid X_c) \leq E(SC_3 \mid X) \tag{6}$$

where $E(SC_3 \mid X)$ is the expected SC travel time given X. For simplicity, we use "conditional" instead of "conditional optimal". In the similar manner, the conditional N_1 given N_2, denoted by $N_1^c(N_2)$, can be determined such that

$$\text{PAI}(N_1^c(N_2)) \geq \text{CAI}(1, N_2) > \text{PAI}(N_1^c(N_2) + 1) \tag{7}$$

Now, we provide a different exact algorithm, called as ALGCONV[3]. Let $N_k(T)$ be the number of products assigned to classes 1 through k at the end of T-th iteration for k=1, 2. Let L_1 be the lower bound of N_1^* suggested by Yang and Kim [8]. Consider our algorithm with two phases; initialization phase and converging phase as follows.

ALGCONV[3]

Step 1. (Initialization phase)

Take a PAI-decreasing ordering, $O=\{1,2,\cdots,n\}$.

$N_1(0) \leftarrow L_1$

$N_2(0) \leftarrow N_2^c(L_1)$

$T \leftarrow 0$

Step 2. (Converging phase)

$T \leftarrow T+1$

Determine $N_1(T)$ by solving PTL[2] with the first $N_2(T-1)$ products;

i.e., $N_1(T) \leftarrow N_1^c(N_2(T-1))$

If $N_1(T) = N_1(T-1))$, then $(N_1^e, N_1^e) \leftarrow (N_1(T), N_2(T-1))$ and stop.

Determine $N_2(T)$ by solving PTL[2] with the last $(n-N_1(T))$ products;

i.e., $N_2(T) \leftarrow N_2^c(N_1(T))$

Go to Step 2.

At step 1, we obtain an O and start with any conditional solution, $(N_1(0), N_2(0))$, where $N_1(0) = L_1$ and $N_2(0) = N_2^c(L_1)$. Note that any conditional solution with $1 \le N_1(0) \le (n-2)$ can be used as the starting solution. We set T=0.

At step 2, we increase T by one, and by holding class 3 fixed, we determine $N_1(T)$ by solving PTL[2] using the first $N_2(T-1)$ products of an O. Using Eq.(7), we determine satisfying the following equation;

$$\text{PAI}(N_1(T)) \ge \text{CAI}(1, N_2(T-1)) > \text{PAI}(N_1(T)+1) \tag{8}$$

That is, $N_1(T) = N_1^c(N_2(T-1))$. We check if there is change in $N_1(T)$. That is, if $N_1(T) = N_1(T-1))$, then our algorithm terminates with $(N_1(T), N_2(T-1))$. Let the terminated solution be $X_E = (N_1^e, N_2^e)$. Clearly, for $i_k \in C_k^*$, k=1, 2, 3, we have,

$$\mathrm{PAI}(i_1) \geq \mathrm{CAI}(1, N_2^e) > \mathrm{PAI}(i_2) \geq \mathrm{CAI}(N_1^e + 1, n) > \mathrm{PAI}(i_3) \tag{9}$$

Note that it is not necessary to check if there is change in $N_2(T-1)$ because $N_2(T-1)$ changes if and only if $N_1(T)$ changes. If $N_1(T) \neq N_1(T-1)$, then we determine $N_2(T)$ by solving PTL[2] using the last $(n - N_1(T))$ products of an O. Using Eq.(5), we determine $N_2(T)$ satisfying the following equation;

$$\mathrm{PAI}(N_2(T)) \geq \mathrm{CAI}(N_1(T)+1, n) > \mathrm{PAI}(N_2(T)+1) \tag{10}$$

That is, $N_2(T) = N_2^c(N_1(T))$. Thus, we have $\left(N_1(T), N_2(T)\right)$ at the end of T-th iteration, with which we repeat step 2.

3.2 Convergence to an Optimal Solution

In order to prove that X_E generated by ALGCONV[3] is an optimal solution, it is enough to prove that (i) $E(SC_3 \mid X_E) \leq E(SC_3 \mid X)$ for any solution X and (ii) the initial conditional solution, $(N_1(0), N_2(0))$, converges to X_E by ALGCONV[3]. Before proving that X_E is an optimal solution, we need to prove some properties as follows.

Property 1. Given a PAI-decreasing ordering, $O=\{1, 2, \ldots, n\}$, and integers a, b, c, and d such that $a \leq b$ and $a \leq c < d$, $\mathrm{CAI}(a,c) > \mathrm{CAI}(b,d)$.

Proof: We consider two cases for the proof; (i) $b \geq a+1$, (ii) $b = a$.
(i) Since $\mathrm{CAI}(a, b-1) > \mathrm{PAI}(i)$ for i=b, b+1,..., d and $\mathrm{CAI}(b,c) > \mathrm{PAI}(i)$ for i=c+1, ... d, we have

$$\begin{aligned}
&\mathrm{CAI}(a,c) - \mathrm{CAI}(b,d) \\
&\quad = \frac{\sum_{i=a}^{b-1} d_i + \sum_{i=b}^{c} d_i}{\sum_{i=a}^{b-1} r_i + \sum_{i=b}^{c} r_i} - \frac{\sum_{i=b}^{c} d_i + \sum_{i=c+1}^{d} d_i}{\sum_{i=b}^{c} r_i + \sum_{i=c+1}^{d} r_i} \\
&\quad = \frac{1}{\sum_{i=a}^{c} r_i \sum_{i=b}^{d} r_i} \left[\sum_{i=a}^{b-1} r_i \sum_{i=b}^{d} r_i \{\mathrm{CAI}(a,b-1) - \mathrm{PAI}(i)\} + \sum_{i=b}^{c} r_i \sum_{i=c+1}^{d} r_i \{\mathrm{CAI}(b,c) - \mathrm{PAI}(i)\} \right] > 0 .
\end{aligned}$$

(ii) Let $D_{ac} = \sum_{i=a}^{c} d_i$ and $R_{ac} = \sum_{i=a}^{c} r_i$. Since $\frac{D_{ac}}{R_{ac}} - \text{PAI}(i) > 0$ for i=c+1, c+2, ..., d, we have,

$$\text{CAI}(a,c) - \text{CAI}(b,d) = \frac{D_{ac}}{R_{ac}} - \frac{D_{ac} + \sum_{i=c+1}^{d} d_i}{R_{ac} + \sum_{i=c+1}^{d} r_i}$$

$$= \frac{1}{R_{ac}\left(R_{ac} + \sum_{i=c+1}^{d} r_i\right)} \sum_{i=c+1}^{d} r_i R_{ac}\left(\frac{D_{ac}}{R_{ac}} - \text{PAI}(i)\right) > 0 . \square$$

Lemma 1. $E(SC_3 \mid X_c)$ is a discrete convex function of N_1. That is,
(i) If $N_1 < N_1^e$, then $E(SC_3 \mid X_c)$ is a decreasing function of N_1.
(ii) If $N_1 > N_1^e$, then $E(SC_3 \mid X_c)$ is an increasing function of N_1.

Proof: Consider two conditional solutions to PTL[3]: $X_{c1} = (N_1, N_2^c(N_1))$ and $X_{c2} = (N_1 + 1, N_2^c(N_1 + 1))$. For simplicity, let $N_2 = N_2^c(N_1)$ and $N_2' = N_2^c(N_1 + 1)$. Let C_k and C_k' be the set of products assigned to class k for k=1, 2, 3, given X_{c1} and X_{c2} respectively. Since $N_2^c(N_1)$ is a nondecreasing function of N_1, we have $N_2' \geq N_2$ and

$$C_2 - C_2' = \{N_1 + 1\}$$
$$C_2 \cap C_2' = \{N_1 + 2, N_1 + 3, \cdots, N_2\}$$
$$C_2' - C_2 = \{N_2 + 1, N_2 + 2, \cdots, N_2'\}$$

Let R_k and D_k be the number of storage locations and average retrieval rate for class k respectively given X_{c1}. Let $d = d_{N_1+1}$ and $r = r_{N_1+1}$. Since X_{c1} and X_{c2} are conditional solutions, using Eq.(5), we have for $i \in C_2' - C_2$,

$$\text{PAI}(i) < \text{CAI}(N_1 + 1, n) \text{ and} \quad (11)$$
$$\text{PAI}(i) \geq \text{CAI}(N_1 + 2, n) \quad (12)$$

Define $Z(X)$ as the objective function of PTL[3] given a solution X and let $g(N_1) = Z(X_{c1})$ and $g(N_1 + 1) = Z(X_{c2})$. It is enough to show that $g(N_1) \geq$

$g(N_1+1)$ in the range of $N_1 < N_1^e$ and $g(N_1) < g(N_1+1)$ in the range of $N_1 > N_1^e$. Since $g(N_1)$ and $g(N_1+1)$ can be rewritten as

$$g(N_1) = \{R_1(D_2+D_3) - D_1(R_2+R_3)\} + \{R_2(D_2+D_3) - D_2(R_2+R_3)\}$$
$$= R_1 D - D_1 R + \sum_{i \in C_2} \{r_i(D_2+D_3) - d_i(R_2+R_3)\} \tag{13}$$

$$g(N_1+1) = (R_1+r)D - (D_1+d)R + \sum_{i \in C_2} \{r_i(D_2+D_3-d) - d_i(R_2+R_3-r)\} \tag{14}$$

we have

$$g(N_1) - g(N_1+1) = (R_1+R_2)r\{\mathrm{PAI}(N_1+1) - \mathrm{CAI}(1,N_2)\}$$
$$+(R_2+R_3-r)\sum_{i=N_2+1}^{N_2'} r_i\{\mathrm{PAI}(i) - \mathrm{CAI}(N_1+2,n)\} \tag{15}$$

Suppose that $N_1 < N_1^e$. Since we have $\mathrm{PAI}(i) \ge \mathrm{CAI}(1,N_2)$ for I $i \in \{1,2,\cdots,$ $N_1^c(N_2^c(N_1))\}$ from Eq.(8) and since $N_1+1 \le N_1^c N_2^c(N_1)$ from Property 2-(i), it follows that $\mathrm{PAI}(N_1+1) \ge \mathrm{CAI}(1,N_2)$. Thus the first term in the right side of Eq.(15) is nonnegative. From Eq.(12), the second term in the right side of Eq.(15) is nonnegative. It follows that $g(N_1) \ge g(N_1+1)$.

Now, suppose that $N_1 > N_1^e$. By rearranging Eq.(15), we have

$$g(N_1) - g(N_1+1) = (R_1+R_2+R'')r\{\mathrm{PAI}(N_1+1) - \mathrm{CAI}(1,N_2')\}$$
$$+\sum_{i=N_2+1}^{N_2'} (R_2+R_3)r_i\{\mathrm{PAI}(i) - \mathrm{CAI}(N_1+1,n)\} \tag{16}$$

where $R'' = \sum_{i=N_2+1}^{N_2'} r_i$.

Since $N_1 > N_1^e$, using Property 2-(ii), $N_1^c(N_2') < N_1+1$.

Since $\mathrm{CAI}(1,\ N_2') > \mathrm{PAI}(N_1^c(N_2')+1)$ (using Eq.(7))

$$\ge \mathrm{PAI}(N_1+1) \quad (\because N_1^c(N_2') \le N_1)$$

it follows that $\mathrm{PAI}(N_1+1) < \mathrm{CAI}(1,\ N_2')$. Thus the first term in the right side of Eq.(16) is negative. From Eq.(11), the second term in the right side of Eq.(16) is negative. It follows that $g(N_1) < g(N_1+1)$. □

Theorem 3. For any solution, X=(N_1, N_2), $E(SC_3 | X_E) \le E(SC_3 | X)$.

Proof: For any solution $X=(N_1, N_2)$, and given N_1, there exists a conditional solution, $X_c = (N_1, N_2^c(N_1))$, from Eq.(6) such that $E(SC_3 | X_c) \le E(SC_3 | X)$. Since $E(SC_3 | X_c)$ is a discrete convex function of N_1 from Lemma 1, we have $E(SC_3 | X_E) \le E(SC_3 | X_c)$. Therefore $E(SC_3 | X_E) \le E(SC_3 | X)$. □

Now, we prove that any initial conditional solution, $(N_1(0), N_2(0))$, converges to X_E while $E(SC_3)$ is minimized by ALGCONV[3]. Before proving the convergence, we need to prove properties about the ranges of both $N_2(T-1)$ and $N_1(T)$ given $N_1(T\text{-}1)$ as follows.

Property 2. Range of $N_2(T\text{-}1)$ given $N_1(T\text{-}1)$

(i) If $N_1(T\text{-}1) = N_1^e$, then $N_2(T\text{-}1) = N_2^e$

(ii) If $N_1(T\text{-}1) < N_1^e$, then $N_1^e \le N_2(T-1) \le N_2^e$

(iii) If $N_1(T\text{-}1) > N_1^e$, then $N_2^e \le N_2(T-1) \le (n-1)$

Proof: (i) If $N_1(T\text{-}1) = N_1^e$, then $N_2(T\text{-}1) = N_2^c(N_1(T\text{-}1)) = N_2^c(N_1^c) = N_2^e$.

(ii) Since $\mathrm{PAI}(N_2(T\text{-}1)+1) < \mathrm{CAI}(N_1(T\text{-}1)+1, n)$ (using Eq.(5))

$< \mathrm{CAI}(1, N_2^e)$ (using Property 1)

$\le \mathrm{PAI}(N_1^e)$ (using Eq.(9)),

it follows that $\mathrm{PAI}(N_2(T\text{-}1)+1) < \mathrm{PAI}(N_1^e)$, i.e., $N_2(T-1) \ge N_1^e$. In the similar manner above,

since $\mathrm{PAI}(N_2(T\text{-}1)) \ge \mathrm{CAI}(N_1(T\text{-}1)+1, n)$ (using Eq.(5))

$> \mathrm{CAI}(N_1^e + 1, n)$ (using Property 1)

$> \mathrm{PAI}(N_2^e + 1)$ (using Eq.(9)),

it follows that $\mathrm{PAI}(N_2(T\text{-}1)) > \mathrm{PAI}(N_2^e + 1)$, i.e., $N_2(T-1) \le N_2^e$. Therefore, (ii) holds true.

(iii) Since $\mathrm{PAI}(N_2(T\text{-}1)+1) < \mathrm{CAI}(N_1(T\text{-}1)+1, n)$ (using Eq.(5))

$< \mathrm{CAI}(N_1^e + 1, n)$ (using Property 1)

$\le \mathrm{PAI}(N_2^e)$ (using Eq.(9)), it follows that $\mathrm{PAI}(N_2(T-1)+1) < \mathrm{PAI}(N_2^e)$, i.e., $N_2^e \le N_2(T-1)$. Since $|C_3| \ge 1$, $N_2(T-1) \le (n-1)$. Hence, (iii) holds true. □

Property 3. Range of $N_1(T)$ given $N_1(T-1)$

(i) If $N_1(T-1) = N_1^e$, then $N_1(T) = N_1^e$.

(ii) If $N_1(T-1) < N_1^e$, then $N_1(T-1) < N_1(T) \le N_1^e$.

(iii) If $N_1(T-1) > N_1^e$, then $N_1^e \le N_1(T) < N_1(T-1)$.

Proof: (i) If $N_1(T-1) = N_1^e$, then $N_1(T) = N_1^c(N_2^c(N_1(T-1))) = N_1^c(N_2^c(N_1^e)) = N_1^c(N_2^e) = N_1^e$.

(ii) Since

$$
\begin{aligned}
\mathrm{PAI}(N_1(T)) &\ge \mathrm{CAI}(1, N_2(T-1)) && \text{(using Eq.(7))} \\
&\ge \mathrm{CAI}(1,\ N_2^e) && (\because\ N_2^c(N_1) \le N_2^e \text{ from Property 2-(ii)}) \\
&> \mathrm{PAI}(N_1^e+1) && \text{(using Eq.(9))},
\end{aligned}
$$

it follows that $\mathrm{PAI}(N_1(T)) > \mathrm{PAI}(N_1^e+1)$, i.e., $N_1(T) \le N_1^e$. For the proof that $N_1(T-1) < N_1(T)$, we use an induction method as follows. Suppose that $N_1(T-1) = 1$. Then $N_1(T) = N_1^e = 1$. This is a contradiction to the assumption that $N_1(T-1) < N_1^e$. Thus $1 = N_1(T-1) < N_1(T)$ and (ii) is true for $N_1(T-1) = 1$. Suppose that (ii) is true for $N_1(T-1) = m$, i.e.,

$$
\text{if } m < N_1^e, \text{ then } m < N_1^c(N_2^c(m)) \tag{17}
$$

It is enough to prove that

$$
\text{if } m+1 < N_1^e, \text{ then } m+1 < N_1^c(N_2^c(m+1))
$$

Since

$$
\begin{aligned}
\mathrm{PAI}(N_1^c(N_2^c(m)) &\ge \mathrm{CAI}(1,\ N_2^c(m)) && \text{(using Eq.(7))} \\
&\ge \mathrm{CAI}(1,\ N_2^c(m+1)) && (\because\ N_2^c(m) \le N_2^c(m+1)\,) \\
&> \mathrm{PAI}(N_1^c(N_2^c(m+1)+1) && \text{(using Eq.(7))}
\end{aligned}
$$

it follows that $N_1^c(N_2^c(m)) \le N_1^c(N_2^c(m+1))$. From Eq.(17), we have $m+1 \le N_1^c$ $(N_2^c(m+1))$. If $m+1 = N_1^c(N_2^c(m+1))$, then $N_1(T-1) = m+1 = N_1^e$. This is a contradiction to the assumption that $N_1(T-1) < N_1^e$. Thus $m+1 < N_1^c(N_2^c(m+1))$

and (ii) is true for $N_1(T-1) = m+1$. Therefore (ii) holds true for all $N_1(T\text{-}1) < N_1^e$. In the similar manner above, (iii) can be proved. □

Lemma 2. (Minimization) As T increases, $E(SC_3)$ decreases until ALGCONV[3] is terminated. That is,
(i) $E(SC_3 \mid (N_1(T\text{-}1), N_2(T\text{-}2))) > E(SC_3 \mid (N_1(T\text{-}1), N_2(T\text{-}1)))$
(ii) $E(SC_3 \mid (N_1(T\text{-}1), N_2(T\text{-}1))) > E(SC_3 \mid (N_1(T), N_2(T\text{-}1)))$

Proof: Define $X_{c1} = (N_1(T\text{-}1), N_2(T\text{-}2))$, $X_{c2} = (N_1(T\text{-}1), N_2(T\text{-}1))$, and $X_{c3} = (N_1(T), N_2(T\text{-}1))$. Since given $N_1(T\text{-}1)$, $N_2(T\text{-}1)$ is determined as $N_2^c(N_1(T\text{-}1))$ in ALGCONV[3], we have $E(SC_3 \mid X_{c1}) \geq E(SC_3 \mid X_{c2})$ from Eq.(3). If $E(SC_3 \mid X_{c1}) = E(SC_3 \mid X_{c2})$, then $N_2(T\text{-}2) = N_2(T\text{-}1)$ from Theorem 1. This is a contradiction that ALGCONV[3] is not terminated. Therefore (i) holds true. Similarly, since given $N_2(T\text{-}1)$, $N_1(T)$ is determined as $N_1^c(N_2(T\text{-}1))$ in ALGCONV[3], we have $E(SC_3 \mid X_{c2}) \geq E(SC_3 \mid X_{c3})$ from Eq.(3). Note that equality holds if and only if $N_1(T\text{-}1) = N_1(T)$. □

Lemma 3. (Convergence) Any initial solution, $(N_1(0), N_2(0))$, converges to X_E by ALGCON[3]. That is, for $T \geq 1$,
(i) if $N_1(T\text{-}1) < N_1^e$, then $N_1(T-1) < N_1(T) \leq N_1^e$ and $N_1^e \leq N_2(T-1) < N_2(T) \leq N_2^e$
(ii) if $N_1(T\text{-}1) > N_1^e$, then $N_1^e \leq N_1(T) < N_1(T-1)$ and $N_2^e \leq N_2(T) < N_2(T-1)$
(iii) if $N_1(T\text{-}1) = N_1^e$, then $N_2(T\text{-}1) = N_2(T) = N_2^e$ and $N_1(T) = N_1^e$.

Proof: (i) From Lemma 2-(ii) and Lemma 1-(ii), we have $N_1(T\text{-}1) < N_1(T) \leq N_1^e$ and $N_1^e \leq N_2(T\text{-}1)$. It remains to prove that $N_2(T-1) < N_2(T) \leq N_2^e$. Since $N_2(T) = N_2^c(N_1(T))$ and

Since $\text{PAI}(N_2(T)+1) < \text{CAI}(N_1(T)+1, n)$	(using Eq.(10))
$< \text{CAI}(N_1(T\text{-}1)+1, n)$	($\because N_1(T) > N_1(T\text{-}1)$)
$\leq \text{PAI}(N_2(T-1))$	(using Eq.(10)),

it follows that $N_2(T) \geq N_2(T-1)$. If $N_2(T) = N_2(T-1)$, then $N_1(T) = N_1(T-1) = N_1^e$. This is a contradiction to the assumption that $N_1(T\text{-}1) < N_1^e$. Hence, we have $N_2(T) > N_2(T-1)$.

Since $\text{PAI}(N_2(T)) \geq \text{CAI}(N_1(T)+1, n)$ (using Eq.(10))

$\geq \text{CAI}(N_1^e+1, n)$ ($\because N_1(T) \leq N_1^e$)

$> \text{PAI}(N_2^e+1)$ (using Eq.(9)),

it follows that $N_2(T) \leq N_2^e$. Therefore, (i) is true if $N_1(T-1) < N_1^e$. In the similar manner above, (ii) can be proved.
(iii) $N_2(T-1) = N_2^c(N_1(T-1)) = N_2^c(N_1^e) = N_2^e$. $N_1(T) = N_1^c(N_2(T-1)) = N_1^c(N_2^e) = N_1^e$ $N_2(T) = N_2^c(N_1(T)) = N_2^c(N_1^e) = N_2^e$. □

Theorem 4. ALGCONV[3] solves PTL[3] in $O(n\lceil \log\ n \rceil)$

Proof: From Lemma 2 and Lemma 3, ALGCON[3] gives an optimal solution to PTL[3]. Step 1 requires $O(n\lceil \log\ n \rceil)$, and step 2 requires $O(n\lceil \log\ n \rceil)$ since the maximum number of iterations of step 2 is O(n) from Lemma 1 and finding $N_k(T)$ for k=1, 2 requires $O(\lceil \log\ n \rceil)$ steps. Hence, the time complexity of ALGCONV[3] is $O(n\lceil \log\ n \rceil)$.□

Corollary 2. If there exists P(3) satisfying Eq.(4), then P(3) is optimal to PTL[3].

Proof: From Theorem 2 and 4, it is trivial. □

4. AN EXAMPLE AND COMMENTS

Suppose that the input data are given as $\{(r_i, d_i), i = 1, 2, \cdots, 10\}$ = {(5,3), (15,4), (20,5), (10,2), (15,3), (20,4), (12,2), (7,1), (7,1), (50,6)} and $t_j = j$ for j=1, ..., 161. Since $\text{PAI}(4) = \text{PAI}(5) = \text{PAI}(6)$ and $\text{PAI}(8) = \text{PAI}(9)$, using Corollary 1, we replace products 4, 5, and 6 with product A having PAI(A) as 9/45 and product 8 and 9 with product B having PAI(B) as 2/14. Without loss of generality, the modified data set can be obtained as $\{(r_i, d_i), i = 1, 2, \cdots, 7\}$ ={(5,3), (15,4), (20,5), (45,9), (12,2), (14,2), (50,6)}. Following ALGCONV[3], we have

Step 1: (Initialization Phase) Since {PAI(i), i=1, 2, ...,7} = {0.6, 0.267, 0.25, 0.2, 0.167, 0.143, 0.12},

$O \leftarrow \{1, 2, 3, 4, 5, 6, 7\}$

$N_1(0) \leftarrow 1$

$N_2(0) \leftarrow N_2^c(N_1) = 4$ since PAI(4)=0.25>CAI(2, 7)=0.1795>0.1667= PAI(5)

$(N_1(0), N_2(0)) \leftarrow (1, 4)$

$T \leftarrow 0$

Step 2. (Converging Phase)

$T \leftarrow 1$

We find a conditional solution holding class 3.

$N_1(1) \leftarrow 3$ since PAI(3) = 0.25 > CAI(1, 4) = 0.2471 >0.2 = PAI(4)

Since $N_1(0) = 1 \neq 3 = N_1(1)$, we find a conditional solution holding class 1.

$N_2(1) \leftarrow 5$ since PAI(5) = 0.167 > CAI(4, 7) = 0.1570 > 0.143 = PAI(6)

$T \leftarrow 2$

We find a conditional solution holding class 3.

$N_1(2) \leftarrow 3$ since PAI(3) = 0.25 > CAI(1, 5) = 0.2371 > 0.2 = PAI(4)

Since $N_1(1) = 3 = N_1(2)$, ALGCONV[3] terminates with $(N_1^e, N_2^e) = (N_1^*, N_2^*) =$ (3,5), i.e., based on the modified data set, we assign products 1, 2 and 3 to class 1 and products 4 and 5 to class 2 and the remaining products to class 3, which gives $E(SC_3)$=131.6774. Based on the original data set, this solution corresponds to assigning products 1, 2 and 3 to class 1 and products 4, 5, 6, and 7 to class 2 and the remaining products to class 3.

5. CONCLUSIONS

In this paper, we provide a converging exact algorithm, ALGCONV[3] with $O(n\lceil \log\ n \rceil)$, for solving a 3-class-based dedicated linear storage problem. In addition, we prove that ALGCONV[3] gives an optimal solution and that the existing necessary condition proved by Yang and Kim [8] is also a sufficient condition. Our algorithm is very fast and could be applied to PTL[K] with $K \geq 4$ in order to reduce computational execution time.

Moreover, our algorithm could be utilized to develop a heuristic algorithm for solving a 3-class-based dedicated storage problem, which does not assume the linearity for the one-way travel time. Our strong conjectures are that (i) the augmented Eq.(4) is not a sufficient condition when $K \geq 4$, (ii) there might exist a greedy algorithm for PTL[3] like that for PTL[2]. These conjectures can be further investigated.

REFERENCES

[1] Bozer, Y. A. and M. S. Cho, "Throughput Performance of Automated Storage/Retrieval Systems under Stochastic Demand," Working paper, The University of Michigan, Ann Arbor, MI, 48109-2117, 1998.

[2] Chang, D. T., U. P. Wen, and J. T. Lin, "The Impact of Acceleration Deceleration on Travel Time Models for Automated Storage-Retrieval Systems," *IIE Transactions* 27, 1 (1995), 108-111.

[3] Cho, M. S. and Y. A. Bozer, "Storage Capacity Estimation for Automated Storage/Retrieval Systems under Stochastic Demand," *Journal of Korean Institute of Industrial Engineers* 27, 2 (2001), 169-175.

[4] Hausman, W. H., L. B. Schwartz, and S. C. Graves, "Optimal Storage Assignment in Automatic Warehousing Systems," *Management Science* 22, 6 (1976), 629-638.

[5] Lee, M. K., "An Approach to Determining Storage Capacity of an Automated Storage/Retrieval System under Full Turnover Based Policy," *Journal of Korean Institute of Industrial Engineers* 24, 4 (1998), 579-589.

[6] Tompkins, J. A. and J. A. White, *Facilities Planning*, John Wiley and Sons Inc., NY., 1984, 335-338.

[7] Yang, M., "Analysis and Optimization of a 2-class-based Dedicated Storage System," *Journal of the Korea Institute of Industrial Engineers* 29, 3 (2003), 222-229.

[8] Yang, M. and S. Kim, "Optimization of a 3-class-based Dedicated Linear Storage System," *Journal of the Korea Institute of Industrial Engineers* 30, 3 (2004), 190-196.