Georgia Southern University
Digital Commons@Georgia Southern

13th IMHRC Proceedings (Cincinnati, Ohio. USA – 2014)

Progress in Material Handling Research

2014

Comparison of Alternative Configurations for Dense Warehousing Systems

Kevin R. Gue
Auburn University, kevin.gue@auburn.edu

G. Don Taylor
Virginia Tech

XXIX. COMPARISON OF ALTERNATIVE CONFIGURATIONS FOR DENSE WAREHOUSING SYSTEMS

G. Don Taylor
Virginia Tech

Kevin R. Gue
Auburn University

Abstract

In heavily constrained environments requiring very high density storage, traditional aisle-based warehousing may not provide viable options. One feasible manifestation of high density storage systems is the 'puzzle-based' system, in which unit loads are moved through the system via manipulation of empty (escort) locations to retrieve desired items. Another option would be the use of movable concentric rings, with escorts being utilized to enable lateral movements of unit loads between the rings. In this paper, the authors present analytical results to compare retrieval time performance for these two types of high density storage systems for randomly demanded items under various assumptions regarding the initial placement of escorts. The paper concludes that the use of movable concentric rings results in significant improvement to retrieval time performance in comparison to rectangular, puzzle-based systems, and further concludes that additional research is warranted.

I. Introduction and Related Literature

Many situations exist in which dense storage systems are needed. Part of the need for increased storage density is driven by a natural evolution toward more efficient and competitive systems. As Zaerpour et al. [1] point out, this push toward higher density, automated, intelligently controlled warehousing is due to three factors; the decreasing cost of innovative technology required to handle increased density, the demand responsiveness required by customers, and the high price of land in consumer positioned warehouses. Another driver for increased density is simply the fact that some environments are very tightly space constrained. For example, consider storage on an ocean-going vessel, and aircraft, or a spacecraft.

When required storage density is particularly necessary, aisle-less systems may be required. One manifestation of high density storage systems that is starting to get attention from the research community is the so called 'puzzle-based' storage system, in which unit loads are moved rectilinearly through aisle-less storage systems from an initial storage position to an input/output (I/O) location using one or more empty 'escorts'. This system is similar to the familiar children's game called the 15-puzzle depicted in Figure 1. In this game, 15 'tiles' are moved through the puzzle using a single empty 'escort' slot to create an ordered set or to form a picture.

2	8	10	13
11	1	5	14
9	7	15	6
4		3	12

Figure 1. The 15-Puzzle

The 15-puzzle and related ($n^2 - 1$) puzzles have received attention from the research community for more than a century, but only recently started to receive attention as a potential model for dense storage systems. Great overviews of the 15-puzzle specifically can be found in Archer [2] or Slocum and Sonneveld [3]. Gue and Kim [4] set the stage for puzzle-based storage research with their paper presenting retrieval time performance in a system with a single escort location, and for multiple escorts placed along a single, horizontal axis near a single I/O location. They also provide a comparison of system performance for puzzle-based storage systems in comparison to traditional aisle-based systems of similar size and configuration. This work was extended by Taylor and Gue [5] as they developed heuristic solutions for more complex escort scenarios and tested the performance of the heuristics with simulation. Kota et al. [6] then provided analytical solutions for more complex escort scenarios. Taylor and Gue [7] extended the work into a 3-dimensional system model and examined performance in scenarios that include vertical item movements in addition to rectilinear, horizontal movements. Zaerpour et al. [1] also describe a 3-dimensional system and suggest that it may be a viable option for an automated car parking system.

Some researchers have dealt with the problems of practical implementation of puzzle-based storage systems. Furmans et al. [8], for example, describe 'Flexconveyor' that is a modular, unit-sized conveyor system in which each unit module can convey in the four cardinal directions. The modules can be used in combination to build dense, custom storage solutions. They also describe KARIS, which is an extension of Flexconveyor. KARIS system modules can be used as automated guided vehicles (AGVs) for low volume situations or can be combined to form conveyors or even puzzle-based systems in situations requiring greater storage density. Gue et al. [9] present a system called 'Gridstore' in which decentralized control is used in a puzzle-based system to store and retrieve totes in a densely packed grid.

Other researchers have suggested alternative configurations for dense storage systems. Alfieri et al. [10] propose the use of sliding shelves that can be moved by AGVs. They also suggest dividing the warehouse into a finite number of sub-divisions with each having its own I/O location, and propose a heuristic to reduce retrieval time. Yu and de Koster [11] describe a compact storage system that makes use of an automated storage/retrieval system (AS/RS) platform with rotating conveyors for depth moves. DePuy and Taylor [12] investigate several puzzle-based games and develop integer programming formulations to solve the 'Rush Hour' traffic problem. The Rush Hour problem is similar to the 15-puzzle, but has the significance of requiring movement solutions when items are not of unit size. This adds significant complexity to item retrieval.

In this paper, we seek to introduce and examine another type of aisle-less warehousing concept and compare it to puzzle-based systems of the type introduced by Gue and Kim [4]. The new option would involve the use of movable concentric rings, with escorts being utilized to enable lateral movements of unit loads between the rings. One such configuration is compared to a 5 x 5 puzzle-based system in Figure 2. It is anticipated, and will be demonstrated herein, that the concentric rings system will produce faster retrieval times than similarly sized puzzle-based systems. It is further anticipated that because it would be easier to provide redundant drive mechanisms to the concentric circles that the circle-based system would produce more reliable retrieval time results because it would be much easier to work around failures in individual storage slots. The anticipated trade-offs for this faster retrieval time and increased reliability would be in terms of density and energy use. The circle-based system would have small losses in storage density, especially in interior circles, and moving an entire circle would generally require more energy than single-slot movements.

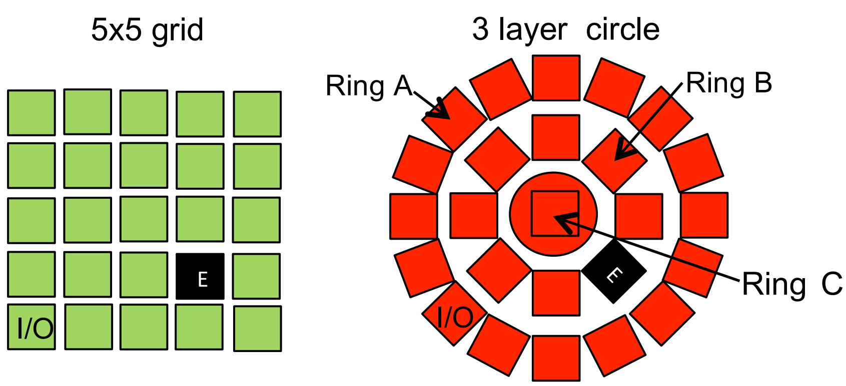

Figure 2. Rectangular Puzzle-Based System Compared to a Circle-Based Storage System.

Like rectangular systems, physical systems similar to the circle-based storage systems proposed herein are currently being designed and implemented, and supporting technology is evolving. Gudehus and Kotzab [13] do a nice job of describing transfer mechanisms, line, ring and star network structures, and specifically discuss circular conveyor systems. Bloss [14] describes the use of circular conveyors on which unit loads are carried during postal operations. Items are conveyed to/from the circular conveyors via tilt-trays or cross-belt conveyors of the type that might be best employed in the system described in this paper. Bastani [15] specifically addresses the reliability of closed-loop conveyor systems. More directly related to implementation of circle-based dense storage systems are systems reported by Woehr [16] and Urban Parking Concepts [17], in which dense urban car parking systems are discussed. Although the concepts presented herein would tend to work with unit-load systems of any size, they are certainly scalable to automobiles.

It is apparent that the technology required to support dense, puzzle-based storage systems is rapidly evolving. Dense storage systems are currently being designed and placed into service. A great deal of work remains to make them cost-effective and reliable, and analytical work remains to optimize their performance in all but the simplest scenarios. Trade-offs between storage density, cost of operation, and system configuration need to be examined. This paper examines one such trade-off and offers new analytical retrieval time results for circular puzzle-based storage systems.

II. Experimental Setting

As noted previously, this paper addresses unit-load systems that are scalable from small parcels to automobiles. For experimental purposes and to facilitate an apples-to-apples comparison, the physical systems used within this paper are a 5 x 5 grid and a 3-ring circle as depicted in Figure 2. Each system has 25 storage locations, and requires at least one empty slot (labeled 'E' for escort in the figure) to facilitate item movement to the I/O location on the lower left of each system.

For an initial comparison, we assume in this paper that maximum system density exists and that a single escort is available in the system. Determining optimal item retrieval policies in multiple escort scenarios becomes quite difficult from a combinatorial viewpoint, and the single escort assumption permits direct comparison of analytical results with earlier work. Gue and Kim [4] present optimal retrieval time results for a single escort initially at the I/O location, but within busy systems it may be difficult to re-organize escorts 'on the fly' to return the escort to a specific location between each retrieval. Researchers like Carlo and Giraldo [18] are making progress with 'rearrange while working' strategies in unit-load warehouses, but our goal is to provide robust results for *any* initial condition that the system may be in at the time of retrieval, regardless of what immediate order or replenishment history may have affected it. Thus, we will compare our analytical results for the circle-based system to the puzzle-based system results presented in Kota et al. [6], in which a single, randomly placed escort exists in the system. In a further effort to enable direct comparisons with Gue and Kim [4] and Kota et al. [6], we will assume that a single I/O position exists in the lower, left corner of the system as depicted in Figure 2.

As indicated in Figure 3, we assume full connectivity between adjacent units in the four cardinal directions in the puzzle-based systems, and inter-ring connectivity in the circle-based system at 8 discrete locations between each ring. We further assume that rings A, B, and C can rotate independently in both directions.

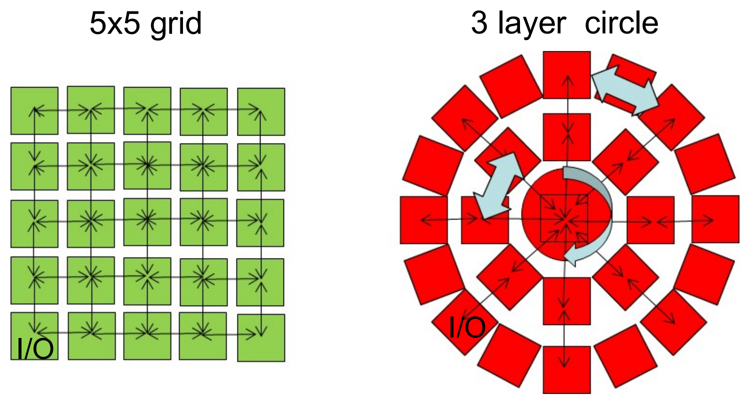

Figure 3. System Connectivity for Puzzle-Based and Circle-Based Storage Systems.

To facilitate discussion and to further ensure an 'apples to apples' comparison between the two system types, we have established positional notation as presented in Figure 4. Within subsequent analytical discussions and proofs, it is perhaps easiest to use (x,y) coordinates to count retrieval time moves, but for problem visualization and general discussion the Ring A, B, and C notation is much simpler.

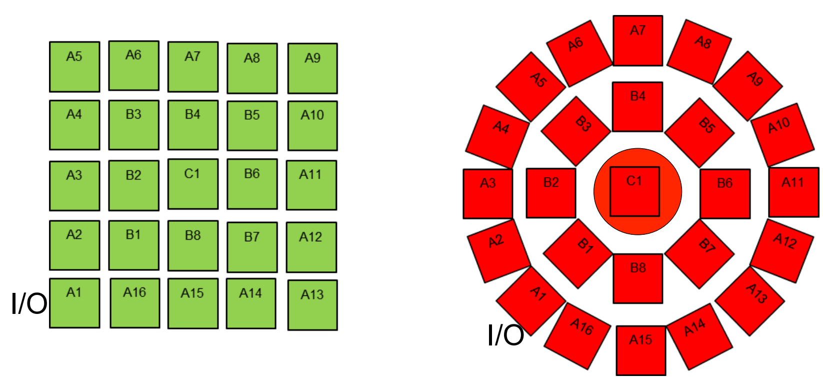

Figure 4. System Notation using Ring Membership

Further assumptions are that in the circle-based system that Rings A, B, and C can move independently, bi-directionally, and concurrently. It is assumed that the rings must be stationary during inter-ring transfers. It is assumed that in the puzzle-based system that each movement from one storage slot to another takes one time unit. In the circle-based system, each inter-ring transfer takes one unit of time, as does the rotation from one initial position to another. For example, rotation of an item initially in position A14 would require 3 time units to reach the I/O point at A1. Thus, we can refer to retrieval time as either 'moves' or 'time units'. Ring C would never be required to rotate more than $1/8^{th}$ of a full rotation in order to align itself for transfer, so the simplifying assumption is made that Ring C rotation time is a part of the one unit of time required to transfer to or from the ring. It is assumed that the initial random position of the single system escort is at (row, column) position (x,y) and that the demanded item is initially at position (i,j). For example, position (3,4) corresponds with position B6 using ring notation, and the I/O position is at position (1,1) or A1.

III. Analytical Estimates of Retrieval Time

In this section, we present analytical results for retrieval time under the experimental setting presented in the previous section. The optimal retrieval strategy in the puzzle-based system when there is one escort located anywhere in the system as depicted in Figure 5 is presented in Kota et al. [6]. To initialize our analytical discussion, that theorem will be presented here as baseline Theorem 0. The proof of the theorem is presented in full in Kota et al. [6] and is therefore not presented herein.

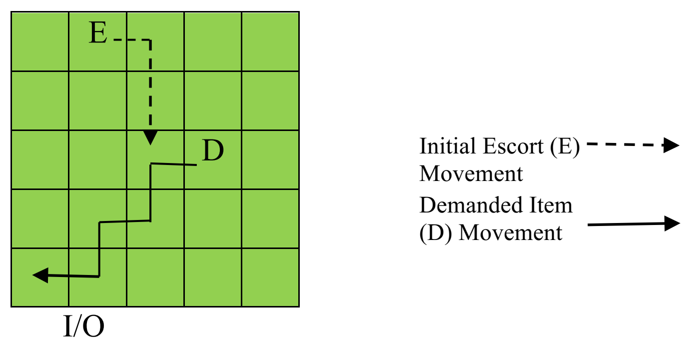

Figure 5. Item Retrieval with a Single Escort Initially in a Random Position.

<u>THEOREM 0:</u> When the demanded item (D), is located at position (i,j), where 'i' is the row and 'j' is the column of the demanded item as depicted in Figure 5, and a single escort is located randomly at position (x,y) in the system, and where m_1 is the number of moves to get the escort from an initial position into a desired 'usable' position for transport when $i > j$, m_2 is the number of moves to get an escort into a desired 'usable' position when $i < j$, and $m_3 = \min(m_1,m_2)$, the optimal retrieval time (RT) is:

$$
\begin{aligned}
RT &= 5i + j - 10 + m_1 && \text{when } i > j \\
&= 5j + i - 10 + m_2 && \text{when } i < j \\
&= 6i - 8 + m_3 && \text{when } i = j
\end{aligned}
$$

Where: $m_1 = |(i-1) - x| + |j - y|$
$m_2 = |i - x| + |(j-1) - y|$
$m_3 = \min(m_1, m_2)$

PROOF: Available in Kota et al. (10).

We now turn our attention to finding the optimal retrieval times for a single demanded item in circle-based systems when a single, randomly placed escort exists. Unfortunately, the optimal policies and proofs are not as simple as they are in the rectangular puzzle-based systems, and many special cases exist. Theorem 1 presents the retrieval time results when the demanded item is initially in Ring A. Theorems 2A-2F will present retrieval time results when the demanded item is in Ring B, and Theorem 3 will present the results when the demanded item is initially in Ring C. There are additional special cases and considerations within some of these theorems for specific combinations of initial escort (E) and demanded item (D) locations.

THEOREM 1: When the demanded item (D) is initially located at position (i,j) on Ring A as depicted in Figure 6, the optimal retrieval time (RT) is:

$$RT = (i-1) + (j-1) = i + j - 2$$

and the optimal strategy is simply to rotate the demanded item to the I/O position A1 on Ring A, rotating in the 'shortest' retrieval direction (select either direction if D is initially at position A9).

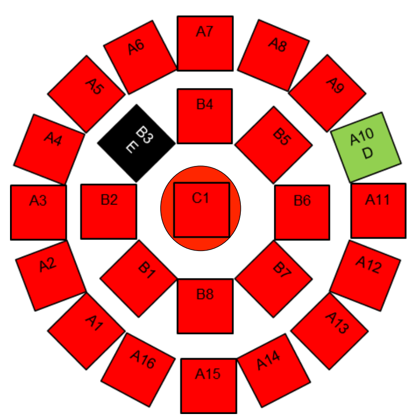

Figure 6. Retrieval from Ring A with a Single Escort Initially in a Random Position.

PROOF: There are 3 possible paths to move the demanded item to the I/O point; rotate on Ring A, use the escort to take a shortcut to the I/O point on ring B, or use multiple moves of the escort to cut through Ring C. If Ring A transport is selected, the location of E is irrelevant, and the retrieval time is simply (i-1) + (j-1) = i + j – 2. Cutting through Ring B can save D a maximum of 2 moves, when D is located at A9 and E is initially located at B5. Unfortunately, it also takes 2 moves to move to reposition E back onto Ring B and then onto Ring A to facilitate transfer of D to the I/O. It also would then take an additional 2 moves to over-rotate D to transfer E to A1, or 2 additional moves on Ring A to move E back to Ring A at B2 or B8, or 2 additional moves to move E to A1 through Ring C (Recall that the escort can move freely through the system). Regardless of how Ring B might be used to move E to A1 to facilitate movement of D, a total of 4 moves are added, resulting in a net loss of two moves in comparison to simply moving D to A1 on Ring A.

Moving D through Ring C can save a maximum of 4 moves when D is initially at A9 and E is initially at B5, but the problems associated with repositioning E for D movements are exacerbated. Because E must be available whenever D changes rings, the optimal path for D would be A9-B5-B4-C1-B2-B1-A1 (or the mirror image on the other side of the diagonal). Additional moves associated with this path defeat the apparent savings of four moves. First, Ring B must sit idle for 4 moves for the E transfers to/from Rings A and C. One move would be required to rotate D out of the way of E at B5. The escort must be rotated from B4 to B2 (or from B6 to B8) on Ring B (2 moves), and D must be rotated to B1 on Ring B (1 move). Thus, a net loss of four moves results in comparison to simply moving D to A1 on Ring A. It is never optimal to move D through Ring C.

<u>THEOREM 2:</u> When the_demanded item resides on Ring B, there are 6 different scenarios that must be considered; 1 scenario in which the escort is on Ring A (Theorem 2A), 4 scenarios in which the escort is on Ring B (THEOREMS 2B-2E), and 1 scenario in which the escort is in Ring C (THEOREM 2F). Proofs for the 6 scenarios follow.

<u>THEOREM 2A:</u> When the demanded item (D) is initially located at position (i,j) on Ring B, and the escort (E) is initially located on Ring A at position (x,y) as depicted in Figure 7, the optimal retrieval time (RT) depends upon the initial positions of D and E. Four cases exist:

<u>CASE 1:</u>	RT = Max[T_A,T_B] + 1	when	E in initial positions A1-A6 or A12-A16,
		or	E in position A7 or A11 and D not in B2 or B8,
		or	E in position A8 or A10 and D not in B1-B3 or B7-B8,
		or	E in A9 and D in B5.
<u>CASE 2:</u>	RT = Max[T_A,T_B]	when	E in position A7 or A11 and D in B2 or B8,
		or	E in position A8 or A10 and D in B1-B3 or B7-B8,
		or	E in A9 and D in B4 or B6.

CASE 3: $RT = Max[T_A,T_B] - 1$ when E in initial position A9 and D in B1, B3 or B7.

CASE 4: $RT = Max[T_A,T_B] - 2$ when E in initial position A9 and D in B2 or B8.

WHERE: $T_A = x + y - 2$ and $T_B = i + j - 4$

In CASE 1, the optimal retrieval policy is achieved in 3 steps:

1) Move E to A1 on Ring A.
2) Move D to B1 on Ring B.
3) Transfer D to the I/O point at position A1

In CASE 2, CASE 3, and CASE 4, the optimal retrieval policy is achieved in 5 steps:

1) If E is not initially at one of the 8 transfer points between Ring A and Ring B, move E on Ring A to the closest transfer point on the shortest path to A1 while concurrently moving D on Ring B one position closer to A1, unless that movement would place D on the path that E must take to A1. If D is initially at position B1, move it to position B2 or B8.
2) Move E through C to A1.
3) Complete rotation of D to B1 on Ring B.
4) Transfer D to the I/O point at position A1.

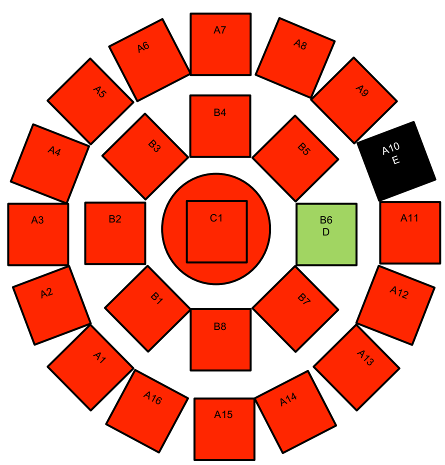

Figure 7. Retrieval from Ring B with a Single Escort Initially in Ring A.

CASE 1 PROOF: T_A clearly represents the time required to position E at A1 via rotation of Ring A. $T_A = (x - 1) + (y - 1) = x + y - 2$. Also, it is clear that T_B represents the time required to move D to B1 via rotation on Ring B. $T_B = (i - 2) + (j - 2) = i + j - 4$. To prove the optimality of the retrieval policy for CASE 1, we must demonstrate that it is never better to move D to Ring A before the I/O point at position A1 and that it is never possible to move E through Ring B or Ring C to achieve a shorter retrieval time.

When $T_A < T_B$, moving D to B1 paces RT. Otherwise, when $T_A \geq T_B$, moving the escort to A1 paces RT. Since transferring D to Ring A requires one unit of time regardless of when the transfer is made, early transfer to Ring A can lead to no reduction in RT when $T_A \geq T_B$ (assuming that the escort can be made available on the same side of a diagonal line from A1 to A9 as the demanded item), because D would then have to travel the exact same distance on Ring A as E would have under the stated optimal policy. When $T_A < T_B$, D movement paces RT and the shortest path to the I/O point for D is on Ring B. We know from THEOREM 1 that moving D from Ring B through Ring C is never optimal because of excessive repositioning requirements for E, and early movement of D to Ring A can only increase the travel time (by 2 to 4 time units depending on the point of transfer) because of the increased circumference of Ring A.

To complete the proof for CASE 1 scenarios, we must also demonstrate that no improvement can be made by moving E to A1 on the shortest path through Ring B or Ring C. This requires the specific consideration of various combinations of initial locations of E and D. When E is initially at A1, $T_A = 0$ and from the discussion above we know that the optimal RT requires D to rotate to B1 on Ring B. When E is initially at positions A2-A6 or A12-A16, E movement through Ring B is greater than or equal to movement through Ring A, so no savings can be accrued by moving E from Ring A.

When E is initially at positions A7 or A11, savings associated with cutting through the interior of the system can be accrued only when D is initially in position B2 or B8 (this situation will be described in the discussion of CASE 2). When E is in A7 or A11, a savings of one time unit can be accrued by moving E on Ring B, and a savings of 2 units can be accrued by moving through Ring C as well, in comparison to movement of E on Ring A alone. If D must be rotated out of the way to permit this movement of E (i.e., the initial position of D is in position B1, or perhaps B4 or B6), then the additional time to rotate D out of the way equals the savings, and the original CASE 1 policy is still optimal. If the initial position of D is in B3, B5, or B7, Ring B must move 2-4 positions to move D to B1 on Ring B, and the savings associated with E movement is negated by delays to Ring B movement caused by the transfers of E to and from Ring B.

When E is initially at A8 or A10, and when D is initially in in B4 or B6, the shortest path for E to A1 is via transfer to Ring B at either A7 or A11. Clearly, while Ring A rotates for E movement to Ring B, Ring B should concurrently rotate D toward B1. In addition to saving subsequent rotation time, this action potentially serves the dual purpose of moving D from B4 or B6 to facilitate E transfer to Ring B. After one movement, the situation is identical to the previously discussed case in which initial conditions are E at A7 or All, and D at B3 or B7, in which the CASE 1 strategy is already proven optimal. Similarly, if E transfer were made at A9 instead of A7 or A11, a savings of two time units is achieved for E movement, but this savings is negated by delays to Ring B movement. Similarly, when D is initially at B5, the savings of 2 units of time created by moving E through the interior of the system is defeated by the increased delays in Ring B movement.

When E is initially in position A9, real savings can be accrued by moving E through the interior of the system in most situations. The exception is when D is initially in position B5. In this case, the savings of 4 units of time accrued by E movement through Ring C is negated by an increase of one unit of time to move D from the initial B5 position to permit E transfer, and by the three subsequent non-concurrent moves for D on Ring B. Similarly, moving E to B1 on Ring B still saves 2 units of time, but still leads to a delay of one time unit to initially move D from B5, and then one more time unit after delivery of the escort to A1 to move D on Ring B from B2 or B8 to B1 for transfer to A1, for no savings. Thus, the optimality of RT for all CASE 1 scenarios is proven, and the optimal policy involves concurrent rotation of E and D in Ring A and Ring B, respectively, with transfer of D to the I/O point from position B1 on Ring B. In all cases in CASE 1, RT = $\mathrm{Max}[T_A,T_B] + 1$, or the maximum of Ring A and Ring B movements plus one additional movement for transfer to the I/O point.

<u>CASE 2 PROOF:</u> In some cases, one unit of time can be saved associated with transfer of the escort through the interior of the system (i.e., through Ring C) for a total retrieval time of $\mathrm{Max}[T_A,T_B]$. One such case occurs when E is initially in position A7 or position A11 and D is initially in position B2 or B8. In this case, the optimal policy involves moving the escort to A1 through Ring C instead of rotating it on Ring A. In this case, a savings of 2 units of time can be saved by moving E through Ring C. With D positioned only one position from B1, only 1 non-current movement of Ring B is required, resulting in a real savings of one unit of time.

When E is initially in position A8 or position A10 and D is initially in B1-B3, or B7-B8, it is also possible to save one unit of time by passing E through Ring C. The first movement of Ring A would be to position E at either A7 or A11 for transfer to the system interior. Concurrently, D can be positioned from any of the initial conditions at B1-B3 or B7-B8 to either position B2 or B8, resulting in the same scenario as presented in the previous paragraph. Once again, the savings of two time units for B movement is off-set by only one additional non-concurrent movement of Ring B for a real savings of one unit of time.

When E is initially at position A9 and D is initially at position B4 or B6, similar results are obtained. In this case, moving E through the interior of the system saves 2 moves if travel is on Ring B and 4 moves if travel is on Ring C in comparison to Ring A movement alone. If E moves on Ring B, only one additional non-concurrent movement of Ring B is required to position D for transfer at B1 for a real savings of one unit. If E moves on Ring C, three additional non-current movements of Ring B are required, once again resulting in a savings of one unit of time, for a total time of $\mathrm{Max}[T_A,T_B]$.

CASE 3 PROOF: When E is in initial position A9 and D is initially in B1, B3, or B7, it is possible to save 2 units of time, making the optimal RT = Max[T_A,T_B] - 1. In this case, the optimal RT policy repositions D to either B2 or B8 in only one move. Moving E through Ring C saves four moves. Subsequently, one non-concurrent move of Ring B to reposition D to position B1 yields a total savings of 1 – 4 + 1 = 2 moves for a total of Max[T_A,T_B] – 1 moves.

CASE 4 PROOF: Finally, to complete the proof of THEOREM 2, it is necessary to discuss the final case in which E is initially in position A9 and D is initially in position B2 or B8. Concurrent rotation of Ring A and Ring B would result in a RT of Max[T_A,T_B] + 1. Moving E directly along the diagonal from A9 to A1 would save four moves in comparison to the concurrent rotation strategy and would require only one additional non-concurrent move of Ring B for a total savings of 3 moves. The resulting RT is therefore {Max[T_A,T_B] + 1} – 4 + 1, or simply RT = Max[T_A,T_B] – 2.

When both E and D reside initially on Ring B, the optimal strategy differs slightly depending upon the initial conditions. THEOREMS 2B-2E present these scenarios.

THEOREM 2B: When the demanded item (D) is initially located at position (i,j) on Ring B, and the escort (E) is also initially located on Ring B at position (x,y), and when E is also initially located on the same side of the diagonal (along a line from A1 to A9), and when E is initially 'ahead' of D (i.e., the distance between E and location B1 is smaller than the distance between D and B1) as depicted in Figure 8, then the optimal retrieval time (RT) is:

$$RT = i + j - 2$$

and the optimal strategy is achieved in 4 steps:

1) Rotate E to B1 on Ring B.
2) Transfer E to the I/O point at position A1.
3) Continue rotating D to B1 on Ring B.
4) Transfer D to the I/O point at position A1.

PROOF: When the optimal policy is followed, [(x – 2) + (y – 2)] time units are required to rotate E to B1 and one additional time unit is required to move E to position A1. By scenario definition, we know that [(x – 2) + (y – 2)] < [(i – 2) + (j – 2)] and that [(i – 2) + (j – 2)] – [(x – 2) + (y – 2)] additional moves are required to transfer D to B1. One additional time unit is required to move D to the I/O point at A1. Summing, we obtain:

$$RT = [(x-2) + (y-2)] + 1 + \{[(i-2) + (j-2)] - [(x-2) + (y-2)]\} + 1$$

$$RT = (i-2) + (j-2) + 2$$

RT = i + j – 2

Due to the larger circumference of Ring A, early movement of E to Ring A can only add to the time required to position E at A. Similarly, early movement of D to Ring A can only add to the retrieval time. Thus, the only possible way to improve is to move E and or D through Ring C, but this is also easily proven to be suboptimal. By definition of the

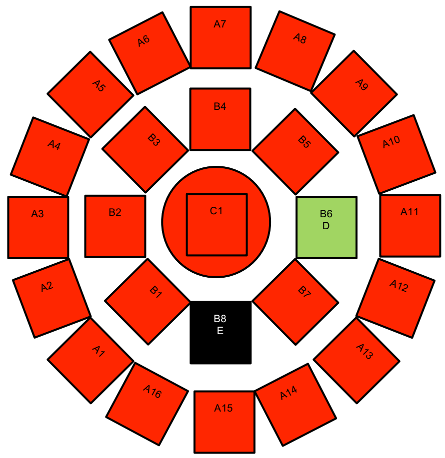

Figure 8. Retrieval from Ring B with Escort Also in Ring B ‘Ahead’ of Demand.

scenario, E can never be initially located farther from the I/O point than either B4 or B6, and the maximum potential savings associated with moving E via Ring C would therefore be 1 time unit, but this savings is defeated by requiring Ring B to sit idle for 2 time units for the transfer instead of concurrently rotating both E and D toward B1 for transfer to the I/O point. As we already know, moving D through Ring C is even worse, and requires additional escort manipulation to continue the movement of D. With no additional options for cost savings, the optimal policy is as stated above, with a total retrieval time of i + j – 2.

<u>THEOREM 2C:</u> When the demanded item (D) is initially located at position (i,j) on Ring B, and the escort (E) is also initially located on Ring B at position (x,y), and when E is also initially located on the same side of the diagonal (along a line from A1 to A9), and when E is initially ‘behind’ D (i.e., the distance between E and location B1 is larger than the distance between D and B1) as depicted in Figure 9, then the optimal retrieval time (RT) depends upon the initial positions of D and E. Three cases exist:

<u>CASE 1:</u>	RT = x + y – 1	when	E in initial positions B2-B3, or B7-B8,
		or	E in initial position B4 or B6 and D not in initial position B2 or B8,

or E in initial position B5 and D in initial position B4 or B6.

CASE 2: RT = x + y – 2 when E in initial position B4 or B6 and D in initial position B2 or B8,

or E in initial position B5 and D in initial position B3 or B7.

CASE 3: RT = x + y – 3 when E in initial position B5 and D in initial position B1, B2 or B8.

In CASE 1, the optimal retrieval policy is achieved in 4 steps:

1) Move E to B2 or B8, rotating Ring B in the direction that moves E toward the initial position of D.
2) Transfer E to Ring A at A3 or A15.
3) Move D to position B1 on Ring B while concurrently rotating Ring A to position E at A1.
4) Transfer D to the I/O point at position A1.

In CASE 2, and CASE 3, the optimal retrieval policy is achieved in 4 steps:

1) If D is initially in location B1, rotate it on Ring B one position to either B2 or B8. This will also rotate E to position B4 or B6.
2) Move E to A1 via the shortest path through Ring C.
3) Rotate D to B1 on Ring B.
4) Transfer D to the I/O point at position A1.

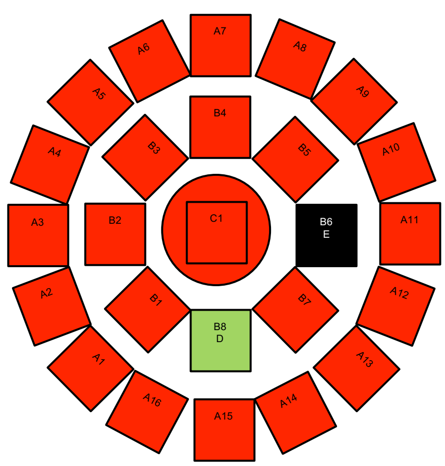

Figure 9. Retrieval from Ring B with Escort Also in Ring B 'Behind' Demand.

CASE 1 PROOF: In situations in which $x \leq y$, moving E to Ring A at position A15 requires $(x - 2) + (y - 3) + 1$ or $x + y - 4$ moves. By scenario definition, absolute values are not required. It takes 2 additional time units to move E from A15 to A1 on Ring A, potentially while D is concurrently rotated to B1 on Ring B. It takes one additional time unit to move D to A1. Thus, under the optimal CASE 1 policy, $RT = (x + y + 4) + 2 + 1 = x + y - 1$. Similar results are obtained when $x > y$.

When E is initially in positions B2-B3 or B7-B8, E cannot reach A1 faster by moving through Ring C, and previous scenarios reveal that there can be no savings associated with moving D through Ring C. Thus, the only way to gain savings in comparison to rotation via Ring B alone is to make an early transfer of E or D to Ring A. Because D is ahead of E, retrieval via rotation of Ring B alone requires over-rotation of D past B1 to position E at B1. If E is initially in B2 or B8, D must be at B1 by definition, and the optimal retrieval time can be achieved by either the optimal policy as stated for CASE 1, or via over-rotation and return of D to B1 for transfer to A1. If E is initially in B3 or B7 and D is initially in B2 or B8, the situation is the same and the optimal policy for CASE 1 performs identically to Ring B rotation alone. If, however, E is initially in B3 or B7 and D is initially in B1, the CASE 1 optimal strategy results in a better RT, because of the excessive over-rotation of D on Ring B. When E transfers at A3 or A15, D can be repositioned to B1 on Ring B concurrent to Ring A movements of E.

When E is in initial positions B4 or B6 and D is not in initial positions B2 or B8, the CASE 1 policy is also optimal. If D is in initial positions B3 or B7, savings are possible when E moves via Ring C, but these savings are exactly off-set by Ring B delays. Similarly, savings in movements associated with rotating E all the way to B1 on Ring B are exactly off-set by over-rotation delays, and no advantage can be gained by departing from the CASE 1 optimal policy. If D is initially in B1, the larger number of slots between (i,j) and (x,y) make over-rotation of D on Ring B unattractive. In fact, transfer of E at B1 would require two additional non-concurrent repositioning moves for D for a loss of two time units in comparison to the optimal policy. To accrue savings by moving E through Ring C, it is first necessary to remove D from position B1. In this case, savings for E movements can only be accrued by first moving Ring B in the 'wrong' direction, which would move E to B5 for a diagonal transfer along B5-C1-B1-A1. Even so, the savings associated with E movements are overcame by Ring B delays to rotate D away from, and then back to B1. The optimal CASE 1 policy is therefore as good as any other retrieval options for this situation.

Finally, when E is in initial position B5 and D is in initial position B4 or B6, the CASE 1 optimal strategy is at least as good as any other strategy. Moving E through C to A1 saves 3 time units in moving E to A1, but delays Ring B movement by 3 time units. Similarly, moving E all the way to B1 on Ring B saves one time unit in moving E to A1, but over-rotates D by one time unit for no gain. Thus, the CASE 1 optimal strategy is at least as good as any other.

CASE 2 PROOF: In CASE 2 scenarios, movement of E through Ring C is optimal. When E is initially in position B4 or B6 and D is in position B2 or B8, respectively, such a situation exists. In this situation, we have three options; (1) use the CASE 1 strategy, (2) over-rotate D past B1 on Ring B and return it to B1 on Ring B, or (3), move E through Ring C. Using the CASE 1 methodology would result in RT = x + y – 1. Over-rotating D past B1 on Ring B saves 1 time unit in moving E to A1, but requires 2 non-concurrent movements to get D back to B1 to result in RT = x + y. Finally, moving E through Ring C saves 2 moves getting E to A1, and due to the initial position of D requires only one non-concurrent movement of Ring B to position D at B1 for transfer to the I/O point for a real savings of one unit and an RT = x + y – 2.

Similarly, when E is initially in position B5 and D is initially in position B3 or B7, movement of E to B1 on Ring B saves 1 unit of transfer time for E, but adds two units of time to return D to B1 following over-rotation for a net loss of one unit of time in comparison to the CASE 1 optimal strategy. Moving E through Ring C saves 3 units of time for E movement, but subsequently requires 2 non-concurrent Ring B movements for a savings of one unit and an RT = x + y – 2.

<u>CASE 3 PROOF:</u> In CASE 3 scenarios, movement of E through Ring C is also optimal, but when E is initially in B5, a real savings of 2 units is possible in some cases in comparison to CASE 1 scenarios. When E is initially at B5 and D is initially in B2 or B8, the RT situation is similar to the CASE 2 scenario with initial positions of E at B4 or B6 and D at B2 or B8, except that D is one position farther away from the I/O. Thus, over-rotation of D on Ring B would be 2 units of time worse than the CASE 1 optimal strategy instead of one, and the savings associated with movement through Ring C would be 2 units of time better instead of only one, for an RT = x + y – 3.

When E is initially at B5 and D is initially at B1, similar results are obtained except D must be moved one position away, to either B2 or B8, to facilitate movement of E through Ring C to the I/O point. In this case, even the CASE 1 optimal strategy would require one additional non-concurrent movement of Ring B to compensate for over-rotation of D. Rotation of E all the way to B1 saves one unit of time but adds four non-concurrent Ring B movements for a total loss of two units of time. Moving E through Ring C saves only 2 time units (because of one additional Ring B movement to move D from B1), but results in over-rotation of only one unit as in the CASE 1 strategy. Thus, the 2 units of savings are true savings and the RT = x + y – 3.

<u>THEOREM 2D:</u> When the demanded item (D) is initially located at position (i,j) on Ring B, and the escort (E) is also initially located on Ring B at position (x,y), and when E is also initially located on the opposite side of the diagonal (along a line from A1 to A9), and when E is initially 'ahead of' D (i.e., the distance between E and location B1 is smaller than the distance between D and B1) as depicted in Figure 10, then the optimal retrieval time (RT) depends upon the initial positions of D and E. Two cases exist:

<u>CASE 1:</u> RT = i + j – 2 when E in initial position B2 or B8.

<u>CASE 2:</u> RT = x + y when E in initial position B3 (or B7).

In both CASE 1 and CASE 2, the optimal retrieval policy is achieved in 3 steps:

1) Transfer E to Ring A on the first move (by scenario definition, E always initially resides on a transfer point).
2) Rotate D to B1 on Ring B while concurrently rotating E to A1 on Ring A.
3) Transfer D to the I/O point at position A1.

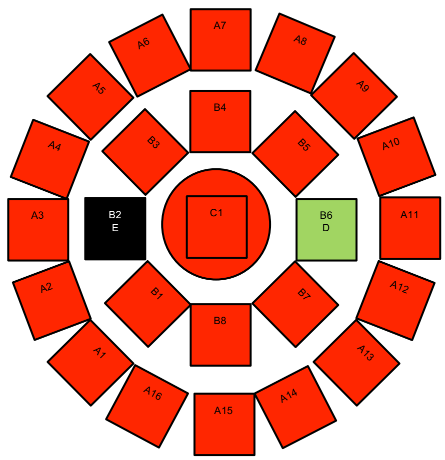

Figure 10. Retrieval from Ring B when E is Closer to I/O but on Other Side of System.

<u>CASE 1 PROOF:</u> In THEOREM 2D, $[(x - 2) + (y - 2)] < [(i - 2) + (j - 2)]$, or equivalently, $(x + y) < (i + j)$. When E is initially at B2 and D at B6 (or equivalently when E is at B8 and D at B4), it takes $(x - 1) + (y - 1) = (x + y - 2)$ moves to get E to A1 and it takes $(i - 2) + (j - 2) = (i + j - 4)$ moves to get D to B1 for transfer to A1 plus the additional initial delay of one unit for the initial E transfer to Ring A for a total of $(i + j - 3)$ moves. In this case, $RT = \max[(x + y - 2),(i + j - 3)] + 1$ moves. By scenario definition, $(x + y - 2) < (i + j - 3)$, so the retrieval time is $(i + j - 3) + 1$ additional move for transferring D to A1, or $RT = i + j - 2$ when utilizing the THEOREM 2D stated optimal policy. We know from earlier discussion that it is never optimal to move D through Ring C. Rotating E to B1 for transfer delays the movement of D toward the I/O point by one unit and adds one additional non-concurrent Ring B movement for a loss of 2 units compared to the stated optimal strategy. There are no other options for improvement. When E is initially at B2 and D is at B7 (or E is at B8 and D is at B3), the proof is identical, but the RT values are one unit better.

<u>CASE 2 PROOF:</u> The only other scenario allowable within THEOREM 2D occurs then E is at B3 (or B7) and D is at B6 (or B4). In comparison to the situation in the previous paragraph, the transfer of E at A5 adds one unit of time and transfer of E to A1 now takes $(x + y - 1)$ moves. The transfer of D to B1 remains at $(i + j - 3)$ moves. By scenario definition we know that $(x + y - 1) > (i + j - 3)$, so the retrieval time is $x + y - 1$ plus one

additional unit of time for D transfer at A1, or simply RT = x + y. Again, movement of D through C is never optimal and any savings associated with transfer of E at B1 (or at any point between the original position of B and B1) are defeated by either over-rotation of D on Ring B or by lack of savings due to concurrent movement of Ring A and Ring B. The optimal THEOREM 2D policy (transfer of E at B3 (or B7)) results in an RT value of (x + y) while transfer at either B2 (B8) or B1 results in RT values of (x + y + 1).

THEOREM 2E: When the demanded item (D) is initially located at position (i,j) on Ring B, and the escort (E) is also initially located on Ring B at position (x,y), and when E is also initially located on the opposite side of the diagonal (along a line from A1 to A9), and when E is initially 'behind' D (i.e., the distance between E and location B1 is larger than the distance between D and B1) as depicted in Figure 11, then the optimal retrieval time (RT) depends upon the initial positions of D and E. Three cases exist:

CASE 1: RT = x + y – 1 when E in initial position B2 or B8.

CASE 2: RT = i + j when E in initial position B3 or B7.

CASE 3: RT = I + j when E in initial location B4 or B6.

In both CASE 1 and CASE 2, the optimal retrieval policy is achieved in 3 steps:

1) If E not initially in position B2 (or B8), rotate D to B2 (or B8) on Ring B.
2) Move E to Ring A at position A3 (or A15).
3) Rotate D to A1 on Ring A while concurrently rotating D to B1 on Ring B.
4) Transfer D to the I/O point at position A1.

In CASE 3, the optimal retrieval policy is also achieved in 3 steps:

1) Move E to A1 through Ring C.
2) Rotate Ring B to position D at B1.
3) Transfer D to the I/O point at Al.

CASE 1 PROOF: By scenario definition, we know that [(x – 2) + (y – 2)] ≥ [(i – 2) + (j – 2)], or equivalently (x + y) ≥ (i + j). Therefore, whenever E is initially in position B2 (or B8), D must be at B8 (or B2). Following the stated optimal policy for CASE 1, it would take one unit of time to move E to A3, two units of time to concurrently rotate E to A1 and D to B1, and one unit of time to transfer D to the I/O point. Thus, moving E to A1 takes x + y – 2 movements and D is waiting at B1 when E arrives. One additional time unit for transfer of D to A1 results in an RT value of x + y – 1. Moving E on Ring B to B1 saves one unit of time for E transfer, but creates 2 subsequent moves associated with moving D in the wrong direction. Clearly, no savings are available through use of Ring C, and CASE 1 is proven.

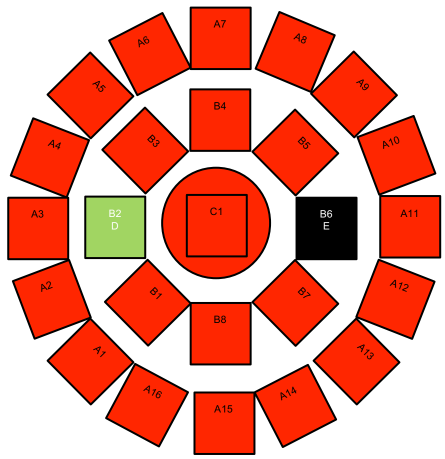

Figure 11. Retrieval from Ring B when D is Closer to I/O but on Other Side of System.

<u>CASE 2 PROOF:</u> When E is initially in position B3 (or B7), then D must reside in either B7 (or B3) or B8 (or B2). When E is initially in B3, Ring C offers no advantage in CASE 2 scenarios. Thus, E can be transferred to Ring A at B3, B2 or B1 (or B7, B8 or B1). If we use the optimal CASE 2 strategy, D movement is the pacer. It takes (i + j – 4) movements to move D to B1, but four time units of additional delays are added to initially rotate D in the wrong direction to move E to B2 (or B8), move E to Ring A at B2 (or B8), subsequently move D back to its initial position, and to move D from B1 to A1. Thus the total RT = i + j. Moving E to Ring A early at position B3 prevents D from rotating one position in the wrong direction, but makes E movement the retrieval time pacer and results in no overall gain in retrieval time. Moving E all the way to B1 for transfer to Ring A causes D to rotate two positions in the wrong direction, making the retrieval time increase by 2 units of time. Three are no additional options for savings.

<u>CASE 3 PROOF:</u> When E is initially in position B4 (or B6), then D can initially reside in B6, B7 or B8 (or B2, B3 or B4), and the use of Ring C for E movement becomes optimal. By definition of the scenario, D is not initially in a blocking position for E movement and E can reach the I/O point via Ring C in 3 units of time (B4-C1-B1-A1 or B6-C1-B1-A1). Subsequent rotation of D to B1 on Ring B takes [(i – 2) + (j – 2)] = i + j – 4 moves. One additional move to transfer D to A1 results in RT = 3 + (i + j – 4) + 1 or RT = i + j.

If the optimal CASE 3 policy of moving E through Ring C is not used, and if D is initially at B8 (or B2), it is possible to transfer E to Ring A at A7, A5, A3, or A1 (or the equivalent on the other side of the system). Doing so would add 3, 2, 2 or 4 additional moves, respectively. A7 transfer adds 4 units of transport time for E, but prevents 1 non-concurrent movement of Ring B for a cost of 3 units of time in comparison to the optimal policy. A5 transfer adds 3 units of transport time for E, but also prevents 1 non-current movement of Ring B for a cost of 2 units of time. A3 transfer adds 2 units of transport time for E, and rotation of D in the wrong direction removes the benefits of subsequent concurrent Ring Movements and the total cost is 2 units of time. Finally, rotating E all the way to A1 for transfer moves D all the way to B5, and it must subsequently be rotated all the way back to B1 for a cost of 4 units in comparison to the optimal policy. There are no other options for savings.

When E is initially in B4 (or B6) and D is in B7 (or B3), similar results are obtained, but the costs associated with early E transfer at positions A7, A5, A3, and A1 are reduced to 2, 1, 2 and 2, respectively. A7 transport adds four units to E transport, but prevents 2 non-concurrent Ring B movements for a net cost of two units of time. A5 transfer adds 3 units to E transport, and prevents 2 non-current Ring B movements for a net cost of one unit of time. A3 transfer adds 2 units of transport time to E, but does not allow for savings in non-concurrent Ring B movements due to over-rotation of D, for a net cost of 2 units. Finally, rotating E all the way to B1 adds 1 unit of time for E transport and repositions D all the way to B4 (or B6) and three non-concurrent Ring B movements are required instead of two, for a net cost of 2 units of time. There are no additional savings options for RT.

When E is initially in B4 (or B6) and D is in B6 (or B4), the optimal policy is still to move E through Ring C, but the time savings associated with this policy instead of E transfer directly to Ring A at positions A7, A5, A3 and A1 are reduced to 1, 0, 0, and 0 units, respectively. When moving E through Ring C, 3 non-concurrent Ring B movements remain after E is positioned at A1. Using A7 transfer, 4 units of time are added to E transport and all 3 non-concurrent moves are eliminated for a cost of 1 unit of time. Using A5 transfer, 3 units of time are added to E transport and all 3 non-concurrent moves of Ring B are eliminated for a net savings of 0 units of time. Using A3 transfer, 2 units of time are added to E transport and one non-concurrent Ring B movement remains, for a net savings of 0 units of time. With A1 transfer, 1 unit of time is added to E transport time, and 2 non-concurrent Ring B movements remain, again for no savings. Thus, the CASE 3 optimal policy of moving E to A1 via Ring C is at least as good as any other policy.

Only one additional possibility remains when the demanded item (D) rests initially in Ring B.

THEOREM 2F: When the demanded item (D) is initially located at position (i,j) on Ring B, and the escort (E) is initially located on Ring C at position (x,y) as depicted in Figure 12, then the optimal retrieval time (RT) is determined by two cases:

CASE 1:	$RT = i + j - 1$	when	E is in any initial position except B1.
CASE 2:	$RT = i + j + 1$	when	E is in initial position B1

The optimal retrieval time for both CASE 1 and CASE 2 can be stated in the following 4 steps:

1) If D is initially in position B1, rotate it to either B2 or B8.
2) Move E to A1 through B1.
3) Rotate D to position B1 on Ring B.
4) Transfer D to the I/O point at position A1.

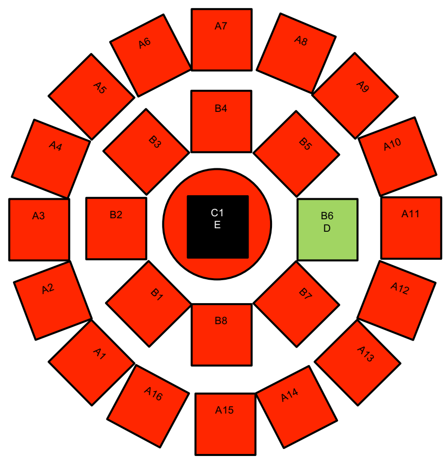

Figure 12. Retrieval when D is initially on Ring B and E is initially on Ring C.

CASE 1 PROOF: We know from Theorem 1 that it is never better to move D through C, so the shortest path for D movement will always be on Ring B for all THEOREM 2F scenarios. Regardless of how E moves to Ring A, it requires holding Ring B idle for 2 units of time. Thus, moving E to any position other than A1 would lengthen RT for any situation in which E movement to A1 on Ring A cannot be handled concurrently to Ring B rotation. Thus, the optimal policy as stated is at least as good as the alternatives in all cases, and better in most.

Moving E from C1 to A1 requires 2 time units. Rotating D to B1 requires an additional $[(i - 2) + (j - 2)]$ or $(i + j - 4)$ time units and transfer of D to A1 requires 1 movement. Thus, the total retrieval time is $2 + (i + j - 4) + 1 = (i + j - 1)$.

CASE 2 PROOF: When D is initially at position B1, it is tempting to move E to either A3 or A15, but this results in 2 additional moves for E. Rotating D to either B2 or B8 also costs two time units to first rotate D out of the way and then back to position B1. Therefore, the optimal policy stated in THEOREM 2F is at least as good as any alternative. In this case, it takes 1 time unit to move D out of the way, 2 time units to move E to A1, one time unit to move D back to B1, and 1 time unit to move D to the I/O point at A1, for a total of 5 moves (or i + j + 1).

Finally, we examine the case in which the demanded item is initially in Ring C and the single escort is located anywhere else in the system.

<u>THEOREM 3:</u> When the demanded item (D) is initially located at position (i,j) on Ring C and E is initially anywhere else in the system at position (x,y) as depicted in Figure 13, the optimal retrieval time (RT) is:

$$RT = \mathrm{Min}[T_{B2}, T_{B8}] + 5$$

<u>WHERE:</u> T_{B2} = The shortest path between E and B2

T_{B8} = The shortest path between E and B8

and the optimal retrieval time is achieved in the following 5 steps:

1) Move E to the closer of B2 or B8.
2) Transfer D to Ring B at either B2 or B8.
3) Move E to A1 through B1.
4) Rotate D to position B1 on Ring B.
5) Transfer D to the I/O point at position A1.

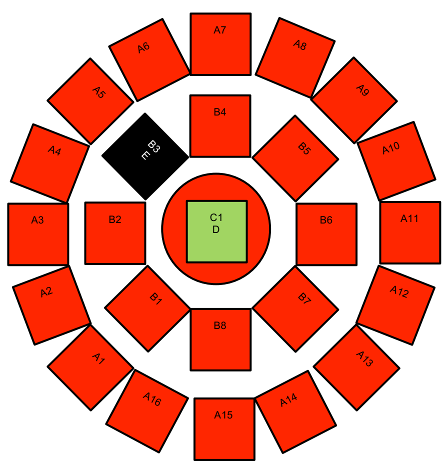

Figure 13. Retrieval when D is initially on Ring C and E is Anywhere in the System.

<u>PROOF:</u> When D is initially in Ring C, the escort must be placed first in Ring B to move D from Ring C, and then must be placed in Ring A in order to move D to the I/O point. We know from Kota et al. [6] that in a rectangular system that it is never better to move D away from the I/O point to obtain earlier access to the escort and the same is true in circular systems. Although all eight Ring B locations are viable transfer points between

Ring C and Ring B, only positions B1, B2 and B8 need be considered. If D were transferred to Ring B in other locations, it would still need to be rotated on Ring B to B2 or B8 on the optimal path. Whether D or E is transported to these locations on Ring B, the exact same time is required, so no benefit can be obtained from early transfer.

Once D is transferred to Ring B, the resulting configuration is identical to the one described in THEOREM 2F. Since we know from THEOREM 2F that movement of D to position B1 would require two additional movements to achieve optimal RT, only positions B2 and B8 should be considered for initial transfer of D to Ring B. Once D is transferred to Ring B, the optimal policy exactly follows from THEOREM 2F.

T_{B2} and T_{B8} values cannot be calculated by merely counting rectilinear distance. When E is initially on Ring A, diagonal transfers can reduce the travel time in some cases, and in all cases the presence of D in Ring C presents an obstacle for free E movement. Thus, it is perhaps best to tabulate T_{B2} and T_{B8} values as in Table 1.

Table 1. Tabulated Values for T_{B2} and T_{B8}.

Location	T_{B2}	T_{B8}		Location	T_{B2}	T_{B8}		Location	T_{B2}	T_{B8}
A1	2	2		A9	4	4		B1	1	1
A2	2	3		A10	5	4		B2	0	2
A3	1	3		A11	5	3		B3	1	3
A4	2	4		A12	5	3		B4	2	4
A5	2	4		A13	4	2		B5	3	3
A6	3	5		A14	4	2		B6	4	2
A7	3	5		A15	3	1		B7	3	1
A8	4	5		A16	3	2		B8	2	0

Finally, RT is calculated by first finding the minimum of T_{B2} and T_{B8} values. Once E is positioned in the closer of B2 or B8, it takes one time unit to move D to Ring B, 2 time units to move E from C1 to A1 through B1, one time unit to rotate D to B1 on Ring B, and 1 unit of time to move D to A1. Thus the total RT = Min[T_{B2},T_{B8}] + 5.

IV. Comparison of System Performance

Now that the analytical results have been presented and proven for the circle-based system, it is possible to compare results with the equivalent rectangular puzzle-based system as presented in the discussion of the experimental design above. RT values for the rectangular puzzle-based system have been calculated using the optimal retrieval time formulas from THEOREM 0. RT values for the circle-based system have been calculated using the optimal retrieval time formulas presented in THEOREM 1, THEOREMS 2A-2F, and THEOREM 3. RT values have been calculated for all combinations of initial positions of D (i,j) and E (x, y). Summary information appears in Table 2.

Table 2 reveals that on the average, the circle-based system results in much better RT performance than the rectangular puzzle-based system of equal size. Across all combinations of (i,j) and (x,y), the rectangular puzzle-based system requires 14.61 units of time per retrieval, and the circle-based system requires only 4.53 units of time, for a savings of 10.08 units of time per retrieval. This represents a retrieval time savings of almost 70%. At worst case, when D is initially in position A9 and E is initially in A1, the circle-based system saves a full 21 units of time in comparison to the rectangular puzzle-based system.

Table 2. Summary Information from Comparison.

Average Overall Puzzle RT	14.61
Average Overall Circle RT	4.53
Maximum Overall Puzzle RT	29
Maximum Overall Circle RT	9
Average Overall Savings for Circle	10.08
Maximum Overall Savings for Circle	21

Table 3 shows average case performance by individual cell. For each initial position of D (for each value of (i,j)), the table shows the average savings provided by the circle-based system in comparison to the rectangular puzzle-based system for each possible value of the initial position of E (x,y). Clearly, the greatest savings are for locations that are far from the I/O point at A1 (i = 1 and j = 1). In these situations, rotation of Ring A provides a much more efficient means of transport than constant manipulation of the escort to enable movement through the interior of a rectangular puzzle-based system. Also, it is very clear that greater savings are accrued when the demanded item is initially along a system edge than when it resides in the interior of the system. We know from Kota et al. [6] that interior movements are far preferable to edge movements in rectangular puzzle-based systems because of the increased escort repositioning moves required to move a demanded item along an edge.

Table 3. Average Case Information by Cell.

	Average Savings by Cell				
	j=1	j=2	J=3	j=4	j=5
i=5	15.50	14.88	14.67	14.88	17.17
i=4	11.29	9.63	9.88	12.13	14.88
i=3	7.50	5.29	5.17	9.88	14.67
i=2	4.13	2.29	5.29	9.63	14.88
i=1	0.00	4.13	7.50	11.29	15.50
	I/O				

Here

Table 4 shows the best case performance (maximum savings) by individual cell. For each initial position of D (for each value of (i,j)), the table shows savings provided by the circle-based system in comparison to the rectangular puzzle-based system for the initial E position (x,y) that provides the greatest savings for that particular (i,j) position. Although the values are higher, the trends remain the same; the greatest savings associated with the use of circle-based systems are for those demand locations that are along system edges or that are a large distance from the I/O point.

Table 4. Best Case Information by Cell.

	Average Savings by Cell				
	j=1	j=2	J=3	j=4	j=5
i=5	19	18	17	18	21
i=4	14	12	13	16	18
i=3	11	7	6	13	17
i=2	8	4	7	12	18
i=1	0	8	11	14	19

I/O
Here

V. Conclusions

As systems continue to evolve in which high density storage is required, researchers will be called upon to design and evaluate system alternatives. Gue and Kim [4] launched a very important alternative by introducing rectangular puzzle-based systems and by providing analytical retrieval time performance results. That seminal paper also inspired other researchers to examine similar systems, and subsequent study has extended that work into larger and more realistic scenarios.

The use of a circular system with movable concentric circles and cross-over points is one such alternative system that seems to hold promise. Certainly, it offers significant improvements to retrieval times, averaging almost 70% improvement in comparison to rectangular puzzle-based systems. It is also quite possible that circle-based systems may be more reliable than puzzle-based alternatives, because the rotation of the various rings may be achieved during periods of partial failure. Each ring could easily be fitted with multiple drive units, and working around a failed unit location would be far easier in a circle-based system than one in which failed units are stationary.

Even so, this improvement comes at some cost. Circular systems are slightly less dense, particularly for smaller systems of the type introduced in this paper, and likely use more energy than rectangular, puzzle-based systems with individual drive units.

Regardless of the configuration selected, additional research is likely warranted. This is particularly true as manufacturers are starting to embrace the dense storage concept and as prototype systems are beginning to emerge. Additional study of 3-dimensional systems is needed in rectangular puzzle-based systems, and initial work needs to be done in 3-dimensional systems for circle-based systems. Also, optimal strategies for more complex, multi-escort scenarios are now being developed for puzzle-based systems, and similar analysis needs to be completed for circle-based systems. Finally, the analysis of systems during use is an important consideration that has not received sufficient attention. How do we replenish such systems while we concurrently continue retrieval operations? These and many other questions and considerations need evaluation before aisle-less storage systems will be ready for widespread implementation.

References

[1] Zaerpour, N., Yu, Y. and de Koster, R. (2014), "Optimizing layout of a live-cube compact storage system", available at: http://www.pom2012.org/fullpapers/add184054_9jD5tm9Pmb.docx.pdf (accessed 17 April 2014).

[2] Archer, A.F. (1999), "A Modern Treatment of the 15 Puzzle", *The American Mathematical Monthly,* Vol. 106 No. 9, pp. 793-799.

[3] Slocum, J., and Sonneveld, D. (2006), *The 15 Puzzle Book*, The Slocum Puzzle Foundation, Beverly Hills, CA.

[4] Gue, K. R., and Kim, B. S. (2007), "Puzzle-based storage systems", *Naval Research Logistics,* Vol. 54 No. 5, pp. 556– 567.

[5] Taylor, G.D., and Gue, K.R. (2008), "The effects of empty storage locations in puzzle-based storage systems", in Fowler, J., and Mason, S. (Eds.), *Proceedings of the 2008 Industrial Engineering Research Conference in Vancouver, BC, Canada*, *May 17-21, 2008*, Institute of Industrial Engineers, Norcross, GA, pp. 519-523.

[6] Kota, V.R., Taylor, G.D., & Gue, K.R. (2010), "Retrieval time performance in puzzle-based storage systems", in Johnson, A., and Miller, J. (Eds.), *Proceedings of the 2010 Industrial Engineering Research Conference in Cancun, Mexico, June 5-9, 2010*, Institute of Industrial Engineers, Norcross, GA, 6p.

[7] Taylor, G.D., & Gue, K.R. (2009), "Design and performance of multi-level puzzle-based storage systems", in Heragu, S., Ashayeri, J., and Yurt, O. (Eds.), *Proceedings of the 2009 International Conference on Value Chain Sustainability in Louisville, KY,*

Oct. 19-21, 2009, Logistics and Distribution Institute (LODI), University of Louisville, Louisville, KY, pp. 180-185.

[8] Furmans, K., Schönung, F. and Gue, K. (2010), "Plug-and-work material handling systems", in Ellis, K.P., Gue, K.R., de Koster, R., Meller, R.D., Montreuil, B., and Ogle, M.K. (Eds.), *Progress in Material Handling Research: 2010,* Material Handling Industry of America, Charlotte, NC, pp. 132-142.

[9] Gue, K.R., Furmans, K., Seibold, Z., and Uludağ, O. (2013), "GridStore: A puzzle-based storage system with decentralized control", forthcoming in *IEEE Transactions on Automation Science and Engineering.*

[10] Alfieri, A., Cantamessa, M., Monchiero, A. (2012), "Heuristics for puzzle-based storage systems driven by a limited set of automated guided vehicles", *Journal of Intelligent Manufacturing*, Vol. 23 No. 5, pp. 1965-1705.

[11] Yu, Y., and de Koster, R. (2009), "Optimal zone boundaries for two-class-based compact three-dimensional automated storage and retrieval systems", *IIE Transactions,* Vol.41 No. 3, pp. 194-208.

[12] DePuy, G. W., and Taylor, G.D. (2007), "Using board puzzles to teach operations research", *INFORMS Transactions on Education*, Vol. 7. No. 2, pp. 160-171.

[13] Gudehus, T., and Kotzab, H. (2012), *Comprehensive Logistics, 2nd Ed.,* Springer-Verlag, Berlin.

[14] Bloss, R. (2013), "Automation pushes the envelope of postal mail handling efficiency", *Assembly Automation,* Vol. 33, No. 1, pp. 3-7.[15]

[15] Bastani, A.S. (2007), "Closed-Loop Conveyor Systems with Breakdown and Repair of Unloading Stations", *IIE Transactions,* Vol. 22, No. 4, pp. 351-360.

[16] Woehr (2013), available at: http://www.woeher.de/en/product/items/multiparker-730.html (accessed 5 August 2013).

[17] Urban Parking Concepts (2013), available at: http://www.urbanparkingconcepts.com/docs/tech.html (accessed 5 August 2013).

[18] Carlo, H.J. and Giraldo, G.E. (2012), "Toward perpetually organized unit-load warehouses", *Computers & Industrial Engineering,* Vol. 63 No. 4, pp. 1003-1012.