

## Reducing Costs of Managing Medication Inventory in Automated Dispensing System in Hospital Units

Gregory Dobson  
Simon School

Vera Tilson  
Simon School

Sandy Sullivan  
URMC Pharmacy

Dave Webster  
URMC Pharmacy

University of Rochester, Rochester, NY 14627, USA

{Greg.Dobson, Vera.Tilson} @simon.rochester.edu

{Sandy\_Sullivan, Dave\_Webster} @urmc.rochester.edu

### Abstract

*Hospital pharmacies utilize automated dispensing systems (ADS) to store and dispense medication in hospital wards. We propose an algorithm to minimize the inventory management costs incurred in holding and refilling the medications stored in an ADS. The algorithm is a linear integer programming formulation to compute how to configure an ADS, and how to allocate medications into capacity-constrained containers. Using a numerical example with real hospital data we demonstrate significant percentage savings relative to the status quo.*

### 1. Introduction

A wide variety of medication is administered to hospital patients. Various regulations govern the management of medication inventory in institutional settings: for example, in the state of New York, potentially addictive medications must be carefully accounted for at each transfer. Automated medication dispensing systems facilitate compliance with such regulations. The key parts of such dispensing systems are a locking storage cabinet with electronic access control and a database for logging the opening and closing of the cabinets, as well as for recording the changes in the inventory levels of the items stored in the drawers. Most systems require user identifiers and passwords, and internal electronic devices track nurses accessing the system, track the patients for whom medications are administered, and provide usage data to the hospital's financial office for the patients' bills [1].

Figure 1 shows a storage drawer of a Pyxis MedStation system. Pyxis MedStations represent 70% of the automated dispensing systems currently deployed in the US hospitals. Each station has multiple drawers, and each drawer is configured with multiple containers as shown in Figure 1. To retrieve medication from an

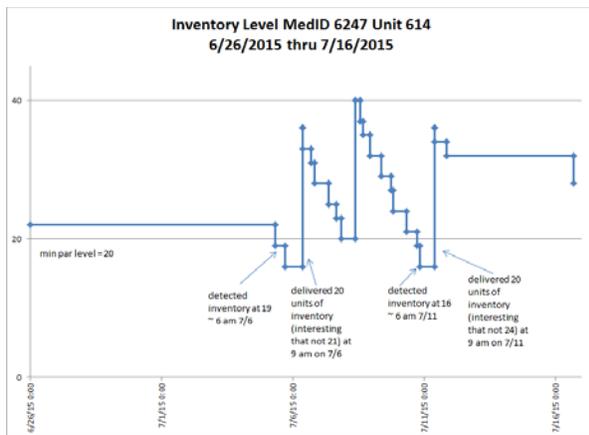
ADS, a healthcare provider selects the patient's name from the computer screen of the system. That action brings up the list of medications prescribed for the patient. Selecting a particular medication from the prescribed list, automatically opens the drawer that contains the container with the medication. The container lid is automatically unlocked and opened. The removal of the medication dose is confirmed by the nurse using the system's computer screen, and this event is then recorded into the ADS database.



**Figure 1: An Example of an Automated Dispensing System: Medication Vial is Removed from a Pyxis MedStation (manufactured by CareFusion)**

Typically, hospitals centralize the management and procurement of pharmaceuticals. Because it is impractical to deliver each medication dose directly from the central pharmacy to the patient in a care unit, medications deliveries are performed in batches. The majority of non-controlled medications are either picked or compounded in the central hospital pharmacy and then delivered to the hospital units with batch deliveries in containers designated for a particular patient. The controlled medications are delivered to the ADS. In addition to controlled medications, some non-controlled medications may also be stored in ADS when there is room available. These include medications that should be immediately available when needed (e.g. in response to an allergic reaction), the frequently ordered medications (e.g. Tylenol), and medications that are

frequently prescribed on per-needed basis, rather than on a regular schedule. The pharmacy department configures which medication is placed in which ADS. The decision is made at the level of the container within a drawer. The pharmacy department sets the minimum and maximum par level for each medication. Typical inventory management of this distributed inventory is periodic review: the inventory database of the ADS is queried periodically generating a report of inventory levels for medications that are at the time of the report below the minimum par level. For these medications, doses are pulled from the central hospital pharmacy storage to bring the ADS inventory up to the maximum par level. The pulled doses are delivered as a batch to the ADS in the hospital wards.



**Figure 2: Inventory Level over Time for a Medication Dispensed via a Pyxis Machine**

Figure 2 illustrates inventory levels over a period of twenty days of a sample medication stored in a Pyxis unit of our hospital. The data for the plot was obtained from the system's database. Note, that the refill amount is not always consistent with the formula

*maximum par level - inventory level at the time the report is run*

We do not know the exact reason for this particular discrepancy, but pharmacy technicians refilling the machines do have some discretion in the operation, and preference to refill with complete packs of medication is one plausible explanation in this case.

A question of interest to pharmacy management is how to utilize the dispensing systems efficiently. One way to measure efficiency is as the cost of keeping and managing the inventory of medications in the Pyxis machines. Assuming the hospital continues to follow the periodic review policy described earlier, the minimum and maximum par levels should be set to trade off the holding, restocking and shortage costs.

The problem of determining the minimum and maximum par levels for a Pyxis machine was previously considered by Kelle, Woosley and Schneider [2]. They

proposed three mathematical programming formulations for setting the par levels under different optimization objectives and constraints. One of the formulations involves minimization of expected ordering costs subject to availability constraint for each medication stored in the machine. The availability constraint is formulated as the limit on the probability of a stockout. The second formulation has a similar availability constraint but the objective is to minimize the expected sum of ordering and holding costs. The third formulation considers the expected sum of ordering, holding and shortage costs and eliminates the availability constraint. All three formulations impose the total volume constraint on the solution, treating the storage space available to each medication as continuous.

The major contribution of our paper is the recognition of the discrete nature of the storage space constraint. Our algorithm provides pharmacies with a complete workable solution that not only recommends the minimum and maximum par level for each medication, but also the size of the storage container that should be used for housing the medication, and assures that the storage containers can fit the cabinet configuration. The second contribution of our paper is the inclusion of inventory holding costs related to the holding of controlled substances. Finally, our numeric example based on the data and operations of a large US hospital quantifies the size of cost savings that can be achieved with an optimal allocation.

## 2. Literature Review

Literature on joint replenishment under stochastic demand (SJRP) is applicable in our setting. Johansen and Melchior [3] noted that in the academic SJRP literature, the continuous review ( $s,c,S$ ) can-order policy [4] and the periodic replenishment policy [5] received considerable attention. Other policies have been proposed, for example Feng, Wu, Muthuraman and Deshpande [6] introduced a continuous-review ( $s,c,d,S$ ) policy for managing multi-product inventory with correlated demand. While in the ( $s,c,S$ ) an order up to  $S_i$  is made for all items  $i$  with inventory level below  $c_i$ , whenever the inventory level of at least one item falls below  $s_i$ , the order level in the ( $s,c,d,S$ ) policy differs.

The joint replenishment literature was recently reviewed by Khouja and Goyal [7], who argued for shifting research focus from the search for the optimal solution to the classic JRP problem to developing applicable models for the real life inventory problems, for examples problems that take into consideration storage and transport constraints. Including these constraints can significantly affect recommendations. For example, the ( $s,S$ )-replenishment policy has been proven to minimize costs in a variety of single-item

inventory settings [8-10]. However, Chen and Lambrecht [11] demonstrated that the  $(s,S)$  policy may not be optimal when inventory storage capacity is constrained.

Given the real-world nature of our problem, we treat the  $(s,S)$  periodic replenishment as given. Our algorithm consists of calculating a cost-minimizing  $(s,S)$  level when total storage capacity is not constrained, and then calculating the cost-minimizing minimum par level when available capacity is below the optimal  $S$  level. Kelle, Woosley and Schneider [2] provide extensive discussions related to the computational challenges of finding the cost minimizing par levels. Given the extensive literature on  $(s,S)$  policies and their variants, and the algorithms to solve them, our focus here is different, namely combining an  $(s,S)$  method with the constraint on storage. Analytic algorithms or faster approximations are left for future work.

### 3. Mathematical Programming Formulation

As described in the introduction, a standard Pyxis machine holds multiple drawers. A standard configuration is six full-height drawers, but two half-height drawers can be used in place of any single full-height drawers. The full-height and the half-height drawers are further configured as a set of rows of containers. The full-height drawers can hold five rows of containers, where each row has a width of five units. So, a row in a full-height drawer can hold a container of width 5, or a container of width 2 and a container of width 3, or three containers of width 1 and one container of width 2, etc. The half-height drawers can also hold five rows of containers, but the total width of each row is six units. The half-height containers are manufactured in widths of 1, 2, and 3 units.

Let  $M$  denote the set of medications to be dispensed from a Pyxis machine, and let  $m \in M$  denote a particular medication. Let  $J$  denote the set of feasible containers. In the hospital where we conducted this project the set of utilized containers is as follows:  $\{(1x1), (1x2), (1x3), (2x1), (2x2), (2x3), (2x5)\}$ , where  $(1x3)$  is a half-height container of width 3,  $(2x5)$  is a full-height container of width 5, etc. A medication can be placed in multiple containers, and the dispensing system configured to open the containers in order: after the first container is emptied, the second one is open whenever the needed medication is requested. So we let  $K$  denote a set of feasible container combinations, for example a feasible combination may consist of two  $(1x3)$  containers, or of a  $(1x2)$  and a  $(1x3)$  container, etc. Let  $k \in K$  denote a particular container combination, and let  $r_j^k$  denote the number of containers of type  $j$  included in container combination

$k$ . Not all combinations are feasible for all medications, for example containers of width of one or two are too small to hold syringes, so we let  $K_m \subseteq K$  be set of container combinations that is feasible for medication  $m$ . Finally, let  $v_m^k$  be the cost associated with stocking medication  $m$  in container combination  $k$ . The problem of finding an allocation of the height of Pyxis into drawers  $(d_1, d_2)$ , an assignment of containers to fit in those drawers  $\{b_j\}$ , and an assignment of medications to those Pyxis containers that minimizes costs can be formulated as an integer linear programming problem:

$$\min_{\{z_m^k\}, \{b_j\}, d_1, d_2, e_{(2x2)}} \sum_{m \in M} \sum_{k \in K_m} v_m^k z_m^k \quad (1)$$

subject to constraints:

$$z_m^k \in \{0,1\} \forall k \in K, m \in M \quad (2)$$

$$\sum_{k \in K_m} z_m^k = 1 \forall m \in M \quad (3)$$

$$\sum_{m \in M} \sum_{k \in K_m} r_j^k z_m^k = b_j \forall j \quad (4)$$

$$d_1 + 2d_2 \leq 2d \quad (5)$$

$$b_{(1x1)} + 2b_{(1x2)} + 3b_{(1x3)} \leq 30d_1 \quad (6)$$

$$b_{(2x1)} + 2b_{(2x2)} + 3b_{(2x3)} + 5b_{(2x5)} \leq 25d_2 \quad (7)$$

$$e_{(2x2)} \geq b_{(2x2)} - b_{(2x3)} \quad (8)$$

$$e_{(2x2)} + 2b_{(2x3)} + 2b_{(2x5)} \leq 10d_2 \quad (9)$$

$$d_1, d_2, b_{(1x1)}, b_{(1x2)}, b_{(1x3)}, b_{(2x1)}, b_{(2x2)}, b_{(2x3)}, b_{(2x5)}, e_{(2x2)} \geq 0, \text{ integer} \quad (10)$$

Decision variable  $z_m^k$  is equal to 1, when container combination  $k$  is utilized for medication  $m$ . Constraint (3) assures that only one container combination is used for a medication. In constraint (5) decision variables  $d_1$  and  $d_2$  are used for counting the number of half-height, and full-height drawers, while parameter  $d$  specifies the total number of full-height drawer slots available. Constraint (4) defines decision variables  $\{b_j\}$  that are used for counting the number of containers of each type used in the solution. Constraint (6) guarantees that the half-height containers fit into the space available in the half-height drawers. Similarly constraint (7) checks that the total space used by the full-height containers is not larger than the space available in the full-height drawers.  $e_{(2x2)}$  defined with constraint (8) is the number of full-height, width 2 cubies in a solution in excess of the number of full-height width 3 cubies. Because the 5-unit width of the full-height drawers is not the divisible by 2 and 3, constraint (9) is needed to assure a feasible solution. Problem formulation notation is listed in Table 1. The proof of the formulation is in Appendix A.

**Table 1: Notation for the Container Assignment Formulation**

Variable	Description
$m \in M$	The index on medications
$J$	The set of feasible container dimensions. The first dimension is the height of the container, the second is the width of the container
$J = \left\{ \begin{array}{l} (1x1), (1x2), \\ (1x3), \\ (2x1), (2x2), \\ (2x3), (2x5) \end{array} \right\}$	
$k \in K$	The index on allowed container combinations
$r_j^k$	The number of containers of type $j$ included in combination $k$ .
$K_m \subseteq K$	The subset of container combinations that is feasible for medication $m$
$v_m^k$	The inventory cost associated with using container combination $k$ for medication $m$
$d$	The total number of available for full-height drawers in a Pyxis unit
$d_1$	Decision variable: Number of half-height drawers
$d_2$	Decision variable: Number of full-height drawers
$b_j$	Decision variables: Total quantity of containers of dimensions $j \in J$
$z_m^k \in \{0,1\}$	Decision variable: Indicator variable for whether medication $m$ is stored in container combination $k$

### 3.1. Computation of the Cost Coefficients

The cost,  $v_m^k$ , of placing medication  $m$  into container combination  $k$  is the cost of holding and managing the medication inventory in that container combination over some period  $T$ . This cost, in turn, is determined by the minimum and maximum par levels assigned to this combination. Let  $\bar{\Sigma}_m^k$  be the maximum number of doses of medication  $m$  that can be held in container combination  $k$ . Let  $h_m^k(\sigma, \Sigma)$  denote the expected holding cost associated with the decision to set the minimum par level at  $\sigma$  and the maximum par level at  $\Sigma$  for medication  $m$  in container combination  $k$ . Let  $s_m^k(\sigma, \Sigma)$  be the expected annual cost of refilling the containers when there is a stock-out, and  $f_m^k(\sigma, \Sigma)$  be the expected annual cost of periodic refills, not related to stock-outs. The expected minimum cost is found as

$$v_m^k = \min_{\sigma, \Sigma} h_m^k(\sigma, \Sigma) + f_m^k(\sigma, \Sigma) + s_m^k(\sigma, \Sigma) \quad (11)$$

subject to

$$0 \leq \sigma < \Sigma \leq \bar{\Sigma}_m^k. \quad (12)$$

Kelle, Woosley and Schneider [2] provide extensive discussions related to the computational challenges of finding an optimal solution to (11). For our work, we used simulation optimization to find the values of  $v_m^k$  and of the corresponding minimum and maximum par levels

$$(\sigma_m^k, \Sigma_m^k) \in \operatorname{argmin}_{\sigma, \Sigma} \{h_m^k(\sigma, \Sigma) + f_m^k(\sigma, \Sigma) + s_m^k(\sigma, \Sigma)\}. \quad (13)$$

We sampled single-period demand values from historical data. Figure 2 is an inventory level diagram for one of the medications dispensed via a Pyxis machine in a hospital ward. The minimum par level is set at 20 doses, the maximum par level is set at 40 doses. The diagram shows that it is not uncommon to observe prolonged periods of time when no doses are dispensed, so we have empirical evidence that daily demand is not normally distributed. Figure 2 illustrates a policy where the central hospital pharmacy checks twice a day (about 6 am and 2 pm) whether there are medications at or below their minimum par levels. Shortly after the below par level reports are generated enough doses are picked to bring the level up to the maximum par level. These additional doses are delivered to the units some time later. If there were additional dispenses between the time the level was checked and the doses were delivered, at the time when the refills are delivered, the inventory level may not reach the maximum par level.

To speed up the search for optimal par levels we limited the minimum par level search to the interval  $[0, 2D_m]$ , where  $D_m$  is the historical maximum one-period demand for medication  $m$ . We limited the maximum par levels to the interval  $[\sigma + 1, \min\{\sigma + EOQ(m), \bar{\Sigma}_m^k\}]$ . For every medication, we first computed the values of  $\sigma_m^k$  and  $\Sigma_m^k$  for container combinations with the largest capacity  $\bar{\Sigma}_m^k$  ( $\bar{\Sigma}_m^k \geq \bar{\Sigma}_m^k$  for all  $k$ ). We then set  $v_m^k = v_m^k$ ,  $\sigma_m^k = \sigma_m^k$  and  $\Sigma_m^k = \Sigma_m^k$  for all container combinations  $k$  that had capacity greater than  $\bar{\Sigma}_m^k$ . For container combinations with smaller volumes, i.e., such that  $\bar{\Sigma}_m^k < \bar{\Sigma}_m^k$ , we set maximum par level equal to capacity ( $\Sigma_m^k = \bar{\Sigma}_m^k$ ) and used simulation optimization to find the corresponding  $\sigma_m^k$  and  $v_m^k$ .

## 4. Computational Experiments

### 4.1. Quantification of Optimization Parameters

We classified medications into categories related to the packaging of unit doses and measured the number of doses that can fit into each container type. A sample of our data is shown in Table 2. Note, that full-height containers of width 1, (2x1), are not used in our hospital. As can be seen in the case of gel capsules, the geometric shape of dose packaging means that doubling or tripling the total container volume by increasing one of the container dimensions does not necessarily result in proportional increase in container capacity for that type of dose.

We limited the number of container combinations we considered, so that no combination had more than two containers. We further limited these combinations so that, if a combination consists of two different containers, these containers are close in size. These assumptions were in line with preferences of the pharmacy management. The seventeen container combinations are listed in Table 3. While these assumptions were appropriate for the computational experiment in our setting, additional or different combinations can be enumerated to define a set of feasible combinations  $K$  in a different hospital.

**Table 2: Pyxis Container Capacity by Dose Type**

Dose Type Description	Qty	Container Size					
		(1x1)	(1x2)	(1x3)	(2x2)	(2x3)	(2x5)
blister pack pill	69	30	100	150	240	360	600
gel capsule	11	12	20	25	40	60	100
medium vial	24	3	6	9	14	21	36
ampule	1	None	10	20	32	48	80
small cup 0-10ml	5	None	None	None	14	23	38
large cup round 15-30ml	4	None	None	None	7	12	20
syringe	8	None	None	14	None	35	58

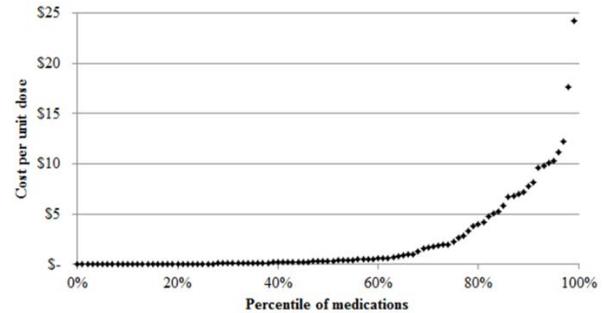
**Table 3: Pyxis Container Combinations Considered in Numerical Experiments: Number of Containers is Followed by the Dimensions of the Container.**

	Description		Description		Description
1	1:(1x1)	7	1:(1x2),1:(1x3)	13	1:(2x2),1:(2x3)
2	2:(1x1)	8	1:(2x2)	14	1:(2x5)
3	1:(1x2)	9	2:(1x3)	15	2:(2x3)
4	1:(1x1),1:(1x2)	10	1:(2x3)	16	1:(2x3),1:(2x5)
5	1:(1x3)	11	2:(2x2)	17	2:(2x5)
6	2:(1x2)	12	1:(1x3),1:(2x3)		

Pharmacy technicians refill machines, and from our observation of the technicians' work we noted that a technician spent about two minutes retrieving the medication from the central pharmacy, and about one minute storing a medication into a Pyxis machine. Assuming an hourly wage (including benefits) of \$20,

we estimated the cost of a non-stockout-related refill, at \$1. We assumed that a stockout-related refill takes ten additional minutes of the pharmacy technician's time, so that the cost associated with a stockout is \$4.33.

Next, we estimated the costs of holding the medications in ADS. We obtained current unit cost data on medications, and assumed financial holding costs of 10% per year. Distribution of per unit costs is shown in Figure 3. The median cost is \$0.35 per dose. Our estimates for the financial holding cost and the cost of refills is in close agreement with the estimates used by Gebicki, Mooney, Chen and Mazur [12] in their simulation study examining medication supply chain in a mid-size hospital.

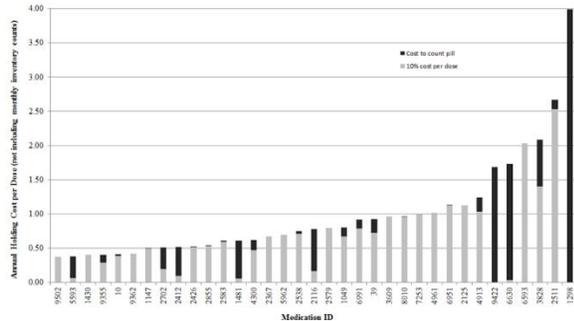


**Figure 3: Distribution of Per Unit Costs for the Medications Included in the Numerical Experiment**

In addition to financial opportunity costs, there are holding costs related to counting medication doses. There are two such holding costs associated with counting. First, the inventory is counted monthly. We estimated that it takes a technician 1/20<sup>th</sup> of a second to count an average dose (assuming 1/2 of a second per 10-pill blister pack). Assuming \$20/hour cost rate for the technician, and 12 counts per year, the annual holding cost for this counting is 1/3 of a penny per unit dose per year. For most medications, this holding cost is negligible compared to the 10% of the purchasing cost, which is what we assumed for the financial opportunity cost.

The majority of medications stored in a Pyxis machine are controlled medications, such as narcotic painkillers. Controlled medications are 62% of the medications in our sample. All the doses of the controlled medication in a container need to be counted by a nurse each time a dose is dispensed from the container. So, we estimated the second component of holding cost related to counting, as the equivalent of 1/20 second of a nurse's time *multiplied by the annual demand*. The fact that this admittedly small cost to count a pill was incurred every time a dose was administered made it significant for some drugs. We assumed the nurse's wage of \$40 per hour, which leads to the estimated of \$0.0006 per unit dose *per unit of annual*

demand. Figure 4 illustrates the impact of this counting costs, comparing it to the 10% cost of purchase. For some medications, the counting cost (shown as the dark part of a stacked bar) constitutes a significant portion of the holding cost.



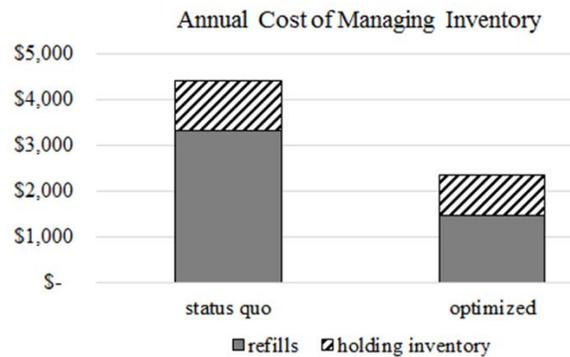
**Figure 4: Comparing Components of the Holding Cost**

One option we considered in the optimization was to split the inventory for a drug across two containers. Roughly speaking, if it was split equally across two containers, then the number of units that a nurse would need to count upon opening only one container was half of what it would have been if all the units were stored in a single larger container. We assumed that the controlled medication counting cost is only half of what we described above, i.e., \$0.0003 per unit dose per unit of annual demand if the medication is distributed between two containers rather than stored in a single container.

**4.2. Comparison of the Optimized Solution with the Status Quo**

To quantify the cost savings that can be achieved using optimized allocation of the medications to Pyxis containers we compared it with the status quo allocation. We obtained historical transaction data, as well as the minimum and maximum par levels for a set of 109 medications stored in a Pyxis machine in one of the hospital units. A minimum of four full-height and five half-height Pyxis drawers is needed to store this set of medications. We used simulation to estimate the inventory management cost for the status quo minimum and maximum par levels for these medications. For the simulation, we assumed restocking and counting costs as discussed in the previous section, and a single regular restocking per day. We compared the number of refills and the average inventory level calculated directly from transaction data (with regular refills happening twice a day) and with our simulation estimates. Over the 109 medications our simulation estimated 10% fewer refills than shown in the transaction data, the average inventory level was estimated as 1% lower than the level calculated from the transaction data.

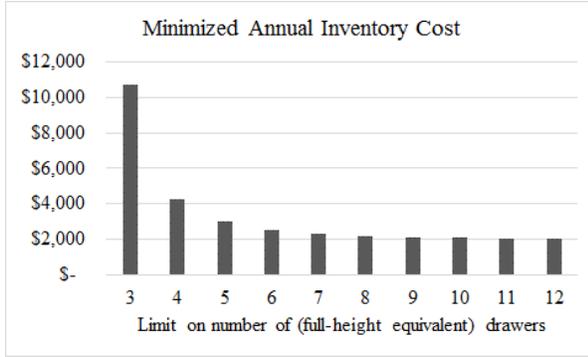
Figure 5 shows the result of our numerical analysis. We estimate that continuing with the current par levels will result in approximately \$4,500 as the total annual cost of holding and restocking the inventory in a single Pyxis machine. Approximately one-fourth of these costs is the cost of holding the inventory, and the rest is the cost of restocking. On average, 6.6 medications will be restocked per day, with approximately 10% of these expedited due to stock-outs. (Note, the current number of stock-outs is significantly smaller, but that is due to restocking twice a day, rather than once a day – the frequency we used in the simulation.)



**Figure 5: Cost Savings Status Quo vs. Optimized Solution for a Single Pyxis System**

The optimized solution results in approximately \$2,000 annual cost savings. The savings are mostly due to fewer refills, with 4 medications refilled on average per day, and an insignificant percentage of stock-outs. Fewer refills imply higher average inventory, so in total the inventory holding costs do not change significantly with the optimized solution. The savings in the inventory holding costs is approximately \$200 per year. Some of the savings in holding costs result from splitting medications over two containers. In the status quo solution, 21 medications are stored in two containers, while in the optimized solution 41 medications are stored in two containers. To allow for more containers two half-height drawers replace one of the full-height drawers.

Figure 6 shows how inventory management costs decrease as a function of the overall system capacity. Better allocation of medication within a Pyxis can allow for the same inventory costs as with the status quo but with 50% less space (4 instead of 6.5 full-height drawers). Using the space used currently can decrease the inventory management costs for this Pyxis unit by \$2,000 a year. Past a certain total capacity, additional capacity has little effect on the costs related to inventory management – assuming the capacity will not be used to manage other medications.



**Figure 6: Minimized Inventory Costs as a Function of Capacity**

## 5. Discussion and Conclusion

With increased pressure on operating costs, managers in the healthcare industry seek to understand whether they are using resources efficiently. Sometimes one of the challenges is simply to define efficiency. Quantification of the relevant costs can be another challenge. In this paper we examined inventory management costs related to using automated medication dispensing systems. We defined efficiency as minimizing costs related to refilling, storing, and dispensing medication using these systems. To estimate cost parameters, we observed employees involved in the tasks related to using the systems and translated their work into monetary costs. Using these costs and historical transactions data on stocking and dispensing medication using an ADS, we estimated annual ADS inventory unit management costs for an ADS in a typical hospital ward as approximately \$4,500 per year.

We proposed a linear integer programming formulation that will allow pharmacy managers to make better decisions about par levels for medications they store in an ADS, and how to configure an ADS to store the medication. With the numeric example based on actual hospital data we showed that optimization could result in significant cost savings in percentage terms: close to 50%. However, in absolute numbers these savings are not large relative to annual expenses of a hospital pharmacy. In a hospital with twenty ADS units, total annual savings will be less than the salary of one full-time pharmacy employee. In addition, a significant effort will be required to populate a database with medication dose types, and with the quantities that fit into various containers. The effort in populating such a database may be worthwhile for an ADS manufacturer, who could implement the proposed algorithms as part of their par level recommendation engine, a feature that may make its product more competitive.

## 6. Appendix A: Proof of the ILP Formulation

Consider the set of constraints (5)-(10). Let  $V = \{d_1, d_2, b_{(1x1)}, b_{(1x2)}, b_{(1x3)}, b_{(2x1)}, b_{(2x2)}, b_{(2x3)}, b_{(2x5)}\}$  be a solution that picks the number of drawers of the two heights and the number of containers of various heights and widths. Let  $C = \{V | V \text{ satisfies (5) - (10)}\}$ .

Let  $P = \{V | \begin{array}{l} \text{the containers represented by } V \\ \text{can be placed in a Pyxis} \end{array}\}$ . First, if  $V \in C$  and  $e_{(2x2)} > (b_{(2x2)} - b_{(2x3)})^+$  then we can set  $e_{(2x2)} = (b_{(2x2)} - b_{(2x3)})^+$  so we do that without loss of generality.

**Proposition:**  $C = P$ .

**Proof:** First we show that  $C \subset P$ . We do this by providing an algorithm that feasibly places the containers into rows of the appropriate drawers. Note that (5) and the integrality of  $d_1$  and  $d_2$  exactly defines the collections of drawer combinations that are feasible for a Pyxis. For the half-height drawers, there are  $5d_1$  rows in the  $d_1$  drawers and each drawer is of width 6. Place the  $b_{(1x3)}$  containers of width 3 into the first  $\lfloor b_{(1x3)}/2 \rfloor$  rows. If there are an odd number of containers then the last one will be placed in row  $\lfloor b_{(1x3)}/2 \rfloor$ . Now place the  $b_{(1x2)}$  containers of width 2 into the last  $\lfloor b_{(1x2)}/3 \rfloor$  rows. It is possible that there is one width 2 container that will need to go in row  $\lfloor b_{(1x3)}/2 \rfloor$  with the odd width 3 container. This placement is possible because by (6)  $2b_{(1x2)} + 3b_{(1x3)} \leq 6 \cdot 5 \cdot d_1$ . Finally place the containers of width 1 in the remaining space which is possible because the number of width 1 containers,  $b_{(1x1)} \leq 6 \cdot 5 \cdot d_1 - 2b_{(1x2)} - 3b_{(1x3)}$ , i.e., the remaining space. This completes placing the half-height containers.

For the full-height containers there are  $5d_2$  rows of width 5 in which to place them. Fill the first  $b_{(2x5)}$  rows each with one container of width 5. Now fill the next  $b_{(2x3)}$  rows with one container of width 3. There are enough rows because  $b_{(2x3)} + b_{(2x5)} \leq 5d_2$  by (7). Next, for these same rows that contain a container of width 3 add a container of width 2. If there are not enough we are fine; if there are additional containers of width 2, i.e.,  $e_{(2x2)} > 0$ , then place two of these containers in each of the next  $\lfloor e_{(2x2)}/2 \rfloor$  rows. Note that constraint (9) ensures that there are enough rows to do this. We have shown that  $C \subset P$ .

To show  $P \subset C$ , let  $V = \{d_1, d_2, b_{(1x1)}, b_{(1x2)}, b_{(1x3)}, b_{(2x1)}, b_{(2x2)}, b_{(2x3)}, b_{(2x5)}\}$  be the vector of drawers and containers that fits in this Pyxis. Clearly  $d_1$  and  $d_2$  satisfy (5). Similarly,

$b_{(1x1)}, b_{(1x2)}$ , and  $b_{(1x3)}$  satisfy (6) and  $b_{(2x1)}, b_{(2x2)}, b_{(2x3)}$ , and  $b_{(2x5)}$  satisfy (7) because they fit in the number of rows given. The only step that remains is to ensure that (8) and (9) are satisfied. To do this we sort the full height drawers as follows: We place the rows with width 5 containers first, call them  $R_5$ , next the rows that have a container of width 3 and a container of width 2, call them  $R_{32}$ , next the rows that have a container of width 3, but not a container of width 2, call them  $R_{3-}$ . Call the remaining rows  $R$ . If  $R_{3-} \neq \emptyset$  and there is a row in  $R$  with a container of width 2, move that container out of that row and into a row in  $R_{3-}$  which places the row into  $R_{32}$ . If there was anything else (width 1 container(s)) in that row put it/them in the row from which you removed the container of width 2. Repeat this process until either  $R_{3-}$  becomes empty or there are no more width 2 containers in  $R$ . The number of rows,  $d_2$ , is greater than or equal to the number of containers of width 5, i.e.,  $|R_5|$ , plus the number of containers of width 3, i.e.,  $|R_{32}| + |R_{3-}|$ , plus the number of rows in  $R$ , i.e.,  $|R|$ . Thus,  $5d_2 = b_{(2x3)} + b_{(2x5)} + |R|$ . Define  $e_{(2x2)} = (b_{(2x2)} - b_{(2x3)})^+$  to be the number of excess width-2 containers not in a row with a container of width 3. Because all of these fit into the rows,  $R$ , we must have that  $e_{(2x2)} \leq 2|R|$ . Therefore,  $e_{(2x2)} + 2b_{(2x3)} + 2b_{(2x5)} \leq 2|R| + 2b_{(2x3)} + 2b_{(2x5)} = 2(5d_2)$  which gives us (9).

## 7. References

[1] M.D. Murray, Automated Medication Dispensing Devices, in: J.D. Amy J. Markowitz (Ed.) A Critical Analysis of Patient Safety Practices, AHRQ Publication No. 01-E058, 2001.

[2] P. Kelle, J. Woosley, H. Schneider, Pharmaceutical supply chain specifics and inventory solutions for a hospital case, *Operations Research for Health Care*, 1 (2012) 54-63.

[3] S. Johansen, P. Melchior, Can-order policy for the periodic-review joint replenishment problem, *Journal of the Operational Research Society*, 54 (2003) 283-290.

[4] J.L. Balintfy, On a basic class of multi-item inventory problems, *Management Science*, 10 (1964) 287-297.

[5] D.R. Atkins, P.O. Iyogun, Periodic versus “can-order” policies for coordinated multi-item inventory systems, *Management Science*, 34 (1988) 791-796.

[6] H. Feng, Q. Wu, K. Muthuraman, V. Deshpande, Replenishment Policies for Multi-Product Stochastic Inventory Systems with Correlated Demand and Joint-Replenishment Costs, *Production and Operations Management*, 24 (2015) 647-664.

[7] M. Khouja, S. Goyal, A review of the joint replenishment problem literature: 1989–2005, *European journal of operational Research*, 186 (2008) 1-16.

[8] D.L. Iglehart, Optimality of (s, S) policies in the infinite horizon dynamic inventory problem, *Management Science*, 9 (1963) 259-267.

[9] H. Scarf, The optimality of (s, S) policies in the dynamic inventory problem, *First Stanford Symposium on Mathematical Methods in the Social Sciences*, Stanford University Press, 1960, pp. 196-202.

[10] Y.-S. Zheng, A Simple Proof for Optimality of (s, S) Policies in Infinite-Horizon Inventory Systems, *Journal of Applied Probability*, 28 (1991) 802-810.

[11] S. Chen, M. Lambrecht, X-Y Band and Modified (s, S) Policy, *Operations Research*, 44 (1996) 1013-1019.

[12] M. Gebicki, E. Mooney, S. Chen, L. Mazur, Evaluation of hospital medication inventory policies, *Health Care Manag Sci.*, 17 (2014) 215-229.