Technical Report

# Optimizing Cash Management for Large Scale Bank Operations

Revised April 10, 2009

## Abstract

The Federal Reserve System (Fed) provides currency services to banks. These services include sorting currency into fit and non-fit bills and repackaging bills for redistribution. To reduce their own cost of currency management operations, many banks would make Fed deposits and withdrawals of the same denomination during a given week. In July of 2007, the Fed introduced fees for making both deposits and withdrawals during a given Monday through Friday period to encourage banks to recycle their currency. Recognizing the market opportunity, Fiserv Corporation initiated a project to optimize bank vault inventories across time and space for a major bank who previously suffered from over 7 billion dollars sitting idle either in trucks, vaults, branches, or ATMs.

This manuscript presents the integer programming model developed to assist Fiserv clients reduce the logistics cost component of cash management. The model has been implemented in software using OPL and CPLEX and has been successfully integrated within their suite of product offerings. The underlying configuration is a time-space multi-commodity network with a fixed-charge cost structure. We report on a successful six month pilot study and present an efficient heuristic procedure that can be used to reduce computational solution times from hours to only a few minutes.



**Keywords:** Cash inventory, multi-commodity, cross-shipping, custodial inventory, time-space network

## 1. Introduction

The major banks in the United States own cash distribution centers called *vaults*. A vault is a large warehouse with no windows and lots of security. Deposits and withdrawals are made using armored vehicles carrying millions in cash. Vaults contain equipment that counts, sorts, and packages currency for redistribution. The fit currency can be distributed to their branch banks, can be used to supply ATMs, or can be returned to the Fed. Currency that is determined to be unfit for further circulation is returned to the Fed for destruction. Large national banks have created a network of cash warehouses (vaults) to manage their cash requirements.

Vaults may experience either a deficit or a surplus of currency of the various denominations. In the past, the Fed supplied deficits and accepted surplus cash from the various vaults at a nominal fee. Beginning in July 2007, the Fed imposed penalties on certain types of cash transfers called *cross-shipping fees*. Since counterfeit bills are usually 100s and 50s, these were excluded from cross-shipping fees. That is, the Fed wants to continue frequent checks of these denominations so that counterfeit bills could be quickly removed from circulation. For other denominations, cross-shipping fees apply when both an order and a deposit occur during the same week. That is, placing an order for 20s on Monday and making a deposit of 20s on Thursday results in a cross-shipping fee on the smaller of the two transactions.

In order to help the vaults avoid excessive cross-shipping fees, the Fed allows a vault to designate a certain area with a given capacity for what is known as *custodial inventory*. Any currency placed in custodial inventory belongs to the Fed but is physically located at the vaults. The advantage to the vault is that this currency has been taken off the books of the vault and essentially returned to the Fed. In addition, the Fed excuses a certain amount of cross-shipping without penalty during each quarter. This value is known as *de minimus* (Fedfocus 2007). Suppose de minimus is 20 million for a given vault. Then no cross-shipping penalty is paid until the 20 million in cross-shipping transactions is exceeded during the current quarter.

Fiserv Corporation is a major player in banking software. One of their products, iCom, helps banks manage the logistics of cash with respect to vaults, branches, and

ATMs. In 2005 they began to develop a new module that would optimize logistical operations at vaults under the new cross-shipping environment to be imposed in the summer of 2007. This manuscript presents the linear multi-commodity integer programming model developed to support the logistical operations of their clients. This model is written in OPL 5.0 (http://www.ilog.com/products/oplstudio/ ) and uses the CPLEX 10.1 (http://www.ilog.com/products/cplex/ ) solver. The optimization model was implemented as a new module within iCom and was used in a six month pilot study by one of their bank clients. Since computational time is a major issue for this project, we also developed a new heuristic that mitigates the computational time issue.

Due to the different possible combinations of cash transfers that can occur, certain assumptions were made in the model. Any cash transfer from the Fed on a particular day is available at the vault on the same day for distribution. A cash transfer between vaults on the same day implies the cash is available at the destination vault on that day. A one day cash transfer between vaults implies that the money is available on the very next day at the destination vault for distribution. The upper bound on the bundles of cash that can be transferred is infinity, i.e. there are always trucks available for cash transfers. The variable cost on cash transfers is linear.

## 2. Survey of Literature

The use of optimization models to assist management in the area of cash management dates back to the 1960s. Orgler [1969] presents a multi-period linear programming model for cash management. Decisions regarding payments, short-term financing, the cash balance, and securities transactions are variables in his model. He attempts to maximize revenues over the planning period subject to institutional business rules. One unique feature of his model is the use of unequal length planning periods. Hence, a one-year model can begin with daily transactions and end with monthly transactions.

Constantinides [1976] uses stochastic calculus to analyze a special cash management problem with stochastic demand. He attempts to minimize the expected cost in the time interval [0,T] as T goes to infinity. Several special cases are analyzed in his exposition. An extension of this work may be found in Constantinides and Richard

[1978]. They formulate a cash management model for a single entity (such as a bank or retail outlet) where the demand for cash in a given time interval is given by a random variable. Their model uses fixed and variable costs for transferring funds as well as a holding-penalty cost for maintaining the given cash balance. They provide a policy analysis for a manager who continuously monitors the cash position and only intervenes at optimal points in time.

Mensching et al. [1978] consider a cash management problem using techniques from inventory theory. Their model is a deterministic economic lot size model in which negative demands are allowed. They also use a linear inventory holding cost and fixed and variable costs for changing inventory levels. They present a forward algorithm to obtain the minimum cost solution to a t-period problem.

More recent work in the area of cash management in fast growing firms may be found in Buzacott and Zhang [2004]. They address the complicated relationship between finance and production. They developed models that attempt to capture the complex trade-offs that must be addressed by these fast growing firms.

An excellent description of the new Fed guidelines may be found in Geismar et al. [2007]. They also present procedures that could assist bank managers in scheduling deposits and withdrawals to the Fed as well as deciding whether or not to purchase expensive equipment needed to sort cash into fit and non-fit groups. They provide guidance on the use of a custodial inventory facility. Brown and Rosenthal [2008] provide an excellent guide for using optimization to solve real-world problems such as the one described in this investigation.

## 3. Contribution to the Literature

This manuscript reports on a successful application of large-scale optimization to the problem of minimizing logistics costs for large national banks. The problem was motivated by a rule change by the Federal Reserve System, but can lead to major cost savings related to both transportation and opportunity cost. Our experience with a six month pilot study is reported. Since computational time is one of the major concerns of clients for this system, we also developed a heuristic strategy that provides near optimal solutions in a fraction of the time required for achieving an optimum. Our heuristic strategy should also be applicable to other problems of this type.

## 4. The Cash Flow Model

A small example of the cash flow time-space network for a single denomination is illustrated in Figure 1. The nodes in the network represent the vault-day combinations, and the Fed. The Fed is a super source with infinite supply and also a super sink. An arc in Figure 1 denotes the flow of cash between nodes. The units of cash flow in the network are called bricks. A brick is composed of 1000 bills of a given denomination. Hence, a brick of 20s is 20,000 dollars. The permissible cash flows (arcs) include same-day trades, one-day trades, overnight transfers, Fed to vault trades, vault to Fed trades, and custodial inventory. A same-day trade is represented by an arc between different nodes corresponding to different vaults on the same day. A one-day trade is represented by an arc between different vaults on successive days. An arc between nodes corresponding to the same vault on successive days represents an overnight transfer or change in the custodial inventory. The overnight transfers can also occur over the weekend or over holiday periods. The arc in Figure 1 from the Fed to vault 2 represents orders placed by vault 2 on day 2. The arc from vault 1 to Fed denotes a deposit made by vault 1 on day 3. The requirement at a vault on a particular day is denoted either by a positive value for surplus or a negative value for deficit in cash. Vault 1 on day 1 has a surplus of 100 bricks while vault 3 has a deficit of 30 bricks.

Note that in Figure 1, vault 2 places an order to the Fed on day 2 and makes a deposit on day 3. Beginning in July of 2007, the Fed began imposing a cross-shipping penalty for placing orders and making deposits of the same denomination during a given Monday through Friday period. There is no penalty for orders only or for deposits only. A cross-shipping penalty is imposed only when both occur during the same week. An order on Friday followed by a deposit on Monday would not incur a penalty.

Figure 1: About Here

A mixed integer multi-commodity linear model for transferring bricks of different denominations in a time-space network has been developed. Let $V$ denote the set of vaults, and let $D$ denote the set of days on which transactions may occur. Let $K$ denote the set of denominations for the model. A description of the all the sets used in the model is presented in Table 1. A description of all costs used in the objective function may be found in Table 2. The capacity related constants, and inventory related constants are described in Table 3. Lists of all the variables in the model are described in Tables 4-9. The objective is to minimize the total transportation and cross-shipping costs, and is presented below.

Tables 1 to 9: About Here

Minimize $$\underbrace{\sum_{u\in V}\sum_{d\in A}\sum_{k\in K}\tau_{kd}\,\text{◐}_{udk}}_{\text{overnight carrying costs}} + \underbrace{\sum_{u\in V}\sum_{v\in V}\sum_{d\in D}\sum_{k\in K}p_{uvdk}\,w_{uvdk}}_{\text{variable costs for same-day transfers}} +$$

$$\underbrace{\sum_{u\in V}\sum_{v\in V}\sum_{d\in A}\sum_{k\in K}(\tau_{kd}+q_{uvdk})\,y_{uvdk}}_{\text{variable costs for one-day transfers}} + \underbrace{\sum_{u\in V}\sum_{v\in V}\sum_{d\in D}b_{uvd}\,\omega_{uvd}}_{\text{fixed costs for same-day transfers}} +$$

$$\underbrace{\sum_{u\in V}\sum_{v\in V}\sum_{d\in A}e_{uvd}\,\boldsymbol{y}_{uvd}}_{\text{fixed costs for one-day transfers}} + \sum_{u\in V}\sum_{d\in D}(\underbrace{i_{ud}\,\varkappa_{ud}}_{\text{fixed costs for Fed to vault transfers}} + \underbrace{j_{ud}\,\mathfrak{z}_{ud}}_{\text{fixed costs for vault to Fed transfers}}) +$$

$$\sum_{u\in V}\sum_{d\in D}\sum_{k\in K}(\underbrace{m_{udk}\,x_{udk}}_{\text{variable costs for Fed to vault transfers}} + \underbrace{n_{udk}\,z_{udk}}_{\text{variable costs for vault to Fed transfers}}) + \sum_{k\in S}(f_k\underbrace{(\bar{b}_k+\hat{b}_k)}_{\text{Fed to vault cross shipping fees}} + g_k\underbrace{(\tilde{b}_k+b'_k)}_{\text{vault to Fed cross shipping fees}}) +$$

$$\underbrace{\sum_{u\in V}\sum_{d\in D}\sum_{k\in K} \mathbf{M}(\bar{\phi}_{udk} + \bar{\xi}_{udk})}_{\text{penalty for artificial flows to make the solution feasible}}$$

There are approximately fifty different types of constraints needed to describe the various business rules modeled. Flow conservation constraints are required for each vault-day-denomination combination. Numerous constraints are needed to obtain the proper relationship between continuous and binary variables used to model the fixed-cost structure. Constraints are needed to account for the arrival time of cash at the various vaults. That is, cash that arrives at noon cannot be shipped out until the next morning. Space and insurance based capacity rules must also be enforced as well as the de minimus and custodial inventory exclusions.

All binary variables have bounds that may be set to 0. This allows the client to close potential arcs (transfers). Hence, the basic model may not have a feasible solution. Expensive artificial variables have been appended to the model so that a feasible solution is always available. An artificial variable with a value greater than zero implies that either a deficit is not satisfied or surplus cash could not be disbursed.

We now provide a discussion of the various types of constraints needed for the application. The following constraints ensure that flow is conserved at each vault on the first day for denominations that are subject to a cross-shipping fee.

$$\text{◐}_{u1k} + \sum_{v\in V}(w_{uv1k} - w_{vu1k} + y_{uv1k}) - x_{u1k} + z_{u1k} + \boldsymbol{c}_{u1k} - t_{uk} + \bar{\phi}_{u1k} - \bar{\xi}_{u1k} = r_{u1k} \qquad (1)$$

$$\forall u \in V, \forall k \in S$$

The following constraints ensure that binary variables associated with the flow variables for various transfers are set to one if the corresponding flow variable is greater than zero. The binary variables in constraints (2) are used to impose fixed costs for the transfers. The binary variables in constraints (3) are used to impose transfer restrictions.

$$\sum_{k\in K} w_{uvdk} \le \mathbf{M}\ \omega_{uvd} \qquad \forall u, v \in V, \forall d \in D \qquad (2)$$

$$\sum_{u \in V} w_{uvdk} \leq \mathbf{G}_k \, \psi_{vdk} \qquad \forall v \in V, \forall d \in D, \forall k \in K \qquad (3)$$

where $\mathbf{M}$ and $\mathbf{G}_k$ are large positive constants.  Constraints of type (4) restrict certain transactions from occurring on the same-day.

$$\bar{\phi}_{udk} + \psi_{udk} \leq 1 \qquad \forall u \in V, \forall d \in D, \forall k \in K \qquad (4)$$

Equations (5) calculate the total number of bricks held in the vaults for each day. The numbers of bricks are bounded below by 0 and above by $\kappa_u$ .

$$\sum_{k \in K} (\alpha_{udk} + \mathbf{C}_{udk}) = \boldsymbol{\ell}_{ud} \qquad \forall u \in V, \forall d \in A \qquad (5)$$

The following equations calculate the cash value held in the vaults for each day. The cash values are bounded below by 0 and above by $\vartheta_u$ .

$$\sum_{k \in K} \beta_k (\alpha_{udk} + \mathbf{C}_{udk}) = d_{ud} \qquad \forall u \in V, \forall d \in A \qquad (6)$$

Equations (7) calculate the change in the custodial inventory each day. This is used only for reporting purposes.

$$\mathbf{t}_{u1k} = \boldsymbol{c}_{u1k} - t_{uk} \qquad \forall u \in V, \forall k \in S \qquad (7)$$

The following constraints calculate the total Fed to vault values for denominations that are subject to a cross-shipping fee.

$$\bar{c}_k = \sum_{v \in V} \sum_{d \in C} x_{vdk} + o_k \qquad \forall k \in S \qquad (8)$$

Constraints (9) are used to calculate the Fed to vault values that will be subject to a cross-shipping fee. The fee is applied to a given denomination if that denomination has both orders and deposits from one or more vaults within the same week.  The penalty is applied to the smaller of the total deposits and total orders.

$$\overline{b}_k \geq \overline{a}_k - \sum_{u \in V} \sum_{d \in C} |r_{udk}| \hbar_k \qquad \forall k \in S \qquad (9)$$

The boundary conditions are as follows:

$$0 \leq \overline{\phi}_{udk} \leq \phi_{udk} \qquad \forall u \in V, \forall d \in D, \forall k \in K$$

$$0 \leq \overline{\xi}_{udk} \leq \xi_{udk} \qquad \forall u \in V, \forall d \in D, \forall k \in K$$

$$0 \leq c_{udk} \leq h_{uk} \qquad \forall u \in V, \forall d \in D, \forall k \in K$$

$$0 \leq \mathscr{b}_{ud} \leq \kappa_u \qquad \forall u \in V, \forall d \in D$$

$$0 \leq \mathscr{d}_{ud} \leq \vartheta_u \qquad \forall u \in V, \forall d \in D$$

There are a few more constraints needed to complete the model. For example, flow conservation constraints were only given for day 1. Similar constraints are required for the middle days and the last day. Some additional constraints are also needed to describe the de minimus exclusion policy.

## 5. A Pilot Study

In this section, we provide the results of a pilot study by a large US bank. Due to confidentially requirements, this bank prefers to remain anonymous. This bank has over 20,000 retail outlets (branches and ATMs) and dispenses over $200 million daily. Prior to the pilot study, they maintained a cash inventory of approximately $7 billion. The pilot study involved 58 vaults.

After the 6 month pilot study, cash inventory was reduced by 35%. Transportation costs were decreased by 55% and cross-shipping fees were projected to decrease by 63% over the previous year. The project was rated “highly successful” by the bank’s internal Six Sigma Unit. Based on this pilot, the system was scheduled for rollout to the entire enterprise.

The model described in this pilot study was implemented as a new module within an existing software product, iCom, offered by Fiserv Corporation (http://www.carreker.com/main/solutions/cash/icom.htm). It was originally developed by Carreker Corporation prior to purchase by Fiserv. The iCom system is designed to help manage a client's cash supply chain. Among other things, the iCom product produces cash forecasts by denomination for all client vaults. Hence, most of the data needed for the optimization model was available in the iCom database.

Special routines were created to pass the input data directly into the OPL model and retrieve the solutions after optimization. Data is loaded directly into memory and no files are created. The system is designed to run daily around 4:00 am so that the results are available when employees arrive for work each day.

## 6. A Heuristic Procedure

Since the cost structure for deliveries involves a fixed-cost and a variable cost, there are numerous constraints that relate continuous variables to binary variables of the form (2), where $w_{uvdk} \geq 0$, $\omega_{uvd}$ is binary, and $M$ is a large positive value (called BigM). Hence, if $\sum_{k \in K} w_{uvdk} > 0$, then $\omega_{uvd}$ must take the value of 1 and the fixed-cost ($b_{uvd}$) must be paid. The BigM value used in constraints of this type must be sufficiently large so as not to place an artificial bound on $\sum_{k \in K} w_{uvdk}$. For this model the BigM values are critical in determining the strength of the continuous relaxation and consequently to solution times for the original integer programming model. The BigM values can be interpreted as truck (arc) capacities and can be calculated based on the requirements of the vaults connected by the arcs.

In order to help determine the relationship between the value of BigM and solution time, we conducted an empirical evaluation using four different strategies for calculating BigM values. In strategy 1, the BigM was set to the sum of the absolute values of the supplies and demands. For the problem illustrated in Figure 1, BigM using strategy 1 is 100+70+30+80+40 = 320. Strategy number 2 uses the maximum of total surplus values or total deficits. For the Figure 1 problem this will be max{180,140} = 180. The third strategy uses a different BigM value for each arc. Consider the arc from v1 to v2 on day 1. The flow on this arc will be no more than the total supply at day 1 or the total demand on days 1, 2, and 3. Hence, BigM for this arc is max {100,140} = 140. That is, for a 10 day model, the arc capacity on an arc at day d can be set to the total supply on days 1,…,d or the total demand on days d,…,10, whichever is larger. For the arc (v3, v2) on day 2, BigM can be set to max {180, 40} = 180. All three of these strategies result in problem instances with identical optimal objective values.

The fourth strategy for selecting the BigM values results in problem instances whose optimal objective values may exceed those of the first three strategies. That is, using the fourth strategy may impose too small an arc capacity resulting in a heuristic algorithm for this problem. For each arc, the BigM is set to the larger of the supply/demand at the end nodes for this arc. For arc (v1, v2) on day 1 in Figure 1, BigM for this arc is set to max {100, 70} = 100. For the arc (v3, v2) on day 2, BigM is set to max {80, 40} = 80. Clearly this strategy may exclude an optimal solution.

Our four strategies have been tested in an empirical evaluation. Seven problem instances of varying sizes were solved using OPL Studio 5.0 and CPLEX 10.1. All test cases were run on a Dell PE 2850, 2.8 GHz dual core Intel Xeon processors with 8 GB of RAM. The problem characteristics are illustrated in Table 10. Problem size is a function of the number of vaults and the number of denominations considered. All problem instances were 7 day models.

Table 10: About Here

Problems P01 and P02 were solved in only one second using all four strategies and P03 only required 11 minutes using strategy 1. Problems P04 through P07 were quite difficult to solve as illustrated in Table 11. The cost reported is a scaled cost and the time is CPU time. All runs were given a 6 hour time limit and the best solution found along with the current optimality gap was reported. Note that strategy 2 obtained the same solutions as strategy 1, but the final optimality gaps were slightly smaller. The third strategy also yielded the same solutions, but the gaps were substantially reduced.

Since all runs were allowed to terminate if a solution was found that was guaranteed to be within 10% of an optimum, problem P06 did not require the full 6 hours of CPU time. Clearly, as BigM values are reduced, convergence improves. The heuristic procedure only requires a few minutes and the objective values were identical on two of the problems, only 0.4% higher on one problem, and 10.9% higher on the other. Since, the surplus and deficit values for the seven day models are forecast values, solutions within 10% are considered satisfactory for this application. Our basic procedure used in strategy 4 should also be applicable for other time-space network problems involving fixed-costs.

Table 11: About Here

These problems are sensitive to the BigM values because the fixed-costs are large compared to the variable costs. For problem P05, most of the fixed-costs are in the range of 200 to 750 and the variable costs are all 0.05. Hence, the linear programming relaxations are very weak for this problem instance. A weak LP relaxation results in excessive branching in a branch-and-bound procedure.

The OPL Studio and CPLEX have several features that can be exploited in an attempt to improve computational efficiency. These include the use of logical constraints that eliminates the need for the BigM values, priority branching, solution polishing, and application of strong branching. In additional empirical tests with the seven test problems, various combinations of these features have been evaluated. Strong branching was the only feature that reduced computational time and that feature is standard in our production system.

## 7. Summary and Conclusions

It's always important for banks to have efficient cash management procedures. The global financial crisis of 2009 has placed additional pressure on banks to manage these operations as efficiently as possible. For large banks, the costs of maintaining high inventory levels of cash along with the costs for transporting cash can be substantial. Fiserv Corporation developed an optimization based software system to assist their banking clients in this difficult problem of cash management. This manuscript presents the mathematical model developed for this project along with the results of a six month pilot study. The client that conducted the pilot study reports that they were able to remove over a billion dollars from their cash inventory.

The complex model involves multiple concepts which have appeared in the literature. The underlying structure is a time-space multi-commodity network flow model. However, most of the arcs represent transfer by armored trucks with both a fixed and variable cost structure. Hence, most problem instances have a large number of binary variables as well as a substantial number of integer variables. The real problem may not have a feasible solution. To ensure that a report can be provided to the client, artificial variables with large costs were used so that feasibility is guaranteed.

An investigation of various fine tuning parameters did not lead to substantial improvement in solution time. Strong branching did yield improvements on our test set. The most important parameters that influence computation time are the various BigM values needed to force binary variables associated with arcs to assume the value of 1 when the corresponding arc has a flow value greater than 0. The smaller the BigM values, the faster CPLEX can solve a problem instance. Since the underlying structure is a time-space network, it was possible to determine fairly small values that will still guarantee optimality. Since our arcs were unbounded, the BigM values are artificial bounds that must be large enough to not alter the optimal solution. However, the heuristic that imposes the bound on arc (i, j) as the maximum of the absolute requirements at node i and node j worked very well for all our test cases.

Our major challenge was to understand all the business rules associated with cash management operations. The most interesting of these were the de minimus forgiveness rule and the rule for imposing cross-shipping fees on the smaller of deposits and orders during a given week. The OPL modeling language and the CPLEX solver have been invaluable tools in the development of this application.

We are aware of at least two limitations of the current implementation. Both are related to the cost structure of contracts with armored carriers. A standard trip will involve a truck and two guards. However, high volume transfers may require a third guard and extremely high volumes may also involve a second vehicle with additional personnel. Other contracts are written for round trips as opposed to single segment trips. That is, the fixed-cost is paid for a trip involving multiple stops. We know how to model round trips and expect that feature to appear in the next version of the model. Additional research is required to model the more complicated contracts.

## References

Buzacott, J., R. Zhang. 2004. Inventory management with asset-based financing. *Management Science* **50** 1274-1292.

Brown, G., R. Rosenthal. 2008. Optimization Tradecraft: Hard-won insights from real-world decision support. *Interfaces* **38** 356-366.

Constantinides, G. 1976. Stochastic cash management with fixed and proportional transaction costs. *Management Science* **22** 1320-1331.

Constantinides, G., S. Richard. 1978. Existence of optimal simple policies for discounted-cost inventory and cash management in continuous time. *Operations Research* **26** 620-636.

Fedfocus, 2007. Federal Reserve Financial Services. http://www.frbservices.org/files/communications/pdf/fedfocus/2007/fedfocus0707.pdf retrieved April 01, 2008.

Geismar, N., M. Dawande, D. Rajamani, C. Sriskandarajah. 2007. Managing a bank's currency inventory under new Federal Reserve guidelines. *Manufacturing & Service Operations Management* **9** 147-167.

Mensching, J., S. Garstka, T. Morton. 1978. Protective planning-horizon procedures for a deterministic cash balance problem. *Operations Research* **26** 637-652.

Orgler Y. 1969. An unequal-period model for cash management decisions. *Management Science* **16** B77-B92.

**Table 1 – Sets**

| Sets | Description |
|---|---|
| $V$ | Set of vaults |
| $D$ | Set of days over a ten day period when the currency transfers may occur ($D = \{1, 2, \ldots, 10\}$) |
| $A$ | Set of days except the last day ($A = D \setminus \{10\}$) |
| $Y$ | Set of days except the first and last day ($Y = A \setminus \{1\}$) |
| $C$ | Set of days in week 1 when a cross-shipping fee may be assessed |
| $X$ | Set of days in week 2 when a cross-shipping fee may be assessed |
| $K$ | Set of denominations |
| $S$ | Set of denominations for which a cross-shipping fee will be assessed (50s and 100s are excluded) |
| $N$ | Set of denominations for which no cross-shipping fee will be assessed ($N = K \setminus S$) |

**Table 2 – Costs**

| Constants | Description |
|---|---|
| $\tau_{kd}$ | The cost to carry one brick of denomination $k \in K$ on day $d \in D$ overnight |
| $f_s$ | Cross-shipping fee charged for transfer of bricks of denomination $s \in S$ from the Fed to any vault (deposits) |
| $g_s$ | Cross-shipping fee charged for transfer of denomination $s \in S$ from a vault to the Fed |
| $b_{uvd}$ | Fixed cost for shipment from vault *u* to vault *v* where $u, v \in V$ for a same-day trade on day $d \in D$ |
| $e_{uvd}$ | Fixed cost for shipment from vault *u* to vault *v* where $u, v \in V$ for a one day trade i.e. from day $d \in D$ to day $d$+1 |
| $p_{uvdk}$ | Variable cost for shipment of one brick of denomination $k \in K$ from vault *u* to vault *v* where $u, v \in V$ for a same-day trade on day $d \in D$ |
| $q_{uvdk}$ | Variable cost for shipment of one brick of denomination $k \in K$ from vault *u* to vault *v* where $u, v \in V$ for a one-day trade on day $d \in D$ |
| $i_{vd}$ | Fixed cost for shipment from the Fed to vault $v \in V$ on day $d \in D$ |
| $j_{vd}$ | Fixed cost for shipment from vault $v \in V$ to the Fed on day $d \in D$ |
| $m_{vdk}$ | Variable cost for shipment of one brick of denomination $k \in K$ from the Fed to vault $v \in V$ on day $d \in D$ |
| $n_{vdk}$ | Variable cost for shipment of one brick of denomination $k \in K$ from vault $v \in V$ to the Fed on day $d \in D$ |

**Table 3 – Constants**

| Constants | Description |
|---|---|
| $r_{vdk}$ | Requirement of denomination $k \in K$ at vault $v \in V$ on day $d \in D$ in bricks. A positive value of *r* indicates surplus and a negative value indicates deficit |
| $\phi_{vdk}$ | Surplus of denomination $k \in K$ at vault $v \in V$ on day $d \in D$ in bricks i.e. $\phi_{vdk} = r_{vdk}$ if $r_{vdk} > 0$; and 0, otherwise |
| $\xi_{vdk}$ | Deficit of denomination $k \in K$ at vault $v \in V$ on day $d \in D$ in bricks i.e. $\xi_{vdk} = - r_{vdk}$ if $r_{vdk} < 0$; and 0, otherwise |
| $t_{vs}$ | Initial custodial inventory of denomination $s \in S$ at vault $v \in V$ |
| $h_{vs}$ | Maximum capacity of custodial inventory for vault $v \in V$ of denomination $s \in S$ in bricks |
| $o_s$ | Total orders placed at the Fed by the vaults on the previous days of the current week for denomination $s \in S$ in bricks |
| $\boldsymbol{l_s}$ | Total deposits of denomination $s \in S$ made to the Fed on previous days of the current week |
| $\alpha_{vdk}$ | Process balance at vault $v \in V$ on day $d \in D$ of denomination $k \in K$ in bricks |
| $\kappa_v$ | Capacity of vault $v \in V$ in bricks |
| $\vartheta_v$ | Capacity of vault $v \in V$ in dollars |
| $\beta_k$ | Value of a brick of denomination $k \in K$ in dollars |
| $I$ | The number of bricks of orders and deposits excused from the cross-shipping penalty |
| $l$ | The last day of transaction ($l = \lvert D \rvert$ ) |

**Table 4 – Overnight, Reporting, Capacity, and Insurance Variables**

| Variables | Description |
|---|---|
| $◐_{vdk}$ | The number of bricks of denomination $k \in K$ carried overnight in vault $v \in V$ from day $d \in D$ to $d$+1 |
| $\ell_{vk}$ | The number of bricks of denomination $k \in K$ remaining at vault $v \in V$ on the last day |
| $c_{vas}$ | The flow of bricks of denomination $s \in S$ through the custodial inventory of vault $v \in V$ on day $a \in A$ |
| $t_{vas}$ | The change in custodial inventory of denomination $s \in S$ at vault $v \in V$ on day $a \in A$ |
| $b_{vd}$ | The total number of bricks of various denominations stored in vault $v \in V$ on day $d \in D$ |
| $d_{vd}$ | The total value in dollars of various denominations stored in vault $v \in V$ on day $d \in D$ |
| $b_{vd}$ | The total number of bricks of various denominations stored in vault $v \in V$ on day $d \in D$ |
| $d_{vd}$ | The total value in dollars of various denominations stored in vault $v \in V$ on day $d \in D$ |

**Table 5 – Same-Day Transaction Variables**

| Variables | Description |
|---|---|
| $w_{uvdk}$ | The number of bricks of denomination $k \in K$ shipped from source vault $u$ to destination vault $v$ on day $d \in D$ |
| $\omega_{uvd}$ | The binary decision variable $\omega_{uvd}$ = 1 if bricks are shipped from source vault $u$ to destination vault $v$ on day $d \in D$; and $\omega_{uvd}$ = 0, otherwise |
| $\psi_{vdk}$ | The binary decision variable $\psi_{vdk} = 1$ if bricks of denomination $k \in K$ are shipped into destination vault $v \in V$ on day $d \in D$; and $\psi_{vdk} = 0$, otherwise |
| $\varpi_{udk}$ | The binary decision variable $\varpi_{udk} = 1$ if bricks of denomination $k \in K$ are shipped from source vault $u \in V$ on day $d \in D$; and $\varpi_{udk} = 0$, otherwise |

**Table 6 – One-Day Transaction Variables**

| Variables | Description |
|---|---|
| $y_{uvdk}$ | The number of bricks of denomination $k \in K$ shipped from source vault $u \in V$ on day $d \in A$ to destination vault $v \in V$ on day $d+1$ |
| $\boldsymbol{y}_{uvd}$ | The binary decision variable $\boldsymbol{y}_{uvd}$ = 1 if bricks are shipped from source vault $u \in V$ on day $d \in A$ to destination vault $v \in V$ on day $d+1$; and $\boldsymbol{y}_{uvd}$ = 0, otherwise |
| $\mathscr{y}_{vdk}$ | The binary decision variable $\mathscr{y}_{vdk}$ = 1 if bricks of denomination $k \in K$ are shipped from any source vault on day $d-1$ into destination vault $v \in V$ on day $d \in D$; and $\mathscr{y}_{vdk} = 0$, otherwise |
| $\mathbf{y}_{udk}$ | The binary decision variable $\mathbf{y}_{udk}$ = 1 if bricks of denomination $k \in K$ are shipped from source vault $u \in V$ on day $d \in A$ to any destination vault on day $d$+1; and $\mathbf{y}_{udk}$ = 0, otherwise |

**Table 7 – Fed-to-Vault and Vault-to-Fed Transaction Variables**

| Variables | Description |
|---|---|
| $x_{vdk}$ | Number of bricks of denomination $k \in K$ ordered by vault $v \in V$ from the Fed on day $d \in D$ |
| $\varkappa_{vd}$ | The binary decision variable $\varkappa_{vd}$ = 1 if there is an order from vault $v \in V$ on day $d \in D$; and $\varkappa_{vd}$ = 0, otherwise |
| $\boldsymbol{x}_{vdk}$ | The binary decision variable $\boldsymbol{x}_{vdk}$ = 1 if there is an order for denomination $k \in K$ from vault $v \in V$ on day $d \in D$; and $\boldsymbol{x}_{vdk}$ = 0, otherwise |
| $z_{udk}$ | Number of bricks of denomination $k \in K$ deposited by vault $u \in V$ in the Fed on day $d \in D$ |
| $\mathfrak{z}_{ud}$ | The binary decision variable $\mathfrak{z}_{ud}$ = 1 if there is a deposit to the Fed from vault $u \in V$ on day $d \in D$; and $\mathfrak{z}_{ud}$ = 0, otherwise |
| $\boldsymbol{z}_{udk}$ | The binary decision variable $\boldsymbol{z}_{udk}$ = 1 if there is a deposit of denomination $k \in K$ to the Fed from vault $u \in V$ on day $d \in D$; and $\boldsymbol{z}_{udk}$ = 0, otherwise |

**Table 8 – Artificial Variables**

| Variables | Description |
|---|---|
| $\bar{\phi}_{udk}$ | Number of bricks of denomination $k \in K$ transferred from vault $u \in V$ to a dummy destination on day $d \in D$ in order to generate a feasible solution |
| $\hat{\phi}_{udk}$ | The binary decision variable $\hat{\phi}_{udk}$ = 1 if bricks of denomination $k \in K$ are transferred from vault $u \in V$ to a dummy destination on day $d \in D$; and $\hat{\phi}_{udk}$ = 0, otherwise |
| $\bar{\xi}_{vdk}$ | Number of bricks of denomination $k \in K$ transferred from a dummy source to vault $v \in V$ on day $d \in D$ in order to generate a feasible solution |
| $\hat{\xi}_{vdk}$ | The binary decision variable $\hat{\xi}_{vdk}$ = 1 if bricks of denomination $k \in K$ are transferred from a dummy source to vault $v \in V$ on day $d \in D$; and $\hat{\xi}_{vdk}$ = 0, otherwise |

**Table 9 – De minimus Variables**

| Variables | Description |
|---|---|
| $\bar{c}_k$ | The total number of bricks of denomination $k \in S$ transferred from Fed to vaults during week 1 |
| $\hat{c}_k$ | The total number of bricks of denomination $k \in S$ transferred from Fed to vaults during week 2 |
| $\tilde{c}_k$ | The total number of bricks of denomination $k \in S$ transferred from vaults to Fed (deposits) during week 1 |
| $c'_k$ | The total number of bricks of denomination $k \in S$ transferred from vaults to Fed during week 2 |
| $\bar{b}_k$ | The total number of bricks of denomination $k \in S$ transferred from Fed to vaults during week 1 subject to cross-shipping fee |
| $\hat{b}_k$ | The total number of bricks of denomination $k \in S$ transferred from Fed to vaults during week 2 subject to cross-shipping fee |
| $\tilde{b}_k$ | The total number of bricks of denomination $k \in S$ transferred from vaults to Fed during week 1 subject to cross-shipping fee |
| $b'_k$ | The total number of bricks of denomination $k \in S$ transferred from vaults to Fed during week 1 subject to cross-shipping fee |
| $\bar{a}_k$ | The total number of bricks of denomination $k \in S$ transferred from Fed to vaults during week 1 adjusted by de minimus $I$ |
| $\hat{a}_k$ | The total number of bricks of denomination $k \in S$ transferred from Fed to vaults during week 2 adjusted by de minimus $I$ |
| $\tilde{a}_k$ | The total number of bricks of denomination $k \in S$ transferred from vaults to Fed during week 1 adjusted by de minimus $I$ |
| $a'_k$ | The total number of bricks of denomination $k \in S$ transferred from vaults to Fed during week 2 adjusted by de minimus $I$ |
| $\hbar_k$ | $\hbar_k = 1$ if deposits are smaller than orders for denomination $k \in S$ during week 1; $\hbar_k = 0$, otherwise |
| $\bar{\lambda}_k$ | $\bar{\lambda}_k = 1$ if deposits are smaller than orders for denomination $k \in S$ during week 2; $\bar{\lambda}_k = 0$, otherwise |

**Table 10. Problem Characteristics**

| Problem Instance | # Vaults | # Denominations | Rows | Columns | | | Non-zeroes |
|---|---|---|---|---|---|---|---|
| | | | | Continuous Variables | Binary Variables | Integer Variables | |
| P01 | 1 | 7 | 1253 | 144 | 396 | 268 | 3239 |
| P02 | 3 | 7 | 3787 | 430 | 1180 | 1380 | 11224 |
| P03 | 8 | 7 | 10577 | 1145 | 3329 | 7611 | 57885 |
| P04 | 3 | 5 | 2809 | 376 | 967 | 929 | 10705 |
| P05 | 3 | 5 | 2779 | 340 | 965 | 917 | 10537 |
| P06 | 3 | 5 | 2779 | 340 | 965 | 917 | 10537 |
| P07 | 3 | 7 | 3787 | 430 | 1215 | 1345 | 13757 |

**Table 11. Comparison of Various Strategies**

| Problem Instance | Strategy 1 | | | Strategy 2 | | | Strategy 3 | | | Strategy 4 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Cost ($) | Time to Solve (hh:mm:ss) | Final Gap | Cost ($) | Time to Solve (hh:mm:ss) | Final Gap | Cost ($) | Time to Solve (hh:mm:ss) | Final Gap | Cost ($) | Time to Solve (hh:mm:ss) |
| P04 | 100 | 06:00:00[1] | 34% | 100 | 06:00:00[1] | 30% | 100 | 06:00:00[1] | 13% | 100.0 | 00:09:04 |
| P05 | 100 | 06:00:00[1] | 31% | 100 | 06:00:00[1] | 30% | 100 | 06:00:00[1] | 12% | 100.0 | 00:14:17 |
| P06 | 100 | 06:00:00[1] | 25% | 100 | 06:00:00[1] | 20% | 100 | 03:58:39 | 5% | 100.4 | 00:02:25 |
| P07 | 100 | 06:00:00[1] | 15% | 100 | 06:00:00[1] | 15% | 100 | 06:00:00[1] | 9% | 110.9 | 00:00:02 |

[1]terminated due to a 6 hour CPU time limit

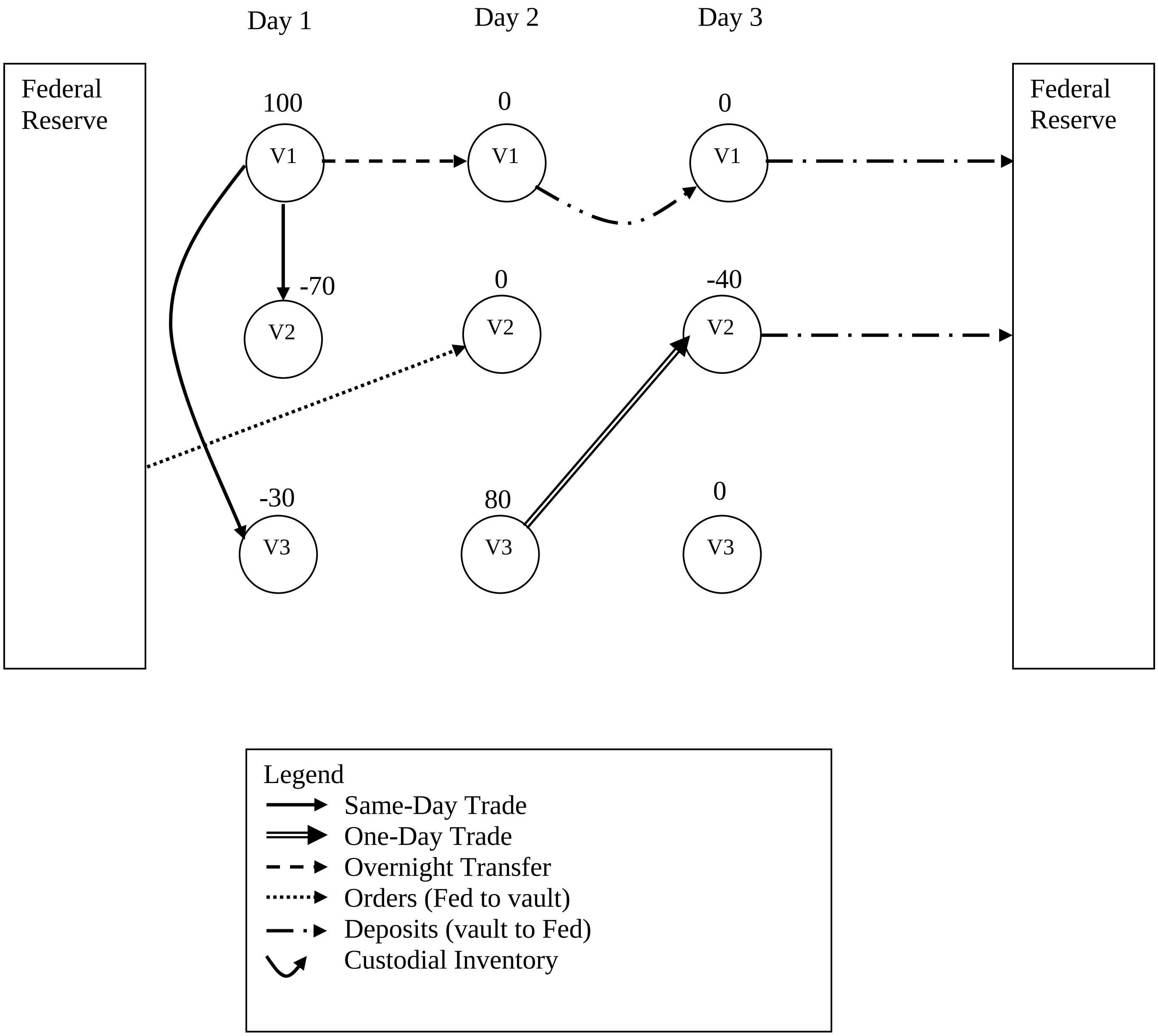


**Figure 1: Time-Space Cash Flow Network**