Fitting random cash management models to data

Francisco Salas-Molina[a,*]

[a] *Universitat de València, Av. Tarongers s/n, 46022 València, Spain*

Abstract

Organizations use cash management models to control balances to both avoid overdrafts and obtain a profit from short-term investments. Most management models are based on control bounds which are derived from the assumption of a particular cash flow probability distribution. In this paper, we relax this strong assumption to fit cash management models to data by means of stochastic and linear programming. We also introduce ensembles of random cash management models which are built by randomly selecting a subsequence of the original cash flow data set. We illustrate our approach by means of a real case study showing that a small random sample of data is enough to fit sufficiently good bound-based models.

Keywords: Machine learning, stochastic programming, data-driven models, ensembles, control bounds.

1. Introduction

A wide range of economic organizations manage cash for operational, precautionary and speculative purposes (Keynes, 1936). Cash management models help decision-makers in their daily job of controlling cash balances. Since the seminal works by Baumol (1952) and Miller and Orr (1966), cash management models follow a inventory control approach. Within this framework, cash balances are allowed to wander around until some control bounds, usually a higher bound and a lower bound, are reached. Then, a control action is made to restore the balance to a given target level. The set of control actions deployed

[*] Corresponding author

Email address: `francisco.salas-molina@uv.es` (Francisco Salas-Molina) over a period of time is called a policy and it is usually determined by simple rules derived from the set of bounds of the model.

The bound-based control approach is based on the strong assumption of a particular probability distribution for cash flows, which is usually assumed to be a normal, independent and stationary cash flow as in Miller and Orr (1966), Baccarin (2009), Premachandra (2004). Surprisingly, the use of empirical data sets in cash management research is limited to recent contributions such as Gormley and Meade (2007) and Salas-Molina et al. (2017), in which alternative forecasters are used to obtain predictions as a key input to cash management models, and Salas-Molina et al. (2016), in which a multiobjective approach to the cash management problem is proposed. We here follow a different data-driven approach. Instead of fitting forecasters as in Gormley and Meade (2007) and Salas-Molina et al. (2017), we here fit cash management models. Furthermore, Gormley and Meade (2007) proposed a bound-based model using forecasts as a key input and they used genetic algorithms to find sufficiently good bounds. On the other hand, Salas-Molina et al. (2016) used simulation techniques and the Miller and Orr (1966) model within a multiobjective framework considering not only the cost but also the risk of alternative policies. In this paper, we first propose a mixed-integer linear program that derives from a general stochastic programming problem to obtain a bound-based model from available data. In order to avoid the computational burden of possibly very large mixed-integer linear programs, we propose a method based on random cash flow subsequences.

We here follow a data-driven approach that mainly focuses on models for cash managers. However, we rely on several machine learning concepts to build our approach. Machine learning covers a wide range of algorithms that can learn from data to make better decisions. A paradigmatic case is deep learning because it requires very little engineering and it can take advantage of computational power and data availability (LeCun et al., 2015, Schmidhuber, 2015). Deep learning models are built through multiple processing layers that are able to discover how computers should modify or adapt their actions to be more precise. On the other hand, random forests are also an interesting technique based on an ensemble of slightly different decision trees (Ho, 1998, Breiman, 2001). The output of the global model is then obtained by combining the output of multiple randomly trained models. In this paper, we use this particular feature to fit random cash management models.

Fitting a model to data means finding the best parameters of the model according to an objective function and a given data set. We here describe a general procedure to fit a set of control bounds to a given data set of cash flows. We make no assumption on the form of the cash flow process under consideration. Then, our approach accepts as a key input a data set with either: (i) past cash flow observations; (ii) cash flows sampled from a particular probability distribution; or (iii) a set of cash flow predictions. Similarly to the ordinary least squares method used in linear regression, we rely on an optimization procedure to produce the solution that minimizes some objective function for a given data set. Instead of a data set with previous examples, we use a data set with cash flow observations. Instead of minimizing the sum of squared deviations, we minimize the sum of costs. And instead of obtaining a set of regression coefficients, we obtain a set of control bounds ready to be used by cash managers.

Relevant related works are those based on stochastic programming (SP) to address different cash management problems as in Golub et al. (1995), Gardin et al. (1995), Gondzio and Kouwenberg (2001) and Castro (2009). In these works, the authors rely on SP and a set of previous realizations of random variables to improve current techniques to manage cash (e.g. in automatic teller machines in the case of Castro (2009)). Thus, this paper is an extension of this body of SP works to fit bound-based models in cash management. A further advantage of our proposal is that the solution provided is a cash management model of the Miller and Orr (1966) type. In other words, we provide cash managers with a set of simple decision rules based on a set of control bounds that proved to be optimal for a given cash flow data set. This fact avoids the drawback of solving a new (possibly large) problem at each time step when information about a new initial condition is available according to the so-called receding horizon philosophy (Bemporad and Morari, 1999, Camacho and Bordons, 2007). In addition, our approach can be extended to fit other cash management models such as the one proposed by Stone (1972) or by Gormley and Meade (2007).

In order to speed up computations, we describe a procedure to randomly select a subsequence of the original cash flow data set to construct an ensemble of random cash management models. The policy to deploy is elicited by averaging the output of randomly trained models similarly to the methods used in machine learning to train random forests (Ho, 1998, Breiman, 2001). To illustrate our approach, we present a case study with real data from an industrial company in Spain. As a benchmark, we use the equations proposed by the Miller and Orr (1966) model to obtain a set of three bounds. The reason to select this model is twofold. First, its relevance. The Miller and Orr model was the first stochastic cash management model and it has become a framework for subsequent research in cash management (some recent examples are Premachandra (2004), da Costa Moraes and Nagano (2014)). Second, its simplicity. While other models propose a higher number of bounds (see e.g. Eppen and Fama (1969), Stone (1972)), the Miller and Orr model is based on only three control bounds allowing us to limit the number of decision variables and constraints for illustrative purposes.

Summarizing, we propose a general methodology to fit cash management models based on control bounds to data as a feasible way to solve the cash management problem (CMP). More precisely, we highlight three main contributions:

1. We provide a method to solve the CMP when using bound-based models without making any assumption on the underlying cash flow process.
2. We construct data-driven cash management models by fitting parameters to data.
3. We introduce ensembles of random cash management models.

This paper is organized as follows. In Section 2, we provide useful background on the Miller and Orr model, the usual cost functions used in cash management and about stochastic programming in cash management. In Section 3, we introduce our method to fit models to cash flow data sets. Next, we present a case study using real data in Section 4. Finally, we conclude in Section 5 suggesting natural extensions of our work.

2. Background

In this section, we first provide useful background on the Miller and Orr model that we later use in a case study to illustrate our data-driven approach. Next, we describe the usual cost functions used in cash management. Finally, we describe a related approach to cash management based on stochastic programming.

2.1. The Miller and Orr model

Consider a cash management system for a typical company as shown in Figure 1. This system comprises two accounts (depicted as circles), a control action x_t between accounts and an external cash flow f_t summarizing both inflows from debtors and outflows to creditors at each time step t. Cash managers can adjust cash balances b_t in account 1 for operational purposes by selling available investments in account 2 through control action x_t at a fixed cost γ_0. On the other hand, idle cash balances in account 1 can be allocated in investment account 2 in exchange for a given return v per money unit when $x_t < 0$. Miller and Orr (1966) proposed a model to control balances for the two-assets system in Figure 1 by assuming that stochastic cash flows f_t are generated by a stationary random walk with standard deviation σ.

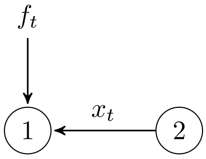

Figure 1: A cash management system with two assets.

Under this framework, cash managers seek to find sequence $x = \{x_1, x_2, \dots, x_\tau\}$ to minimize long-rung average daily cost of managing their cash balances over any planning horizon of τ days given by:

$$\min \left[\gamma_0 \cdot \frac{\mathrm{E}(n)}{\tau} + v \cdot \mathrm{E}(b) \right] \tag{1}$$

where $\mathrm{E}(n)$ is the expected number of transfers $n = \{|x| : x_i \neq 0\}$ during planning horizon τ, and $\mathrm{E}(b)$ is the average daily balance derived from sequence $b = \{b_1, b_2, \dots, b_\tau\}$ containing cash balances at each time step. Following the recommendations in Gormley and Meade (2007), we use an indicator function I_q that takes value one when condition q holds, zero otherwise, to rewrite objective function in equation (1) as follows:

$$\min \left[\frac{\gamma_0}{\tau} \sum_{t=1}^{\tau} I_{x_t \neq 0} + \frac{v}{\tau} \sum_{t=1}^{\tau} b_t \right] \tag{2}$$

subject to the following state transition law with an initial state b_0:

$$b_t = b_{t-1} + f_t + x_t. \tag{3}$$

Miller and Orr (1966) proposed a bound-based model by showing that the optimal policy (when cash flows follow a stationary random walk with standard deviation σ) is obtained by defining three control bounds L, Z and H. These bounds allow determining specific control action x_t and cash balance b_t at each time step t from a previous balance b_{t-1} and an external cash flow f_t. Formally, control action x_t is elicited by comparing the current cash balance derived from previous balance b_{t-1} and actual cash flow f_t to the lower and upper bounds as follows:

$$x_t = \begin{cases} Z - b_{t-1} - f_t, & \text{if} \quad b_{t-1} + f_t > H \\ 0, & \text{if} \quad L < b_{t-1} + f_t < H \\ Z - b_{t-1} - f_t, & \text{if} \quad b_{t-1} + f_t < L. \end{cases} \tag{4}$$

Although Miller and Orr set lower limit L to zero in their work, a real cash manager should set a lower limit above zero for precautionary motives as recommended in Ross et al. (2002). This lower limit represents a safety cash buffer and its selection will depend on the level of risk the company is willing to accept. As a result, when the cash balance reaches L, a positive transfer is made to restore the balance to Z (from account 2 to account 1 in Figure 1). Similarly, when H is reached a negative transfer (from account 1 to account 2 in Figure 1) is made to restore the balance to a target level Z as shown in Figure 2. After setting lower limit L for precautionary purposes and by minimizing objective function (2) with respect to policy x, Miller and Orr (1966) showed that the optimal policy is given by equation (4) with parameters Z and H set as follows:

$$Z = L + \left(\frac{3 \cdot \gamma_0 \cdot \sigma^2}{4 \cdot v} \right)^{1/3} \tag{5}$$

and

$$H = 3 \cdot Z - 2 \cdot L. \tag{6}$$

The reasoning behind the optimality of these bounds requires rewriting objective function (1) in terms of bounds H and Z. By setting $R = H - Z$ and $L = 0$, the problem can be stated in terms of the variance of the net cash flows as:

$$\min_{R,Z} \left\{ \frac{\gamma_0 \sigma^2}{RZ} + \frac{v(R + 2Z)}{3} \right\}. \tag{7}$$

The first term of objective function (7) relates transaction cost γ_0 with the inverse value of the expected duration of a random walk of variance σ starting at level Z and

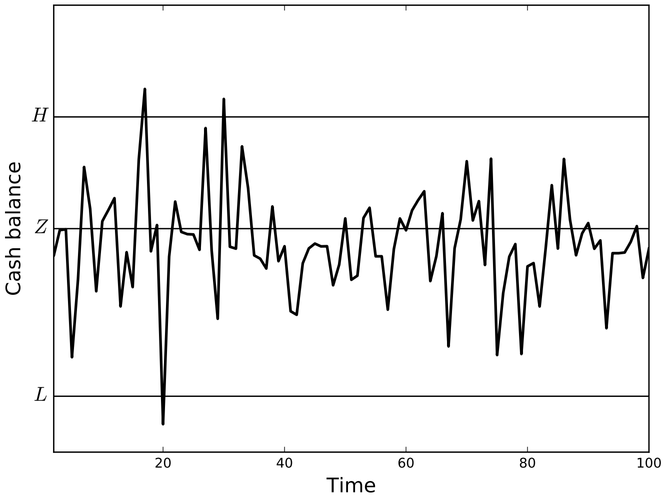

Figure 2: The Miller and Orr model.

ending at bounds H or L. The second term relates the holding cost with the average cash balance given by $(H + Z)/3$. The necessary conditions for a minimum are that the partial derivatives of objective function (7) with respect to R and Z are equal to zero that ultimately lead to the bound expressions described in equations (5) and (6). These results imply that the greater the transfer cost (γ_0), the higher the target cash balance (Z), and the greater the holding cost (v), the lower the target cash balance (Z). However, the greater the uncertainty of net daily cash flows, measured by σ, the higher the target cash balance (Z).

2.2. Holding and transaction costs in cash management

In what follows, we consider a more general approach than Miller and Orr (1966) with respect to cost functions as described in recent cash management works (see e.g. Gormley and Meade (2007), Salas-Molina et al. (2016)). Any positive transaction adding cash to an account may have a cost, which may include a fixed part (γ_0^+) and a variable part (γ_1^+). On the other hand, a negative transaction removing cash from an account may also have a cost with a fixed part (γ_0^-) and a variable part (γ_1^-). Furthermore, at the end of the day, a holding cost (v) per money unit is charged if a positive cash balance occurs, or a penalty cost (u) per money unit is charged if a negative cash balance occurs. According to this cost structure, a general daily cost function is defined as:

$$c(x_t) = \Gamma(x_t) + \Lambda(b_t) \tag{8}$$

where $\Gamma(x_t)$ is a transfer cost function, and $\Lambda(b_t)$ stands for a holding/shortage cost function. The transfer cost function $\Gamma(x_t)$ is defined as:

$$\Gamma(x_t) = \begin{cases} \gamma_0^- - \gamma_1^- \cdot x_t & \text{if} \quad x_t < 0, \\ 0 & \text{if} \quad x_t = 0, \\ \gamma_0^+ + \gamma_1^+ \cdot x_t & \text{if} \quad x_t > 0. \end{cases} \tag{9}$$

Additionally, the holding/shortage cost function $\Lambda(b_t)$ is expressed as:

$$\Lambda(b_t) = \begin{cases} -u \cdot b_t & \text{if} \quad b_t < 0; u > 0, \\ v \cdot b_t & \text{if} \quad b_t \geq 0; v > 0. \end{cases} \tag{10}$$

As a result, cash managers aiming to derive cash management policies need to solve the following program:

$$\min \frac{1}{\tau} \sum_{t=1}^{\tau} \left[(\gamma_0^- - \gamma_1^- \cdot x_t) I_{x_t<0} + (\gamma_0^+ + \gamma_1^+ \cdot x_t) I_{x_t>0} + b_t(v \cdot I_{b_t \geq 0} - u \cdot I_{b_t<0}) \right] \tag{11}$$

subject to transition equation (3). Note that the presence of indicator functions I_q in objective function (11) implies non-linearity since these functions depend on the value of decision variables x_t and b_t. This fact complicates the selection of the best policies. In Section 3, we provide a method to derive cash management policies from data that relies on mixed integer linear programming to overcome this problem.

2.3. Stochastic programing and cash management

Within a general formulation of a stochastic problem (Birge and Louveaux, 2011), we have to make decisions under some degree of uncertainty. These decisions are called first-stage decisions and are usually summarized in a vector $\boldsymbol{x}$ of decision variables. When new information on the realization of some random vector $\boldsymbol{\xi}$ is available, second-stage decisions $\boldsymbol{y}$ are taken. The ultimate goal is to find decisions $\boldsymbol{x}$ that minimize average costs according to the distribution of $\boldsymbol{\xi}$ by means of the so-called stochastic program with recourse as follows:

$$\min_{\boldsymbol{x}} c(\boldsymbol{x}) + \mathrm{E}_{\boldsymbol{\xi}} \left[Q(\boldsymbol{x}, \boldsymbol{\xi}) \right] \tag{12}$$

subject to:

$$A\boldsymbol{x} = \boldsymbol{d} \tag{13}$$

$$\boldsymbol{x} \in \mathbb{R}^{n_1}, \;\; \boldsymbol{x} \geq 0 \tag{14}$$

where:

$$Q(\boldsymbol{x}, \boldsymbol{\xi}) = \min_{\boldsymbol{y}} \{q(\boldsymbol{y}, \boldsymbol{\xi}) \;\mid\; W\boldsymbol{y} = h(\boldsymbol{\xi}) - T(\boldsymbol{\xi})\boldsymbol{x}\} \tag{15}$$

$$\boldsymbol{y} \in \mathbb{R}^{n_2} \tag{16}$$

where $q(\boldsymbol{y}, \boldsymbol{\xi})$, $h(\boldsymbol{\xi})$ and $T(\boldsymbol{\xi})$ are usually linear functions of random variable $\boldsymbol{\xi}$, matrix W is assumed to be fixed, and $\mathrm{E}_{\boldsymbol{\xi}}$ denote mathematical expectation with respect to $\boldsymbol{\xi}$.

Cash management usually involves a sequence of decisions over a given planning horizon. Then, we need to formulate a multistage stochastic problem with τ time steps (Castro, 2009):

$$\min_{\boldsymbol{x}_i} c_1(\boldsymbol{x}_1) + \mathrm{E}_{\xi^2}\left[Q(\boldsymbol{x}, \xi^\tau)\right] \tag{17}$$

subject to:

$$W_1\boldsymbol{x}_1 = \boldsymbol{h}_1 \tag{18}$$

$$W_2\boldsymbol{x}_2(\xi^2) + T_1(\xi^2)\boldsymbol{x}_1 = \boldsymbol{h}_2(\xi^2) \tag{19}$$

$$W_i\boldsymbol{x}_i(\xi^i) + T_{i-1}(\xi^{i-1})\boldsymbol{x}_{i-1} = \boldsymbol{h}_i(\xi^i), \;\; i = 3, 4, \ldots, \tau \tag{20}$$

$$\boldsymbol{x}_1 \geq 0, \;\; \boldsymbol{x}_i(\xi_i) \geq 0, \;\; i = 2, 3, \ldots, \tau, \tag{21}$$

where ξ^i denotes the history of random events up to time step i, W_i $(i = 1, \ldots, \tau)$ are known $m_i \times n_i$ matrices, $\boldsymbol{h}_1$ is a known vector in $\mathbb{R}^{m_1}$, T_{i-1} are $m_i \times (n_i - 1)$ random matrices, $\boldsymbol{h}_i$ are random vectors in $\mathbb{R}^{m_1}$, and $\boldsymbol{x}_i$ $(i = 1, \ldots, \tau)$ are vectors of decision variables at time step i that depends on past events. In words, term $\mathrm{E}_{\xi^2}\left[Q(\boldsymbol{x}, \xi^\tau)\right]$ in objective function (17) summarizes the expected cost from the second time step to the end of planning horizon τ. In addition, constraints (18), (19) and (20) define the state transition between stages. In Section 3, we will see that our data-driven proposal is a special case of the general multistage stochastic problem encoded from equation (17) to (21).

3. A data-driven procedure to fit bound-based models

In this section, we introduce our data-driven stochastic approach to select the set of control bounds that determines the policy to be deployed by cash managers when using bound-based models. We first provide some useful definitions; next, we describe a novel method to fit bound-based cash management models; and finally, we introduce the concept of generalization power of cash management models.

3.1. Some useful definitions

As described in Section 2.2, cash managers make decisions under some economic context.

Definition 1. A **cost structure** α is a tuple $\left(\gamma_0^+, \gamma_0^-, \gamma_1^+, \gamma_1^-, v, u\right)$ defining the transaction and holding costs for a given context.

Within this context, we assume that cash managers have been able to observe the external net cash flows during a given period of time. Then, a data set of observed net cash flows is available as a key input to derive a cash management model.

Definition 2. Given a cash flow data set $\boldsymbol{f} = \{f_t : t = 1, 2, \ldots, N\}$ of size $N \in \mathbb{N}$, an initial condition b_0, and a cost structure α, a **cash management model** $h(\boldsymbol{f}, b_0, \alpha)$ returns a policy that optimizes some objective function.

By considering alternative cash flow data sets, cash managers can design a number of different cash management models to define a combined policy.

Definition 3. An **ensemble of cash management models** $\{h_k : k = 1, 2, \ldots, K\}$ of size $K \in \mathbb{N}$ is a collection of models that returns a policy by combining the policy derived from each model.

By randomly selecting a part of a larger data set, cash managers can use alternative cash flow data sets to build a collection of models.

Definition 4. A **random cash management set** (RCMS) is an ensemble of models where each model $h_k(b_0, \boldsymbol{f}_k, \alpha)$ is trained by randomly selecting a subsequence $\boldsymbol{f}_k$ belonging to a larger cash flow sequence $\boldsymbol{f}$.

Given an initial condition b_0, a context α, and a cash flow data set $\boldsymbol{f}$, we next present method to fit a model of the Miller and Orr type as an instance of a general model $h(\boldsymbol{f}, b_0, \alpha)$.

3.2. Fitting bound-based cash management models

Recall from the introduction that we here follow a data-driven approach similar to those used in machine learning to fit models to data. The underlying idea behind the proposed methodology is in line with the main stream of statistical learning approaches ranging from regression analysis to non-parametric estimation. Similarly to linear regression depicted in Figure 3 (a), we here use mixed integer linear programming to produce the optimal bound-based model that minimizes the sum of holding and transactions costs introduced in Section 2.2 for a given data set of cash flows. The most remarkable difference is that instead of obtaining a set of regression coefficients, we obtain a set of control bounds ready to be used by cash managers to deploy a control policy as shown in Figure 3 (b).

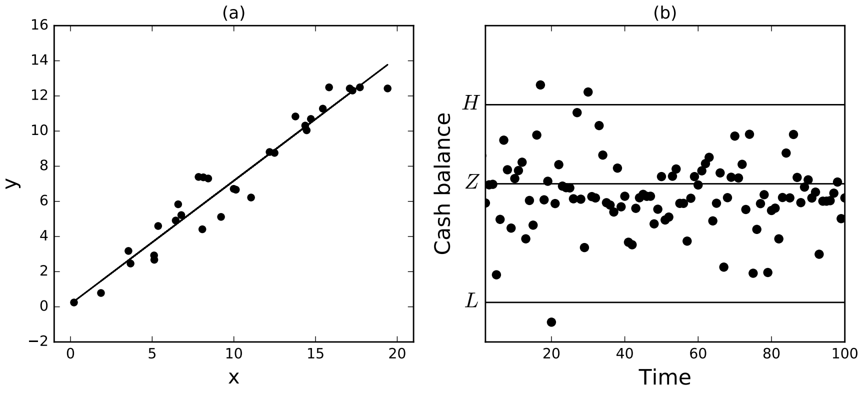

Figure 3: (a) Fitting a linear model to data; (b) Fitting a bound-based model to data.

In order fit a model $h(\boldsymbol{f}, b_0, \alpha)$, we next reformulate objective function (11) with the sum of holding and transaction costs as a mixed integer linear function. To this end, we transform the common two-assets setting shown in Figure 1 into an equivalent configuration as depicted in Figure 4. Let x_t be the difference between control actions $x_t = x_t^+ - x_t^-$ at account 1, with x_t^+ and x_t^- being non-negative real numbers. In this setting, the transfer cost function in equation (9) can be expressed as follows:

$$\Gamma(x_t) = \gamma_0^+ \cdot z_t^+ + \gamma_1^+ \cdot x_t^+ + \gamma_0^- \cdot z_t^- + \gamma_1^- \cdot x_t^- \tag{22}$$

where $z_t^+, z_t^- \in \{0, 1\}$ are binary auxiliary variables satisfying:

$$z_t^+ + z_t^- \leq 1 \tag{23}$$

$$x_t^+ \leq M \cdot z_t^+ \tag{24}$$

$$x_t^- \leq M \cdot z_t^- \tag{25}$$

where M is a very large number. A similar approach can be followed to linearize the holding/penalty cost function in equation (10). For simplicity, we assume $u = \infty$ and $\Lambda(b_t) = v \cdot b_t$ to restrict ourselves to the usual situation in which cash managers discard policies with negative balances due to high penalty costs. Then, we can rewrite the cost function in equation (8) as follows:

$$c(x_t) = \gamma_0^+ \cdot z_t^+ + \gamma_1^+ \cdot x_t^+ + \gamma_0^- \cdot z_t^- + \gamma_1^- \cdot x_t^- + v \cdot b_t. \tag{26}$$

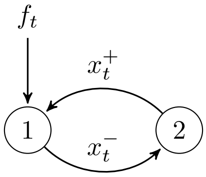

Figure 4: The common two-assets setting in the cash management problem.

Furthermore, we must also rewrite the law of motion in equation (3) as:

$$b_t = b_{t-1} + f_t + x_t^+ - x_t^-. \tag{27}$$

According to the Miller and Orr policy described in equation (4), positive transactions x_t^+ occur when bound L is reached. Thus, $z_t^+ = 1$ when $b_{t-1} + f_t \leq L$, and the amount transferred is given by $x_t^+ = Z - b_{t-1} - f_t$. This can be expressed by the following linear constraints:

$$b_{t-1} + f_t - L \leq M(1 - z_t^+) \tag{28}$$

$$-M(1 - z_t^+) \leq x_t^+ - Z + b_{t-1} + f_t \leq M(1 - z_t^+). \tag{29}$$

Furthermore, negative transactions x_t^- occur when bound H is reached. Thus, $z_t^- = 1$ when $b_{t-1} + f_t \geq H$, and the amount transferred is given by $x_t^- = b_{t-1} + f_t - Z$. This can be expressed by the following linear constraints:

$$H - b_{t-1} - f_t \leq M(1 - z_t^-) \tag{30}$$

$$-M(1 - z_t^-) \leq x_t^- + Z - b_{t-1} - f_t \leq M(1 - z_t^-). \tag{31}$$

A third group of conditions must hold when the cash balance is between bounds L and H. Thus, when $z_t^+ = 0$ and $z_t^- = 0$, no transaction occurs. This can be expressed by the following linear constraints:

$$L - b_{t-1} - f_t \leq M(z_t^+ + z_t^-) \tag{32}$$

$$b_{t-1} + f_t - H \leq M(z_t^+ + z_t^-) \tag{33}$$

$$-M \cdot z_t^+ \leq x_t^+ \leq M \cdot z_t^+ \tag{34}$$

$$-M \cdot z_t^- \leq x_t^- \leq M \cdot z_t^-. \tag{35}$$

As a result, given an initial cash balance b_0 and a sequence of cash flow observations $\{f_t : t = 1, 2, \ldots, N\}$ as a given data set, we are in a position to elicit the set of optimal control bounds for policies of the Miller and Orr type by solving the following mixed integer linear program:

$$\min \sum_{t=1}^{N} \left[\gamma_0^+ \cdot z_t^+ + \gamma_1^+ \cdot x_t^+ + \gamma_0^- \cdot z_t^- + \gamma_1^- \cdot x_t^- + v \cdot b_t\right] \tag{36}$$

subject to:

$$b_t = b_{t-1} + f_t + x_t^+ - x_t^- \tag{37}$$

$$b_{t-1} + f_t - L \leq M(1 - z_t^+) \tag{38}$$

$$-M(1 - z_t^+) \leq x_t^+ - Z + b_{t-1} + f_t \leq M(1 - z_t^+) \tag{39}$$

$$H - b_{t-1} - f_t \leq M(1 - z_t^-) \tag{40}$$

$$-M(1 - z_t^-) \leq x_t^- + Z - b_{t-1} - f_t \leq M(1 - z_t^-) \tag{41}$$

$$L - b_{t-1} - f_t \leq M(z_t^+ + z_t^-) \tag{42}$$

$$b_{t-1} + f_t - H \leq M(z_t^+ + z_t^-) \tag{43}$$

$$-M \cdot z_t^+ \leq x_t^+ \leq M \cdot z_t^+ \tag{44}$$

$$-M \cdot z_t^- \leq x_t^- \leq M \cdot z_t^- \tag{45}$$

$$z_t^+ + z_t^- \leq 1 \tag{46}$$

$$x_t^+ \leq M \cdot z_t^+ \tag{47}$$

$$x_t^- \leq M \cdot z_t^- \tag{48}$$

$$b_t \geq b_{min} \tag{49}$$

where the final decision variables are bounds L, Z and H that determine the optimal control policy. At each time step, four additional auxiliary decision variables, two real (x_t^+ and x_t^-) and two binary (z_t^+ and z_t^-) are necessary to solve the problem. Following the recommendations in Ross et al. (2002) about the Miller and Orr model, we also set a minimum cash balance for precautionary purposes. In practice, setting this value b_{min} is equivalent to set a lower limit for bound L.

Note that the problem encoded from equation (36) to (49) is a special case of the multistage stochastic problem described from (17) to (21) adapted to the characteristics of the Miller and Orr (1966) model where random variable $\xi_i = f_t$. On the one hand, objective function (36) is the expected cost over some time interval which is equivalent to add the cost of first decision $c_1(x_1)$ to the expected cost of the rest of decisions summarized in $\mathrm{E}_{\xi^2}\left[Q(\boldsymbol{x}, \xi^\tau)\right]$ from equation (17). On the other hand, balance transition equations such as (37) or constraint (38) relate current state decision variables with previous ones as in constraint (20). As an illustrative example, if vectors $\boldsymbol{x}_i = \left[b_t, x_t^+, x_t^-, z_t^+, y_t\right]'$ and $\boldsymbol{x}_{i-1} = \left[b_{t-1}, x_{t-1}^+, x_{t-1}^-, z_{t-1}^+, y_{t-1}\right]'$ summarize two subsets of decision variables at two consecutive time steps as in equation (20), where y_t and y_{t-1} are additional non-negative auxiliary variables used to transform an inequality into an equality, we can set W_i, T_{i-1} and $\boldsymbol{h}_i$ as follows:

$$W_i = \begin{bmatrix} 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & M & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} ; T_{i-1} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} ; \boldsymbol{h}_i = \begin{bmatrix} f_t \\ M - f_t + L \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{50}$$

to obtain equation (37) by computing:

$$W_i \boldsymbol{x}_i + T_{i-1} \boldsymbol{x}_{i-1} = \boldsymbol{h}_i. \tag{51}$$

By considering the complete vector of decision variables, we can follow a similar reasoning to represent the rest of constraints of the formulation encoded from equations (36) to (49) as a general stochastic program such as the one described in Section 2.3 through the use of larger matrices W_i, T_{i-1} and vectors $\boldsymbol{h}_i$, which we here omit for economy of space.

The model proposed in this section is based on an observed cash flow data set as a possible realization of an underlying cash flow process. Then, the solution to the program is the best Miller and Orr model in terms of cost adjusted to the observed cash flow. Intuitively, the goodness of the model depends on how representative of the underlying cash flow process is the data used to fit the model. The higher the number of observations N included in the data set used to fit the model, the higher the probability that the model better captures the real characteristics of the underlying cash flow process. For instance, if the sample volatility is a good estimation of the real volatility, the model will handle well cash flows derived from the real process. The rationale is the same that it is behind modern machine learning techniques aiming to obtain models that generalize well the underlying process. However, a balance between representativeness of the model and computational efficiency must be considered.

However, it is important to highlight that the size N of the available data set may limit the utility of this procedure for computational reasons. Due to the presence of binary variables in the minimization problem encoded from equation (36) to (49), computation times may result prohibitive for large data sets. In order to mitigate this effect, we propose ensemble methods and random subsequences to fit cash management models as formally introduced in Definitions 3 and 4. Selecting random subsequences of the input data space has been fruitfully used to construct decision tree models (Ho, 1998, Breiman, 2001). Furthermore, ensemble methods are learning algorithms that construct a set of models and then predict by taking a possibly weighted average of their predictions (Dietterich, 2000). Note that the random selection of a subsequence from a time indexed cash flow may be performed by different methods. Taking a uniform sample with replacement as in bagging (Breiman, 1996), or weighting the observations to produce a biased selection similarly to boosting (Freund and Schapire, 1996), are suitable procedures.

3.3. Generalization power of cash management models

Another key feature in machine learning is the appropriateness of a particular predictive model to a given data set. The goodness of fit is usually evaluated by the predictive accuracy that refers to how well the model is able to reproduce the data used to fit the model (Makridakis et al., 2008). In this paper, we follow the approach of measuring how well a particular cash management model fits to a given data set by computing the sum of holding and transaction costs. Furthermore, we propose to measure the utility of any model derived from the optimization problem described in Section 3.2 in comparison to a benchmark model. An interesting benchmark model is the trivial strategy of taking no control action. By comparing performances of models to trivial strategies, we are implicitly checking if cash management models are worthwhile along the lines of Daellenbach (1974). We can also use alternative cash management models as a benchmark. In the following case study, we use the equations proposed by the Miller and Orr model described in Section 2 for benchmarking purposes.

As a result, given a cash flow data set $\boldsymbol{f}$, we define the goodness of fit $G(\boldsymbol{f})$ as:

$$G(\boldsymbol{f}) = \frac{C(\boldsymbol{f})}{C_0(\boldsymbol{f})} \tag{52}$$

where $C(\boldsymbol{f})$ is the total cost of the fitted model over data set $\boldsymbol{f}$, and $C_0(\boldsymbol{f})$ is the total cost of the benchmark model over the same data set. The lower the value of G, the better the model with respect to the benchmark. However, we are usually more interested in the generalization power of models when dealing with data not used to fit the model. In machine learning, the generalization power of predictive models is estimated by the accuracy of the model in terms of the deviation of forecasts from actual values in a test data set not used to fit the model (Makridakis et al., 2008, Provost and Fawcett, 2013). To this end, the existing data set is usually split in a training set to fit the model (in-sample data) and a test set to evaluate the model (out-of-sample data). Next, we estimate the generalization power of a cash management model by computing its relative performance with respect to a benchmark model over a test set:

$$G(\boldsymbol{f}_{test}) = \frac{C(\boldsymbol{f}_{test})}{C_0(\boldsymbol{f}_{test})} \tag{53}$$

where $\boldsymbol{f}_{test}$ is a cash flow data set not used to fit the model.

A more sophisticated technique for estimating the generalization power is cross-validation. Unlike training and test set splitting, cross-validation estimates the generalization power of a model by performing multiple splits (Hastie et al., 2009, Provost and Fawcett, 2013). This method estimates the generalization power of the model across the available data set obtaining useful statistics such as the mean and variance of the expected performance.

4. Case study

In this section, we describe a case study based on a real cash flow data set from an industrial company in Spain that has been recently used in Salas-Molina et al. (2017). This data set contains 2717 daily net cash flows on working days covering a period of more than ten years with mean 0.009 and a high variability 0.097 in terms of standard deviation, both figures in millions of euros. In what follows, we first describe a method to estimate the generalization power of the RCMS defined in Section 3.1. Next, we study the impact of alternative economic contexts on generalization power, and finally, we explore the influence of the number of cash flow observations to fit a RCMS. All the experiments in this case study are performed on Jupyter Notebooks executed on a CPU Intel Core Duo E8400 at 3 GHz with 4 GB of RAM under operating system Windows 10 Professional 64 bits. Mathematical programs are solved through the Python interface of Gurobi optimization software (Gurobi Optimization, Inc, 2017).

4.1. Estimating the generalization power of random cash management models

Using the common 80/20 % split to produce a training and a test set would force us to solve a mixed integer linear program with almost 8700 decisions variables. One of the main purposes of proposing ensembles of models such as the RCMS introduced in Section 3.2 is to obtain a ready-to-use model without solving the extensive problem encoded from equation (36) to (49) for a large sequence of observations $f_1, f_2, \ldots, f_N$ when N is large as it is the case of our data set with $N = 2717$ cash flows. In order to support this approach with computational results, we next evaluate efficiency in terms of the required computing time to solve the problem for a range of different size samples in which $n \ll N$ holds. The results obtained discouraged us from testing larger values of n since we think that longer computing times are not acceptable in practice from a cash management point of view.

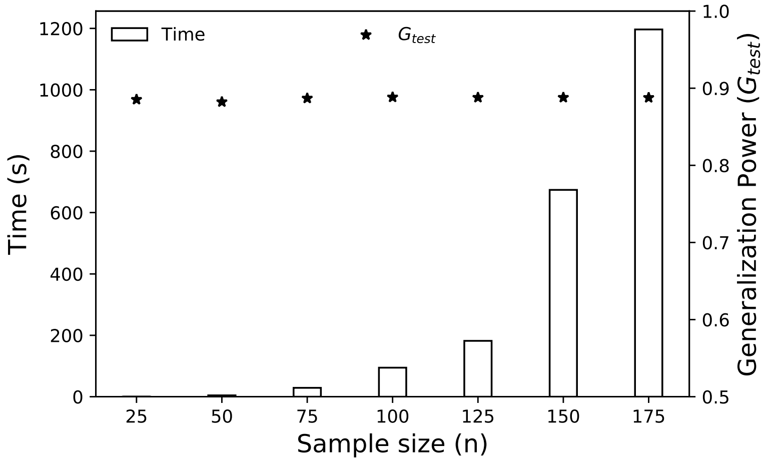

Figure 5: Run time for different size samples.

In Figure 5, we show the average run time required to solve real instances of problem (36)-(49) using a state-of-the-art integer optimization solver such as Gurobi for different size samples. We also depict generalization power $G(\boldsymbol{f}_{test})$ computed using expression (53) to evaluate the trade-off between efficiency and generalization power of the RCMS. The results in Figure 5 show that run times rapidly increase with the number of observations used to fit the model. However, the generalization power of models remains remarkably stable. As a result, we can reasonably infer that solving the extensive formulation for $N = 2717$ observations would lead to prohibitive computing times. To overcome this drawback, we follow the strategy of using a relatively low number of observations fit a good bound-based model. Thus, we here recommend the use of RCMS as a suitable method to find a balance between efficiency and generalization power of bound-based models. In what follows, we use RCMS to fit a cash management model of the Miller and Orr type to the data provided by the company. To this end, we proceed as detailed in Algorithm 1.

Algorithm 1: Generalization power estimation for cash management models

```
Input: Cash flow data set $\boldsymbol{f}$; initial balance $b_0$; number $n$ of observations to fit
  the model; number of randomly trained models $K$; cost structure $\alpha$; benchmark
  model $h_0$; and training/test set rate $r$;
Output: Generalization power for a cash management model;
Split $\boldsymbol{f}$ in $\boldsymbol{f}_{train}$ and $\boldsymbol{f}_{test}$ according to $r$;
for $k = 1, 2, \ldots, K$ do
    Randomly sample $n$ observations from $\boldsymbol{f}_{train}$ to obtain $\boldsymbol{f}_k$;
    Estimate the model $h_k(b_0, \boldsymbol{f}_k, \alpha)$ by solving program (36)-(49) with $N = n$;
end
Obtain model $h(b_0, \boldsymbol{f}_{train}, \alpha)$ by averaging parameters of $h_k$;
Compute $G(\boldsymbol{f}_{test}) = C(\boldsymbol{f}_{test})/C_0(\boldsymbol{f}_{test})$ from the cost of deploying $h$ and $h_0$;
```

Note that Algorithm 1 can be used for a single generalization power estimate, but it can be replicated as many times as needed for cross-validation. In this case study, we use a training set wit the first 80% of the observations and a test set with the remaining 20% since we aim to test the utility of the model with the most recent data. As an illustrative example, let us consider the following cost structure selected from those proposed in da Costa Moraes and Nagano (2014):

$$\alpha_1 = \left(\gamma_0^+ = 20€, \gamma_0^- = 20€, \gamma_1^+ = 0.01\%, \gamma_1^- = 0.01\%, v = 0.02\%, u = \infty\right). \tag{54}$$

In this case study, we use as a benchmark the Miller and Orr model derived from the application of equations (5) and (6) from Section 2. According to the recommendations in Ross et al. (2002), we set a lower bound L for precautionary purposes. A suitable way to do it is setting a value proportional to the empirical standard deviation of cash flows along the lines of Ben-Tal and Nemirovski (1999), Ben-Tal et al. (2009) for robust optimization. Then, we set:

$$L = \delta \cdot \sigma \tag{55}$$

where σ is the empirical standard deviation of the cash flow data set and δ is a parameter reflecting the attitude towards risk of cash managers so that the higher the value of δ, the more averse to risk they are. Let us consider as abnormal cash flows those with absolute value above five standard deviations as recommended in Gormley and Meade (2007). Then, we set $\delta = 5$ for precautionary purposes in order to ensure that only abnormal cash flows (only 0.25% of the observations in this data set) may result in a negative cash balance.

In a first numerical example, we aim to compare the generalization power achieved by our RCMS from Section 3.1 with respect to the Miller and Orr equations described in Section 2. To this end, we first split the whole cash flow data set in a training set with the first 80% of the observations and a test set with the remaining 20%. From the standard deviation of cash flows in the training set and cost structure α_1 in tuple (54), we obtain a Miller and Orr model using equations (55), (5) and (6) to obtain $(L_0, Z_0, H_0) = (0.48, 0.57, 0.75)$, figures in millions of Euros. Starting at an initial stable cash balance for both models equal to Z from equation (5), we use Algorithm 1 with $K = 20$ and $n = 25$ to obtain a RCMS with $(L, Z, H) = (0.46, 0.50, 0.64)$ and $G(\boldsymbol{f}_{test}) = 0.88$. Since the generalization power is below one, our RCMS performs better than the benchmark. Note also that only 25 samples from the training set and 20 randomly trained models are enough for our RCMS to reduce the cost with respect to the Miller and Orr equations evaluated over the test set.

4.2. Generalization versus economic context

Cash managers may be interested in analyzing the impact of alternative economic contexts on the generalization power of RCMS. In this section, we compare the generalization power of a RCMS (with $K = 20$ and $n = 25$) to the Miller and Orr benchmark by applying Algorithm 1 to different cost structures. In addition to cost structure α_1 described in tuple (54), we present in Table 1 the results obtained for different combinations of holding and transaction costs. More precisely, we consider the case when: fixed transaction costs are doubled (α_2); variable transaction costs are set to zero (α_3); fixed transaction costs are doubled and variable transaction costs are set to zero (α_4); holding costs are doubled (α_5); both fixed transaction and holding costs are doubled and variable transaction costs are set to zero (α_6); variable transaction costs are doubled (α_7); both fixed and variable transaction costs and also holding costs are doubled (α_8). As in the example in Section 4.1, we assume $u = \infty$, for all contexts in Table 1.

Table 1: Generalization power of a RCMS with $K = 20$ and $n = 25$ for different economic contexts

Context	γ_0^+(€)	γ_0^-(€)	γ_1^+(%)	γ_1^-(%)	v(%)	$G(\boldsymbol{f}_{test})$
α_1	20	20	0.01	0.01	0.02	0.88
α_2	40	40	0.01	0.01	0.02	0,89
α_3	20	20	0	0	0.02	0,85
α_4	40	40	0	0	0.02	0,85
α_5	20	20	0.01	0.01	0.04	0,88
α_6	40	40	0	0	0.04	0,86
α_7	20	20	0.02	0.02	0.02	0,89
α_8	40	40	0.02	0.02	0.04	0,88

According to equations (52) and (53), the lower the value of $G(\boldsymbol{f}_{test})$, the better the performance of the RCMS with respect to the Miller and Orr benchmark. Since the generalization power is quite stable, we can conclude that the impact of changes in the economic context is reduced.

4.3. Generalization versus number of observations used to fit a RCMS

An interesting additional exercise consists in determining the minimum number of samples that allow our RCMS to improve the generalization power of the benchmark. We may be also interested in the number of samples from which our RCMS does not produce any further improvement or even worsen its performance. Note that in the limit, when the size of the sample approaches the size of the training set, the performance of our RCMS and the Miller and Orr benchmark should be very similar. We answer these questions by means of a learning plot mapping the number of samples n used to train our RCMS to the generalization power G over the test set.

In Figure 6, we represent the generalization power for ensembles of 20 randomly trained cash management models for different small sample sizes of the training set in steps of two. Sizes below 16 produce results much worse than the benchmark that we remove from the plot for scale reasons. From the analysis of this plot, we can highlight three interesting points: (i) our RCMS consistently outperforms the Miller and Orr benchmark for a wide range of small samples; (ii) the best sample size seems to be around sizes from 18 to 22 observations; and (iii) there is a slight increasing trend in the generalization power with the sample size.

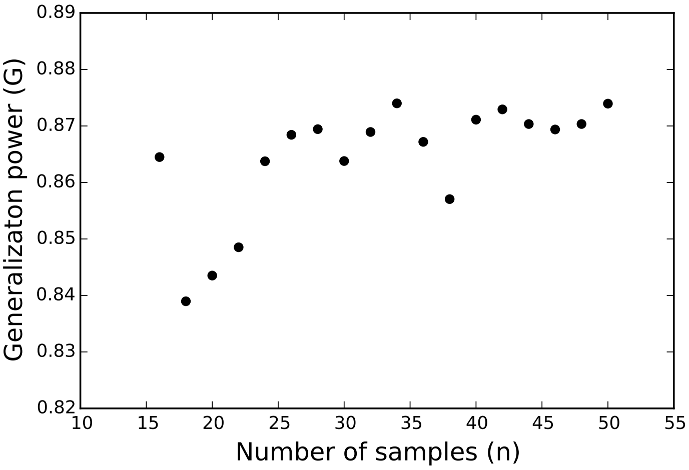

Figure 6: Learning plot for random cash management models.

5. Concluding remarks

Organizations can leverage optimization models to improve data-driven decision-making in finance. Most cash management models are based on the assumption of a particular underlying cash flow process usually assumed to be independent, stationary and Gaussian. In this paper, we relax this strong assumption to fit cash management models to data by relying on both recent machine learning techniques and mathematical programming.

In an attempt to extract useful knowledge from data, we describe a general procedure based on mixed integer linear programs derived from a general stochastic programming approach to elicit the best bound-based model from a given cash flow data set. Our approach is suitable for a wide range of organizations since we use net cash flows that summarize an arbitrary number of flows in a single figure per time step as an input to the model. To reduce the computational effort required to solve large mixed integer linear programs, we also introduce the concept of ensembles of random cash management models. These ensembles are built by randomly selecting a subsequence of the original cash flow data set. Interestingly, the results from a case study using real data show that a small sample size is enough to fit better bound-based models than a benchmark based on the whole training set. These results must encourage cash managers to find a balance between computational effort and generalization power of cash management models deployed to deal with real cash flows.

It is also important to highlight that our approach can be applied to a variety of data sets including past cash flow observations, draws from a particular probability distribution or even forecasts. Summarizing, we show that stochastic and mixed integer linear programming can be used to fit cash management models to data as way to solve the cash management problem without making any assumption on the available data. Natural extensions of our work may explore more sophisticated methods to randomly train cash management models.

References

Baccarin, S. (2009). Optimal impulse control for a multidimensional cash management system with generalized cost functions. *European Journal of Operational Research*, 196(1):198–206.

Baumol, W. J. (1952). The transactions demand for cash: An inventory theoretic approach. *The Quarterly Journal of Economics*, 66(4):545–556.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.

Ben-Tal, A. and Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13.

Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Camacho, E. F. and Bordons, C. (2007). *Model predictive control*. Springer.

Castro, J. (2009). A stochastic programming approach to cash management in banking. *European Journal of Operational Research*, 192(3):963–974.

da Costa Moraes, M. B. and Nagano, M. S. (2014). Evolutionary models in cash management policies with multiple assets. *Economic Modelling*, 39:1–7.

Daellenbach, H. G. (1974). Are cash management optimization models worthwhile? *Journal of Financial and Quantitative Analysis*, 9(04):607–626.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer.

Eppen, G. D. and Fama, E. F. (1969). Cash balance and simple dynamic portfolio problems with proportional costs. *International Economic Review*, 10(2):119–133.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156.

Gardin, F., Power, R., and Martinelli, E. (1995). Liquidity management with fuzzy qualitative constraints. *Decision Support Systems*, 15(2):147–156.

Golub, B., Holmer, M., McKendall, R., Pohlman, L., and Zenios, S. A. (1995). A stochastic programming model for money management. *European Journal of Operational Research*, 85(2):282–296.

Gondzio, J. and Kouwenberg, R. (2001). High-performance computing for asset-liability management. *Operations Research*, 49(6):879–891.

Gormley, F. M. and Meade, N. (2007). The utility of cash flow forecasts in the management of corporate cash balances. *European Journal of Operational Research*, 182(2):923–935.

Gurobi Optimization, Inc (2017). Gurobi optimizer reference manual.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning*. Springer New York.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence*, 20(8):832–844.

Keynes, J. M. (1936). *General theory of employment, interest and money*. Atlantic Publishers & Dist.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Makridakis, S., Wheelwright, S. C., and Hyndman, R. J. (2008). *Forecasting methods and applications*. John Wiley & Sons.

Miller, M. H. and Orr, D. (1966). A model of the demand for money by firms. *The Quarterly Journal of Economics*, 80(3):413–435.

Premachandra, I. (2004). A diffusion approximation model for managing cash in firms: An alternative approach to the miller–orr model. *European Journal of Operational Research*, 157(1):218–226.

Provost, F. and Fawcett, T. (2013). *Data Science for Business: What you need to know about data mining and data-analytic thinking*. O'Reilly Media, Inc.

Ross, S. A., Westerfield, R., and Jordan, B. D. (2002). *Fundamentals of corporate finance*. McGraw-Hill, sixth edition.

Salas-Molina, F., Martin, F. J., Rodriguez-Aguilar, J. A., Serra, J., and Arcos, J. L. (2017). Empowering cash managers to achieve cost savings by improving predictive accuracy. *International Journal of Forecasting*, 33(2):403–415.

Salas-Molina, F., Pla-Santamaria, D., and Rodriguez-Aguilar, J. A. (2016). A multi-objective approach to the cash management problem. *Annals of Operations Research*, pages 1–15.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Stone, B. K. (1972). The use of forecasts and smoothing in control-limit models for cash management. *Financial Management*, 1(1):72–84.