ARTICLE IN PRESS

Omega ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Sound branch cash management for less: A low-cost forecasting algorithm under uncertain demand ☆

Julia García Cabello [a,*], F.J. Lobillo [b]

[a] Dpto. de Matemática Aplicada, Facultad Ciencias Económicas y Empresariales, Universidad de Granada, Campus de Cartuja, Granada 18071, Spain
[b] CITIC and Dpto. de Álgebra. Universidad de Granada, Spain

ARTICLE INFO

Article history:
Received 6 November 2015
Accepted 13 September 2016

Keywords:
Cash management at banking branch-level
Real bank data processing
Demand forecasting algorithm
Stochastic processes

ABSTRACT

This paper deals with cash management for bank branches, under the assumption that branches have a role to play in the improvement of global bank institution performance. In the current scenario of unprecedented pressure amongst banks to keep costs under control, our contribution is the design of a sound and low-cost algorithm to optimize branch cash holdings using software implementation in SageMath. It is accompanied by data processing based on 60,000 real banking records. This is the first academic paper to run such an extensive database at branch level.

We find that our algorithm by and large performs well when forecasting the cash amounts that the branch might require from the central hub to satisfy all branch necessities, avoiding the generation of either surplus or shortage of cash. It is also extremely easy to implement in daily branching practice, leading to an overall reduction in operating costs. In addition, our algorithm may be easily adjusted as required and be tailor-made to the special requirements of each banking institution.

1. Introduction

The banking industry has, throughout History, constantly searched for accurate management tools to improve its performance. Additionally, in the current scenario of economic crisis, bank entities are under unprecedented pressure to keep costs under control, while improving customer service. In this situation of fierce competition, where improving efficiency seems to be a primary objective for banks, there is a body of research that argues that bank *branches* have a role to play through the assumption that *branching* efficiency significantly helps improve the performance of a global bank institution: see [8,13,30].

The importance of bank branches is increasingly recognized, in particular due to their potential in developing customer relationships as well as being one of the most effective sales channels for bank institutions. Although self-service banking (ATMs) and internet banking offer customers convenient real-time access, bank branches provide a more convenient and people-friendly service. This local customer service continues to fulfil a critical role in new customer acquisition and cross-sales, in particular for more complex financial products, while it is the preferred way of doing business of customers of different ages.[1] The increased competition amongst bank entities "has engaged in a proactive, differentiated and customer-based strategy on the retail side where the sales component of *branch* activity is emphasized" [11]. According to this philosophy, Ioannou and Mavri [22] present a decision support system in order to reconfigure and improve the bank branch network. Other evidence of the current importance of bank branches is the growing supply of proposals from financial/consulting services that are focused on bank branches. For instance, in [28], IBM addresses the introduction of banking software in new markets (Chinese banking) by boosting Chinese banking development through the improvement of the operational efficiency of Chinese bank branches.

When attempting to improve banking performance, an efficient cash management is crucial. Why? On one hand, since liquidity comes at a cost, banks might decide how much liquidity is enough while avoiding dormant liquid sources. In this sense, sound cash management reduces financing costs and optimizes the return on the cash position. On the other hand, the importance of designing methods of improving cash management relies on the increased interest of accurate liquidity management for banks in the actual context of

☆This manuscript was processed by Associate Editor Rosenberger.

* Corresponding author.
E-mail addresses: cabello@ugr.es (J. García Cabello), jlobillo@ugr.es (F.J. Lobillo).

[1] In the U.S., commercial banking enhances its competitiveness against non-bank financial institutions by selling financial advice products through bank branches.

http://dx.doi.org/10.1016/j.omega.2016.09.005
0305-0483/

Please cite this article as: García Cabello J, Lobillo FJ. Sound branch cash management for less: A low-cost forecasting algorithm under uncertain demand. Omega (2016), http://dx.doi.org/10.1016/j.omega.2016.09.005

uncertainty and instability ([1] or [14]). Actually, the recent financial crisis has demonstrated the financial system's fragility *if banks do not have sufficient safety liquidity levels*. Thus, an efficient cash management is aimed at helping banks to provide a cushion of capital – available to cover losses of any kind – in order to comply with those regulatory reforms which set the *safety liquidity levels* that banks must attain. Amongst this legislation, importantly, Basel III rules, with two minimum ratios, "Liquidity Coverage Ratio" (LCR) which is a kind of stress test, and "Net Stable Funding Ratio" (NSFR) which tries to ensure that a bank's assets would be adequately supported by stable funding sources. Authors who agree on the key role that a precise cash management plays in the general health of banking institutions are [12,15,24], amongst others.

The present paper may be placed within the literature related to the design of managerial measures for improving *branch* cash management. This paper attempts to provide a new tool to upgrade cash management at daily branch level by improving their cash forecasting processes. At present, a precise liquidity management *forecasting* plays an essential role, since deciding which part of the liquid sources should be kept as ready money, as opposed to being invested in other products, is not trivial at all. On the one hand, banking institutions might have some money stock in order to face short-term obligations at the aggregate and branch level, which should prevent risking bankruptcy in long-term projections. On the other hand, when banking firms keep liquid assets in cash, they give up a part of their profitability, i.e., the opportunity costs of not investing in other alternatives which generate health. Throughout the literature, models of cash management (amongst other approaches) have attempted to solve this problem.

The branch total amount of ready cash is known as cash holdings. Such branch liquid assets are the sum of (a) the deposits made by customers and (b) the cash which might be required from the central hub in order to satisfy all of the cash needs by the branch, such as customers withdrawals. However, while controlling branch cash holdings is pressing for the banking institutions' performance, the key point in the day to day operations of a branch is to guarantee an optimum level of cash inside the branch (1) which satisfies all the needs a branch may have (2) without generating either a shortage or a surplus of money. In addition, note that avoiding generating a surplus of money allows branches to considerably decrease the risk of theft.

This paper focuses on computing the (b)-summand of branch cash holdings: the amount of cash which might be required from the central hub in order to meet all the branch cash needs. In most cases, branches do not have in practice well-functioning systems of computing such quantities of cash, apart from some intuitive routines based on branch history data. That means that branch managers require similar quantities of cash corresponding to weeks with similar features. However, during the decision-making processes, the staff in charge usually reaches a decision with only partial information. The absence of more precise and inexpensive procedures generates inconsistencies intrinsic to the operational rules, whilst non-consistent methods as well as human errors are amongst these. This paper attempts to fill this gap by proposing a method of branch-specific computation that should be valid for all of them. A sound and low-cost method.

Actually, the main contribution of this paper is an algorithm designed to be an accurate tool to improve the performance of branches with respect to its cash management. This algorithm will significantly reduce cash holdings at branches, thereby providing efficient improvements in liquidity management. More specifically, it is a monitoring program to guide short-term corrective cash management actions of the branch's staff. The theoretical fundamentals of the proposed algorithm are some notional studies on the cash requirements of branches from their central hubs developed under "the demand for cash" scope (see [6] and [33] for the deterministic model, [27] for the introduction to the stochastic model). These fundamentals were reported by the first author of this paper in [18].[2]

We find that our algorithm performs well across the forecasting of cash amounts that the branch might require from the central hub to satisfy all branch necessities, avoiding having to generate either a surplus or a shortage of cash. In this regard, two algorithms have been designed depending on the unit of time considered. The first one corresponds to a daily computation, suitable for internal branch adjustments, whereas the second one performs for weekly cash forecasts. Many other units of time could be considered as part of the algorithm setting options. Besides, the algorithm proposed is very easy to implement in daily branching practices. Hence it involves an overall reduction in operating costs since this may be implemented without extra cost either in personnel training or in the implementation of the program itself. Thus, this algorithm is a sound and low-cost method that is also appropriate for all kind of branches, not only for those that can be considered candidates for increased supervision.

The algorithm is accompanied by a complete database processing of real branch-level records (more than 60,000 excel multicolumn cells have been processed), using software implementation in SageMath [32], to prove its accuracy as well as to derive to other conclusions. The data processing corresponds to two excel files which contain all daily branch operations from June 2012 to March 2013 of some representative Spanish branches belonging to a well known Spanish banking company.[3] As mentioned in [13], branch literature is much less complete than banking literature due to the lack of easy access to branch-level data. As a matter of fact, there are only a few studies supported by real banking records *based on data transactions*, due to the existing difficulties when accessing real sufficiently detailed banking data. To the authors' knowledge, the processing of such a (huge) real banking database has not been carried out till now.

The system can also be expanded to incorporate a cost structure in such a way that the algorithm forecasted amounts which should be required from the central hub in order to comply with all branch cash needs, also minimize a given costs function. One of the advantages of this complementary cost structure is that the cost function could be modified as needed provided only that it verifies some slight requirements. As the cost specification is an important characteristic of inventory management problems, the inclusion of the cost structure allows to locate the cash management problem described in this paper within the broader context of the optimal inventory literature (see [34] for an up-to-date and complete review of such literature).

The remainder of the paper is organized as follows. While Section 2 presents a literature review, Section 3 gathers the main outcome of the theoretical framework. Section 4 presents the software algorithm. Section 5 contains the data description and data-processing of real banking records, with numerical experiments devoted to back testing the algorithm. In Section 6 a corrective coefficient is designed, aimed at explicitly incorporating local demographics to the algorithm's forecasting. Section 7 is devoted to the development of a cost structure. Section 8 concludes the paper. Finally, to facilitate the exposition, the appendix contains the algorithm (daily and weekly) in flowchart form. It should be noticed that the algorithm has been presented in the main text (Section 4) in (pseudo)code with flowcharts in an

[2] Since the theoretical method proposed in [18], pointed in the right direction, has resulted in an effective forecasting system for bank branches, a patent has been requested for the paper [18] by the University of Granada, "Method for managing liquidity in bank branches" number ES201431094, United States.

[3] Due to the existing difficulties imposed by the EU legislation, both the names of the people and the banks must be kept confidential.

appendix. With this, we aim at catching the attention of as many researchers as possible, not only those who are familiar with coding.

2. Literature review

Models of cash management (or of demand for money) are approaches to determine the optimal investments that organizations should make in cash. These models can be categorized based on several criteria. One of them categorizes these models into two types: those with demand by households, pioneered by the Baumol–Tobin model [6], and continued by Frenkel and Jovanovic [17], Bar-Illan [5] and more recently Álvarez and Lippi [3,4], and those which concern cash management by firms pioneered by the paper of Miller and Orr [27]. Other authors categorize these models into deterministic versus stochastic (see [20] for a review of deterministic cash flow models) while other classifications have been established according to their mathematical fundamentals. As a matter of fact, while the cash management issue has often been treated as an inventory problem, following [26], the models of cash management can be grouped into Inventory Theory models (now both Baumol–Tobin and Miller–Orr belong to the same category), those developed with Linear Programming and those which are based on Dynamic Programming. The literature on cash management has been extended in so many directions that it is almost impossible to mention all of them. Amongst those papers that study cash supply chains, special mention should be given to those under the Operations Management (OM) scope: Nair and Anderson [29], who study the sweep programs, that is, procedures which allow banks to move funds from demand deposits or other checkable deposits into money market deposit accounts as part of savings accounts in a process generally invisible to the bank customers. Geismar et al. [19], Mehrotra et al. [25], Rajamani et al. [31], and more recently Zhu et al. [35], are papers that analyse the changes in cash circulation due to new currency guidelines implemented by the Federal Reserve System. Particularly, in [19] a model which describes the flow of currency between the bank and the Fed is designed.

However, at the branch level, the literature on research devoted to designing techniques to improve branch cash management specifically for branches, is quite scarce[4] apart from those papers which develop regulatory measures for the under-performing branches. To the best of the authors' knowledge, for the specific context of bank branches, this only comes to a total of three references: the first one is [10], where authors presented a model aimed at reducing cash management costs in a bank's branch using data mining. Secondly, the work [18] written by the first author of the present paper, where a theoretical model under "the transaction demand for the cash" scope recreates the setting of cash requirements at the branch level. The third reference is a subsequent work, [9] (2014) ([18] was published on 2013), whose authors apply GA (genetic algorithm) and particle swarm optimization (PSO) in cash balance management using multiple asset investments. In a theoretical context very similar to the one developed in [18], the authors minimize the total costs of cash maintenance in order to obtain cash management policies on three assets (cash management and two investments).

Finally, this section is concluded with a brief note on the different methods for estimating bank branch performance, as individual bank success and profitability may depend on its evaluation systems to measure bank branch financial performance ([sic], see [16]). The performance measurement approaches which have been developed to deal explicitly with branch performance evaluation (optimization techniques, simulations, stochastic tools, fuzzy logics and decision support systems) may be mainly categorized into *traditional ratios*, which are the most commonly used, *parametric models*, which require the existence of cost or production functions, *non-parametric techniques*, with data envelopment analysis (DEA) as focal point, and *integrated systems for performance evaluation* with the balanced scorecard (BSC) as main exponent. Others, as the evaluation index for bank branch financial performance developed in [16], are based on the integrated use of cognitive mapping and MACBETH.

3. Theoretical framework: branch cash requirements

3.1. Dynamics of the funds of a branch

The functioning of the branches as far as their liquid funds are concerned is detailed here. Every branch has certain daily available funds. At heart, there are two ways *cash enters the branch*: the branch requests for cash from the central hub and the deposits made by individual users/companies. Periodically, the branch adjusts its cash levels to its necessities – deposits and withdrawals – avoiding generating dormant money. When necessary, the branch requests a certain amount of cash. Let C_0 stands for the total amount of money that the branch requests from its cash central. We use $C_0{}^i$ instead when the unit of time when computing C_0, i, needs to be emphasized: that is, $C_0{}^i$ denotes the total amount of money that the branch requests from its cash central at ith-week. While C_0 is usually computed weekly, this unit of time – one week – may be changed without loss of generality.

Habitual bank branches cash management routines on the computation of C_0 basically consist of historical data handling: the branch registers the amount of cash on some particular week (work days, holidays, beginning/end-of month, etc.) and the result obtained (exceed or short), and copies the successful amounts. Despite there are other more precise procedures to help computing C_0, such as computer technologies, presently the current computation of C_0 is mainly based on the branch managers' expertise. However, in the decision-making process of identifying C_0, the staff in charge often reaches a decision with only partial information.

On the other hand, as an internal banking control mechanism, every branch must perform under a *cash upper bound* fixed by the bank entity according to the volume of the branch's turnover, denoted by C_{max}. This cash upper bound acts as a benchmark for each branch in such a way that it could be considered as a measurement of the size of the branch.[5] Once the branch liquid funds exceed this margin, the branch must request an armoured van from the cash hub to evacuate the surplus. As far as the branch *cash needs* are concerned, these are those that can be anticipated and those of random nature. The same classification can be established for deposits. As an example of anticipated branch cash needs, there exists in some countries a fixed-by-law cash threshold for big withdrawals. This means that branch users are required by law to give advanced notice to the branch in case of withdrawing quantities of cash that exceed this threshold. Actually, Governments throughout the EU are tightening up the control over branch big withdrawals as an effective fraud-prevention measure. In general terms, there may be limits on random withdrawals fixed by law or by the banks (Fig. 1).

[4] "Scarce" unless the strand of research focused on improving the performance of automatic teller machines, ATMs, would be considered as part of the literature to improve branch cash management.

[5] As mentioned later, there are many criteria to quantify the size of a branch amongst bank managers: the volume of credits, the number of business/private clients, the number of staff or the volume of deposits, amongst others. Actually, branch size is not a closed concept: on the contrary, it may be measured by means of several parameters.

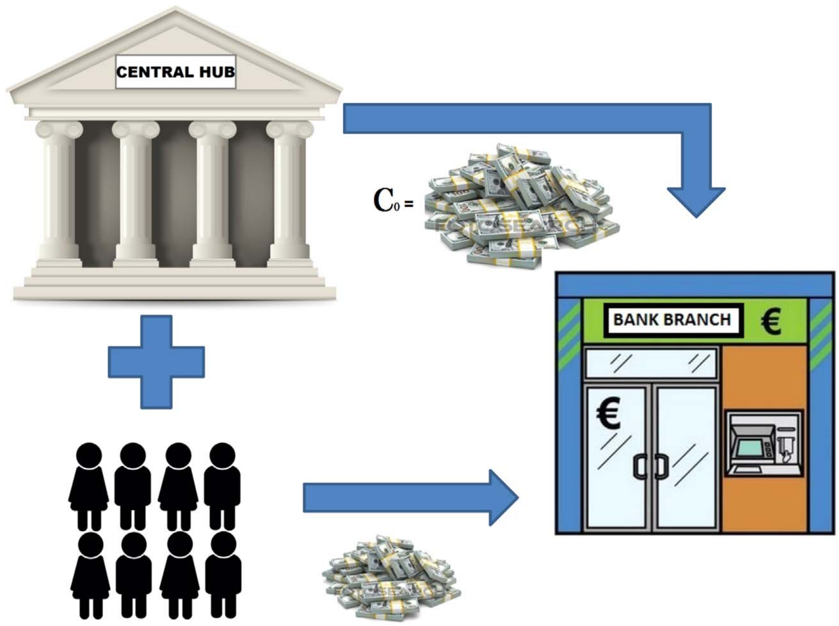

Fig. 1. Branch cash entries.

3.2. Fundamentals

We briefly summarize now the main outcome related to the computation of C_0. In order to model intraday liquidity, the consumer's withdrawals and deposits were described by means of some compound Poisson processes, as if they were arrivals in a counting process. For this, let X_t represents the withdrawals process while the deposit process was designed as Y_t, where t represents time. This leads to the main result of our theoretical setting related to the computation of C_0:

Theorem 1 ([18]). *The total cash amount C_0 which provides enough service to cover the branch demand of cash, C_0, may be computed as $C_0 = E_{X_1} - E_{Y_1} + K$, where K is the expected branch cash needs/deposits for a unit of time.*

4. The algorithm

This section is devoted to the algorithm that forecasts the cash amount that the branch might require from the central hub to satisfy all the needs the branch might have. As mentioned in the previous section, this is C_0 (or ${C_0}^i$, when the computing unit of time i needs to be emphasized). Two algorithms have been designed depending on the unit of time considered. The first one corresponds to a daily computation of C_0 (i.e., ${C_0}^i$ where i is a counter of days) for internal adjustment purposes whereas the second one is intended for routine weekly forecasts of cash (${C_0}^i$, where i is a counter of weeks).

For both algorithms, W_L denotes the *cash threshold for big withdrawals*. That means that users are required to give advanced notice to the branch in case of withdrawals which exceed W_L. There is also a *cash threshold for big deposits*, denoted by D_L. It is considered as a branch size benchmark (i.e., a key sign of the branches' ability to manage their liquid resources).[6] Such threshold does not mean that the branch would refuse a deposit that was too large or force the depositor to schedule making the deposit. On the contrary, any branch would accept any deposit. However, as each branch must perform under a cash upper bound, deposits above the threshold may distort any branch computations (for instance, the computation of C_0) because branch cash holdings would increase above the standard values. As such, deposits exceeding the corresponding threshold shall not be considered by the algorithm computations. Nevertheless, as part of the algorithm setting options, both threshold values (for both withdrawals and deposits) may be modified up or down depending on needs. For further details, see the Conclusions section.

4.1. Daily algorithm

For the daily time unit, stated in Algorithm 1, we assume that deposits are processed at the end of the day. Hence, they can be considered as part of the expected deposits in the next day. Thus, the inputs of Algorithm 1 consist of

- the current time T before new withdrawals and deposits;
- the mean λ^w of number of withdrawal operations per time unit until time T;
- the mean E^w of allowed withdrawals per operation until time T;
- the list of requested withdrawals $X = [(W_1, t_1), \ldots, (W_{N^w}, t_{N^w})]$ in this day, arranged according to time;
- the list of deposits $Y = [(D_1, t_1), \ldots, (D_{N^d}, t_{N^d})]$ in this day, arranged according to time;
- the total expected cash needs/deposits for the next day.

The outputs are

- the amount C_0 of cash to be requested the next day;
- new values of T, E^w, λ^w achieved at the end of the current day.

A few words about Algorithm 1. Lines 1 and 2 register the total number of withdrawal operations and cash needs before the process of today's operations. Next lines, until the end of the **for** loop beginning in line 5, are devoted to computing the total amount of withdrawals as well as the withdrawal amounts. In addition, if there is a withdrawal request which is bigger than the withdrawal threshold, it is deleted from the withdrawals counter and added to K. The reason for this is that, as mentioned before,

[6] A low threshold for deposits should match those branches that are unable to handle big deposits (small branches) while large sized branches, able to handle high volumes of cash, correspond to high thresholds for deposits.

there exists in Spain a fixed-by-law cash threshold for big withdrawals which implies that branch users are required by law to give advanced notice to the branch in case of withdrawing quantities of cash that exceed this threshold. Hence, the above-described banks' functioning is the Spanish usual one: any withdrawal request which is *over the threshold* is satisfied the day after in order to comply with legislation. This is the reason as to why withdrawals exceeding the stated threshold can be considered as *expected* ones: because branch managers already know of their existence.

For the same reason (Spanish bank's rules of functioning) deposits are considered expected ones in the daily algorithm. Following the Spanish branches usual functioning, they are all processed at the end of the day: as a matter of fact, all deposits are stored to be jointly controlled at the end of the journey. Again, this is the reason as to why they should be considered as *expected* deposits (since branch managers already know of their existence) and they may be consequently added to K and processed in the **for** loop in line 16 after deleting them from K.

Algorithm 1. Branch money request. Daily input and output.

Input: T, E^w, λ^w
Input: $X=[(W_1,t_1),\dots,(W_{N^w},t_{N^w})]$, list of withdrawals.
Input: $Y=[(D_1,t_1),\dots,(D_{N^d},t_{N^d})]$, list of deposits.
Input: K, both branch expected cash needs/deposits.
Output: C_0, the total amount of money that the branch should request from it cash central for a unit of time.
Output: T, E^w, λ^w
1: $N^w_{st} \leftarrow \lambda^w T$
2: $W_{st} \leftarrow E^w N^w_{st}$
▹ Processing of withdrawals
3: $N^w_t \leftarrow N^w + N^w_{st}$
4: $W_t \leftarrow W_{st}$
5: **for** $i \leftarrow 1$ **to** N^w **do**
6: **if** $W_i \geq W_L$ **then**
7: $K \leftarrow K + W_i$
8: $W_i \leftarrow 0$
9: $N^w_t \leftarrow N^w_t - 1$
10: **else**
11: $W_t \leftarrow W_t + W_i$
12: $E_X \leftarrow W_t - W_{st}$
13: $E^w \leftarrow W_t / N^w_t$
14: $T \leftarrow T+1$
15: $\lambda^w \leftarrow N^w_t / T$
▹ Processing of deposits
16: **for** $i \leftarrow 1$ **to** N^d **do**
17: **if** $D_i < D_L$ **then**
18: $K \leftarrow K - D_i$
19: **return** $E_X + K, T, E^w, \lambda^w$

4.2. Weekly algorithm

The second algorithm considers a week as the time unit. This leaves the door open to other units of time. From a computational point of view, these preferences (including time unit) may be arranged as part of the settings options. Algorithm 2 is much more clear if it is partitioned into three parts: the pre-processing of withdrawals, that of deposits, and the main core (which gathers what remains of the weekly algorithm after the pre-processing of withdrawals and that of deposits were completed, see Fig. 2).

Let us explain how the pre-processing parts work. We begin with deposits, although both procedures are interchangeable and can actually be executed in parallel. We assume that deposits are processed at the end of the day. Hence, from the standpoint of

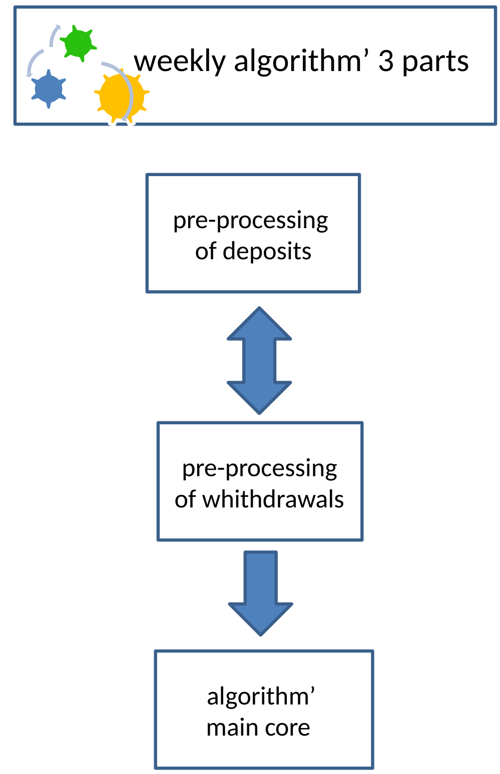

Fig. 2. Weekly algorithm's parts.

branch managers, all deposits made in one day can be considered as a unique deposit made the day after. In our pre-processing, deposits on Monday are accumulated as one deposit on Tuesday. The same works from Tuesday to Thursday. However, deposits made from Friday to Sunday are accumulated in K since they will be processed the week after. This is the Spanish banks' usual functioning. With regard to the weekly case, branch managers concentrate at the end of the day all deposits made throughout the full day in order to consider them as just one deposit to be processed the day after: hence from Monday to Thursday, daily deposits must be replaced by just one deposit the day after. On the contrary, deposits made from Friday to Sunday are accumulated in K since they will be processed the week after (and they can be considered as expected ones in consequence). Furthermore, if a deposit exceeds the deposits threshold, it is not taken into consideration, if a *really big* deposit takes place, since the branches usually evacuate it directly to the main office. In summary, this performance is intended to be coherent with the banks working standards described above, but may be altered as needed.

Algorithm 2. Branch money request. Weekly input and output. Pre-processing deposits.

Input: $T, E^w, \lambda^w, E^d, \lambda^d, ctr$
Input: $X=[(W_1,t_1),\dots,(W_{N^w},t_{N^w})]$, list of withdrawals.
Input: $Y=[(D_1,t_1),\dots,(D_{N^d},t_{N^d})]$, list of deposits.
Input: K, both branch expected cash needs/deposits.
Output: C_0, the total amount of money that the branch should request from it cash central every day.
Output: $T, E^w, \lambda^w, E^d, \lambda^d$
1: **procedure** PREPROCESSING_DEPOSITS (Y, K)
2: $D'_1 = D'_2 = D'_3 = D'_4 \leftarrow 0$
3: $Y' \leftarrow [(D'_1, \frac{1}{7}), (D'_2, \frac{2}{7}), (D'_3, \frac{3}{7}), (D'_4, \frac{4}{7})]$
4: **for** $i \leftarrow 1$ **to** N^d **do**
5: **if** $D_i < D_L$ **then**
6: **if** $t_i < \frac{1}{7}$ **then** ▹Monday's deposits

7: $D'_1 \leftarrow D'_1 + D_i$
8: **else if** $t_i < \frac{2}{7}$ **then** ▹Tuesday's deposits, i.e., $\frac{1}{7} \le t_i < \frac{2}{7}$
9: $D'_2 \leftarrow D'_2 + D_i$
10: **else if** $t_i < \frac{3}{7}$ **then** ▹Wednesday's deposits, i.e., $\frac{2}{7} \le t_i < \frac{3}{7}$
11: $D'_3 \leftarrow D'_3 + D_i$
12: **else if** $t_i < \frac{4}{7}$ ▹Thursday's deposits, i.e., $\frac{3}{7} \le t_i < \frac{4}{7}$
13: $D'_4 \leftarrow D'_4 + D_i$
14: **else** ▹Friday to Sunday's deposits, i.e., $\frac{4}{7} \le t_i$
15: $K \leftarrow K - D_i$
16: $Y \leftarrow Y'$
17: **return** Y, K

Algorithm 2. Branch money request. Weekly input and output. Pre-processing withdrawals.

18: **procedure** Preprocessing_Withdrawals X, K
19: **for** $i \leftarrow 1$ **to** N^w **do**
20: **if** $W_i \ge W_L$ **then**
21: **if** $t_i < \frac{4}{7}$ **then**
22: $t_i \leftarrow t_i + \frac{1}{7}$
23: **else**
24: $K \leftarrow K + W_i$
25: $X \leftarrow X - [(W_i, t_i)]$ ▹Element added to the set expected cash needs and deleted from the list of withdrawals.
26: Sort_ON_TIME (X) ▹Sorting on time
27: **return** X, K

Note that a meter *ctr* has been designed (see previous Algorithm 2, pre-processing deposits). This *automatic counter* is intended to be a *control mechanism which should provide a real-time warning for possible underperforming branches*. It should be noticed that a negative value of C_0 is not a good deal for a bank branch, especially if this situation is repeated in time. If this contingency takes place, the bank should consider if this particular branch is labelled as underperforming one in order to study either if the branch remains open or not. In this sense, the automatic counter *ctr* informs about the number of weeks with negative cash holdings in order to give notice to branch managers of possible underperforming cases once a threshold-of-negative-cash holdings (fixed by branch managers) is exceeded.

Algorithm 2. Branch money request. Weekly input and output. Main algorithm.

28: $N^w_{st} \leftarrow \lambda^w T$
29: $E^w_{st} \leftarrow E^w N^w_{st}$
30: $N^d_{st} \leftarrow \lambda^d T$
31: $E^d_{st} \leftarrow E^d N^d_{st}$
32: $T_t \leftarrow T + 1$
33: Preprocessing_Withdrawals (X, K)
34: $N^w_t \leftarrow N^w + N^w_{st}$ ▹Total number of withdrawals
35: $E^w_t \leftarrow E^w_{st}$ ▹Total quantity of withdrawals
36: **for** $i \leftarrow 1$ **to** N^w **do**
37: $E^w_t \leftarrow E^w_t + W_i$
38: $E_X \leftarrow E^w_t - E^w_{st}$
39: $E^w_t \leftarrow E^w_t / N^w_t$
40: $\lambda^w \leftarrow N^w_t / T_t$
41: Preprocessing_Deposits (Y, K)
42: $N^d_t \leftarrow N^d + N^d_{st}$ ▹Total number of deposits
43: $E^d_t \leftarrow E^d_{st}$ ▹Total quantity of deposits
44: **for** $i \leftarrow 1$ **to** N^d **do**
45: $E^d_t \leftarrow E^d_t + D_i$
46: $E_Y \leftarrow E^d_t - E^d_{st}$
47: $E^d_t \leftarrow E^d_t / N^d_t$
48: $\lambda^d \leftarrow N^d_t / T_t$
49: $C_0 = E_X - E_Y + K$
50: **if** $C_0 < 0$ **then**
51: $ctr \leftarrow ctr + 1$
52: **return** $C_0, T, E_w, \lambda_w, E_d, \lambda_d, ctr$

The pre-processing program for withdrawals runs as follows: if a requested withdrawal from Monday to Thursday is *over the threshold*, then it is considered as a new withdrawal made the day after, since this expense is going to be satisfied the day after. If one of these comes in Friday, then it will be processed the week after and consequently, it may be added to K as being considered expected one. Note that it is easy to change the algorithm's thresholds by modifying line 20.

Once deposits and withdrawals have been pre-processed, the main part of the algorithm can be performed. The former mean values are used to estimate the total number of withdrawals, deposits and amounts. They are included on the list of new data to compute the new mean values.

4.3. Learning from past failings

It is important to remark here on one of the algorithm properties (common to both algorithms, daily and weekly). As part of its sequential code, *the algorithm attempts to minimize the cumulative error while computing the forecasting cash amounts*, learning from past errors. Note first that the theoretical model (see Theorem 1, [18]) sets up how to compute the expected cash amount for the i-th week starting from data for general former week $i-r$ (it could be prior step $i-1$ or not), thus establishing a dynamic process in which the precise previous stage is not specified. This feature left possibilities open depending on needs.

The fact is that this computation, when based only on prior step – the week before – may cause misleading peaks-and-troughs. To avoid this, as well as to induce the algorithm to minimize cumulative error, the computation of the sequence of C_0 has been done as follows: each iteration uses as **inputs** the mean of cash withdrawn E_w, the mean of number of withdrawals, λ_w, the mean of deposits E_d and the mean of number of deposits made to date, λ_d, together with K (both branch expected cash needs/deposits for the unit of time). The corresponding **outputs** of our forecasting algorithm, apart from C_0, are new data E_w, λ_w, E_d and λ_d for next week which perform as inputs for next iteration. That way, the mean values E_w, λ_w, E_d and λ_d meet a temporal sequence, with every step becoming broader. Hence, the algorithm minimizes the cumulative error while computing the forecasting cash amounts as well as stabilizing the method.

Note also that the data corresponding to the first week (the data to start the computation) is unknown. We shall refer to these as initial values. In the subsequent section of numerical experiments, we will assess the sensitivity of the algorithm to changes in the set of initial values, analyzing the results (a) with a set of initial values equal to zero and (b) with a set of initial values equal to final outputs from (a) – computation.

5. Data description and data-processing of real banking records

In this section, some numerical experiments in order to back test the weekly algorithm (as it is the most commonly used in actual banking) are performed. Our experiments are based upon two excel

Please cite this article as: García Cabello J, Lobillo FJ. Sound branch cash management for less: A low-cost forecasting algorithm under uncertain demand. Omega (2016), http://dx.doi.org/10.1016/j.omega.2016.09.005

files that contain all daily branch operations from June 2012 to March 2013 of some representative Spanish branches of a well known Spanish bank. Our initial data set was originally written using the entity's specific code (see attached files as supplementary material). As part of the database processing, significant external operations (such as withdrawals and deposits) have been extracted/separated from those internal organizational orders (accounting entries).

There are additional problems with the real branch database. Apart from difficulties in achieving the proper authorizations for accessing branch material, *the information that branches register is not complete in most cases*: to a great extent, bank branches have no thorough record of all remaining amounts of cash from previous weeks. Nonetheless, cash holdings remaining from previous weeks influence the branch managers' decision on cash request for the following week. The information regarding cash amounts remaining from previous weeks is, therefore, crucial when establishing "real cash needs for every branch", since they might be "deposits minus withdrawals plus cash remaining from the prior period". This lack of information forces us to consider "real cash needs" as simply "deposits minus withdrawals".[7] This situation implies that, inevitably, the whole attempt to back test the algorithm's accuracy when comparing the forecasted cash amounts with real cash needs could however be affected by an error margin caused by this lack of branch information.

The following diagrams outline the results over almost a year of comparison between (1) weekly real cash needs by branch consumers (defined as "deposits minus withdrawals" with the misgivings highlighted above) and (2) algorithm weekly forecasted amounts of cash which should be required from the central hub in order to comply with all branch cash needs. This context also defines some parameter sensitivity tests. In Section 5.1, different assumptions relating to branch size (different deposits threshold values) are used to determine their incidence on the algorithm's accuracy. In Section 5.2, the algorithm's sensitivity to variations in the set of initial values is tested. In this regard, we consider a two stage data processing: the first one starts from $E_w = 0, \lambda_w = 0, E_d = 0, \lambda_d = 0$ while the second one starts from the final values obtained from the first stage of data processing. The latter is aimed at imitating the real branch routine, where the initial values considered are likely to be the mean ones.

5.1. Algorithm effectiveness depending on the branch size

Here, the accuracy of the algorithm for branches of different sizes is tested. At present, there are many criteria to quantify the size of a branch amongst bank managers: the volume of credits, the number of business/private clients, the number of staff or the volume of deposits, amongst others. The most accepted is to consider the size of a branch as an increasing function of the total branch cash needs: the bigger branch sizes correspond to bigger movements – entries and exits – of liquid resources.

Thus, in order to measure the algorithm's sensitivity to changes in branch sizes, we consider different values for the *threshold for deposits* since, as mentioned before, the threshold for deposits is a key sign of the branches' ability to manage their liquid resources and, in consequence, a measurement of the branch size. While we vary the threshold for deposits, we keep constant the threshold for withdrawals in all cases, since this should be fixed by banks or by law. However, both threshold values may be modified up or down according to needs (see Conclusions section).

In order to cover the whole range of branch sizes, we categorize them into three general types: small, medium and large. The cut-off points for classifying the branches into these three groups depend on each specific context/country, and they may be modified as part of the settings options. We shall assume that Small branch size equals to "deposits threshold" = 10,000€, Medium size equals to "deposits threshold" = 100,000€ and Large size equals to "deposits threshold" = 1,000,000€.

For all branch sizes (small, medium and large) the following graphs represent the difference between "real cash needs"(deposits minus withdrawals) and algorithm forecasted cash amounts with initial values $E_w = 0, \lambda_w = 0, E_d = 0, \lambda_d = 0$. Black bars represent real cash needs and white ones the forecasts. The x-axis shows weeks of the year 2013/2014 starting the first week of June, while cash amounts in Euros appear on the y-axis.

5.1.1. Small branch size: "deposits threshold" = 10,000 €

It becomes apparent that the forecasted cash amounts are *above* the real necessities in most cases. As mentioned before, this deviation is caused by the definition of "real cash needs" (as "deposits minus withdrawals" instead of "deposits minus withdrawals plus remaining money from former week"). In consequence, the "real cash needs" volumes are *lower* than what they should be. Hence, with the corrective coefficient applied with respect to the assumed definition of "real cash needs", we may conclude that the algorithm works well (Fig. 3).

5.1.2. Medium branch size: "deposits threshold" = 100,000 €

All prior comments for small branch size apply here too. In as far as the comparison between graphs of small and medium branch size we note that, in general terms, there are no apparent differences between them, with the exception of week 24 (4th week of November). For this week, the algorithm's forecasted amount is more suitable for smaller rather than for medium branch size. In fact, in cases of medium size branches, the algorithm forecasts an amount for week 24 (4th week of November) that far exceeds the "real" demand of cash, while the prediction in the case of the small branch size is more commensurate with real needs. *This is because, importantly, "real" demand of cash is much lower for medium size branches than it is for small ones in the particular case of week* 24 (*4th week of November*). In as far as this peculiarity (4th week), in light of the branch managers' experience regarding consumers' habits, one explanation could be that consumers at small branches (more so than medium/large ones) tend to withdraw cash instead of using other kinds of payment services such as credit transfers or card payments for the scheduled supply of funds corresponding to the beginning of the Christmas holidays – a period of high demand of cash. Exactly the same applies to week 24 in case of large size branches (medium = large as far as week 24 is concerned, see next paragraph) (Fig. 4).

5.1.3. Large branch size: "deposits threshold" = 1,000,000 €

All above general comments for small/medium size branches apply here too. In light of the above outcomes, we may conclude that the algorithm performs well for all kinds of branches regardless of their size (Fig. 5). We take now a closer look at the details of the features of the best and worst-fit weeks. The following can be found in Tables 1–4.

If we heed the branch managers' experience (ex post) in order to explain these singularities, one might take into account economic, social and cultural factors: periods where spending tends to increase, corresponding to pre-holidays (beginning of July and December); the so-called 'hard January' as a financial rupture from personal habits of moderation and austerity; the different degrees of impact of the financial crisis, a factor that distorts the branch managers' predictions for cash in as far as the consumers' habits are concerned; work schedules, with hot dates such as tax revenue deadlines (June); forthcoming school enrolment dates (end of August/beginning of

[7] What we mean here by withdrawals is cash withdrawals, an e-transfer through the interbank system would not be taken into consideration (e.g., from one bank to another or from one account to another).

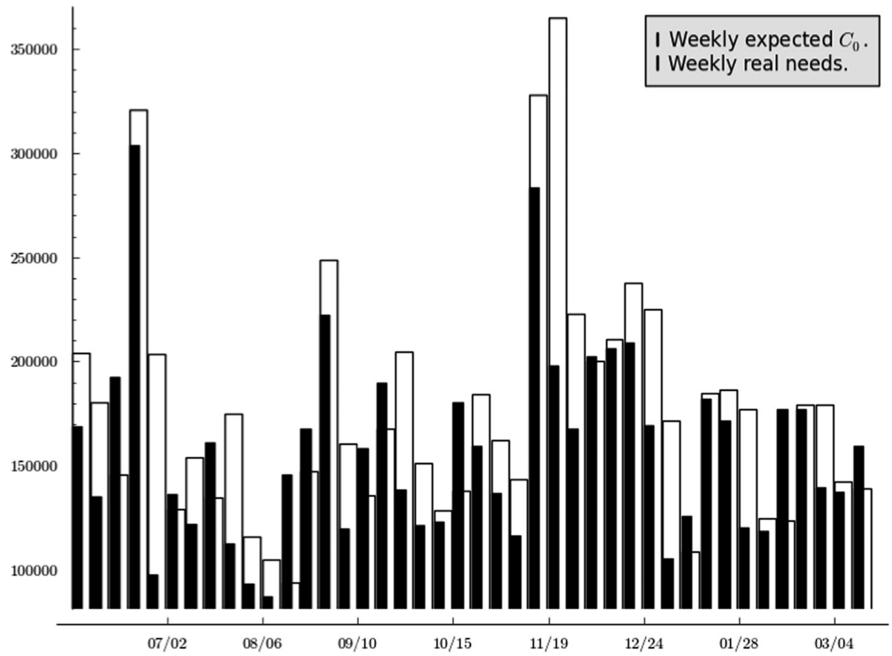

Fig. 3. Weekly expected C_0. Small branch size.

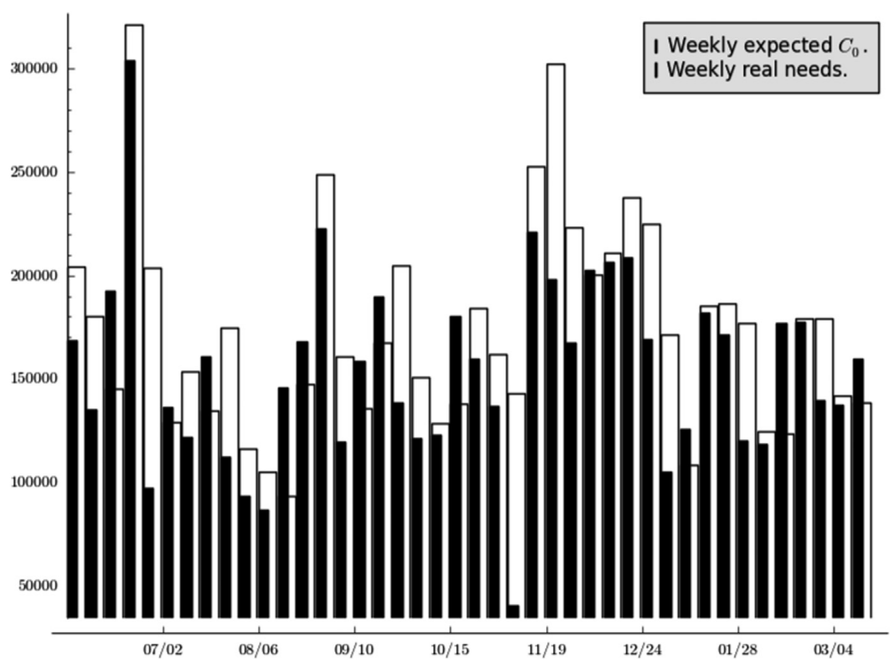

Fig. 4. Weekly expected C_0. Medium branch size.

September), as a peak in expenditure. And many other factors that vary depending on each country's habits and traditions, socio-economic circumstances and particular banking conditions that belong to the private sphere. That is to say, the main determinants that should be taken into account in order to further fine-tune the algorithm predictions are special conditions pertaining to the socio-economic and cultural scope of each country/region/state or are of confidential nature, except for banking branch managers. Hence, it would not be possible to go any further in designing a *general* algorithm which would be valid for all cases, but rather we should apply our proposal to each particular context with its own peculiarities (see Conclusions for more detailed explanations).

5.2. The sensitivity of the algorithm to changes in the set of initial values

We will test here the sensitivity of the algorithm to changes in various points in time. In particular, we will show that the algorithm is robust in the sense that its forecasts do not depend on the point in time in which they are made. To achieve this, let us recall the algorithm's set of initial values: the mean of the cash withdrawn E_w, the mean of the number of withdrawals, λ_w, the mean of the deposits E_d and the mean of number of deposits made to date, λ_d. Note that all these parameters are time dependent. Actually, these settings correspond to observations made at different times, one to one: each set of

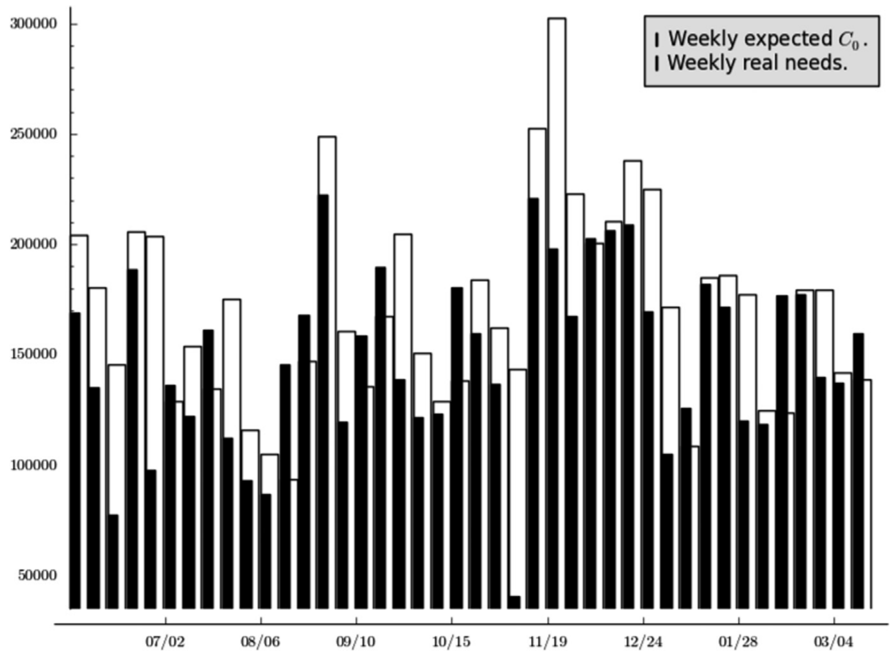

Fig. 5. Weekly expected C_0. Large branch size.

Table 1
Best-fit weeks.

Number	Specific week
6	2nd July
13	1st September
20	4th October
28	4th December
29	1st January
34,35	2nd, 3rd February
37, 39	1st, 3rd March
41	1st April

Table 2
Second bests.

Number	Specific week
4	4th June
10, 11	2nd, 3rd August
14, 16	2nd, 4th September
17, 19	1st, 3rd October
22, 23, 24	2nd, 3rd, 4th November
30	2nd January
33	1st February

Table 3
Worst-fit weeks: forecasts in excess.

Number	Specific week
5	1st July
26	2nd December

Table 4
Worst-fit weeks: forecasts short-sighted.

Number	Specific week
12	4th August
38	2nd March

initial values corresponds to a point in time while each point in time corresponds to a set of initial values. Thus, we shall identify each set of initial values with the temporal point in time where they were considered.

The following graphs (Figs. 6–8) stand for data processing with a set of initial values $E_w = 140.47$, $\lambda_w = 1384.21$, $E_d = 4802.28$, $\lambda_d = 17.31$ for the three cases of branch sizes (small, medium and large). As these graphs show, data set produces identical results. Hence, the algorithm does not depend on the set of initial values. Since they represent temporal data, the conclusion is that the algorithm's forecasts do not depend on points in time. The method, then, is robust.

6. Local demographics

As mentioned before, branch size is not a closed concept: on the contrary, it may be measured by means of several parameters. Actually, *there exists a relationship between branch size and local demographics*: branch size depends on branch cash transactions – number and amounts – while branch cash transactions depend on branch customer's needs for cash, which are strongly related to local demographics (a heavy retail stores area will require much more cash than a heavy industrial area where firms do not deal with much cash). This section is devoted to explicitly incorporating branch's local demographics to the algorithm's code.

Let us first point out that local demographics has already been *implicitly* considered when using the above categorization of branches for model validation because grouping the branches into city centre, rural or business centre by practitioners implicitly include their demographic features inside. Evidence of this is the fact that branch managers, in practice, categorize branches on city centre, rural or business centre depending, not (only) on their branch geographical location but on their number and amount of transactions, *on a not clearly defined basis*: i.e., although a branch is geographically located at a rural area, it could be considered by practitioners as city centre if its number and amount of transactions exceed the internal benchmarks for rural branches.

Demographic parameters have to be carefully managed due to there being major variations on specifying demographics on "local" – as opposite to "internationally accepted" – parameters. As

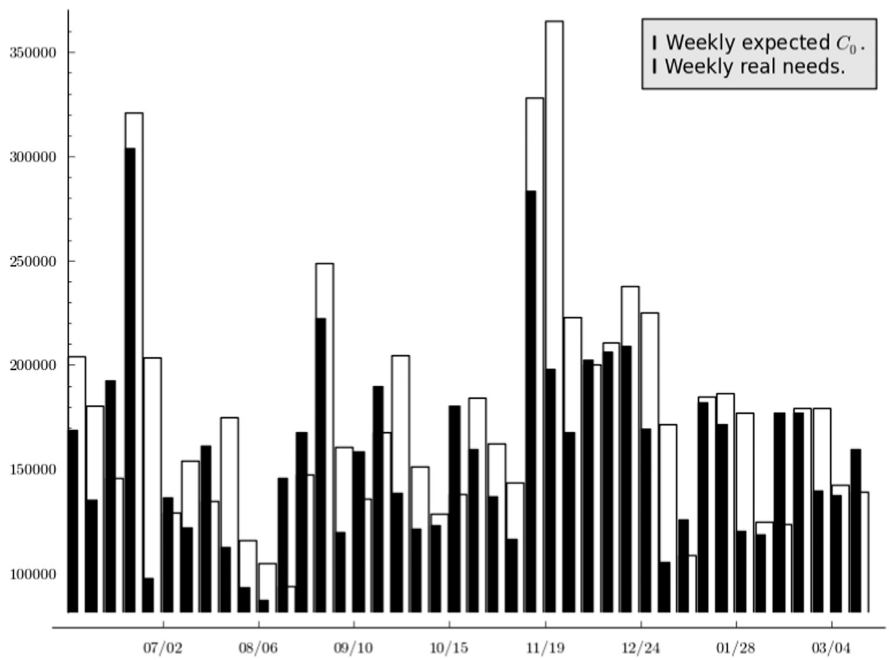

Fig. 6. Weekly expected C_0. Small branch size.

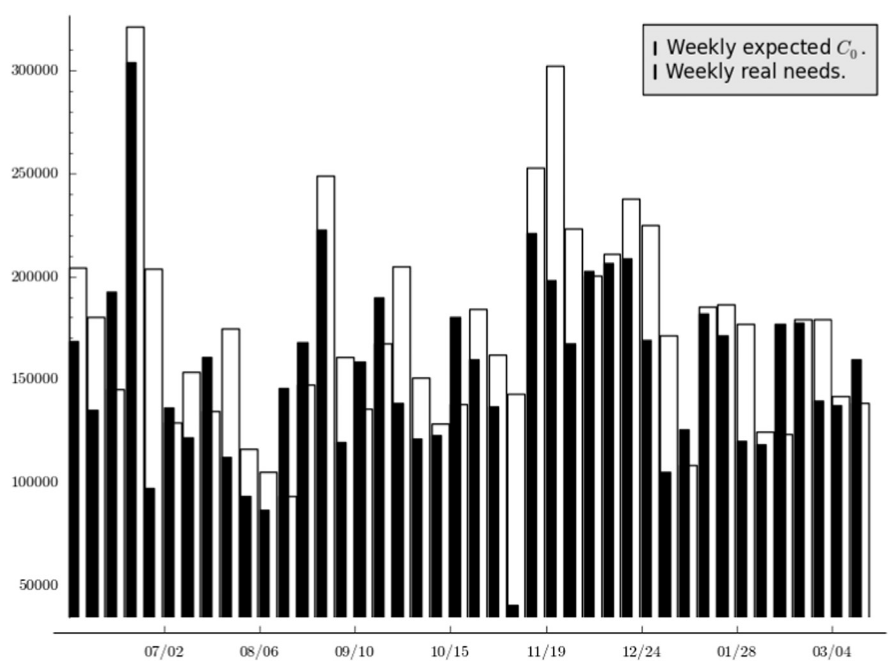

Fig. 7. Weekly expected C_0. Medium branch size.

a matter of fact, the above categorization of branches by branch managers is quite unclear because the distinction between urban and rural areas is growing *fuzzy*: while the main criteria to define them commonly include population size/density and availability of some support services such as secondary schools and hospitals, the combination of criteria applied can vary greatly: even the population thresholds used across countries can be different. As a result of such existence of wide fluctuation in definitions, international comparisons are very difficult (see [2], where explicitly it is explained that "Scientists from different disciplines diverge when defining these zones – rural/urban – or their limits; they even often mention the zones without any definition. This practice excludes comparison between studies.", [sic]) or [21], where the authors, in order to develop a *local* indicator of financial development, chose to carry out their study within a single country rather than across countries – as they sought an unified vision.

In order to overcome the aforementioned difficulties, a weighting (demographic correction coefficient) d has been designed to be added to the algorithm's code. This has been carried out in a *fuzzy* way: i.e., a range of values $d \in [d_{min}C_0, d_{max}C_0]$ has been considered instead of a single one. For this reason, notations d or $[d_{min}C_0, d_{max}C_0]$ can be used interchangeably to refer to the "demographic correction coefficient". Moreover, in order to cover the widest range of cases, d would depend on some binary variables which take value equal to 1 if it applies, 0 if not. This way these determinants could be processed only when applicable.

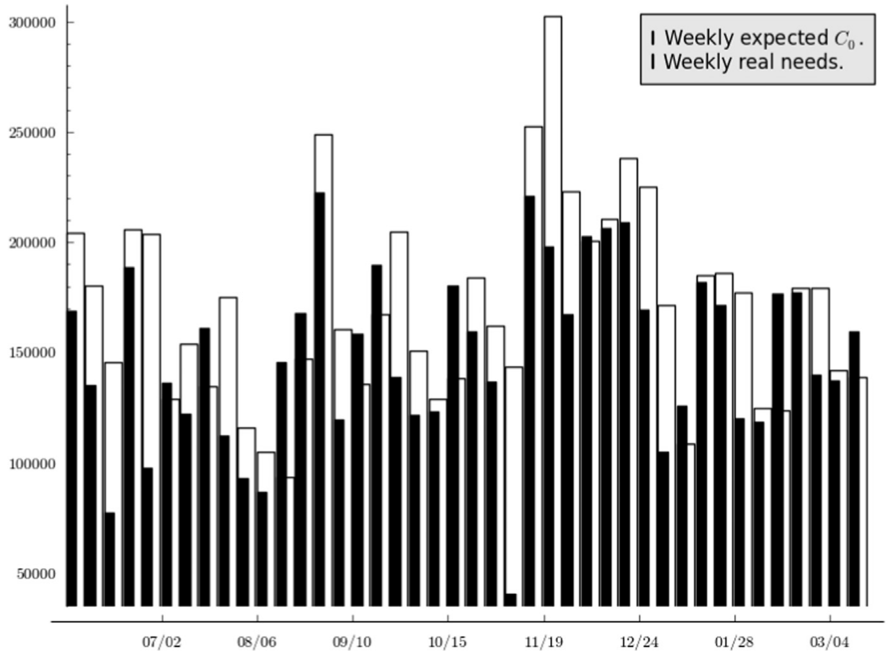

Fig. 8. Weekly expected C_0. Large branch size.

Furthermore, both determinants and number of them can be freely selected depending on the considered scenario.

Specifically, the viewpoint adopted here is that the demographic coefficient increases the uncertainty in the output C_0 in such a way that it can be resolved with more than one choice: this is the reason as to why a range of values[8] $d \in [d_{min}C_0, d_{max}C_0]$ has been considered instead of a single one.

As an illustrative example, we consider the following three geographical variables as determinant of branch location: $z_1 =$ unemployment, $z_2 =$ density of population and $z_3 =$ percentage of foreign population. Our selection has been made according to the following criteria: apart from the frequency with which they are used in literature, "data availability" has been taken into account since, for instance, the variable *per capita income* is quite common in the literature as determinant of branch location but often this information is not available. This is as to why we propose *unemployment* instead, which is strongly correlated (negatively) with per capita income. But, as mentioned before, both variables and number of them can be freely selected.

Thus, the values of the coefficient $d \in [d_{min}, d_{max}], d_{min} \le 1 \le d_{max}$ are determined by the values of the chosen geographic variables z_i, according to Table 5.

This argument can be easily extended either when more variables are needed or a wider range of variable's values is considered. Finally, once the demographic corrector coefficient has been computed, it may be added to Algorithm 2 in line 52 in such a way that this line would finally be presented as

$$52 : \textbf{return } [d_{min}, d_{max}] \times C_0, T, E_w, \lambda_w, E_d, \lambda_d, ctr$$

where the notation $[d_{min}, d_{max}] \times C_0$ means the interval $[d_{min}C_0, d_{max}C_0]$.

[8] In the choice of d inside $[d_{min}C_0, d_{max}C_0]$, the expertise eye of the office director can help leading to the best option.

7. A cost structure

As mentioned in the Introduction, besides the literature related to the design of managerial measures for improving branch cash management, this paper may be also discussed within the broader context of the optimal inventory literature. In short, the problem of inventory control is aimed at a successful management of stocks of goods in order to meet the demand by seeking for an inventory policy that will make profits as large as possible or *costs as small as possible*.

This section is thus devoted to developing a complementary cost structure. The starting point is the cost function stated in [18], designed by following the normal practice in inventory theory of assuming that the bank seeks to minimize the long-run average cost of managing the cash balance under some policy of simple form. This cost function is as follows:

$$\varepsilon(C_0, C_{max}) = \gamma \underbrace{\frac{A}{(C_0 - C_{max})C_{max}}}_{\text{costs due to cash flow}} + \nu \underbrace{\frac{C_0 + C_{max}}{3}}_{\text{opportunity costs}} + \underbrace{BC_{max}}_{\text{insurance costs}}$$

where C_{max}[9] is the branch cash upper bound fixed by the bank entity, γ is the unitary cost per transfer, A is the variance of daily changes in the cash balance,[10] ν is the daily rate of interest earned on portfolio (e.g., other banks products which yield higher benefits), and B is the constant of proportionality due to that the payment on the bank's company to a theft insurance policy, which should be directly proportional to C_{max}. $\varepsilon(C_0, C_{max})$ was considered in [18] as objective function in the following constrained optimization programme:

$$\text{Minimize :} \quad \gamma \frac{A}{(C_0 - C_{max})C_{max}} + \nu \frac{C_0 + C_{max}}{3} + BC_{max}$$

$$\text{s.a.} \quad \begin{cases} C_0 = E_{X_1} - E_{Y_1} + K \\ C_0 \le C_{max}. \end{cases}$$

[9] The notation C_{max} replaces the former notation of C_z of [18] for clarity purposes.

[10] Specifically, considering that the random behaviour of the cash balance can be characterized as a sequence of independent Bernoulli trials, if μ denotes the amount of euros that the branch cash balance increases or decreases in some small fraction of a working day $\frac{1}{t}$, thus $A = \mu^2 t$.

Table 5
Example of demographic coefficient.

z_1	z_2	z_3	d_{min}	d_{max}
0	0	0	1.0	1.0
0	0	1	1.0	1.1
0	1	0	1.0	1.1
0	1	1	0.9	1.1
1	0	0	1.0	1.1
1	0	1	0.9	1.1
1	1	0	0.9	1.1
1	1	1	0.8	1.2

$z_1 =$ unemployment $z_2 =$ density of population $z_3 =$ percentage of foreign population.

in such a way that the algorithm's output C_0 was intended to satisfy the following requirements: C_0 (by performing under C_{max}) is enough to cover both expected and unexpected branch cash needs while minimizing the banking cost function. Here, we return to this question as follows:

On one hand, we consider the corrective coefficient for local demographics, denoted either d or $[d_{min}C_0, d_{max}C_0]$, as stated in the previous section. If more than one corrective coefficient would be added, thus the maximum of them should be considered instead. We detailed this at the Conclusion section, where the possibility of adding other corrective coefficients – such a seasonality – has been discussed.[11] Thus, the first constraint $C_0 = E_{X_1} - E_{Y_1} + K$ is turning into

$$C \in [d_{min}(E_{X_1} - E_{Y_1} + K), d_{max}(E_{X_1} - E_{Y_1} + K)],$$

where C is an unknown representing the new value of the cash holding under cost minimization. On the other hand, as each branch might observe its cash upper bound, C_{max}, once this numerical value is substituted in the minimization program, it hence performs under a single variable, C, and a (new) single constraint, as follows:

$$\text{Minimize}: \quad \gamma \frac{A}{(C - C_{max})C_{max}} + \nu \frac{C + C_{max}}{3} + BC_{max}$$
$$\text{s.a.} \quad C \in [d_{min}(E_{X_1} - E_{Y_1} + K), d_{max}(E_{X_1} - E_{Y_1} + K)].$$

According to Weiertrass's Extreme Value Theorem, for a real-valued continuous function ($C \neq C_{max}$, $C_{max} \neq 0$) on a non-empty compact domain, there exists global minimum. The extremum occurs either at critical points inside (in the interior) of the interval or at the end points of the interval. Once the algorithm output $E_{X_1} - E_{Y_1} + K$ and the corrective coefficient $[d_{min}, d_{max}]$ are computed, algorithm evaluates the cost function both at the critical and at the end points of the interval in order to identify the smallest value. This may be achieved by adding Algorithm 3 to the algorithm code.

Algorithm 3. Cash holdings under cost estimation.

Input: $\gamma, \nu, A, B, C_{max}$ as described in this section.
Input: $C_0 = E_{X_1} - E_{Y_1} + K$ the output of Algorithm 2.
Output: A new C_0 minimizing the cost function.

$\Phi(C) = \gamma \frac{A}{(C - C_{max})C_{max}} + \nu \frac{C + C_{max}}{3} + BC_{max}$
$Z \leftarrow \{z \in [d_{min}(E_{X_1} - E_{Y_1} + K), d_{max}(E_{X_1} - E_{Y_1} + K)] \mid \Phi'(z) = 0\}$
$output \leftarrow d_{min}(E_{X_1} - E_{Y_1} + K)$
for $z \in Z$ **do**
  if $\Phi(z) < \Phi(output)$ **then**
    $output \leftarrow z$
if $\Phi(d_{max}(E_{X_1} - E_{Y_1} + K)) < \Phi(output)$ **then**
  $output \leftarrow d_{max}(E_{X_1} - E_{Y_1} + K)$
return $output$

Table 6
Seasonality coefficient.

T	1–4	5–13	14-16	17–26	27–31	32–40
$s_c(T)$	1.0	1.1	1.2	1.0	1.3	1.0

It should be noticed that one of the advantages of the developed cost structure is that the cost function may be changed as needed provided only that it verifies the conditions stated at Weiertrass's Extreme Value Theorem (which are very mild indeed).

8. Conclusions and directions for future research

This paper provides a new tool – a forecasting algorithm under uncertain demand – to improve cash management at branch level by updating the branches' cash forecasting processes. We have shown that the algorithm is robust in the sense that its predictions do not depend on the point in time in which they were made. Also, we have found that it performs well for all kinds of branches, regardless of their size. Besides being sound, it is a low-cost method. Given these reasons, this algorithm is appropriate for all types of branches, not only for those which may be candidates for increased supervision.

The algorithm is designed to work well at the greatest possible number of scenarios with minimum cost. While being a low-cost tool is very attractive, the generality of the algorithm is one of its better features, since it may be adjusted as required and can be tailor-made to suit the specific requirements of each banking institution (or each kind of branch), subject to minor fine-tunings. In fact, since only minor adjustments would need to be made on the algorithm's code, these may be carried out throughout the banking institutions' own computer services.

This is the case, for instance, of the *cash threshold for big withdrawals*, W_L. This issue has been considered in this paper since there may be limits on random withdrawals fixed by law or by the banks own internal policies.[12] Each banking institution may fix its own, or a disproportionately large amount (aimed at approaching $W_L \to \infty$) in the case that there is no threshold for big withdrawals. In this specific case, adapting lines 6th and 17th for the daily algorithm and/or the 5th and 20th for the weekly algorithm would modify both thresholds for deposits and withdrawals as required.

As a second example of adjustments to comply with other requirements, a seasonality coefficient may be designed as follows: in the first place, it should be noticed that the seasonality coefficient is time-dependent over the week number, which is stored in T. Table 6 shows thus how this corrector coefficient can be stored.

This table has been developed on the basis of data provided in this paper, the first log corresponding to June. A seasonality coefficient over 1.0 is established in Summer, September, and December, when an increment of cash is presumed. Finally, the seasonality coefficient can be added to Algorithm 2 in line 52 in a similar way that the demographic corrector coefficient was added. Thus, including both coefficients, the line would finally be presented as

$$52: \textbf{return } s_c(T) \times C_0, T, E_w, \lambda_w, E_d, \lambda_d, ctr$$

[11] Actually, in Section 8, a seasonality corrective coefficient has been designed. This may also serve as an illustrative example of how many other corrective coefficients – relative to special conditions pertaining to the socioeconomic and cultural scope of each country/region/state – may be designed.

[12] In Spain, for instance, users are required by law to give advanced notice to the branch in case of withdrawals which exceed the threshold W_L.

As it has been shown, the algorithm's generality is one of its better features, since it may be adjusted as needed. Moreover, it guarantees not only full effectiveness (since the algorithm would fit the best to each branch's necessities) but also lower costs (since the adjustments could be implemented by their own central computer). Both features, low-cost and generality, would provide a competitive advantage.

The usefulness of our results depends to some extent on the regulatory requirements on cash holdings. Specifically, reserve requirements are binding in the U.S. while in Europe there are no such requirements. In principle, this may suggest that our model and findings are more relevant in the case of European banks as holding cash in excess is not driven by a binding regulation. However, the results could be also relevant for the U.S. in a number of dimensions. First of all, even in the presence of binding reserve requirements, banks may hold cash in excess and may be in need of improving their cash management. Secondly, banks may hold excess reserves in the light of market conditions and they may look in particular to the evolution of inflation and interest rates [23]. Third, as noted, inter alia, by Bennett and Peristiani, [7], there is evidence that reserve requirements constrain U.S. commercial banks and other depository institutions to a much smaller degree than in the past. One of the reasons is the spread of "sweep" arrangements – a banking innovation that allows depository institutions to shift funds out of customer accounts subject to reserve requirements. All in all, this evidence suggests that U.S. banks now appear to be managing their cash flows more in accordance with business needs than just with regulatory obligations.

Management insight and experience of branch financial officers are not incorporated into the algorithm's code. As mentioned in the Introduction, the proposed algorithm is a monitoring program to guide short-term corrective cash management actions of the branch staff. Thus, on one hand, the algorithm could work as an expert system when complemented by the branch managers' expertise. On the other hand, the expertise of each banking institution could also be incorporated into the algorithm's code. For instance, in the day to day activities of certain banks' branches, the computation of C_0 is usually done using only the previous step with similar features.[13] To this regard, recall that the theoretical model (Theorem 1) sets up how to compute the expected cash amount for the i-th week starting from previous data $i-r$ (it could be prior step $i-1$ or not), without specifying the precise previous stage. While our algorithm performs well across the mean of all previous data as previous stage, it might work just as well (or even better, depending on bank entities/kind of branch needs) with other possibilities. These other possibilities include using the previous step with similar features (simulating the aforementioned branches' routine in the computation of C_0) or even, the mean of those previous stages with similar features.

In this sense, the terms and conditions of a possible agreement between the University of Granada and some Spanish banking institutions are currently in negotiation in order to carry out some experiments to best-fit the algorithm to the needs of each banking institution. This is a future research project within a foreseeable period of time.

Acknowledgements

The first author would like to deeply thank the people who have allowed access to banking records as they have made the analysis stated in the present paper possible. Unfortunately, both the names of these persons and the banking institutions must be kept confidential.

Moreover, both authors are indebted to the anonymous reviewers for the further suggestions to improve our manuscript.

Financial support from the excellence project of the Spanish Ministry of Science and Innovation "Mecanismos de resolución de crisis: cambios en el sistema financiero y efectos en la Economía real" (P12-SEJ-2463). The first author also thanks the financial support for the project received from by of the Regional Government of Andalusia "GAMMA (Grupo de Análisis Microeconómico y Macroeconómico Aplicado)" (SEJ340).

Appendix A. The algorithms in flowchart form

See Figs. 9–12.

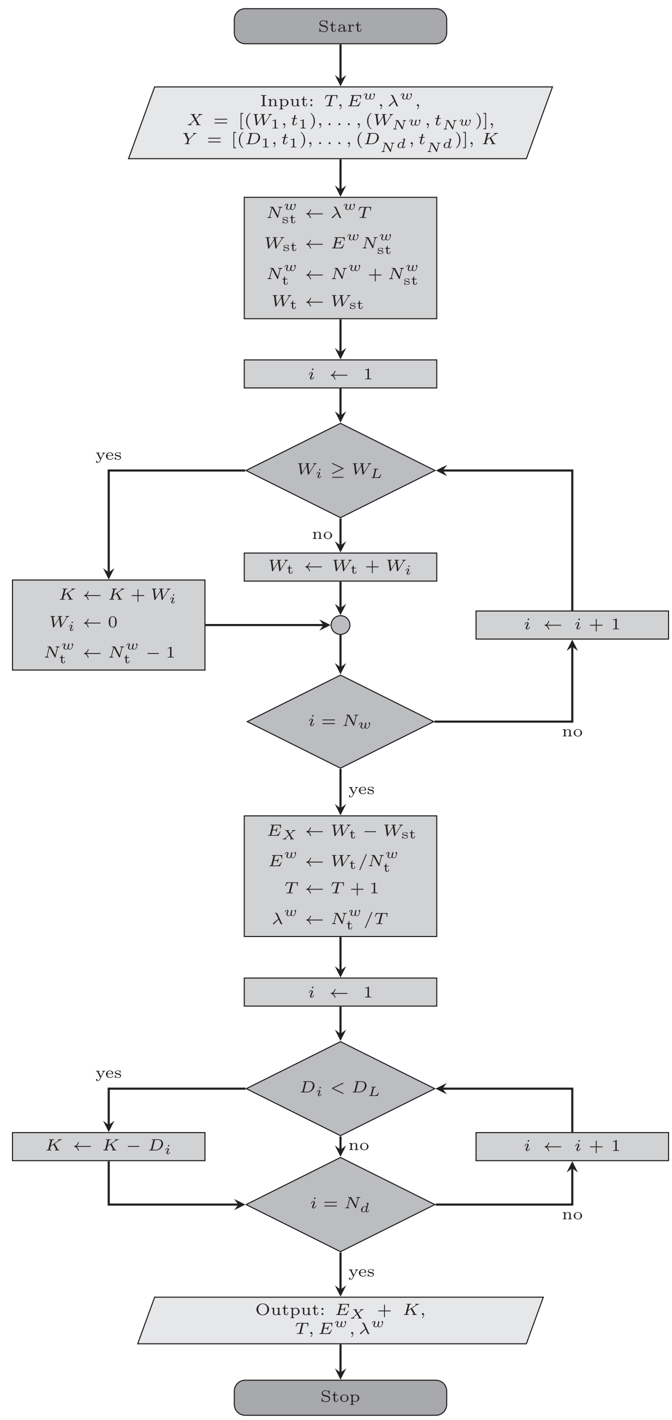

Fig. 9. Daily forecasting algorithm for C_0 (Algorithm 1).

[13] As an example of weeks with similar features, let us think of first weeks of each month.

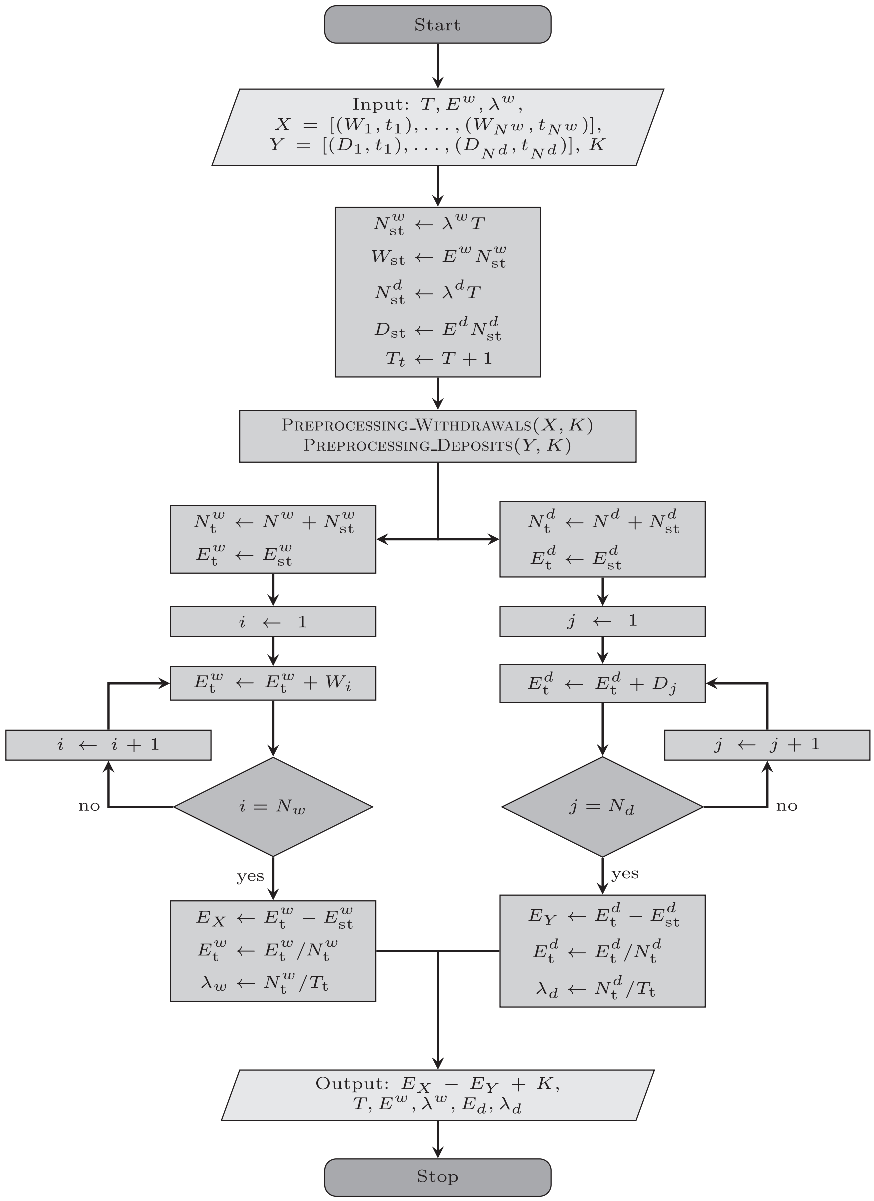

Fig. 10. Weekly forecasting algorithm for C_0 (Algorithm 2).

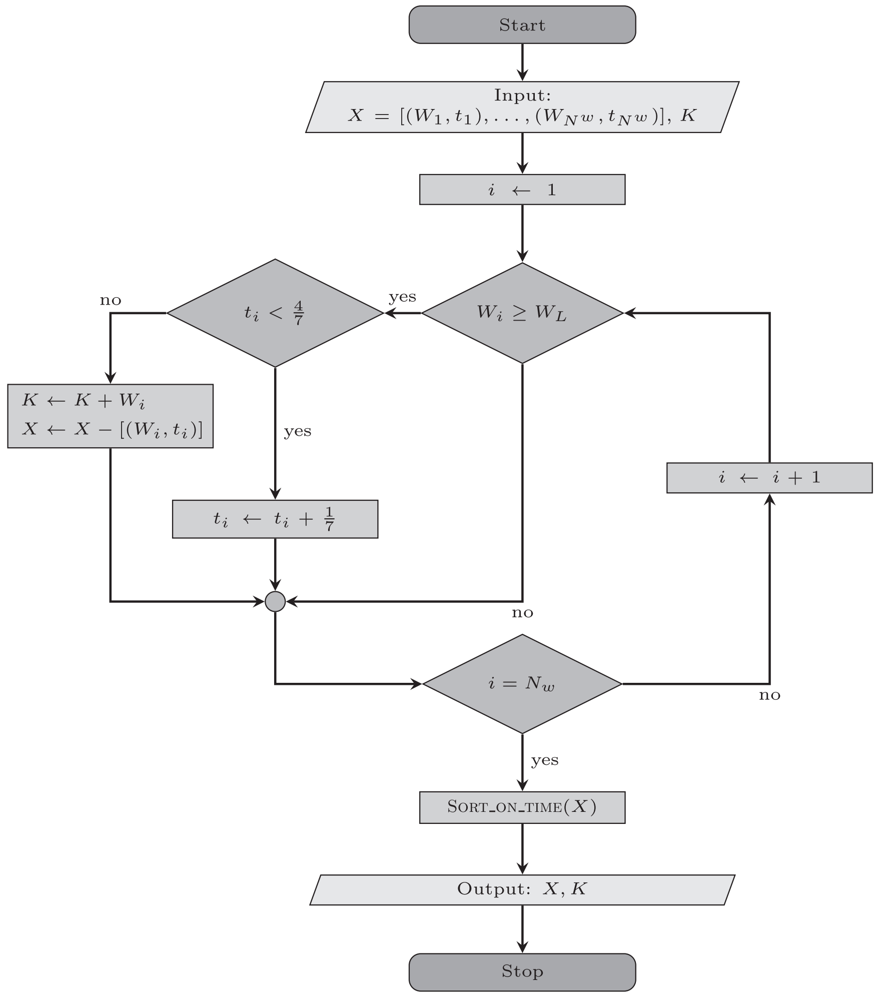

Fig. 11. Pre-processing steps for withdrawals. Weekly algorithm.

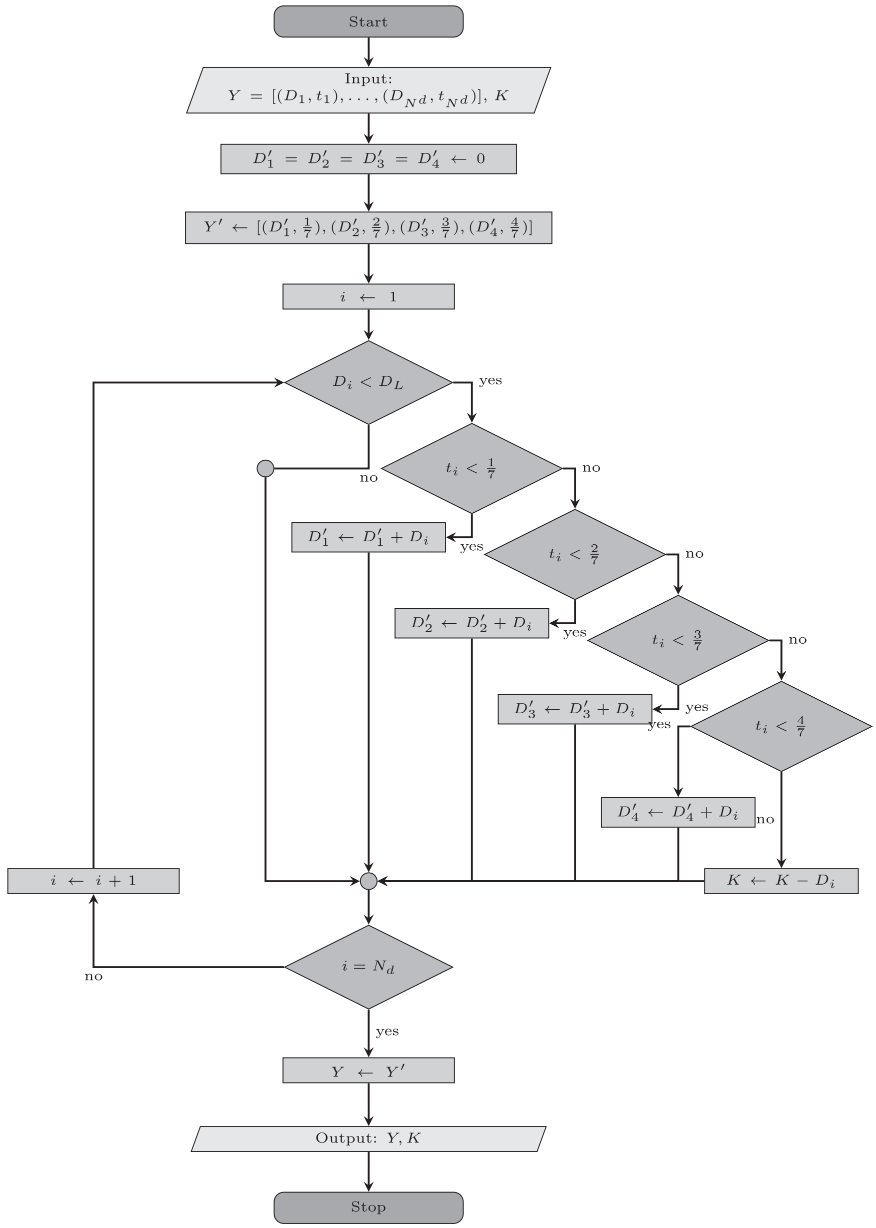

Fig. 12. Pre-processing steps for deposits. Weekly algorithm.

Appendix B. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.omega.2016.09.005.

References

[1] Allen F, Gale D. Financial intermediaries and markets. Econometrica 2004;72:1023–61. http://dx.doi.org/10.1111/j.1468-0262.2004.00525.x.

[2] André M, Mahy G, Lejeune P, Bogaert J. Toward a synthesis of the concept and a definition of the zones in the urban-rural gradient. Biotechnology, Agronomy and Society and Environment 2014;18(1):61–74.

[3] Álvarez F, Lippi F. Financial innovation and the transactions demand for cash. Econometrica 2009;77(2):363–402. http://dx.doi.org/10.3982/ECTA7451.

[4] Álvarez F, Lippi F. The demand of liquid assets with uncertain lumpy purchases. Journal of Monetary Economics 2010;60:753–70. http://dx.doi.org/10.1016/j.jmoneco.2013.05.008.

[5] Bar-Illan A. Overdraft and the demand for money. The American Economic Review 1990;80:201–1216. http://dx.doi.org/10.1002/mde.973.

[6] Baumol WJ. The transaction demand for the cash: an inventory theoretic approach. Quarterly Journal of Economics 1952;66:545–56. http://dx.doi.org/10.2307/1882104.

[7] Bennett P, Peristiani S. Are reserve requirements still binding? Economic Policy Review 2002;8(1), Federal Reserve Bank of New York.

[8] Berger AN, Leusner JH, Mingo JJ. The efficiency of bank branches. Journal of Monetary Economics 1997;40:141–62. http://dx.doi.org/10.1016/S0304-3932(97)00035-4.

[9] Botelho M, Seido MN. Evolutionary models in cash management policies with multiple assets. Economic Modelling 2014;39:1–7. http://dx.doi.org/10.4301/S1807-17752013000300006.

[10] Cardona L, Moreno LA. Cash management cost reduction using data mining to forecast cash demand and LP to optimize resources. Memetic Computational Intelligence and Complexity 2012;4:127–34. http://dx.doi.org/10.1007/s12293-012-0080-4.

[11] Cook DW, Hababou M. Sales performance measurement in bank branches. Omega 2001;29:299–307. http://dx.doi.org/10.1016/S0305-0483(01)00025-1.

[12] Chiang DT. Liquidity planning in a small bank. Omega 1979;4(7):287–95. http://dx.doi.org/10.1016/0305-0483(79)90034-3.

[13] Das A, Ray S, Nag A. Labor-use efficiency in Indian banking: a branch-level analysis. Omega 2009;37:411–25. http://dx.doi.org/10.1016/j.omega.2007.05.002.

[14] Diamond DW, Rajan RC. Fear of fire sales, illiquidity seeking, and credit freezes. Quarterly Journal of Economics 2011;126:557–91. http://dx.doi.org/10.1093/qje/qjr012.

Please cite this article as: García Cabello J, Lobillo FJ. Sound branch cash management for less: A low-cost forecasting algorithm under uncertain demand. Omega (2016), http://dx.doi.org/10.1016/j.omega.2016.09.005

[15] Fecht F, Nyborg KG, Rocholl J. The price of liquidity: the effects of market conditions and bank characteristics. Journal of Financial Economics 2011;102:344–62. http://dx.doi.org/10.1016/j.jfineco.2011.05.015.

[16] Ferreira F, Santos S, Rodrigues P, Spahr R. Evaluating retail banking service quality and convenience with MCDA techniques: a case study at the bank branch level. Journal of Business Economics and Management 2014;15(1):1–21. http://dx.doi.org/10.3846/16111699.2012.673504.

[17] Frenkel JA, Jovanovic B. On transactions and precautionary demand for money. The Quarterly Journal of Economics 1980;94:24–43. http://dx.doi.org/10.3386/w0288.

[18] García Cabello J. Cash efficiency for bank branches. Springerplus 2013;2(1):1–15. http://dx.doi.org/10.1186/2193-1801-2-334.

[19] Geismar N, Dawande M, Rajamani D, Sriskandarajah C. Managing a bank's currency inventory under new Federal Reserve guidelines. Manufacturing & Service Operations Management 2007;9(2):147–67. http://dx.doi.org/10.1287/msom.1060.0124.

[20] Gregory G. Cash flow models: a review. Omega 1976;6(4):643–56. http://dx.doi.org/10.1016/0305-0483(76)90092-X.

[21] Guiso L, Sapienza P, Zingales L. Does local financial development matter? The Quarterly Journal of Economics 2004;119(3):929–69.

[22] Ioannou G, Mavri M. Performance-net: a decision support system for reconfiguring a bank's branch network. Omega 2007;35:190–201. http://dx.doi.org/10.1016/j.omega.2005.05.007.

[23] Keister T, McAndrews J. Why are banks holding so many excess reserves? Federal Reserve Bank of new york staff reports, no. 380; 2009, July.

[24] Loutskina E. The role of securitization in bank liquidity and funding management. Journal of Financial Economics 2011;100:663–84. http://dx.doi.org/10.1016/j.jfineco.2011.02.005.

[25] Mehrotra M, Dawande M, Sriskandarajah C. A depository institution's optimal currency supply network under the Fed's new guidelines: operating policies, logistics, and impact. Production and Operations Management 2010;19(6):709–24. http://dx.doi.org/10.1111/j.1937-5956.2010.01153.x.

[26] Melo MAS, Bilich F. Expectancy balance model for cash flow. Journal of Economics and Finance 2011;37(2):240–52. http://dx.doi.org/10.1007/s12197-011-9180-0.

[27] Miller M, Orr D. A model of the demand of money by firms. Quarterly Journal of Economics 1966;80:413–35. http://dx.doi.org/10.2307/1880728.

[28] Ming CX, Hao W, Peng ZZ. IBM reports ⟨http://www.ibm.com/developerworks/websphere/library/techarticles/ind-wmbtt⟩; 2010.

[29] Nair SK, Anderson RG. A specialized inventory problem in banks: optimizing retail sweeps. Production and Operations Management 2008;17(3):285–95. http://dx.doi.org/10.3401/poms.1080.0028.

[30] Paradi JC, Zhu H. A survey on bank branch efficiency and performance research with data envelopment analysis. Omega 2013;41:61–79. http://dx.doi.org/10.1016/j.omega.2011.08.010.

[31] Rajamani D, Geismar HN, Sriskandarajah C. A framework to analyze cash supply chains. Production and Operations Management 2006;15(4):544–52. http://dx.doi.org/10.1111/j.1937-5956.2006.tb00162.x.

[32] SageMath. The Sage mathematics software system (version 6.6). The Sage developers ⟨http://www.sagemath.org⟩. http://dx.doi.org/10.5281/zenodo.17093; 2015.

[33] Tobin J. The interest-elasticity of transactions demand for cash. Review of Economics and Statistics 1956;38(3):241–7. http://dx.doi.org/10.2307/1925776.

[34] Zhiukov S. A literature review on models of inventory management under uncertainty. Business Systems and Economics 2015;5(1):26–35. http://dx.doi.org/10.13165/VSE-15-5-1-03.

[35] Zhu Y, Dawande M, Sriskandarajah CH. Value of local cash reuse: inventory models for medium-size depository institutions under the new federal reserve policy. Manufacturing & Service Operations Management 2011;13(4):508–24. http://dx.doi.org/10.1287/msom.1110.0336.

Please cite this article as: García Cabello J, Lobillo FJ. Sound branch cash management for less: A low-cost forecasting algorithm under uncertain demand. Omega (2016), http://dx.doi.org/10.1016/j.omega.2016.09.005