



HAL
open science

Optimal cash management with uncertain, interrelated and bounded demands

Pascal Schroeder, Imed Kacem

► **To cite this version:**

Pascal Schroeder, Imed Kacem. Optimal cash management with uncertain, interrelated and bounded demands. *Computers & Industrial Engineering*, 2019, 133, pp.195-206. 10.1016/j.cie.2019.04.052 . hal-03487003

HAL Id: hal-03487003

<https://hal.science/hal-03487003>

Submitted on 20 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Optimal cash management with uncertain, interrelated and bounded demands

Pascal Schroeder, Imed Kacem

LCOMS EA7306

Université de Lorraine, Metz, France

Abstract

We consider the cash management problem with uncertain demands. These demands may be interrelated and globally bounded. Most research done in this field assumes a given information about a probability distribution of demands. In practice, however, it is difficult to correctly estimate this distribution. Here, we report on an optimal online algorithm called *balanced cash supply for interrelated demands* (BCSID) which is free of any distribution assumption. BCSID has information about the minimal and maximal ratio of two consecutive demands. The new algorithm has the smallest relative difference in relation to the optimal result when demands are adverse. Hence, our algorithm is particularly relevant for risk averse cash managers. Furthermore, we provide a heuristic algorithm called *approximated balanced bounded cash supply for interrelated demands* (aBBCSID). This algorithm has the same information as does BCSID; it also has information about the global bounds of the demands. In addition, we carry out extensive numerical testing and compare these two algorithms to simple benchmarks and the solution of [11]. The testing proves the practical relevance of BCSID and aBBCSID.

Keywords: combinatorial optimization, cash management problem, online algorithm, interrelated demands, bounded demands, competitive analysis

Email addresses: pascal.schroeder@univ-lorraine.fr (Pascal Schroeder),
imed.kacem@univ-lorraine.fr (Imed Kacem)

Preprint submitted to Computers and Industrial Engineering

April 28, 2019

Optimal cash management with uncertain, interrelated and bounded demands

Abstract

We consider the cash management problem with uncertain demands. These demands may be interrelated and globally bounded. Most research done in this field assumes a given information about a probability distribution of demands. In practice, however, it is difficult to correctly estimate this distribution. Here, we report on an optimal online algorithm called *balanced cash supply for interrelated demands* (BCSID) which is free of any distribution assumption. BCSID has information about the minimal and maximal ratio of two consecutive demands. The new algorithm has the smallest relative difference in relation to the optimal result when demands are adverse. Hence, our algorithm is particularly relevant for risk averse cash managers. Furthermore, we provide a heuristic algorithm called *approximated balanced bounded cash supply for interrelated demands* (aBBCSID). This algorithm has the same information as does BCSID; it also has information about the global bounds of the demands. In addition, we carry out extensive numerical testing and compare these two algorithms to simple benchmarks and the solution of [11]. The testing proves the practical relevance of BCSID and aBBCSID.

Keywords: combinatorial optimization, cash management problem, online algorithm, interrelated demands, bounded demands, competitive analysis

1. Introduction

The main cash management problem was introduced by [3]. Here, an investor holds two distinct types of assets, the first one being a "cash balance into which periodic receipts of income are deposited and from which a steady flow of expenditures are made" ([12], p. 413); the second asset is a non-cash asset which bears interest at a given constant rate. The investor is allowed to convert one asset into the other at any period; regardless of the direction

of the conversion, the investor must pay a "broker fee". Also, conversion takes place instantaneously such that there is no need for a "buffer stock". Baumol's model in essence answers the question of when and how many units of the non-cash asset to convert into cash.

From a recent review by [14], we observe that most current contributions to the cash management problem focus on stochastic programming and robust optimization. Stochastic programming assumes that information about the distribution of the uncertain input is given. In [4], the authors derive the optimal transaction policy by applying a stochastic maximum principle. In [7], a multi-stage stochastic linear program is formulated and various kinds of assets are considered. In [2], a multidimensional cash management system where cash balances fluctuate as a homogeneous diffusion process is considered and an optimal solution is developed. Another reference for a stochastic model is presented in [9]; the authors develop a solution which minimizes the transaction costs when cash flows are not independently or identically distributed.

In contrast to stochastic models where distributions are required, robust optimization utilizes a set for the uncertain data ([10]). [15] proposed an effective solution for the cash management problem of stationary companies based on mixed-integer linear programming and robust optimization. Moreover, [16] incorporated risk preferences for cash management decisions; they developed a multi-objective cash management model based on compromise programming. In [13], particle swarm optimization and genetic algorithms are applied to the cash management problem using various assets; again, the objective is to minimize the total costs of cash management. More recently, [17] provided a formal definition of cash management systems with multiple accounts.

Our focus, however, is not on stochastic means nor robust optimization to solve the cash management problem; rather, we focus on developing algorithms which minimize the competitive ratio and are thus optimal online. [20] established this ratio which is widely used for other online financial problems, e.g. online portfolio selection [5] and automated credit rating prediction [8]. In simple terms, the competitive ratio of an online algorithm problem is the maximum possible quotient in terms of costs incurred by the online algorithm and the best possible solution.

Finding optimal online algorithms is also referred to as online optimization and is more appropriate for sequential decision making problems for which no distributions are available (see [10]). According to [6], an algorithm is

online if it computes and proposes an output o_t before knowing about the input i_t at each period t .

[11] transformed the cash management problem into an online problem. They considered the cash management problem under uncertain demands; the demands are uncertain, since they are revealed sequentially and are not known in advance. A demand D_t is an amount of cash needed to pay expenditures during a time period t . An investor has two types of assets, cash and earning assets. All expenditures must be paid in cash; only earning assets bear interests. Both types of assets can be converted into the other type for a fixed price. [11] assume that the investor owns a sufficient amount of earning assets. Therefore, we disregard cash-to-asset conversion.

If the supply of cash S_t is larger than the cash demand D_t during period t , then the surplus is transferred back into the earning asset. The investor incurs opportunity costs for every unit that was converted too early. In the case of a cash deficit, the investor has to borrow the needed additional cash. The deficit is made up at the beginning of the next period when the investor converts into cash.

The investor wants to minimize the total costs associated with the expenditures. For this to happen, information of the exact amount of future demands is needed. In online problems, it is common to have some information about the input sequence which is to be revealed. In [11], hindsight information about the minimum (m) and maximum (M) amount of needed cash is given, i.e. $D_t \in [m, M] \forall t$. In other words, the demands are bounded from above (below) by M (m). We refer to this model as the model for bounded demands.

Inspired by the work of [21], [19] and [18] considered interrelated prices in the field of online conversion problems. In essence, they derived online algorithms that always select the price which implies the lowest competitive ratio as defined by [20]. The competitive ratio is proven to be a coherent risk measure according to [1]; furthermore, it takes worst-case scenarios into account, which can be essential for decision making.

We model the demands to be interrelated such that demands vary between the factors θ_1 and θ_2 for two consecutive periods; the overall number of periods is T . In other words, $D_t \in [D_{t-1}\theta_1, D_{t-1}\theta_2]$ for $t = 1, \dots, T$. We refer to this model as the model for interrelated demands. Here we have information about θ_1 and θ_2 . We actuate the assumption of demands being interrelated by the following idea. In real-world applications, cash managers often use the so called scenario planning technique in order to make decision. Among other

ideas, this technique uses the current position (the currently needed cash) and predicts extreme scenarios (the maximum relative increase and decrease of needed cash) until the next point of decision making. Another reason for modeling demands to be interrelated is the fact that for many departments within a business, the needed cash is allowed to only differ slightly from the previously demanded cash. In addition, we only consider cash outflows and disregard any inflows, i.e. $\theta_1, \theta_2 > 0$. The reason for this can be found in the fact that there is a substantial amount of departments within a company that do not achieve a positive net income. For instance, the human resource, R&D and marketing department of a company are usually not the recipients of income cash flows, although their costs are substantial. Since we consider only cash outflows, it must follow that we assume the number of earning assets to be sufficiently large to cover all possible future demands. This is justified by the observation, that many businesses constantly invest part of their earned cash inflows in some kind of earning asset (stocks, bonds, etc.); only a small portion of their actual wealth is kept as a liquid asset. A third demand model is the combination of the model used by [11] and our model with interrelated demands. In other words,

$$D_t \in [\min(M, \max(D_{t-1}\theta_1, m)), \max(m, \min(D_{t-1}\theta_2, M))] \quad (1)$$

for $t = 1, \dots, T$.

All three models are illustrated in Figure 1. The red (blue) line illustrates that demand sequence where the highest (lowest) demands occur; the green line illustrates an arbitrary sequence which is different from those two sequences. All models with bounded demands have a previously given maximum M (minimum m) threshold value that cannot be exceeded; in our example, we have $m = 1$ and $M = 10$. For the model with bounded demands, an arbitrary sequence can contain arbitrary demands between m and M , i.e. $D_t \in [m, M]$. As opposed to the other models, the demand can surge (fall) from m (M) to M (m) between two arbitrary periods. The red (blue) line remains constant at M (m), while the arbitrary demand sequence (green line) contains the following values: $D_0 = 5, D_1 = 1, D_2 = 10, D_3 = 1$ and $D_4 = 10$.

For the model with interrelated demands, a demand D_t cannot go outside the value of $D_{t-1}\theta_1$ and $D_{t-1}\theta_2$; in our example, we have $\theta_1 = 0.5$ and $\theta_2 = 1.5$. The red line illustrates the demand sequence with the highest possible demands $D_0 = 5, D_1 = 7.5, D_2 = 11.25, D_3 = 16.88$ and $D_4 = 25.31$.

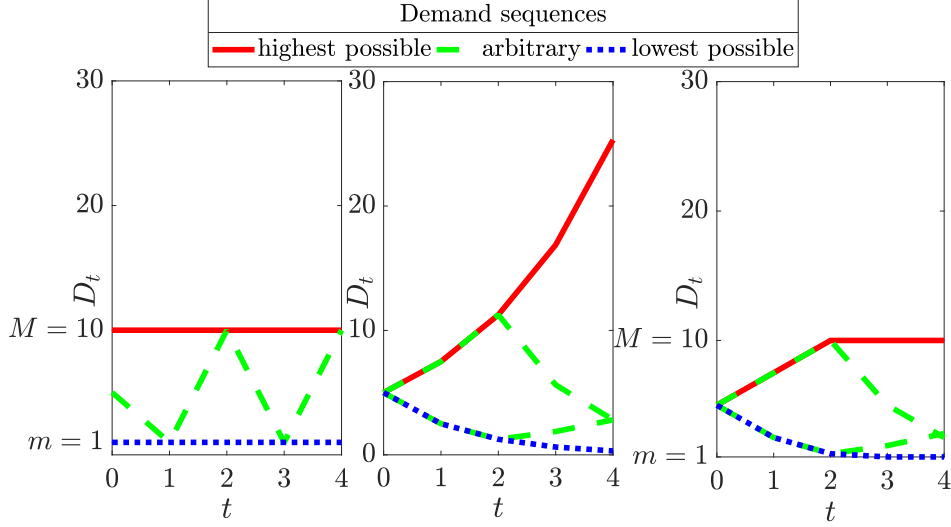


Figure 1: Model with bounded demands used by [11] (left), model with interrelated demands (middle) and model with interrelated, bounded demands (right); the red (blue) line gives the highest (lowest) possible demand sequence, the green line represents an arbitrary demand sequence

The blue line represents the demand sequence with the lowest possible demands $D_0 = 5, D_1 = 2.5, D_2 = 1.25, D_3 = 0.63$ and $D_4 = 0, 31$. Two arbitrary sequences (green lines) are illustrated, one rising from $D_0 = 5$ to $D_2 = D_0\theta_2^2 = 11.25$ and then falling to $D_4 = D_2\theta_1^2 = 2.81$; the other one falls from $D_0 = 5$ to $D_2 = D_0\theta_1^2 = 1.25$ and then rises to $D_4 = D_2\theta_2^2 = 2.81$.

For the model with interrelated, bounded demands, a demand D_t cannot go outside the value of $D_{t-1}\theta_1$ and $D_{t-1}\theta_2$ nor m and M . The red line illustrates the demand sequence with the highest possible demands $D_0 = 5, D_1 = 7.5, D_2 = M = 10 = D_3 = D_4$. The blue line represents the demand sequence with the lowest possible demands $D_0 = 5, D_1 = 2.5, D_2 = 1.25, D_3 = m = 1 = D_4$. Two arbitrary sequences (green lines) are illustrated, one rising from $D_0 = 5$ to $D_2 = \min(D_0\theta_2^2 = 11.25, M = 10) = 10$ and then falling to $D_4 = D_2\theta_1^2 = 10 \cdot 0.5^2 = 2.50$; the other one falls from $D_0 = 5$ to $D_2 = \max(D_0\theta_1^2 = 1.25, m = 1) = 1.25$ and then rises to $D_4 = D_2\theta_2^2 = 1.25 \cdot 1.5^2 = 2.81$.

The contributions of this paper are the following:

1. We present the online algorithm *balanced cash supply with interrelated demands* (BCSID) for given θ_1, θ_2 and T
2. We prove the competitive ratio and optimality of BCSID
3. We present the heuristic online algorithm called *approximated balanced bounded cash supply with interrelated demands* (aBBCSID) for given $\theta_1, \theta_2, m, M, T$ and initial demand D_0
4. We carry out a numerical testing to compare the performance of BCSID and aBBCSID to some simple heuristics and the optimal solution for bounded demands given by [11]

The paper is organized in five sections. In Section 2, the formal problem description and the concepts of competitive ratio and optimality are given. In Section 3, we first prove the worst-case sequence $\mathbf{D}^{wc}(\mathbf{S})$ for a selected supply sequence \mathbf{S} . Based on this, we derive BCSID and provide proofs of its competitive ratio, optimality, and behavior for uncertain θ_1 and θ_2 . We present the heuristic algorithm aBBCSID. In Section 4, we present and execute the numerical testing. Section 5 concludes this paper.

2. Preliminaries

In the cash management problem under uncertain demands an online player (ON) must make a decision on how much cash S_t to convert (or extract) from its (limitless) earning assets at the beginning of every period $t = 1, \dots, T$. If the number of earning assets were finite, then some S_t would require taking out a loan which has to be paid back with interests. This would enhance the complexity of the problem. The assumption of the earning assets being limitless is made to avoid this complexity.

The demand for cash D_t is revealed at the end of t . Note that at $t = 1$ we know the exact value of D_0 . The sequences of all supplies S_t and all demands D_t are denoted as \mathbf{S} and \mathbf{D} respectively. The sets of all the feasible sequences of \mathbf{S} and \mathbf{D} are denoted as \mathcal{S} and \mathcal{D} . All demand and supply sequences must originate from these sets. Formally, we have

$$\mathbf{S} = S_1, \dots, S_T, \quad \mathbf{S} \in \mathcal{S} \text{ and } \mathbf{D} = D_1, \dots, D_T, \quad \mathbf{D} \in \mathcal{D}. \quad (2)$$

For every converted unit of cash, ON incurs a transaction cost c . Having selected S_t , ON incurs accumulated transaction costs cS_t at the beginning of every period t . When D_t is revealed at the end of t , there are three possible scenarios:

1. The supply of cash extracted equals the demand of cash, $S_t = D_t$. In this case, the incurred total costs of ON are cD_t .
2. Too much cash has been extracted, i.e. $S_t > D_t$. We first incur the transaction costs cS_t , which can be separated into cD_t and $c(S_t - D_t)$ (because $S_t > D_t$) for the excess cash. An excess unit cannot yield the interest rate i and must be transferred back into the earning asset for cost c_{back} . Hence, every excess unit incurs opportunity cost h with $h = c + i + c_{back}$. In this scenario, the incurred total costs of ON are

$$cD_t + c(S_t - D_t) + i(S_t - D_t) + c_{back}(S_t - D_t) = cD_t + h(S_t - D_t). \quad (3)$$

3. Too little cash has been extracted, i.e. $S_t < D_t$. Then we incur credit costs $(D_t - S_t)j$ at the end of t ; j is the difference of interest rate paid to the creditor and interest rate of the earning asset per unit. We set this difference to be always non-negative. In other words, j is the shortage cost factor for each unit of excess demand. At the end of t , we have to convert the missing $D_t - S_t$ and incur also transaction costs $c(D_t - S_t)$. In this scenario, the incurred total costs of ON are

$$cS_t + c(D_t - S_t) + j(D_t - S_t) = cD_t + j(D_t - S_t). \quad (4)$$

Contrary to other models, we only apply opportunity costs to the excess cash not invested; the rationale for this choice is the point of view that if S_t meets exactly the demand D_t , then ON incurs the lowest costs possible. A positive deviation excludes ON from gaining further interest payments. A negative deviation, however, allows ON to receive more interest payments. This is overcompensated by the incurred credit costs; in other words, the interest bearing rate of the earning asset is assumed to be inferior or equal to the interest rate paid to a creditor. If this was not the case, then the solution to never extract any cash at the beginning of t would always incur the lowest cost possible.

From the above points, we see that it is always best to convert exactly D_t at t since ON only incurs the costs cD_t and avoids any additional costs. If we knew the future demands, then we could always ensure that $S_t = D_t$ for $t = 1, \dots, T$, or in other words, $\mathbf{S} = \mathbf{D}$. We denote such a player with perfect information as OPT. The costs of ON for extracting the cash supply S_t at t for an uncertain demand D_t is (regardless of the occurring scenario)

$$cD_t + j \max(0, D_t - S_t) + h \max(0, S_t - D_t), \quad (5)$$

while OPT's incurred costs are cD_t .

The cumulated costs $ON(\mathbf{D}, \mathbf{S})_t$ for ON after an arbitrary period t are

$$ON(\mathbf{D}, \mathbf{S})_t = \sum_{\tau=1}^t cD_\tau + j \max(0, D_\tau - S_\tau) + h \max(0, S_\tau - D_\tau), \quad (6)$$

with $ON(\mathbf{D}, \mathbf{S})_T = ON(\mathbf{D}, \mathbf{S})$ being the total costs of ON using \mathbf{S} on sequence \mathbf{D} . OPT's accumulated costs $OPT(\mathbf{D})_t$ after period t are

$$OPT(\mathbf{D})_t = c \sum_{\tau=1}^t D_\tau, \quad (7)$$

with $OPT(\mathbf{D})_T = OPT(\mathbf{D})$ being the total costs of OPT on sequence \mathbf{D} . The ratio of $ON(\mathbf{D}, \mathbf{S})_t$ and $OPT(\mathbf{D})_t$ is denoted as $r(\mathbf{D}, \mathbf{S})_t$.

There exists many different types of costs which find relevance in literature, e.g. fix and variable holding costs. In this work, we focus merely on variable credit and opportunity costs.

The ratio of ON using \mathbf{S} and OPT on demand sequence \mathbf{D} is

$$\begin{aligned} \frac{ON(\mathbf{D}, \mathbf{S})}{OPT(\mathbf{D})} &= 1 + \frac{\sum_{t=1}^T j \max(0, D_t - S_t) + h \max(0, S_t - D_t)}{\sum_{t=1}^T cD_t} \\ &= r(\mathbf{D}, \mathbf{S})_T. \end{aligned} \quad (8)$$

We want to minimize $r(\mathbf{D}, \mathbf{S})_T$, its value being greater than or equal to 1. Since the performance of a generic online algorithm ALG depends merely on the sequence \mathbf{D} , on which ALG has no influence, and the decision variable \mathbf{S} , we state the following: the worst-case scenario ratio, i.e. the competitive ratio r^{ALG} (see [20]), of ALG using \mathbf{S}^{ALG} is

$$r^{ALG} = \max_{\mathbf{D} \in \mathcal{D}} r(\mathbf{D}, \mathbf{S}^{ALG})_T. \quad (9)$$

Another way to interpret above equation is the following: independent of the demand sequence $\mathbf{D} \in \mathcal{D}$, ALG always incurs at most r^{ALG} times of the costs which OPT achieves, formally

$$ON(\mathbf{D}, \mathbf{S}^{ALG}) \leq r^{ALG} OPT(\mathbf{D}) \forall \mathbf{D} \in \mathcal{D}. \quad (10)$$

An online algorithm is optimal if it incurs the smallest r possible, i.e. there exists no other online algorithm which guarantees a smaller r than the one

guaranteed by ALG'. In other words, ALG' is optimal if it always selects the decision variable $\mathbf{S}^{ALG'}$ such that

$$\mathbf{S}^{ALG'} = \arg \min_{\mathbf{S} \in \mathcal{S}} \max_{\mathbf{D} \in \mathcal{D}} r(\mathbf{D}, \mathbf{S})_T. \quad (11)$$

An online problem can be considered solved (from a worst-case perspective) if there exists an optimal online algorithm. It is not possible to create another algorithm with a lower competitive ratio for these problems. The only room for improvement might be time complexity and average-case performance. However, online algorithms with better time complexity and/or average-case performance tend to have a higher competitive ratio and are thus not desirable.

3. Cash Management with uncertain, interrelated demands

In this section we first consider the worst-case demand sequence $\mathbf{D}^{wc}(\mathbf{S})$ for an arbitrary cash supply sequence \mathbf{S} . In other words, we prove the existence and the value of the worst-case demand for a selected S_t in an arbitrary period t regardless of any online algorithms. We find that there is a supply S_t^* which minimizes the total costs (relative to OPT) for this worst-case demand in t . The online algorithm *balanced cash supply for interrelated demands* (BCSID) always selects S_t^* in every period t . Knowing the worst-case demand sequence when selecting S_t^* in each period, we prove the competitive ratio and subsequently the optimality of BCSID.

We assume the demands to be interrelated, that is, $D_t \in [D_{t-1}\theta_1, D_{t-1}\theta_2]$ for $t = 1, \dots, T$ and $0 < \theta_1 \leq \theta_2 < \infty$. In Figure 2, we illustrate possible demand corridors for various parameter combinations of θ_1 and θ_2 , $D_0 = 1$ and $T = 4$. A demand corridor contains all possible demand sequences for given θ_1, θ_2, T and D_0 . As can be seen, using $0 < \theta_1 \leq \theta_2 < \infty$, it is also possible to model demands which are always growing (falling) over time; this can be seen in the red (blue) lines. Demand bounds diverge if $\theta_1 < 1$ and $\theta_2 > 1$ (green lines); in limit, this implies the lower (upper) bound on D_t to be zero (infinity).

We now find the worst-case demand sequence $\mathbf{D}^{wc}(\mathbf{S})$.

Lemma 1. *The elements D_1, \dots, D_T of the worst-case demand sequence $\mathbf{D}^{wc}(\mathbf{S})$ for a selected cash supply strategy $\mathbf{S} \in \mathcal{S}$ obey the following rule*

$$D_t = \begin{cases} D_{t-1}\theta_2 & \text{if } S_t \leq D_{t-1}\theta_1\theta_2 \frac{j+h}{j\theta_1+h\theta_2} \\ D_{t-1}\theta_1 & \text{else} \end{cases} \quad (12)$$

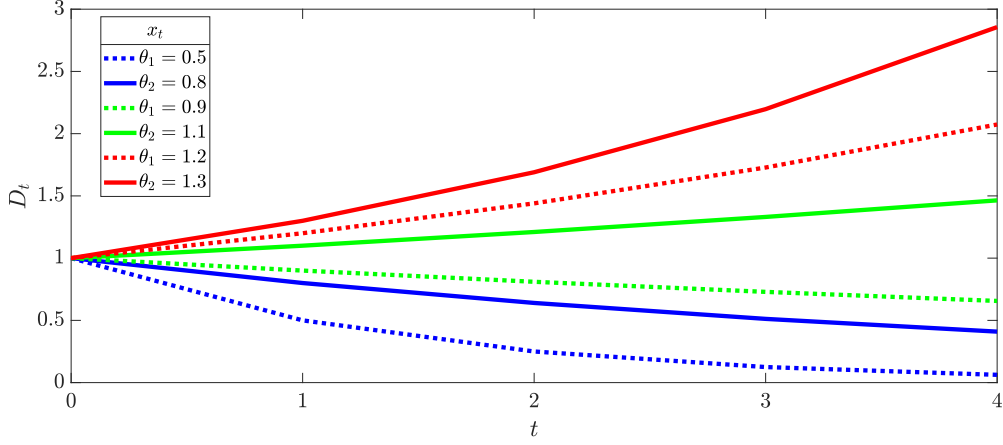


Figure 2: Possible demand corridors for variable parameter combinations of θ_1 and θ_2 , $D_0 = 1$ and $T = 4$

for $t = 1, \dots, T$ and given D_0 .

Proof. After an arbitrary period $t = 1, \dots, T$ the ratio of ON and OPT (denoted as ρ) is at most

$$\rho = \max_{D_t} \frac{cD_t + j \max(0, D_t - S_t) + h \max(0, S_t - D_t)}{cD_t}. \quad (13)$$

Note that only D_t is considered. We know that demands are interrelated; this alters ρ such that

$$\begin{aligned} \rho &= \max_{x_t \in [\theta_1, \theta_2]} \frac{cD_{t-1}x_t + j \max(0, D_{t-1}x_t - S_t) + h \max(0, S_t - D_{t-1}x_t)}{cD_{t-1}x_t} \\ &= 1 + \frac{1}{c} \max_{x_t \in [\theta_1, \theta_2]} \left(j \max\left(0, 1 - \frac{S_t}{D_{t-1}x_t}\right) + h \max\left(0, \frac{S_t}{D_{t-1}x_t} - 1\right) \right). \end{aligned} \quad (14)$$

For $x_t = \theta_2$ the term $\max(0, 1 - S_t (D_{t-1}x_t)^{-1})$ is maximized (compared to $\theta_1 \leq x_t < \theta_2$), while $x_t = \theta_1$ maximizes the term $\max(0, S_t (D_{t-1}x_t)^{-1} - 1)$ (compared to $\theta_1 < x_t \leq \theta_2$). Hence, instead of considering the impact on ρ for $x_t \in [\theta_1, \theta_2]$, we merely need to investigate the impact on ρ for $x_t = \theta_1$ and $x_t = \theta_2$. Rationally, S_t should not be higher than $D_{t-1}\theta_2$, nor lower than

$D_{t-1}\theta_1$. Therefore, ρ simplifies to

$$\begin{aligned}
\rho &= 1 + \frac{1}{c} \max_{x_t \in \{\theta_1, \theta_2\}} \left(j \max \left(0, 1 - \frac{S_t}{D_{t-1}x_t} \right) + h \max \left(0, \frac{S_t}{D_{t-1}x_t} - 1 \right) \right) \quad (15) \\
&= 1 + \frac{1}{c} \max \left(j \max \left(0, 1 - \frac{S_t}{D_{t-1}\theta_1} \right) + h \max \left(0, \frac{S_t}{D_{t-1}\theta_1} - 1 \right), \right. \\
&\quad \left. j \max \left(0, 1 - \frac{S_t}{D_{t-1}\theta_2} \right) + h \max \left(0, \frac{S_t}{D_{t-1}\theta_2} - 1 \right) \right) \\
&= 1 + \frac{1}{c} \max \left(0 + h \left(\frac{S_t}{D_{t-1}\theta_1} - 1 \right), j \left(1 - \frac{S_t}{D_{t-1}\theta_2} \right) + 0 \right).
\end{aligned}$$

We now search for that S_t^* which minimizes ρ . Since lowering (raising) S_t implies raising the right (left)-hand side of the max expression, there must exist one S_t^* for which both elements have the same value, formally

$$h \left(\frac{S_t^*}{D_{t-1}\theta_1} - 1 \right) = j \left(1 - \frac{S_t^*}{D_{t-1}\theta_2} \right) \text{ and finally } S_t^* = D_{t-1}\theta_1\theta_2 \frac{j+h}{j\theta_1+h\theta_2}. \quad (16)$$

For $S_t = S_t^*$, ρ becomes

$$\begin{aligned}
\rho &= 1 + \frac{1}{c} \max \left(h \left(\theta_2 \frac{j+h}{j\theta_1+h\theta_2} - 1 \right), j \left(1 - \theta_1 \frac{j+h}{j\theta_1+h\theta_2} \right) \right) \quad (17) \\
&= 1 + \frac{1}{c} \max \left(h j \frac{\theta_2 - \theta_1}{j\theta_1+h\theta_2}, h j \frac{\theta_2 - \theta_1}{j\theta_1+h\theta_2} \right) \\
&= 1 + \frac{hj}{c} \frac{\theta_2 - \theta_1}{j\theta_1+h\theta_2}.
\end{aligned}$$

We now consider a deviation from S_t^* and show that the **resulting ρ is** not lower. Assume we employ $S_t = S_t^* + \epsilon_t$. Then we have

$$\begin{aligned}
\rho &= 1 + \frac{1}{c} \max \left(h \left(\frac{S_t^* + \epsilon_t}{D_{t-1}\theta_1} - 1 \right), j \left(1 - \frac{S_t^* + \epsilon_t}{D_{t-1}\theta_2} \right) \right) \\
&= 1 + \frac{1}{c} \max \left(h \left(\frac{D_{t-1}\theta_1\theta_2 \frac{j+h}{j\theta_1+h\theta_2} + \epsilon_t}{D_{t-1}\theta_1} - 1 \right), j \left(1 - \frac{D_{t-1}\theta_1\theta_2 \frac{j+h}{j\theta_1+h\theta_2} + \epsilon_t}{D_{t-1}\theta_2} \right) \right) \\
&= 1 + \frac{1}{c} \max \left(h \left(\theta_2 \frac{j+h}{j\theta_1+h\theta_2} + \frac{\epsilon_t}{D_{t-1}\theta_1} - 1 \right), j \left(1 - \theta_1 \frac{j+h}{j\theta_1+h\theta_2} - \frac{\epsilon_t}{D_{t-1}\theta_2} \right) \right) \quad (18)
\end{aligned}$$

with ϵ_t being the absolute difference between the selected S_t and S_t^* ($\epsilon_t \in \mathbb{R} \forall t$). For $\epsilon_t < 0$, we find that

$$h \left(\theta_2 \frac{j+h}{j\theta_1+h\theta_2} + \frac{\epsilon_t}{D_{t-1}\theta_1} - 1 \right) < h j \frac{\theta_2 - \theta_1}{j\theta_1+h\theta_2}. \quad (19)$$

However, we also find that

$$j \left(1 - \theta_1 \frac{j+h}{j\theta_1 + h\theta_2} - \frac{\epsilon_t}{D_{t-1}\theta_2} \right) > hj \frac{\theta_2 - \theta_1}{j\theta_1 + h\theta_2}. \quad (20)$$

In practical words, if ON possesses less cash than S_t^* , then the demand D_{t-1} will rise by the factor θ_2 in the worst-case. For $0 < \epsilon$, we observe the opposite. If ON's amount of cash is higher than S_t^* , then the demand will fall by the factor θ_1 in the worst-case. Since we consider the demand and supply for an arbitrary period t , we conclude that the worst-case demand sequence continues to behave as mentioned above, i.e. if the actual cash supply is below or equal to S_t^* , then the demand rises by the factor θ_2 and falls by the factor θ_1 otherwise. This finishes the proof. \square

One simple online algorithm for the cash management problem with uncertain demands is called *learning cash supply* (LCS). The idea behind LCS is the assumption that the demand of yesterday resembles the demand of today.

Algorithm LCS: At period t , convert into cash until you have the amount of S_t^{LCS} available, with

$$S_t^{LCS} = D_{t-1} \quad (21)$$

for all $t = 1, \dots, T$. Note that we do not provide the competitive ratio for LCS since it is not optimal (see example in Table 1) and does not have any hindsight information about the demand sequence; thus, LCS is not in the focus of this work.

We illustrate the importance of Lemma 1 with an example. We use different cash supply strategies, the first one being $\mathbf{S}^* = S_1^*, \dots, S_T^*$, followed by $\mathbf{S}^\epsilon = S_1^* - \epsilon, \dots, S_T^* - \epsilon$ with $\epsilon = \{-0.01, 0.01\}$, \mathbf{S}^- with $S_t^- = 0 \forall t$ and the last one being \mathbf{S}^{LCS} . We set the parameters as follows: $\theta_1 = 0.90, \theta_2 = 1.05, T = 6, D_0 = 10, j = 0.10, h = 0.08$ and $c = 0.01$. Clearly, the worst-case sequence is different for each strategy. Therefore, the worst-case demand for period t using S_t is denoted as $D_t^{wc}(S_t)$.

The resulting supplies, the worst-case demands and the accumulated costs for ON and OPT are illustrated in Table 1 for each strategy. Furthermore, since we deal with worst-case demand sequences for each cash supply sequence, the ON-to-OPT ratio becomes the competitive ratio for the given parameter combination. This ratio is also given in the table. We see that the supply strategy \mathbf{S}^* **does not** incur the lowest overall costs for ON; **however**, it achieves

	t					
	1	2	3	4	5	6
S_t^*	9.78	8.80	7.92	7.13	6.41	5.77
$D_t^{wc}(S_t^*)$	10.50	11.03	11.58	12.16	12.76	13.40
$ON(\mathbf{D}^{wc}(\mathbf{S}^*), \mathbf{S}^*)_t$	0.18	0.36	0.56	0.76	0.98	1.21
$OPT(\mathbf{D}^{wc}(\mathbf{S}^*))_t$	0.11	0.22	0.33	0.45	0.58	0.71
$r(\mathbf{D}^{wc}(\mathbf{S}^*), \mathbf{S}^*)_t$	1.69	1.69	1.69	1.69	1.69	1.69
S_t^-	0	0	0	0	0	0
$D_t^{wc}(S_t^-)$	10.50	11.03	11.58	12.16	12.76	13.40
$ON(\mathbf{D}^{wc}(\mathbf{S}^-), \mathbf{S}^-)_t$	1.16	2.37	3.64	4.98	6.38	7.86
$OPT(\mathbf{D}^{wc}(\mathbf{S}^-))_t$	0.11	0.22	0.33	0.45	0.58	0.71
$r(\mathbf{D}^{wc}(\mathbf{S}^-), \mathbf{S}^-)_t$	11.00	11.00	11.00	11.00	11.00	11.00
$S_t^{-0.01}$	9.77	10.25	10.77	11.31	11.87	12.47
$D_t^{wc}(S_t^{-0.01})$	10.5	11.03	11.58	12.16	12.76	13.40
$ON(\mathbf{D}^{wc}(\mathbf{S}^{-0.01}), \mathbf{S}^{-0.01})_t$	0.18	0.37	0.56	0.77	0.99	1.21
$OPT(\mathbf{D}^{wc}(\mathbf{S}^{-0.01}))_t$	0.11	0.22	0.33	0.45	0.58	0.71
$r(\mathbf{D}^{wc}(\mathbf{S}^{-0.01}), \mathbf{S}^{-0.01})_t$	1.70	1.70	1.70	1.70	1.70	1.70
$S_t^{0.01}$	9.79	8.81	7.93	7.14	6.42	5.78
$D_t^{wc}(S_t^{0.01})$	9.00	8.10	7.29	6.56	5.90	5.31
$ON(\mathbf{D}^{wc}(\mathbf{S}^{0.01}), \mathbf{S}^{0.01})_t$	0.15	0.29	0.41	0.53	0.63	0.72
$OPT(\mathbf{D}^{wc}(\mathbf{S}^{0.01}))_t$	0.09	0.17	0.24	0.31	0.37	0.42
$r(\mathbf{D}^{wc}(\mathbf{S}^{0.01}), \mathbf{S}^{0.01})_t$	1.70	1.70	1.70	1.70	1.70	1.70
S_t^{LCS}	10.00	9.00	8.10	7.29	6.56	5.91
$D_t^{wc}(S_t^{LCS})$	9.00	8.10	7.29	6.56	5.91	5.31
$ON(\mathbf{D}^{wc}(\mathbf{S}^{LCS}), \mathbf{S}^{LCS})_t$	0.17	0.32	0.46	0.59	0.70	0.80
$OPT(\mathbf{D}^{wc}(\mathbf{S}^{LCS}))_t$	0.09	0.17	0.24	0.31	0.37	0.42
$r(\mathbf{D}^{wc}(\mathbf{S}^{LCS}), \mathbf{S}^{LCS})_t$	1.89	1.89	1.89	1.89	1.89	1.89

Table 1: Worst-case demand sequences and incurred ON to OPT ratios for various cash supply strategies with given $\theta_1 = 0.90, \theta_2 = 1.05, T = 6, D_0 = 10, j = 0.10, h = 0.08$ and $c = 0.01$

the lowest competitive ratio compared to all other strategies (1.69) for all t . If ON extracts more than S_t^* (i.e. $\epsilon > 0$), then the worst-case demands will always fall by the factor θ_1 ; the worst-case demands always increase by θ_2 if less than S_t^* (i.e. $\epsilon < 0$) has been extracted. Consequently, for $\epsilon \neq 0$ we incur a higher competitive ratio. Furthermore, we see that extracting no cash at all (using \mathbf{S}^-) incurs the highest costs for ON and the highest competitive

ratio (11.00); clearly, it is not a good strategy for the selected parameter combination. Finally, we note that LCS also incurs a higher competitive ratio. Clearly, LCS is not optimal (in the sense that it incurs the lowest competitive ratio). We further conclude that an online algorithm that uses \mathbf{S}^* is a candidate for fulfilling the optimality criteria.

We now define the algorithm BCSID which always uses S_t^* in the beginning of period t .

Algorithm BCSID: In the beginning of period t , convert into cash until you have the amount of S_t^* available.

BCSID is applicable for worst-case and non-worst-case demand sequences. The latter is illustrated in Section 4, where we demonstrate its performance. For instance, a trivial non-worst-case demand sequence might be \mathbf{D} with $D_1, \dots, D_T = D_0$; the resulting S_t^* is $S_t^* = D_0 \theta_1 \theta_2 \frac{(j+h)}{j\theta_1+h\theta_2} \forall t$. We now prove the competitive ratio and optimality of BCSID.

Theorem 1. *BCSID's competitive ratio is*

$$r^{BCSID} = 1 + \frac{hj(\theta_2 - \theta_1)}{c(j\theta_1 + h\theta_2)} \quad (22)$$

for the cash management problem with opportunity costs h , borrowing costs j , transaction costs c and demand change bounds θ_1 and θ_2 .

Proof. Assume we use $\mathbf{S} = \mathbf{S}^*$. Using Lemma 1, the ratio of ON and OPT in the worst-case sequence $\mathbf{D}^{wc}(\mathbf{S}^*)$ after T periods is

$$\begin{aligned} r(\mathbf{D}^{wc}(\mathbf{S}^*), \mathbf{S}^*)_T &= \frac{\sum_{t=1}^T cD_t^{wc}(S_t^*) + j \max(0, D_t^{wc}(S_t^*) - S_t^*) + h \max(0, S_t^* - D_t^{wc}(S_t^*))}{c \sum_{t=1}^T D_t^{wc}(S_t^*)} \quad (23) \\ &= 1 + \frac{\sum_{t=1}^T j \max(0, D_t^{wc}(S_t^*) - S_t^*) + h \max(0, S_t^* - D_t^{wc}(S_t^*))}{c \sum_{t=1}^T D_t^{wc}(S_t^*)}. \end{aligned}$$

We know that for the output sequence \mathbf{S}^* it is of no importance whether the demands fall or rise by the factor θ_1 or θ_2 between consecutive periods. Therefore, we set

$$\begin{aligned} \mathbf{D}^{wc}(\mathbf{S}^*) &= D_1(S_1^*), \dots, D_T(S_T^*) \\ &= D_0\theta_2, D_1^{wc}(S_1^*)\theta_2, \dots, D_{T-1}^{wc}(S_{T-1}^*)\theta_2, \end{aligned} \quad (24)$$

with $D_0\theta_2 = D_0^{wc}(S_0^*)$. Equation (23) then simplifies to

$$\begin{aligned}
r(\mathbf{D}^{wc}(\mathbf{S}^*), \mathbf{S}^*)_T &= 1 + \frac{\sum_{t=1}^T jD_{t-1}^{wc}(S_{t-1}^*)\theta_2 - S_t^* + h0}{c \sum_{t=1}^T D_{t-1}^{wc}(S_{t-1}^*)\theta_2} \\
&= 1 + \frac{j \left(\sum_{t=1}^T D_{t-1}^{wc}(S_{t-1}^*)\theta_2 - \sum_{t=1}^T S_t^* \right)}{c \sum_{t=1}^T D_{t-1}^{wc}(S_{t-1}^*)\theta_2} \\
&= 1 + \frac{j}{c} - j \frac{\sum_{t=1}^T D_{t-1}^{wc}(S_{t-1}^*)\theta_1\theta_2 \frac{j+h}{j\theta_1+h\theta_2}}{c \sum_{t=1}^T D_{t-1}(S_{t-1}^*)\theta_2} \\
&= 1 + \frac{j}{c} - j\theta_1 \frac{j+h}{j\theta_1+h\theta_2} \frac{\sum_{t=1}^T D_{t-1}^{wc}(S_{t-1}^*)\theta_2}{c \sum_{t=1}^T D_{t-1}^{wc}(S_{t-1}^*)\theta_2} \\
&= 1 + \frac{jh\theta_2 + j^2\theta_1 - j^2\theta_1 - j\theta_2h}{c(j\theta_1+h\theta_2)} \\
&= 1 + \frac{hj(\theta_2 - \theta_1)}{c(j\theta_1+h\theta_2)}.
\end{aligned} \tag{25}$$

This finishes the proof, because $r(\mathbf{D}^{wc}(\mathbf{S}^*), \mathbf{S}^*)_T = r^{BCSID}$. \square

We observe that r^{BCSID} is independent of T .

Theorem 2. *BCSID is an optimal online algorithm for the cash management problem with opportunity costs h , borrowing costs j , transaction costs c and demand change bounds θ_1 and θ_2 .*

Proof. Assume there exists an algorithm ALG' which proposes to extract the amount $S_t^{ALG'} = S_t^* + \epsilon_t$ in each period t . Using Equation (17) and (18), the difference δ_t between the ratio of ON and OPT using S_t^* and the one using $S_t^{ALG'}$ is

$$\begin{aligned}
\delta_t &= \frac{\frac{hj(\theta_2 - \theta_1)}{j\theta_1+h\theta_2} - \max \left(h \left(\theta_2 \frac{j+h}{j\theta_1+h\theta_2} + \frac{\epsilon_t}{D_{t-1}^{wc}\theta_1} - 1 \right), j \left(1 - \theta_1 \frac{j+h}{j\theta_1+h\theta_2} - \frac{\epsilon_t}{D_{t-1}^{wc}\theta_2} \right) \right)}{c} \\
&= \frac{\frac{hj(\theta_2 - \theta_1)}{j\theta_1+h\theta_2} + \min \left(-h \left(\theta_2 \frac{j+h}{j\theta_1+h\theta_2} + \frac{\epsilon_t}{D_{t-1}^{wc}\theta_1} - 1 \right), -j \left(1 - \theta_1 \frac{j+h}{j\theta_1+h\theta_2} - \frac{\epsilon_t}{D_{t-1}^{wc}\theta_2} \right) \right)}{c} \\
&= \frac{\min \left(\frac{hj(\theta_2 - \theta_1)}{j\theta_1+h\theta_2} - \frac{h\theta_2(j+h)}{j\theta_1+h\theta_2} - \frac{h\epsilon_t}{D_{t-1}^{wc}\theta_1} + h, \frac{hj(\theta_2 - \theta_1)}{j\theta_1+h\theta_2} - j + \frac{j\theta_1(j+h)}{j\theta_1+h\theta_2} + \frac{j\epsilon_t}{D_{t-1}^{wc}\theta_2} \right)}{c} \\
&= \frac{\min \left(-\frac{h\epsilon_t}{D_{t-1}^{wc}\theta_1}, \frac{j\epsilon_t}{D_{t-1}^{wc}\theta_2} \right)}{c}.
\end{aligned} \tag{26}$$

From the above equation we see that it is not possible to obtain a positive (greater zero) value for any δ_t . This means that the ratio of ON, that deviates

from S_t^* , and OPT in an arbitrary period $t = 1, \dots, T$ is never lower than the one incurred using S_t^* . We conclude that if no algorithm comes relatively closer to the costs incurred by OPT on a worst-case demand sequence than BCSID for an arbitrary $t = 1, \dots, T$, then there exists no algorithm which guarantees a lower competitive ratio than BCSID. Hence, BCSID is optimal. This finishes the proof. \square

From Theorem 2 we see that BCSID is an optimal online algorithm which, contrary to many algorithms in literature, does not need any information about the distribution of future demands. We now consider the case in which we do not know the values of the parameters θ_1 and θ_2 . Especially, we cannot exclude that $\theta_1 \rightarrow 0$.

Proposition 1. *BCSID proposes to not extract any cash supply in an arbitrary period $t = 1, \dots, T$ for $\theta_1 \rightarrow 0$ with arbitrary θ_2 .*

Proof. We rewrite S_t^* in the following way

$$\begin{aligned} S_t^* &= D_{t-1} \theta_1 \theta_2 \frac{j+h}{j\theta_1+h\theta_2} \\ &= D_{t-1} \theta_1 \frac{j+h}{j\theta_1\theta_2^{-1}+h}. \end{aligned} \quad (27)$$

For $\theta_1 \rightarrow 0$, S_t^* becomes

$$\begin{aligned} \lim_{\theta_1 \rightarrow 0} S_t^* &= D_{t-1} \cdot 0 \cdot \frac{j+h}{j \cdot 0 \cdot \theta_2^{-1} + h} \\ &= 0. \end{aligned} \quad (28)$$

This concludes the proof. \square

Proposition 2. *The competitive ratio of BCSID is $1 + jc^{-1}$ for $\theta_1 \rightarrow 0$ with arbitrary θ_2 .*

Proof. This follows directly from Equation (25) w.r.t. $\theta_1 \rightarrow 0$. Notably we have

$$\begin{aligned} \lim_{\theta_1 \rightarrow 0} r^{BCSID} &= 1 + \frac{hj(\theta_2-0)}{c(0+h\theta_2)} \\ &= 1 + jc^{-1}. \end{aligned} \quad (29)$$

This finishes the proof. \square

We provide the approximation algorithm *approximated balanced* BCSID (aBBCSID). The main idea of this algorithm is again the balancing of two demand movements (just as in BCSID). The first movement is a demand

increase $\min(D_{t-1}\theta_1, D_{t-1}\theta_2, M)$; the second one is a decrease in demand $\max(D_{t-1}\theta_1, D_{t-1}\theta_2, m)$. The cash supply $S_t^{aBBCSID}$ equates the ratio of ON and OPT (denoted as ρ) for these movements in period t , formally

$$\frac{j \max(0, \min(D_{t-1}\theta_1, D_{t-1}\theta_2, M) - S_t^{aBBCSID})}{c \min(D_{t-1}\theta_1, D_{t-1}\theta_2, M)} = \frac{h \max(0, S_t^{aBBCSID} - \max(D_{t-1}\theta_1, D_{t-1}\theta_2, m))}{c \max(D_{t-1}\theta_1, D_{t-1}\theta_2, m)}. \quad (30)$$

Thus, we have

$$S_t^{aBBCSID} = \frac{\min(D_{t-1}\theta_1, D_{t-1}\theta_2, M) \max(D_{t-1}\theta_1, D_{t-1}\theta_2, m) (j + h)}{j \max(D_{t-1}\theta_1, D_{t-1}\theta_2, m) + h \min(D_{t-1}\theta_1, D_{t-1}\theta_2, M)}. \quad (31)$$

Algorithm aBBCSID: In every period t , convert into cash until you have the amount of $S_t^{aBBCSID}$ available.

We observe that $S_t^{aBBCSID} = S_t^{BCSID}$ if $\theta_1 \geq mD_{t-1}^{-1}$ and $\theta_2 \leq MD_{t-1}^{-1}$. In other words, the global demand bounds do not have any influence on the supply S_t at that particular period t .

4. Numerical Testing

In this section we examine the actual performance of BCSID and aBBCSID. We further include LCS, the optimal online algorithm *online strategy* (OS) of [11] for given global bounds m and M and the *middle of the expected demand range* (MER) algorithm.

Algorithm OS: In every period t , convert into cash until you have the amount of S_t^{OS} available, with

$$S_t^{OS} = \frac{(j + h) m M}{j m + h M}. \quad (32)$$

Algorithm MER: In every period t , convert into cash until you have the amount of S_t^{MER} available, with

$$S_t^{MER} = D_{t-1} \frac{(\theta_1 + \theta_2)}{2}. \quad (33)$$

We first need a random variable $X \sim \mathcal{U}(0, 1)$ which originates from the standard uniform distribution. The arithmetic mean $E(X)$ is 0.5. The two boundaries of X are 0 and 1. We now define the demand sequences based on $\theta_1, \theta_2, M, m, D_0$ and X . The demands in these sequences are generated in the following way:

$$D_t(X) = D_{t-1}(X) \theta_1^{1-2\min(0.5, X)} \theta_2^{2\max(0.5, X)-1} \text{ and } D_0(X) = 1. \quad (34)$$

For $X = 0.5$, we have $D_t(X) = D_{t-1}(X)$. In other words, the demands remain constant over time on average; this is highly beneficial for LCS. The lower boundary of $D_t(X)$ is $D_{t-1}(X)\theta_1$ and the upper boundary is $D_{t-1}(X)\theta_2$. Globally, the demands are **bounded** from above (below) by $M = \max_{t=1,\dots,T} D_t(X)$ ($m = \min_{t=1,\dots,T} D_t(X)$). Using Equation (34), we generate arbitrary demand sequences \mathbf{D} using $X, T = 250$ and various $M, m, \theta_1, \theta_2, j$ and h . We consider three performance measures, the arithmetic mean (mean), the lowest (low) and the highest (high) costs. All three measures are applied to 1,000 experiments for every unique parameter combination. Formally, we have

$$\begin{aligned} \text{mean}^{ALG} &= \frac{1}{n} \sum_{i=1}^n ON(\mathbf{D}(X_i), \mathbf{S}^{ALG}), \\ \text{low}^{ALG} &= \min_{i=1,\dots,n} ON(\mathbf{D}(X_i), \mathbf{S}^{ALG}), \\ \text{high}^{ALG} &= \max_{i=1,\dots,n} ON(\mathbf{D}(X_i), \mathbf{S}^{ALG}), \end{aligned} \quad (35)$$

for $n = 1,000$. Concerning the costs (i.e. $ON(\mathbf{D}(X_i), \mathbf{S})$), we focus only on the costs for failing to extract D_t in each t , formally

$$ON(\mathbf{D}(X), \mathbf{S}) = \sum_{t=1}^T j \max(0, D_t - S_t) + h \max(0, S_t - D_t). \quad (36)$$

Note that we do not need any information about c , because we are only interested in the additional costs and there is no algorithm that needs c to compute an output.

Note that we set $j = h = 1$ as a point of orientation; in cash management practice these values are unrealistic. However, this setting does not "overemphasize" one cost factor more than the other, i.e. one missing cash unit costs as much as one excess cash unit.

We now consider the performance of the algorithms for various **values of** θ_1 and θ_2 ; as for the selected pairs of θ_1 and θ_2 , we start with $\theta_1 = \theta_2 = 1$. For θ_1 , we let it decrease by the following equation, where $\theta_{i,1}$ denotes the new, lower θ_1 and $\theta_{i-1,1}$ denotes the old, higher θ_1 , formally

$$\theta_{i,1} = ((\theta_{i-1,1})^{-1} + 0.05)^{-1}, \quad (37)$$

for $i = 2, \dots, 181$ and $\theta_{1,1} = 1$. This continues until $\theta_{181,1} = 0.10$; this yields 181 different values for θ_1 . We let θ_2 absolutely increase by 0.05 until $\theta_2 = 10$. Formally, the i -th value $\theta_{i,2}$ of θ_2 is $\theta_{i,2} = \theta_{i-1,2} + 0.05$ for $i = 2, \dots, 181$ and

$\theta_{1,2} = 1$. This yields again 181 different values for θ_2 (overall there are 32,761 combinations). BCSID and MER have information about θ_1, θ_2 and $D_0(X)$; OS has information about m and M of each demand sequence. aBBCSID knows θ_1, θ_2, m, M and $D_0(X)$. All values for θ_1 and θ_2 are shown in Figure 3.

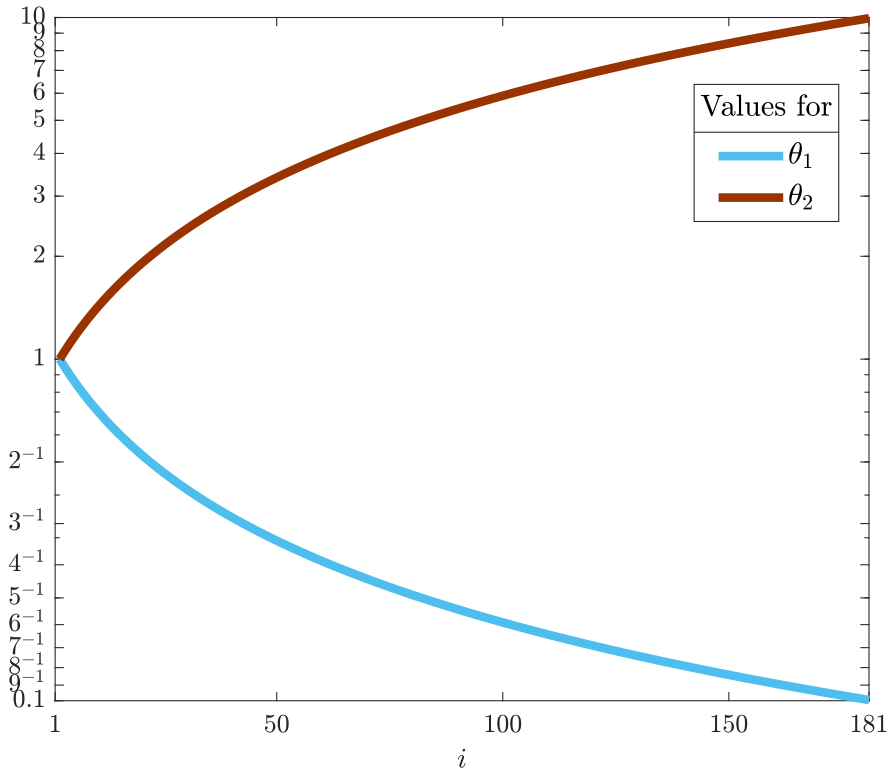


Figure 3: Considered values of θ_1 and θ_2 for the numerical testing

We first consider the performance of the algorithms BCSID, OS and MER in terms of mean performance for various values of θ_1 and θ_2 as depicted in Figure 4. While the left part of the figure illustrates $ON(\mathbf{D}(X), \mathbf{S})$, the right part states which algorithm is best for each combination of θ_1 and θ_2 . We observe that MER incurs the highest costs whenever $\theta_2 \geq 3.5$ regardless of θ_1 . Apparently, MER suffers from extracting too much cash compared to

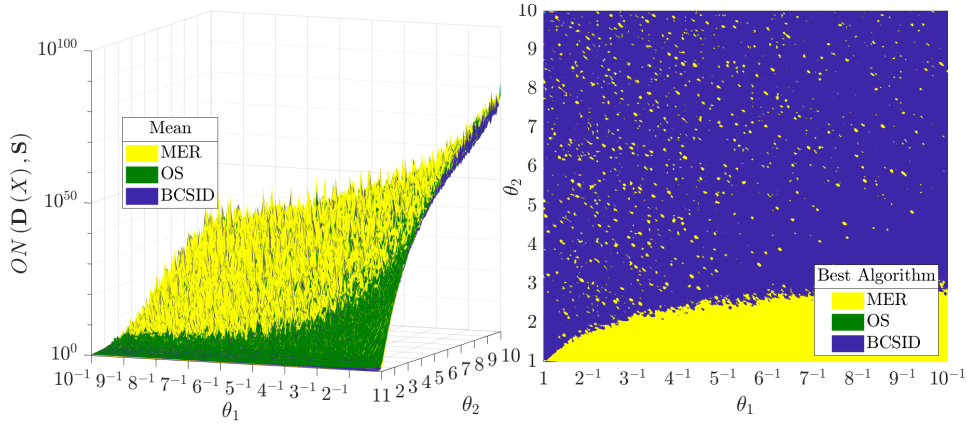


Figure 4: Influence of θ_1 and θ_2 on the mean performance of BCSID, MER and OS

all other algorithms in this area. There are some combinations for which OS performs worse; however, these occasional peaks of OS are due to the number of experiments being limited to 1,000. An increase might eliminate them. For $\theta_2 < 3.5$, we see that OS incurs the highest costs; OS extracts too little cash and incurs very high shortage costs. Interestingly, BCSID is never worst. Looking at the best algorithm, we observe that MER is best, as long as $\theta_1^{-1} \gg \theta_2$. In other words, MER is best whenever the possible relative increase is a lot smaller than the possible relative decrease. Apparently, all other algorithms tend to extract less and thus incur higher shortage costs. For all other parameter combinations we find that BCSID is best. We further observe that there are again some occasional combinations for which MER is best. Again this is due to $n = 1,000$. These findings are also confirmed in Figure 5 where BCSID, MER and OS are compared in terms of highest costs incurred.

We now turn to the performance of BCSID, MER and OS in terms of lowest costs incurred as depicted in Figure 6.

We observe again that MER is clearly outperformed by all other algorithms. OS is worst whenever θ_2 is small and θ_1 is high. MER is best whenever OS is worst; for all other combinations, we find BCSID outperforming MER and OS in terms of lowest costs incurred.

We now turn to the performance of BCSID, aBBCSID and LCS in terms of mean performance as depicted in Figure 7. Looking at the left part of the figure, we observe that aBBCSID incurs higher costs than BCSID and LCS

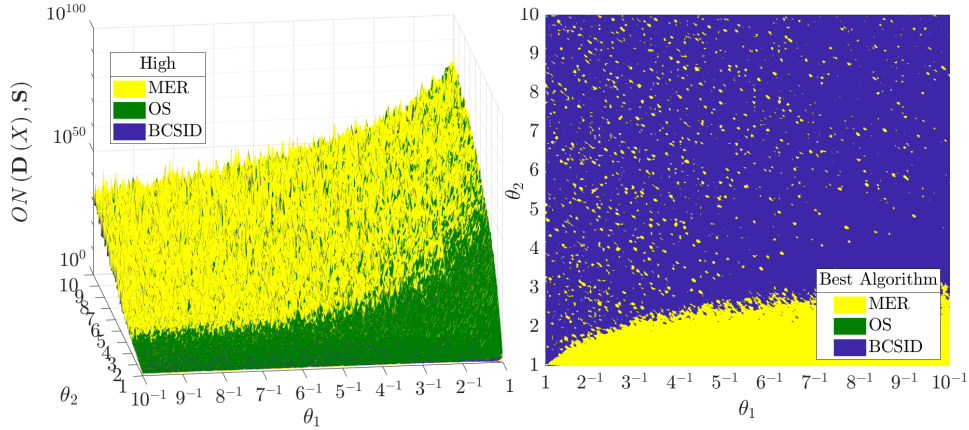


Figure 5: Influence of θ_1 and θ_2 on the highest costs incurred of BCSID, MER and OS

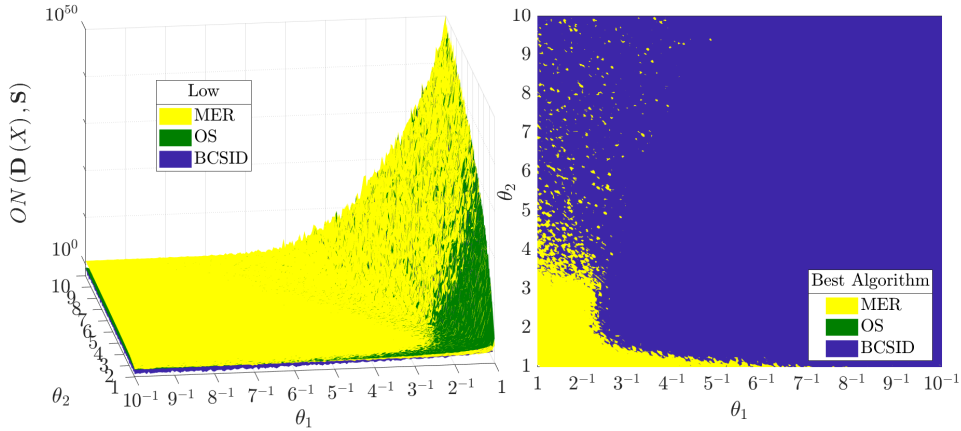


Figure 6: Influence of θ_1 and θ_2 on the lowest costs incurred of BCSID, MER and OS

as soon as $\theta_1^{-1} \gg \theta_2$. BCSID is worst whenever $\theta_2 \gg \theta_1^{-1}$. The remaining large area shows that there is no clear worst performing algorithm. We observe peaks of every algorithm; however, there is a tendency of LCS being the algorithm which incurs the highest costs more often than its competitors do. Looking at the right part of the figure, we observe that aBBCSID tends to be best whenever $\theta_2 \gg \theta_1^{-1}$; still in that area, we observe that there are some parameter combinations of θ_1 and θ_2 for which LCS or BCSID is best. Clearly, for $\theta_1^{-1} \gg \theta_2$ we find that LCS dominates all other algorithms.

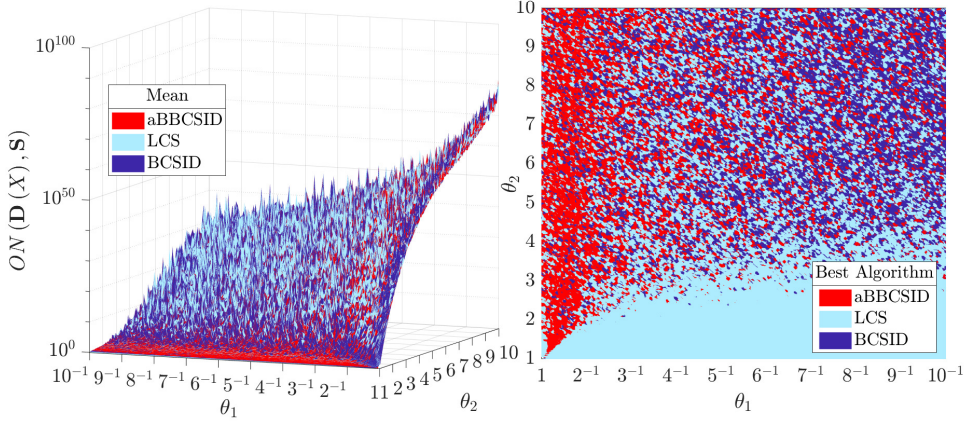


Figure 7: Influence of θ_1 and θ_2 on the mean performance of BCSID, LCS and aBBCSID

Besides these aforementioned areas, there is no clear best algorithm. Every algorithm has a number of parameter combinations for which it is performing best. Finally, putting the right and left part of the figure together, we observe that BCSID and aBBCSID are often the best-performing algorithms whenever $\theta_2 \gg \theta_1^{-1}$ and $\theta_2 \ll \theta_1^{-1}$ does not hold. This is due to the observation of LCS often being the worst-performing algorithm in that area. Hence, BCSID and aBBCSID are clearly preferable to LCS in that area. If $\theta_1^{-1} \gg \theta_2$, then LCS is best and BCSID is better than aBBCSID. If $\theta_2 \gg \theta_1^{-1}$, then aBBCSID tends to be best and LCS tends to be better than BCSID.

These findings are also confirmed in Figure 8 where BCSID, LCS and aBBCSID are compared to one another in terms of highest costs incurred.

We now turn to the performance of BCSID, LCS and aBBCSID in terms of lowest costs incurred as depicted in Figure 9. Interestingly, we observe that LCS is worst unless $\theta_2 \gg \theta_1^{-1}$. Only if θ_1 and θ_2 are high, we find that LCS is best and BCSID is worst. When LCS is worst, aBBCSID tends to be better than BCSID for many combinations of θ_1 and θ_2 . This is particularly true when $\theta_1 \leq \theta_2^{-1}$.

Table 2 compares all five algorithms to one another in terms of mean, highest and lowest cost incurred for $n = 10,000$ experiments, $D_0 = j = h = 1, T = 250$ and all parameter combinations of $\theta_1 = 0.5, 0.25, 0.1$ and $\theta_2 = 2, 4, 10$. **The lowest values are given in bold, the highest ones are given in italic.** From the table we see that LCS and BCSID are better than the other algorithms in terms of mean performance for the most combinations of θ_1 and θ_2 . In

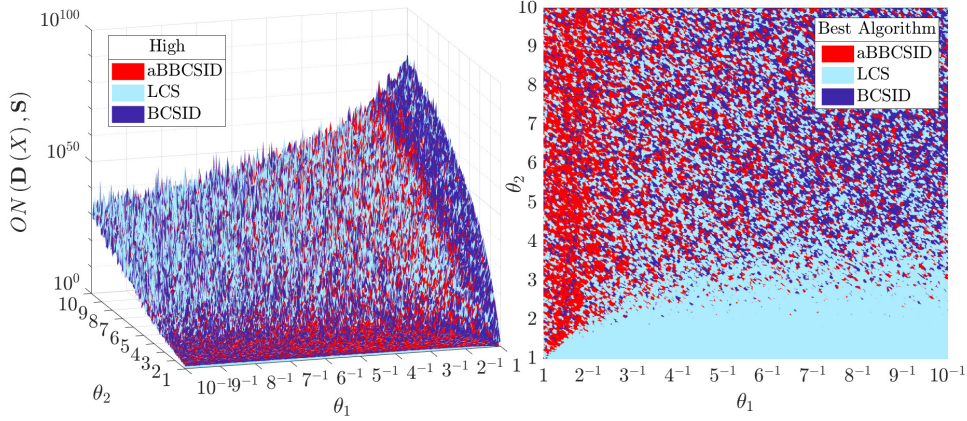


Figure 8: Influence of θ_1 and θ_2 in terms of highest cost incurred of BCSID, LCS and aBBCSID

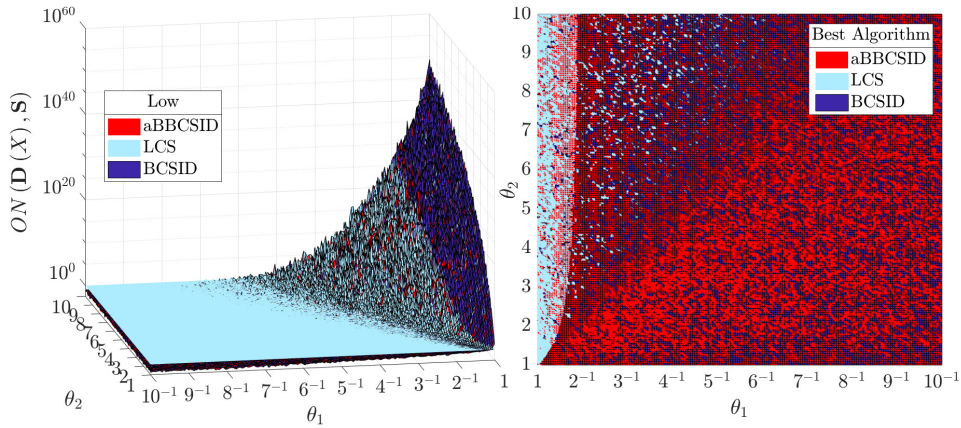


Figure 9: Influence of θ_1 and θ_2 on the lowest cost incurred of BCSID, LCS and aBBCSID

terms of low performance we observe that aBBCSID is better than all other algorithms in the majority of cases. In terms of high performance LCS and BCSID are again best; however, aBBCSID is best for $\theta_1 = 0.1, \theta_2 = 4$ and MER is best for $\theta_1 = 0.5, \theta_2 = 4$. MER and OS are performing worst across all performance measures. However, we note that there is no algorithm which clearly outperforms all the other ones.

We now turn to the performance of all algorithms for various values of h . We fix $j = 1, \theta_1 = 0.5$ and $\theta_2 = 2$. For the sake of interpretability, we now

	(θ_1, θ_2)	BCSID	aBBCSID	LCS	OS	MER
<i>mean</i> ^{ALG}	(0.5, 2)	$2.81E + 7$	$2.82E + 7$	2.76E + 7	$8.40E+7$	$3.36E + 7$
	(0.5, 4)	$3.13E + 30$	$3.15E + 30$	3.11E + 30	$6.05E+30$	$4.27E + 30$
	(0.5, 10)	$6.09E + 62$	$7.26E + 62$	6.08E + 62	$1.50E + 63$	$4.09E+63$
	(0.25, 2)	20.26	26.02	20.28	43.69	20.91
	(0.25, 4)	$5.63E + 17$	$4.26E + 17$	3.92E + 17	$6.93E + 17$	$8.5E+17$
	(0.25, 10)	5.56E + 50	$5.71E + 50$	$6.77E + 50$	$9.13E + 50$	$3.81E+51$
	(0.1, 2)	6.34	7.66	5.53	9.48	5.55
(0.1, 4)	$2.99E + 5$	2.93E + 5	$3.40E + 5$	$3.42E + 5$	$5.05E+5$	
(0.1, 10)	3.70E + 32	$3.70E + 32$	$4.15E + 32$	$4.43E + 32$	$1.83E+33$	
<i>low</i> ^{ALG}	(0.5, 2)	0.54	0.52	1.03	2.23	1.83
	(0.5, 4)	$2.24E + 4$	2.08E + 4	$2.30E + 4$	$4.83E + 4$	$5.85E+4$
	(0.5, 10)	$9.68E + 22$	9.41E + 22	$1.03E + 23$	$1.89E + 23$	$6.91E+23$
	(0.25, 2)	0.86	0.04	1.00	0.42	1.18
	(0.25, 4)	0.33	0.15	1.00	0.64	2.85
	(0.25, 10)	$2.39E + 3$	$2.55E + 3$	2.12E + 3	$4.23E + 3$	$1.78E+4$
	(0.1, 2)	0.10	0.01	1.00	0.12	1.06
(0.1, 4)	0.11	0.01	1.00	0.13	2.18	
(0.1, 10)	0.01	0.02	1.00	0.14	5.62	
<i>high</i> ^{ALG}	(0.5, 2)	$1.25E + 11$	$1.27E + 11$	1.23E + 11	$3.83E + 11$	$1.49E+12$
	(0.5, 4)	$1.88E + 34$	$1.96E + 34$	$1.83E + 34$	$3.37E+34$	1.47E + 34
	(0.5, 10)	$5.60E + 66$	$6.77E + 66$	5.58E + 66	$1.42E + 67$	$3.93E+67$
	(0.25, 2)	2.24E + 4	$2.68E + 4$	$2.25E + 4$	$4.44E+4$	$2.31E + 4$
	(0.25, 4)	$4.86E + 21$	$3.68E + 21$	3.37E + 21	$5.99E + 21$	$7.35E+21$
	(0.25, 10)	5.42E + 54	$5.57E + 54$	$6.60E + 54$	$8.92E + 54$	$3.75E+55$
	(0.1, 2)	$8.36E + 2$	$9.65E + 2$	6.29E + 2	$1.16E+3$	$6.21E + 2$
(0.1, 4)	$2.26E + 9$	2.12E + 9	$2.62E + 9$	$2.44E + 9$	$3.73E+9$	
(0.1, 10)	3.48E + 36	$3.49E + 36$	$3.91E + 36$	$4.16E + 36$	$1.72E+37$	

Table 2: Performance of online algorithms for various combinations of θ_1 and θ_2 with $D_0 = j = h = 1$ and $n = 10,000$

compare all algorithms directly to BCSID; formally, we derive the logarithmic difference $\log \frac{ON(\mathbf{D}(X), \mathbf{S}^{BCSID})}{ON(\mathbf{D}(X), \mathbf{S}^{ALG})}$ with ALG containing all algorithms except BCSID. Clearly, we have $\log \frac{ON(\mathbf{D}(X), \mathbf{S}^{BCSID})}{ON(\mathbf{D}(X), \mathbf{S}^{BCSID})} = 0$. Hence, an algorithm performs better (worse) than BCSID whenever its logarithmic difference is below (above) zero. The results are depicted in Figure 10. Starting with OS we observe that it performs worse and worse as h declines for all three performance measures. Apparently, BCSID and aBBCSID tend to extract more cash than all other algorithms in order to avoid paying shortage costs. In terms of MER and mean and high performance, we state that as $h \rightarrow 1$, the logarithmic difference approaches zero; when $h \approx j$ we find that BCSID is

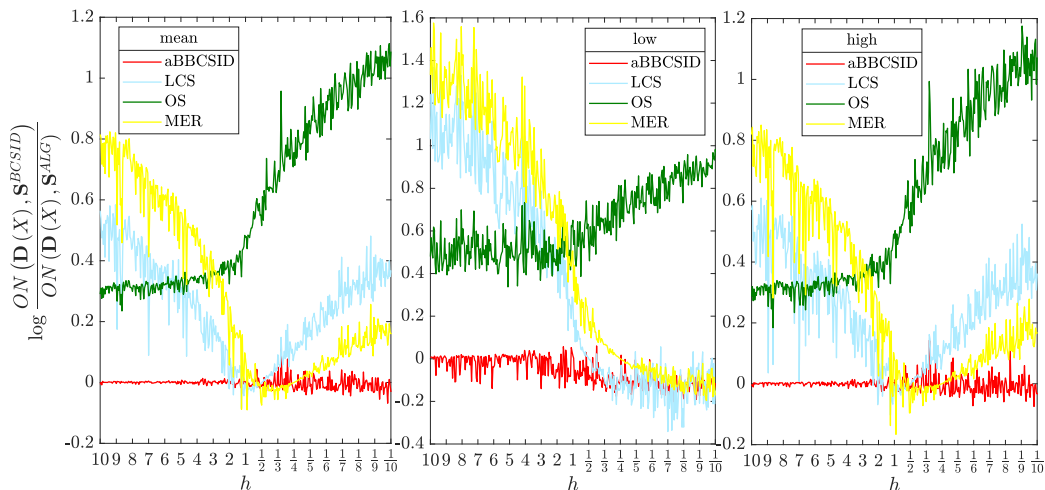


Figure 10: Logarithmic difference between all algorithms and BCSID for various values of h for all three performance measures

often outperformed by MER. However, if $j \gg h$ we observe that BCSID tends to perform better. Looking at MER and the lowest costs incurred, we find that MER outperforms BCSID if $h < 0.25$. Considering LCS, we find the same observation regarding MER for all three performance measures. In addition, LCS tends to be better than all other algorithms (including BCSID and aBBCSID) whenever $h < 0.25$ for the low performance measure. Regarding aBBCSID in terms of mean and high performance measure, we observe that the logarithmic difference to BCSID is around zero when $h > j$. When $j < h$, we observe that results tend to fluctuate more around zero. aBBCSID is not capable to systematically outperform BCSID and vice versa. In terms of low performance however, we observe that aBBCSID outperforms BCSID; this is particularly true when $h < j$.

We now turn to the performance of all algorithms for various $M = 2, \dots, 10,000$ and fix $m = j = h = 1$. We modify the generated demands such that they are merely dependent on m and M , formally

$$D_t(X) = M^X m^{1-X} \forall t. \quad (38)$$

In many instances, we now find that $\theta_1 \rightarrow \frac{m}{M} \leftarrow \theta_2^{-1}$.

We consider the logarithmic distance between an online algorithm ALG and

aBBCSID, i.e. $\log \frac{ON(\mathbf{D}(\mathbf{X}), \mathbf{S}^{ALG})}{ON(\mathbf{D}(\mathbf{X}), \mathbf{S}^{aBBCSID})}$. Concerning MER, BCSID and aBBCSID, we note that they have hindsight information about θ_1, θ_2 and $D_0(X)$ (and m and M in case of aBBCSID). The results for all three performance measures are depicted in Figure 11.

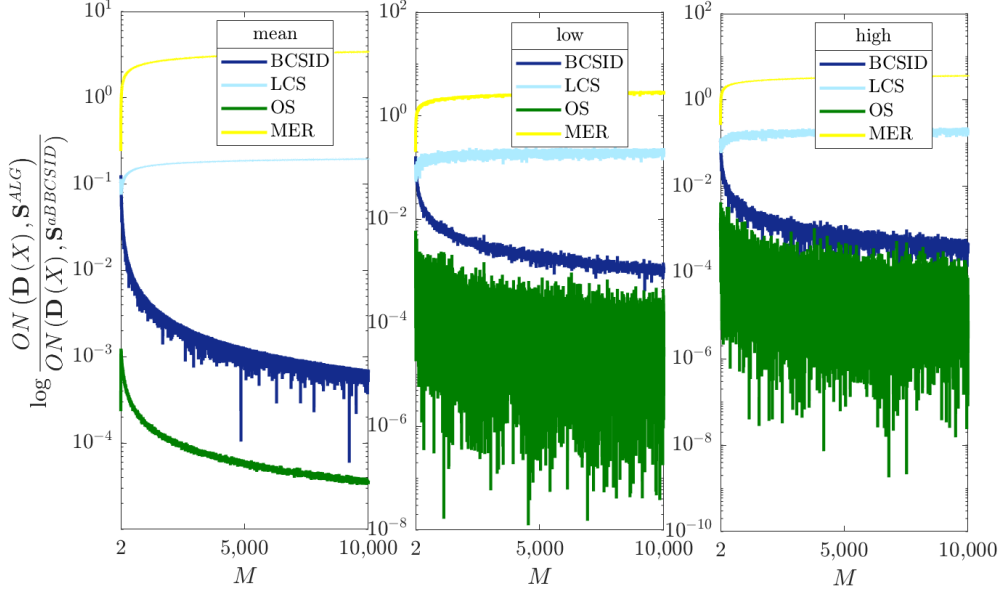


Figure 11: Logarithmic difference between all algorithms and aBBCSID for various values of M on all three performance measures

From the figure, we conclude that MER is worst in all three measures. LCS is also performing worse than OS, aBBCSID and BCSID. We further observe that the logarithmic distance of MER and LCS to aBBCSID increases as M increases. BCSID is worse than aBBCSID and OS, but better than LCS and MER in all three performance measures. Moreover, the logarithmic difference between BCSID and aBBCSID decreases as M increases. Finally, OS performs better than all other algorithms except for aBBCSID. Like BCSID, OS logarithmic difference to aBBCSID declines as M increases. We further note that for low and high performance, the results of OS show a high fluctuation. This finishes the numerical testing. Since we considered the separate influence of the variables $\theta_1, \theta_2, \frac{j}{h}, m$ and M , we cannot state which algorithm is best for a given parameter combination; however, we can state how each algorithm profits from a change in an isolated parameter.

Taking these insights together, a **recommendation** as to which algorithm to use for an arbitrary parameter combination can be made. The findings of this section are as follows:

1. For varying $\frac{h}{j}$
 - for $\frac{h}{j} \gg 1$: BCSID and aBBCSID profit more than all other algorithms (for the mean and high performance measure)
 - for $\frac{h}{j} \gg 1$: aBBCSID profits more than all other algorithms (for the low performance measure)
 - for $\frac{h}{j} < 1$: aBBCSID and LCS profit more than all other algorithms (for the low performance measure)
 - for $\frac{h}{j} > \frac{1}{5}$: aBBCSID profits more than all other algorithms (for the mean and high performance measure)
 - for $1 \geq \frac{h}{j} \geq \frac{1}{5}$: MER profits more than all other algorithms (for the mean and high performance measure)
 - for $h \approx j$: LCS profits more than all other algorithms (for the mean and high performance measure)
2. For $D_t(\mathbf{D}(X), \mathbf{S}) = M^X m^{1-X}$
 - aBBCSID profits most in all three performance measures
 - if mere information about m and M is available, then OS profits most
 - if mere information about θ_1 and θ_2 is available, then BCSID profits most
 - without any information, LCS profits most
 - MER profits less in all three performance measures
3. For demand sequences generated as shown in Equation (34):
 - for $\theta_2 \gg \theta_1^{-1}$: BCSID profits more than MER and OS (for the mean and low performance measure)
 - for $0 < \theta_1 \leq 1 \leq \theta_2 < \infty$: OS never profits more than all other algorithms (for all three performance measures)
 - for $\theta_2 \ll \theta_1^{-1}$: aBBCSID tends to profit less than BCSID and LCS (for the mean and high performance measure)

- for $\theta_2 \gg \theta_1^{-1}$: BCSID (aBBCSID) profits less (more) than LCS and aBBCSID (BCSID) (for the mean and high performance measure)
- unless $\theta_2 \gg \theta_1^{-1}$: LCS **profits** less than aBBCSID and BCSID (for the low performance measure)

5. Conclusion

In this paper we propose and solve the cash management problem with uncertain, interrelated demands that are either globally bounded or not. We first derive the algorithm BCSID which solves this problem optimally online when demands are not globally bounded. Next we derive the heuristic algorithm aBBCSID which solves the problem for bounded demands. We then test these algorithms against the simple but in practice very efficient strategy LCS, the optimal solution OS of [11] and MER. If the demand sequences are derived as in Equation (34), then there is no algorithm which outperforms all others across all parameter combinations and performance measures. If the demand sequences are derived as in Equation (38), then aBBCSID outperforms **other algorithms for most parameter combinations across all performance measures**.

In terms of future research, we state that there exists merely a heuristic solution for the cash management problem with given $\theta_1, \theta_2, D_0, m$ and M . An optimal solution does not exist yet. In addition, we focused on the competitive ratio of an online algorithm, i.e. the maximal relative difference of an algorithm to the best solution for adverse demands. It would be of interest to derive online algorithms which focus on the competitive difference of an online algorithm, i.e. the maximal absolute difference between the performance of an online algorithm and the one of the best solution. This can be seen as minimizing max regret for the online problem setting.

One way of practical improvement for BCSID and aBBCSID would be to adapt the extracted amount of cash to incorporate parameters which change over time, e.g., $\theta_{1,t}, c_t$ and j_t .

Acknowledgements

The authors thank the two anonymous reviewers for their comments and suggestions which greatly improved this paper.

References

- [1] Ahmad, I., Mohr, E., Schmidt, G. (2014). Competitive Ratio as Coherent Measure of Risk. In: Operations Research Proceedings 2012 (pp. 63-69).
- [2] Baccarin, S. (2009). Optimal impulse control for a multidimensional cash management system with generalized cost functions. *European Journal of Operation Research*, 196(1), 198-206.
- [3] Baumol, W.J. (1952). The Transactions Demand for Cash: An Inventory Theoretic Approach. *The Quarterly Journal of Economics*, 66(4), 545-556.
- [4] Bensoussan, A., Chutani, A., Sethi, S. P. (2009). Optimal Cash Management under Uncertainty. *Operations Research Letters*, 37(6), 425-429.
- [5] Borodin, A. El-Yaniv, R. Gogan, V. (2000). On the Competitive Theory and Practice of Portfolio Selection (Extended Abstract). In: LATIN 2000: Theoretical Informatics (pp 173-196).
- [6] El-Yaniv, R., Fiat, A., Karp, R. M., Turpin, G. (2001). Optimal Search and One-Way Trading Online Algorithms. *Algorithmica*, 30(1), 101-139.
- [7] Ferstl, R., Weissensteiner, A. (2010). Cash Management Using Multi-Stage Stochastic Programming. *Quantitative Finance*, 10(2), 209-219.
- [8] Gangolf, C. Dochow, R Schmidt, G. Tamisier, T. (2016) Automated credit rating prediction in a competitive framework. *RAIRO Operations Research*, 50(4-5), 749-765.
- [9] Gormley, F. M., Meade, N. (2007). The utility of cash forecast in the management of corporate cash balances. *European Journal of Operational Research*, 182(2), 923-935.
- [10] Jaillet, P., Wagner, M. R. (2008): Generalized Online Routing: New Competitive Ratios, Resource Augmentation, and Asymptotic Analyses. *Operations Research*, 56(3), 745-757.
- [11] Liu, B., Xin, C. (2008). An Online Model for Managing Cash: An Alternative Approach to the Miller-Orr Model. In: 2008 Fourth International Conference on Natural Computation, (pp 314-317).

- [12] Miller, M. H, Orr, D. (1966). A model of the demand for money by firms. *Quarterly Journal of Economics*, 80(3), 413-435.
- [13] da Costa Moraes, M., Nagano, M. S. (2014). Evolutionary models in cash management policies with multiple assets. *Economic Modelling*, 39, 1-7.
- [14] da Costa Moraes, M., Nagano, M. S., Sobreiro, V. (2015). Stochastic Cash Flow Management Models: A Literature Review Since the 1980s. In Guarnieri, P. (Eds), *Decision Models in Engineering and Management*, (pp 11-28).
- [15] Righetto, G. M., Morabito, R., Alem, D. (2016). A Robust Optimization Approach for Cash Flow Management in Stationary Companies. *Computers and Industrial Engineering*, 99, 137-152.
- [16] Salas-Molina, F., Pla-Santamaria, D., Rodríguez-Aguilar, J. A. (2018). A multi-objective approach to the cash management problem. *Annals of Operations Research*, 267(1), 515-529.
- [17] Salas-Molina, F., Rodríguez-Aguilar, J., Pla-Santamaria, D., García-Bernabeu, A. (2019). On the formal foundations of cash management systems. *Operational Research*, 10.1007/s12351-019-00464-6, in press.
- [18] Schroeder, P., Dochow, R., Schmidt, G. (2018). Optimal solutions for the online time series search and one-way trading problem with interrelated prices and a profit function. *Computers and Industrial Engineering*, 119, 465-471.
- [19] Schroeder, P., Schmidt, G., Kacem, I. (2016). Optimal on-line algorithms for bi-directional non-preemptive conversion with interrelated conversion rates. In: 2016 International Conference on Control, Decision and Information Technologies (CoDIT) (pp 28-33).
- [20] Sleator, D. D., Tarjan, R. E. (1985). Amortized Efficiency of List Update and Paging Rules. *Communications of the ACM*, 28, 202-208.
- [21] Zhang, W. Xu, Y. Zheng, F., Dong, Y. (2012). Optimal Algorithms for Online Time Series Search and One-way Trading with Interrelated Prices. *Journal of Combinatorial Optimization*, 23, 159-166.