Competitive difference analysis of the cash management problem with uncertain demands

Pascal Schroeder, Imed Kacem

Competitive difference analysis of the cash management problem with uncertain demands

Pascal Schroeder*, Imed Kacem

LCOMS EA 7306, Université de Lorraine, Metz, France,
{pascal.schroeder,imed.kacem}@univ-lorraine.fr

Abstract

The cash management problem with uncertain demands belongs to the class of online problems which are free of any distribution assumptions. The demands can either be globally bounded or interrelated. We consider the performance measure competitive difference. This measure can be interpreted as maximum regret for online problems. The minimization of maximum regret is a major point of interest in the area of combinatorial optimization . We derive new algorithms which aim at minimizing the maximum regret incurred. Their experimental performance is compared to already established solutions, which consider the performance measure competitive ratio. From this comparison we confirm the practicability of our solutions for real-case scenarios. Hence, our algorithms are particularly relevant for risk averse cash managers, who are interested in minimizing their maximum regret while ensuring a solid performance in non-worst case scenarios.

Keywords: combinatorial optimization, cash management problem, minmax regret, interrelated demands, bounded demands

1. Introduction

In an online problem a player must take an irrevocable decision without knowing the future at every point of time. Ever since the work of [31], the so-called competitive ratio renders online algorithms comparable to each other. For minimization problems, the competitive ratio of an online algorithm is the relative difference between the value obtained using the solution of the

*Corresponding author

online algorithm and the lowest possible value in a worst-case scenario. The online algorithm with the lowest competitive ratio solves the online problem to optimality. Very famous online problems are the online portfolio selection problem presented by [6] and the online conversion problem presented by [8]. More recently, [7] combined online problems and financial leasing. They developed algorithms that minimize competitive ratio. Because of their uncertainty, financial problems are very suitable for an online setting. Another financial problem is the cash management problem; our work considers an online variant of this problem. The main cash management problem was introduced by [3]. Here, a cash manager 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*" ([15], p. 413); the second asset is a non-cash asset which bears interest at a given constant rate. The cash manager is allowed to convert one asset into the other at any period; regardless of the direction of the conversion, the cash manager must pay a "broker fee". Moreover, 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.

[17] presented a literature review about cash management; they state that most contributions of today focus on robust optimization and stochastic models. Regarding robust optimization, [19] designed an effective solution for the cash management problem of stationary companies; their solution is based on mixed integer linear programming and robust optimization. Moreover, risk preferences for cash management decision were incorporated by [22]; based on compromise programming, a multi-objective cash management model was developed. Genetic algorithms and particle swarm optimization were applied to the cash management problem using various assets in [16]; the objective is to minimize the total costs of cash management. Concerning stochastic programming, we can mention the works of [9] and [4]. In [4] the authors derive the optimal transaction policy by applying a stochastic maximum principle. A multi-stage stochastic linear program is formulated and various kinds of assets are considered in [9]. A multi-dimensional cash management system where cash balances fluctuate as a homogeneous diffusion process is considered in [2] and an optimal solution is developed. [11] presented another stochastic model; they developed a solution which minimizes the transaction costs when cash flows are not independently or identically distributed. Recently, [18] combined the robust optimization with the stochastic program-

ming approach into the distributionally robust optimization approach for the newsvendor problem. [20] applied several data techniques to empirical cash flows in order to select an appropriate cash management model. [25] presented an empirical analysis of daily cash flow time-series. Based on this, [23] presented a framework for implementing machine learning techniques into the cash management problem. [24] addressed the problem of selecting a cash management model in terms of risk and cost. The effect of the accuracy when forecasting the needed cash flow for the coming period on cost minimization has been investigated by [21]. [13] used Monte-Carlo simulation to minimize several different types of risk associated with cash management. Our focus, however, is on neither of the two research directions; rather, we focus on developing algorithms which minimize the maximum regret. The min-max regret criterion was proposed by [26]. It "*aims at obtaining a solution minimizing the maximum deviation, over all possible scenarios, between the value of the solution and the optimal value of the corresponding scenario*" (see [1] p. 427). As an example, the minimization of this criterion can be used to construct robust algorithms for solving combinatorial optimization problems ([12]). A min-max regret version for the scheduling problem with outsourcing decision under processing time uncertainty has been given in [5]. According to [10], combinatorial optimization problems which uses the min-max regret criterion are oftentimes NP-hard. Recently, [32] proposed the so called competitive difference analysis. Compared to the competitive ratio analysis, this analysis aims at creating online algorithms which have the smallest worst-case performance gap to the optimal solution. In other words, the analysis aims at creating algorithms which fulfill the min-max regret criteria. The competitive difference can be seen as a type of maximum regret for online problems. While both types of analysis aim at deriving risk-averse online algorithms, they differ in term of risk-averseness degree ([32]). In a sense, a competitive ratio analysis becomes a competitive difference analysis with a logarithmic utility function, while the competitive difference analysis has a linear utility function. Those functions may reflect the decision maker's attitude towards risk. [32] therefore stated that a "cost-minimization decision maker is more risk-averse (or less risk-seeking) under competitive difference analysis than under competitive ratio analysis." Therefore, online algorithms which minimize competitive difference are more relevant to cash managers with high risk aversion than algorithms which minimize competitive ratio. [14] were the first to combine online problems, the competitive ratio analysis and the cash management problem. They consider

the cash management problem with uncertain, globally bounded demands. A demand is an amount of cash needed to pay expenditures in a time period. A cash manager 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. The model further assumes that the amount of owned earning assets is limitless. Therefore, they disregard cash to earning asset conversion. If more cash is available than actually needed, then the surplus is transferred back into the earning asset. The manager incurs opportunity costs for every unit that was converted too early. In the case of a cash deficit, the manager has to borrow the needed additional cash. The deficit is made up at the beginning of the next period where the cash manager converts into cash. The objective is to minimize the total costs associated with the expenditures. For this to happen, knowledge of the exact amount of future demands is needed just as in the newsvendor problem. However, unlike the newsvendor problem, the cash management problem can become a multi-period problem (see Subsection 3.2); furthermore it does not rely on complete knowledge about the underlying demand distributing. [14] bounded the demands by the minimum (m) and maximum (M) amount of needed cash. Inspired by the work of [33] and [30], [27] 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 [31].
The way the price p_t is derived directly affects the demand of cash D_t to pay this price in period t and thus the cash supply S_t. If p_t is an element of $[m, M]$ or $[p_{t-1}\theta_1, p_{t-1}\theta_2]$ (with $0 < \theta_1 \leq \theta_2 < \infty$), then it must follow that D_t is also an element of these intervals. Within a company for instance, there are costs which vary only little over time (e.g. building maintenance, catering, cleaning) while others vary extremely over time (e.g. heating, litigation). Thus, both demand models have their respective field of application. Combining the model of [14] and [33], [28] and [29] designed an optimal online algorithm for uncertain, interrelated and bounded demands called *balanced cash supply for interrelated demands* (BCSID), gave its competitive ratio and proved optimality. Furthermore, [29] provided an extensive testing for the cash management problem. We consider again both demand models. Hence the demands vary between the factors θ_1 and θ_2 for two consecutive points of time or m and M; the overall number of points of time is T. We motivate this assumption by the following idea. In real-world applications, cash managers oftentimes use the so called scenario planning technique in order to

take decisions. Among other ideas, this technique uses the current position (the currently needed cash) and predicts extreme scenarios (the maximal 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., $\theta_1, \theta_2, m, M > 0$. 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 and so on); only a small portion of their actual wealth is kept as a liquid asset.
The contributions of this work are the following:

1. We present the algorithm *min-max regret for bounded demands* (MRBD) for given m and M,
2. we present the algorithm *heuristic min-max regret for interrelated demands* (HMRID) for θ_1, θ_2 and D_0,
3. we present the algorithm *min-max regret for interrelated demands* (MRID) for θ_1, θ_2, D_0 and T,
4. we prove the competitive difference of MRBD, HMRID and MRID,
5. we prove that MRBD and MRID minimize competitive difference and
6. we carry out a numerical testing to demonstrate the solid performance of the three new solutions to the solutions of [28] and [14] in non-worst case scenarios.

The paper is organized in five sections. In Section 2 the formal description of the cash management problem and the concepts of competitive difference are given. In Section 3, we first consider the cash management problem with given m and M and present the algorithm MRBD, prove competitive difference and show that the solution minimizes competitive difference. We then consider the cash management problem with given θ_1, θ_2 and D_0 and derive the heuristic solution HMRID. We then add knowledge about T and derive the optimal solution MRID. We then prove competitive difference and show that MRID minimizes competitive difference. The section concludes with a numerical example. In Section 4 we present the numerical testing and carry it out. The paper ends with a conclusion in Section 5.

2. The cash management problem

In the cash management problem under uncertain demands, a manager ON must decide on how much cash S_t to convert (or extract) from its (limitless) earning assets at the beginning of every period $t = 1, \ldots, T$. At the end of t the actual cash demand D_t is revealed to ON. At $t = 1$ ON knows the value of D_0. The sequences of all S_t and 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 $\mathcal{D}$. Formally, we have $\mathbf{S} = S_1, \ldots, S_T, \quad \mathbf{S} \in \mathcal{S}$ and $\mathbf{D} = D_1, \ldots, D_T, \quad \mathbf{D} \in \mathcal{D}$.

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

1. $S_t = D_t$, i.e. the supply equals the demand. In this case, the incurred total costs of ON are cD_t.

2. $S_t > D_t$, i.e. too much cash has been extracted. We first incur the transaction costs cS_t; this can be separated into cD_t and $c\left(S_t - D_t\right)$ (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

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

3. $S_t < D_t$, i.e. too little cash has been extracted. ON incurs additional credit costs $\left(D_t - S_t\right) j$ 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. This difference is assumed to be always non-negative. At the end of t,

ON has to extract the missing $D_t - S_t$ and incurs also transaction costs $c(D_t - S_t)$. ON's total costs are

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

In line with [14], [28] and [29], we only apply opportunity costs to the excess cash not invested. If $S_t = D_t$, then ON incurs the lowest costs possible. If $S_t > D_t$, then ON receives less interest payments. If $S_t < D_t$, then ON receives more interest payments; this however is overcompensated by the incurred credit costs. As can be seen, it is always best to convert exactly D_t at t. If ON knew the future demands, then she could always ensure that $S_t = D_t$ for $t = 1, \ldots, T$, or in other words, $\mathbf{S} = \mathbf{D}$. A player with such knowledge is denoted as OPT.

ON's costs for extracting S_t at t for an uncertain demand D_t is (regardless of the occurring scenario) $cD_t + j\max(0, D_t - S_t) + h\max(0, S_t - D_t)$, while OPT's incurred costs are cD_t.

ON's total costs using $\mathbf{S}$ on sequence $\mathbf{D}$ is

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

while OPT's incurred accumulated costs are $OPT(\mathbf{D}) = c\sum\limits_{t=1}^{T} D_t$.

One simple online algorithm which has no knowledge about any parameters is called *learning cash supply* (LCS) (see [28]). 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

$$S_t^{LCS} = D_{t-1} \tag{1}$$

for all $t = 1, \ldots, T$.

ON's regret $R_t(D_t, S_t)$ for using S_t in period t with demand D_t is the difference of the costs incurred by ON compared to the costs incurred by OPT, formally

$$R_t(D_t, S_t) \; = j\max(0, D_t - S_t) + h\max(0, S_t - D_t). \tag{2}$$

In order for ON to minimize the maximal regret (i.e. the additional costs) in period t, she needs to find that S_t which balances two possible max-regret movements. The first one is a maximization of $j\left(D_t - S_t\right)$; the second one is a maximization of $h\left(S_t - D_t\right)$. Since minimizing the max regret when extracting too little implies maximizing the regret when extracting too much and viz versa, there must exists an S_t' for which the following holds,

$$j \max_{D_t}\left(D_t - S_t'\right) = h \max_{D_t}\left(S_t' - D_t\right). \tag{3}$$

This S_t' minimizes the maximal regret in period t. If (2) is independent of t, then always selecting S_t' results in minimizing the competitive difference or maximal regret until T. From this it follows also that S_t' must be independent of time and thus constant.
If the demands are depending on time, then using S_t' in every t does not minimize the competitive difference. Rather, whole sequences of possible cash demands and supplies must be considered. ON's competitive difference (or overall regret) $R\left(\mathbf{D}, \mathbf{S}\right)$ when using $\mathbf{S}$ on sequence $\mathbf{D}$ is merely

$$R\left(\mathbf{D}, \mathbf{S}\right) \;= \textstyle\sum_{t=1}^{T} R_t\left(D_t, S_t\right). \tag{4}$$

In order for ON to minimize the competitive difference after T periods, she needs to use the supply sequence $\mathbf{S}^*$ with

$$\mathbf{S}^* = \arg\min_{\mathbf{S}\in\mathcal{S}} \max_{\mathbf{D}\in\mathcal{D}} R\left(\mathbf{D}, \mathbf{S}\right). \tag{5}$$

3. Min-max regret algorithms for the cash management problem

3.1. Cash management with known m and M

All demands D_t vary between m and M, i.e. $D_t \in [m, M]\,\forall t$, with $0 < m \leq M < \infty$. Clearly, the arbitrary cash supply S_t must also be between m and M. (3) thus becomes

$$\begin{gathered} j \max_{D_t\in[m,M]}\left(D_t - S_t'\right) = h \max_{D_t\in[m,M]}\left(S_t' - D_t\right) \\ j\left(M - S_t'\right) = h\left(S_t' - m\right). \end{gathered}$$

Thus the maximal regret is minimized for an arbitrary period t when selecting S_t' with $S_t' = S' = \dfrac{jM + hm}{j + h}$.
The algorithm *min-max regret for bounded demands* (MRBD) minimizes competitive difference by constantly selecting S'.
Algorithm MRBD: In the beginning of an arbitrary period t convert into

cash until you have the amount of $S^{MRBD} = S'$ available.
We now prove competitive difference when using MRBD followed by the proof of MRBD minimizing competitive difference.

Theorem 1. *The competitive difference of MRBD is*

$$\max_{\mathbf{D}\in\mathcal{D}} R\left(\mathbf{D},\mathbf{S}'\right) = Tjh\frac{M-m}{j+h}.$$

Proof. Assume ON uses S' in period 1. Since it is of no importance whether demands drop to m or surge to M, we assume the latter. Using (2), ON incurs a regret of

$$\begin{aligned} R_1\left(M,S'\right) &= j\max\left(0,M-S'\right) + h\max\left(0,S'-M\right) \\ &= j\left(M-S'\right) \\ &= jh\frac{M-m}{j+h}. \end{aligned}$$

This repeats $T-1$ times. The supply strategy $\mathbf{S}'$ merely contains T times the cash supply S'. Consequently, the competitive difference after T periods using MRBD is (using (4))

$$\max_{\mathbf{D}\in\mathcal{D}} R\left(\mathbf{D},\mathbf{S}'\right) = Tjh\frac{M-m}{j+h}.$$

$\square$

Theorem 2. *MRBD minimizes competitive difference and thus achieves min-max regret.*

Proof. Assume ON deviates from S' by an $\epsilon_t \in [m-S', M-S']$ in an arbitrary period t. Thus, ON incurs the maximal regret

$$\begin{aligned} \max_{D_t\in[m,M]} R_t\left(D_t, S'+\epsilon_t\right) &= j\max\left(0,D_t-S'-\epsilon_t\right) + h\max\left(0,S'+\epsilon_t-D_t\right) \\ &= \max\left(j\left(M-S'-\epsilon_t\right), h\left(S'+\epsilon_t-m\right)\right) \\ &\geq R_t\left(M,S'\right). \end{aligned}$$

We observe that it is not possible to obtain a lower maximal regret in an arbitrary period. Therefore, deviating from $\mathbf{S}'$ implies a higher competitive difference; thus MRBD minimizes competitive difference and thus fulfills the min-max regret criterion. $\square$

The optimal online algorithm *optimal strategy* (OS) of [14] proposes to extract

$$S_t^{OS} = \frac{(j+h)\,mM}{jm+hM} \forall t \tag{6}$$

and incurs a competitive ratio of $1 + \dfrac{jh\,(Mm^{-1}-1)}{c\,(hMm^{-1}+j)}$.

3.2. Cash management with known θ_1, θ_2, D_0 and T

The demands are interrelated with their preceding demands, i.e. $D_t \in [D_{t-1}\theta_1, D_{t-1}\theta_2]\,\forall t$ with θ_1 being the maximal decrease factor and θ_2 being the maximal increase factor between two consecutive periods. Clearly, S_t must also be an element of that interval. Note that in the beginning of $t=1$ we have knowledge about D_0.

Contrary to Subsection 3.1, in order to find an algorithm which minimizes competitive difference, we must solve (5) since demands are interrelated. We start the design of a min-max regret algorithm and use backward induction to derive it. In the last period T ON must consider a demand interval of $D_T \in [D_{T-1}\theta_1, D_{T-1}\theta_2]$. Using (2), we have

$$j \max_{D_T \in [D_{T-1}\theta_1, D_{T-1}\theta_2]} (D_T - S_T^*) = h \max_{D_T \in [D_{T-1}\theta_1, D_{T-1}\theta_2]} (S_T^* - D_T)$$
$$j\,(D_{T-1}\theta_2 - S_T^*) = h\,(S_T^* - D_{T-1}\theta_1)\,,$$

with S_T^* being the value for which the competitive difference in the last period is minimized. Consequently, S_T^* is

$$S_T^* = D_{T-1}\frac{\theta_1 h + \theta_2 j}{j+h}. \tag{7}$$

From the above equation, we derive a heuristic algorithm called *heuristic min-max regret for interrelated demands* (HMRID).

Algorithm HMRID: In the beginning of an arbitrary period t convert into cash until you have the amount of S_t^{HMRID} available with

$$S_t^{HMRID} = D_{t-1}\frac{\theta_1 h + \theta_2 j}{j+h}.$$

We now prove competitive difference when using HMRID.

Theorem 3. *The competitive difference of HMRID is*

$$\max_{\mathbf{D}\in\mathcal{D}} R\left(\mathbf{D}, \mathbf{S}^{HMRID}\right) = \frac{D_0 jh\,(\theta_2-\theta_1)\left(\theta_2^T - 1\right)}{(j+h)\,(\theta_2-1)}.$$

Proof. Assume ON uses HMRID until T. Then it is of no importance whether the demand increases by θ_2 or declines by θ_1. Note that this only concerns the last period. Clearly, the incurred maximal regret during period T is

$$\begin{aligned} R_T\left(D_{T-1}\theta_2, \mathbf{S}_t^{HMRID}\right) &= j\left(D_{T-1}\theta_2 - S_T^{HMRID}\right) \\ &= \frac{D_{T-1}jh}{j+h}\left(\theta_2-\theta_1\right). \end{aligned} \tag{8}$$

Denote $r_t\left(S_t\right)$ to be the ratio of $R_t\left(D_t, S_t\right)$ and D_{t-1}, formally

$$r_t\left(S_t\right) = \frac{R_t\left(D_t, S_t\right)}{D_{t-1}}.$$

Observing (8), we must maximize D_{T-1} in order to maximize the regret during period T. This can only be achieved by a constant increase in demands until D_{T-1} with $D_{T-1} = D_0\theta_2^{T-2}$. Thus, the competitive difference when using HMRID is (according to (4))

$$\begin{aligned} \max_{\mathbf{D}\in\mathcal{D}} R\left(\mathbf{D}, \mathbf{S}^{HMRID}\right) &= \frac{D_0jh}{j+h}(\theta_2-\theta_1) + \frac{D_0jh}{j+h}\theta_2(\theta_2-\theta_1) + \ldots \\ & \quad + \cdots + \frac{D_0jh}{j+h}\theta_2^{T-1}\left(\theta_2-\theta_1\right) \\ &= \frac{D_0jh\left(\theta_2-\theta_1\right)}{j+h}\sum_{t=0}^{T-1}\theta_2^t \\ &= \frac{D_0jh\left(\theta_2-\theta_1\right)\left(\theta_2^T-1\right)}{\left(j+h\right)\left(\theta_2-1\right)}. \end{aligned}$$

□

We continue with the design of an algorithm for minimizing competitive difference. We already found S_T^* (see (7)) and the associated maximal regret (see (8)). We consider now period $T-1$; again, there are the two possible demand movements that maximize regret in that period. They must be balanced against each other again, keeping in mind the respective regret of the later period T. Formally, we solve the following expression

$$\begin{aligned}
S^*_{T-1} &= \arg\min_{S_{T-1}} \max_{D_{T-1}\in[D_{T-2}\theta_1 D_{T-2}\theta_2]} R_{T-1}\left(D_{T-1}, S_{T-1}\right) + R_T\left(D_{T-1}\theta_2, S^*_T\right) \\
&= \arg\min_{S_{T-1}} \max_{D_{T-1}\in[D_{T-2}\theta_1 D_{T-2}\theta_2]} R_{T-1}\left(D_{T-1}, S_{T-1}\right) + D_{T-1} r_T\left(S^*_t\right) \\
&= \arg\min_{S_{T-1}} \max\left(R_{T-1}\left(D_{T-2}\theta_1, S_{T-1}\right) + D_{T-2}\theta_1 r_T\left(S^*_T\right),\right. \\
&\left. R_{T-1}\left(D_{T-2}\theta_2, S^*_{T-1}\right) + D_{T-2}\theta_2 r_T\left(S^*_T\right)\right) \\
&= \arg\min_{S_{T-1}} \max\left(h\left(S_{T-1} - D_{T-2}\theta_1\right) + D_{T-2}\theta_1 r_T\left(S^*_T\right),\right. \\
&\left.\left(D_{T-2}\theta_2 - S_{T-1}\right) + D_{T-2}\theta_2 r_T\left(S^*_T\right)\right).
\end{aligned}$$

Again, we balance the left-hand side against the right-hand side of the max expression to find S^*_{T-1}. Finally, we obtain

$$S^*_{T-1} = D_{T-2}\frac{\theta_1 h + \theta_2 j + r_T\left(S^*_T\right)\left(\theta_2 - \theta_1\right)}{j+h}.$$

The resulting $r_{T-1}\left(D_{T-1}, S^*_{T-1}\right)$ is

$$\begin{aligned}
r_{T-1}\left(D_{T-1}, S^*_{T-1}\right) &= \frac{R_{T-1}\left(D_{T-1}, S^*_{T-1}\right)}{D_{T-2}} \\
&= \frac{j\left(D_{T-2}\theta_2 - S^*_{T-1}\right)}{D_{T-2}} \\
&= \frac{j\left(\theta_2 - \theta_1\right)}{j+h}\left(h - r_T\left(S^*_T\right)\right).
\end{aligned}$$

Using the same approach as before, we obtain S^*_{T-2} in period $T-2$ with

$$S^*_{T-2} = \frac{D_{T-3}\left(\theta_1 h + \theta_2 j + r_{T-1}\left(\theta_2 - \theta_1\right) + r_T\theta_2\left(\theta_2 - \theta_1\right)\right)}{j+h}.$$

The resulting $r_{T-2}\left(D_{T-2}, S^*_{T-2}\right)$ is

$$r_{T-2}\left(D_{T-2}, S^*_{T-2}\right) = \frac{j\left(\theta_2 - \theta_1\right)}{j+h}\left(h - r_{T-1}\left(D_{T-1}, S^*_{T-1}\right) - r_T\left(D_T, S^*_T\right)\theta_2\right).$$

From these above information we derive a recursive formula for S^*_{T-i} and $r_{T-i}\left(D_{T-i}, S^*_{T-i}\right)$ with $i = 0, \ldots, T-1$. Formally, we have

$$S^*_{T-i} = \frac{D_{T-i-1}}{j+h}\left(\theta_1 h + \theta_2 j + \left(\theta_2 - \theta_1\right)\sum_{\tau=0}^{i-1} r_{T-i+\tau+1}\left(D_{T-i+\tau+1}, S^*_{T-i+\tau+1}\right)\theta_2^\tau\right).$$

This can be written as

$$S_t^* = \frac{D_{t-1}}{j+h}\left(\theta_1 h + \theta_2 j + (\theta_2 - \theta_1)\sum_{\tau=0}^{T-t-1} r_{t+\tau+1}\left(D_{t+\tau+1}, S_{t+\tau+1}^*\right)\theta_2^\tau\right).$$

Furthermore, we have

$$r_{T-i}\left(D_{T-i}, S_{T-i}^*\right) = \frac{j(\theta_2-\theta_1)}{j+h}\left(h - \sum_{\tau=0}^{i-1} r_{T-i+\tau+1}\left(D_{T-i+\tau+1}, S_{T-i+\tau+1}^*\right)\theta_2^\tau\right),$$

which can again be written as

$$r_t\left(D_t, S_t^*\right) = \frac{j(\theta_2-\theta_1)}{j+h}\left(h - \sum_{\tau=0}^{T-t-1} r_{t+\tau+1}\left(D_{t+\tau+1}, S_{T-i+\tau+1}^*\right)\theta_2^\tau\right).$$

From the above formulas, we derive the algorithm called *min-max regret for interrelated demands* (MRID).
Algorithm MRID: In the beginning of period 1, calculate $r_{T-i}\left(D_{T-i}, S_{T-i}^*\right)$ for $i = 0$ to $T-1$. Then, in each period $t = 1, \ldots, T$

1. Calculate $S_t^{MRID} = \min\left(D_{t-1}\theta_2, S_t^*\right)$.

2. Receive D_t.

We now prove competitive difference when using MRID followed by the proof of MRID minimizing competitive difference.

Theorem 4. *The competitive difference of MRID is*

$$\begin{aligned}\max_{\mathbf{D}\in\mathcal{D}} R\left(\mathbf{D}, \mathbf{S}^{MRID}\right) \;=\;& \frac{jhD_0(\theta_2-\theta_1)}{j+h}\sum_{t=1}^{T}\theta_2^{t-1}\max\left(0, 1-\right.\\ &\left.\frac{1}{h}\sum_{\tau=0}^{T-t-1} r_{t+\tau+1}\left(D_{t+\tau+1}, S_{t+\tau+1}^{MRID}\right)\theta_2^\tau\right).\end{aligned}$$

Proof. Assume ON uses MRID. In order to derive the competitive difference after T periods, let the demands always increase by θ_2 between two consecutive periods. Using (4), ON's competitive difference is

$$\begin{aligned}
\max_{\mathbf{D}\in\mathcal{D}} R(\mathbf{D},\mathbf{S}) &= \max_{\mathbf{D}\in\mathcal{D}} \sum_{t=1}^{T} R_t\left(D_t, S_t^{MRID}\right) \\
&= \sum_{t=1}^{T} R\left(D_0\theta_2^t, S_t^{MRID}\right) \\
&= j\sum_{t=1}^{T} D_0\theta_2^t - S_t^{MRID} \\
&= \frac{jD_0}{j+h}\sum_{t=1}^{T}\theta_2^t(j+h) - \min\Big(\theta_2^t(j+h), \theta_2^{t-1}(\theta_1 h + \theta_2 j \\
&+ (\theta_2-\theta_1)\sum_{\tau=0}^{T-t-1} r_{t+\tau+1}\left(D_{t+\tau+1}, S_{t+\tau+1}^{MRID}\right)\theta_2^{\tau}\Big)\Big) \\
&= \frac{jD_0}{j+h}\sum_{t=1}^{T}\theta_2^{t-1}\Big(\theta_2(j+h) - \min\big(\theta_2(j+h), \theta_1 h + \theta_2 j \\
&+ (\theta_2-\theta_1)\sum_{\tau=0}^{T-t-1} r_{t+\tau+1}\left(D_{t+\tau+1}, S_{t+\tau+1}^{MRID}\right)\theta_2^{\tau}\big)\Big) \\
&= \frac{jD_0(\theta_2-\theta_1)}{j+h}\sum_{t=1}^{T}\theta_2^{t-1}\max\Big(0, h- \\
&\sum_{\tau=0}^{T-t-1} r_{t+\tau+1}\left(D_{t+\tau+1}, S_{t+\tau+1}^{MRID}\right)\theta_2^{\tau}\Big) \\
&= \frac{jhD_0(\theta_2-\theta_1)}{j+h}\sum_{t=1}^{T}\theta_2^{t-1}\max\Big(0, 1- \\
&\frac{1}{h}\sum_{\tau=0}^{T-t-1} r_{t+\tau+1}\left(D_{t+\tau+1}, S_{t+\tau+1}^{MRID}\right)\theta_2^{\tau}\Big).
\end{aligned}$$

□

Theorem 5. *MRID minimizes competitive difference and thus achieves min-max regret.*

Proof. This follows immediately from the design of MRID. An arbitrary S_t^{MRID} is bounded from above by $D_{t-1}\theta_2$ and represents the cash supply which minimizes the maximal regret incurred in period t while observing future worst-case demands. If ON were to deviate at the beginning of an arbitrary period t from S_t^{MRID}, then the overall incurred competitive difference would increase. □

The optimal online algorithm *balanced cash supply for interrelated de-*

mands (BCSID) of [28] and [29] proposes to extract

$$S_t^{BCSID} = D_{t-1}\theta_1\theta_2\frac{(j+h)}{j\theta_1 + h\theta_2}\forall t \tag{9}$$

and incurs a competitive ratio of $1 + \dfrac{jh\left(\theta_2 - \theta_1\right)}{c\left(j\theta_1 + h\theta_2\right)}$.

3.3. Numerical example

We now consider the following numerical example in order to illustrate the applicability of our algorithms in real world scenarios. A manager must extract cash for five consecutive periods in order to form a budget for a department within a company, i.e. $T = 5$. The department needed $\$1,000,000$ prior to the current period $t = 1$, i.e. $D_0 = 1.00E + 6$. Due to internal regulations within the company, the manager knows that the department will not increase (decrease) their spending by more than 5% (3%) between two consecutive periods, i.e. $\theta_1 = 0.97$ and $\theta_2 = 1.05$. In addition, the manager knows that the demands are globally bounded between $\$920,000 = m$ and $\$1,030,000 = M$. The manager wants to minimize the cumulated additional costs (regret) when extracting the cash supply sequence $\mathbf{S}$. The opportunity cost factor is $h = 0.08$ and the shortage cost factor is $j = 0.05$. The manager uses the BCSID, OS, LCS, MRID, HMRID and MRBD. The results are given in Table 1. The lowest values of regret are given in bold, the highest ones are given in italic.

All supply strategies $\mathbf{S}^{ALG}$ are given in million dollars ($\$1MM = \$1 \cdot 10^6$); all regrets incurred are given in thousand dollars ($\$1M = \$1 \cdot 10^3$). As can be seen in the table, MRID incurs the lowest additional costs in sequence $\mathbf{D}_1$. All other algorithms are compared to MRID by calculating $R\left(\mathbf{D}_1, \mathbf{S}^{ALG}\right) \cdot R\left(\mathbf{D}_1, \mathbf{S}^{MRID}\right)^{-1} - 1$. Although being a heuristic algorithm, HMRID (4%) outperforms all other algorithms (except MRID). LCS (6%) performs well compared to BCSID (8%) and all algorithms which need the knowledge of m and M. OS (39%) and MRBD (31%) are worst. MRBD performs best on sequence $\mathbf{D}_2$, followed by BCSID (10%); here we compare the performance of all algorithms to the one of MRBD by calculating $R\left(\mathbf{D}_1, \mathbf{S}^{ALG}\right) \cdot R\left(\mathbf{D}_1, \mathbf{S}^{MRBD}\right)^{-1} - 1$. Interestingly, MRID (34%) performs worst.

We see that all three algorithms have their advantage; when MRID is performing well, then HMRID can be relatively close to it. If it underperforms,

ALG	BCSID	OS	LCS	MRID	HMRID	MRBD
$\mathbf{D}_1 = 1.01, 0.98, 1.00, 1.03, 1.03$ [in \$1MM]						
in \$1MM						
S_1^{ALG}	1.00	0.98	1.00	1.01	1.00	0.99
S_2^{ALG}	1.01	0.98	1.01	1.02	1.01	0.99
S_3^{ALG}	0.98	0.98	0.98	0.99	0.99	0.99
S_4^{ALG}	1.00	0.98	1.00	1.01	1.01	0.99
S_5^{ALG}	1.03	0.98	1.03	1.03	1.03	0.99
in \$1M						
$R_1\left(D_1, S_1^{ALG}\right)$	0.72	*1.45*	0.69	**0.34**	0.65	1.30
$R_2\left(D_2, S_2^{ALG}\right)$	2.25	**0.01**	2.31	*2.74*	2.37	0.22
$R_3\left(D_3, S_3^{ALG}\right)$	*1.01*	0.99	0.98	**0.79**	0.94	0.84
$R_4\left(D_4, S_4^{ALG}\right)$	1.21	*2.16*	1.17	**1.06**	1.13	2.01
$R_5\left(D_5, S_5^{ALG}\right)$	0.14	*2.26*	0.1	**0.02**	0.07	2.12
$R\left(\mathbf{D}_1, \mathbf{S}^{ALG}\right)$	5.33	*6.87*	5.25	**4.95**	5.15	6.49
in %						
$\frac{R\left(\mathbf{D}_1, \mathbf{S}^{ALG}\right)}{R\left(\mathbf{D}_1, \mathbf{S}^{MRID}\right)} - 1$	8	*39*	6	**0**	4	31
$\mathbf{D}_2 = 0.99, 1.01, 1.02, 1.00, 0.98$ [in \$1MM]						
in \$1MM						
S_1^{ALG}	1.00	0.98	1.00	1.01	1.00	0.99
S_2^{ALG}	0.99	0.98	0.99	0.99	0.99	0.99
S_3^{ALG}	1.01	0.98	1.01	1.02	1.01	0.99
S_4^{ALG}	1.02	0.98	1.02	1.02	1.02	0.99
S_5^{ALG}	1.00	0.98	1.00	1.00	1.00	0.99
in \$1M						
$R_1\left(D_1, S_1^{ALG}\right)$	0.83	0.21	0.89	*1.45*	0.95	**0.06**
$R_2\left(D_2, S_2^{ALG}\right)$	1.30	*1.47*	1.26	**0.99**	1.22	1.32
$R_3\left(D_3, S_3^{ALG}\right)$	0.20	*1.63*	0.16	**0.05**	0.12	1.48
$R_4\left(D_4, S_4^{ALG}\right)$	1.30	0.79	1.36	*1.54*	1.42	**0.64**
$R_5\left(D_5, S_5^{ALG}\right)$	1.55	**0.35**	1.61	*1.80*	1.67	0.59
$R\left(\mathbf{D}_2, \mathbf{S}^{ALG}\right)$	5.18	4.45	5.28	*5.83*	5.39	**4.10**
in %						
$\frac{R\left(\mathbf{D}_2, \mathbf{S}^{ALG}\right)}{R\left(\mathbf{D}_2, \mathbf{S}^{MRBD}\right)} - 1$	10	15	14	*34*	19	**0**

Table 1: Performance of online algorithms for two demands sequences

then HMRID can be substantially better. HMRID appears to be a balanced solution between minimizing the maximal regret and average-case performance. MRBD can also perform well for some sequences of demand.

4. Numerical testing

In this section we conduct a numerical test of all three newly introduced algorithms. We also add the already established algorithms BCSID (see (9)) and LCS (see (1)) of [28] and [29], and OS (see (6)) of [14]. We first present the way the testing is done and then present and interpret the results.

4.1. Test-setup

We first need a random variable $X \sim \mathcal{U}(0,1)$ which originates from the standard uniform distribution. The expected value $E(X)$ is 0.5. The two boundaries of X are 0 and 1. We now define the demand sequences based on θ_1, θ_2, D_0 and X. The demands in these sequences are generated in the following way,

$$D_t(X) = D_{t-1}(X)\,\theta_1^{1-2\min(0.5,X)}\theta_2^{2\max(0.5,X)-1} \text{ and } D_0(X) = 1. \tag{10}$$

The expected value of $D_t(X)$ is $D_{t-1}(X)$. 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 $D_{t-1}(X)\,\theta_2$. In other words, the demands cannot change more than by the factor θ_1 and θ_2; this is consistent with the considered cash management problem with given $\theta_1, \theta_2, D_0, h$ and j. Using (10) we generate arbitrary demand sequences $\mathbf{D}$ using $X, T = 50$ and various $\theta_1\theta_2, j$ and h. To enhance interpretation, we fix $j = 1$. We calculate the median, the 5 permille and 995 permille and the standard deviation σ in terms of regret incurred in $1,000$ experiments for every unique combination. We use the standard parameter setting of $\theta_1 = 0.50, \theta_2 = 2, j = h = 1$ and $T = 50$.
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.
As for the selected pairs of θ_1 and θ_2, we start with $\theta_1 = \theta_2 = 1$ and let θ_2 absolutely increase by 0.05 until $\theta_2 = 10$. This yields 181 different values of θ_2. For θ_1, we let it decrease by the following equation, where θ_1^n denotes the new, lower θ_1 and θ_1^o denotes the old, higher θ_1,

$$\theta_1^n = \left((\theta_1^o)^{-1} + 0.05\right)^{-1}. \tag{11}$$

This continues until $\theta_1 = 0.10$; this yields again 181 different values for θ_1 (overall there are $32,761$ combinations).
For hj we start at 10 and absolutely decline by 0.05 until 1 (yielding 181 values). From that value on, we let the values decline in the same way as θ_1 (as can be seen in (11)) until $hj = 0.10$ (yielding 180 values).
Finally, all algorithms have knowledge about their needed parameters, i.e. the correct values of $\theta_1, \theta_2, T, D_0(X), m, M, j$ and h.

4.2. Results

We now consider the median performance of all six algorithms as depicted in Figure 1. The following observations are made:

- MRID is worse than all other algorithms whenever $\theta_1 \to 10^{-1}$ and $\theta_2 \to 10$. MRBD's median is highest for all other combinations of θ_1 and θ_2.
- MRID is better than all other algorithms (except for LCS) when $\theta_2 \to 1$.
- HMRID and OS are better than MRBD regardless of the selected θ_1 and θ_2.
- OS tends to be better than HMRID whenever $\theta_2 \geq 3$.
- LCS or BCSID always outperform all other algorithms.
- LCS is better than BCSID if either $\theta_2 \to 10$ and $\theta_1 \to 1$ or $\theta_1 \to 10^{-1}$ and $\theta_2 \to 1$.

We now consider the 5 permille performance of all six algorithms as depicted in Figure 2. The following observations are made:

- LCS is best whenever $\theta_2 \to 10$ and $\theta_1 \to 1$. BCSID is best for all other combinations.
- BCSID is better than OS, MRBD and MRID regardless of the selected θ_1 and θ_2.
- HMRID is best whenever $\theta_2 \approx 1$.
- HMRID is better than MRID for arbitrary θ_1 and θ_2.

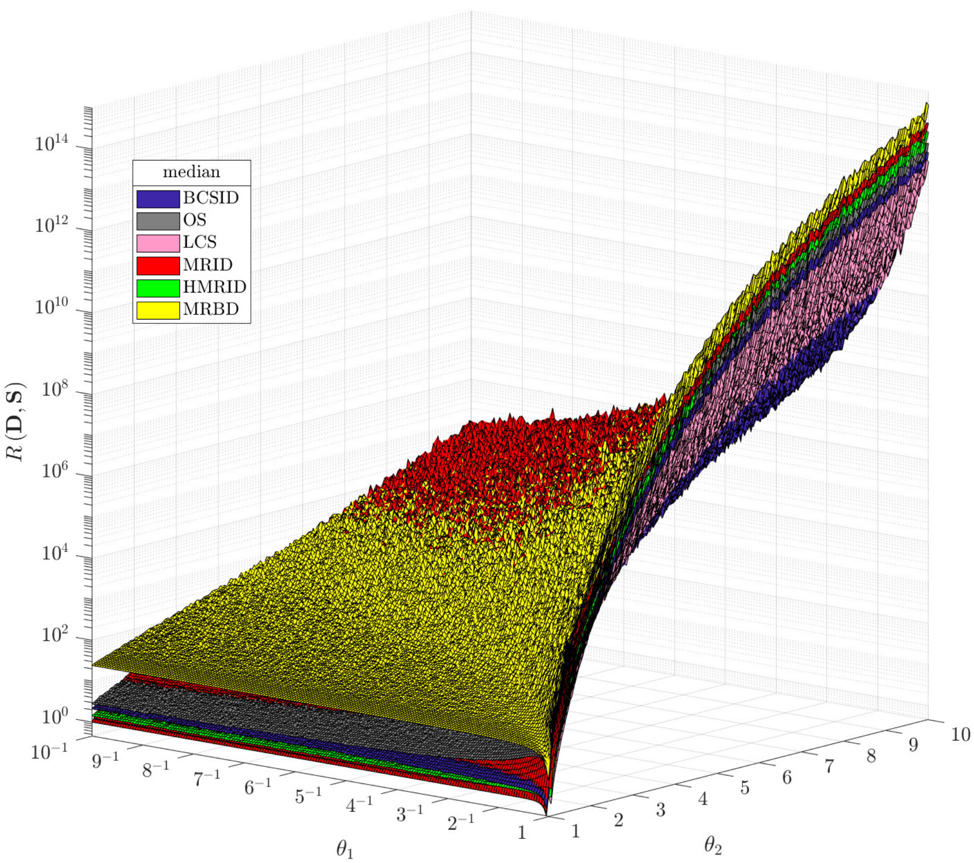

Figure 1: Influence of θ_1 and θ_2 on the median performance of the algorithms using the standard parameter setting with 1,000 experiments and $T = 50$

- MRBD is worst whenever $\theta_2 \to 10$ and $\theta_1 \to 1$ or $\theta_2 < 2\theta_1^{-1}$. MRID is worst for all other combinations.

We now consider the 995 permille performance of all six algorithms as depicted in Figure 3. The following observations are made:

- MRBD is worst, regardless of the selected parameter combination.

- MRID is worse than OS if $\theta_2 > 2$.

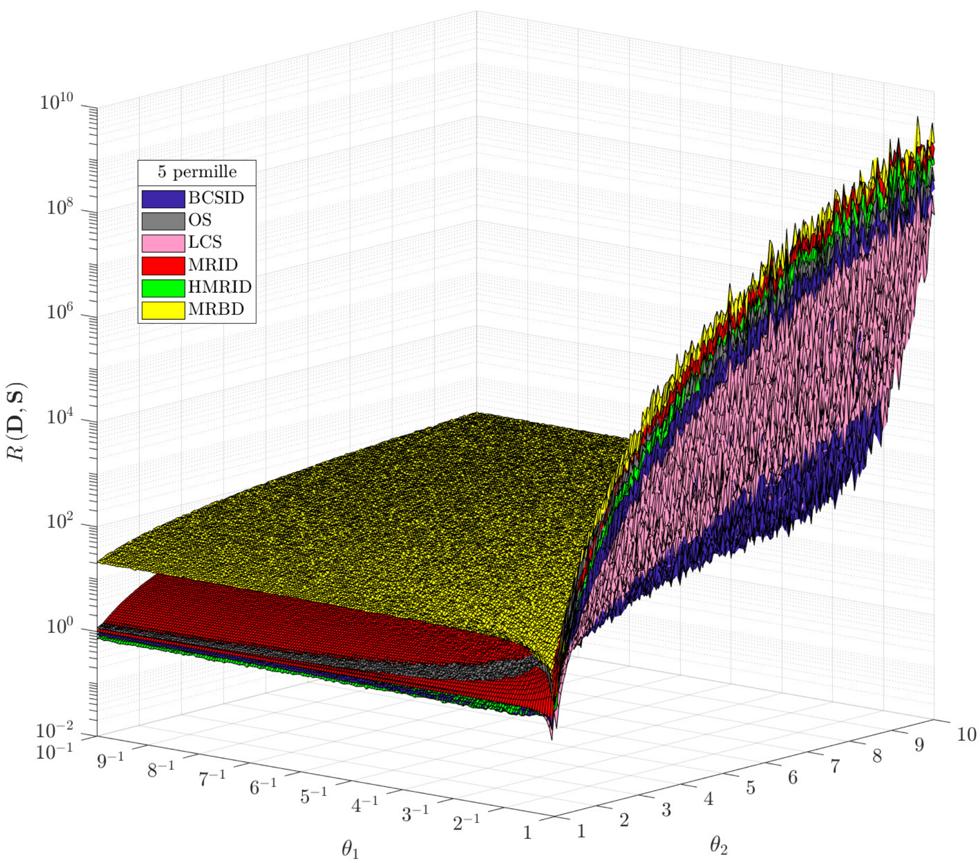

Figure 2: Influence of θ_1 and θ_2 on the 5 permille performance of the algorithms using the standard parameter setting with 1,000 experiments and $T = 50$

- Again, HMRID is better than MRID for arbitrary θ_1 and θ_2.
- Again, OS tends to be better than HMRID whenever $\theta_2 \geq 3$.
- Again, LCS is best whenever $\theta_2 \to 10$ and $\theta_1 \to 1$. For all other combinations, LCS does not outperform BCSID systematically and vice versa.

We now consider the σ performance of all six algorithms as depicted in Figure 4. The following observations are made:

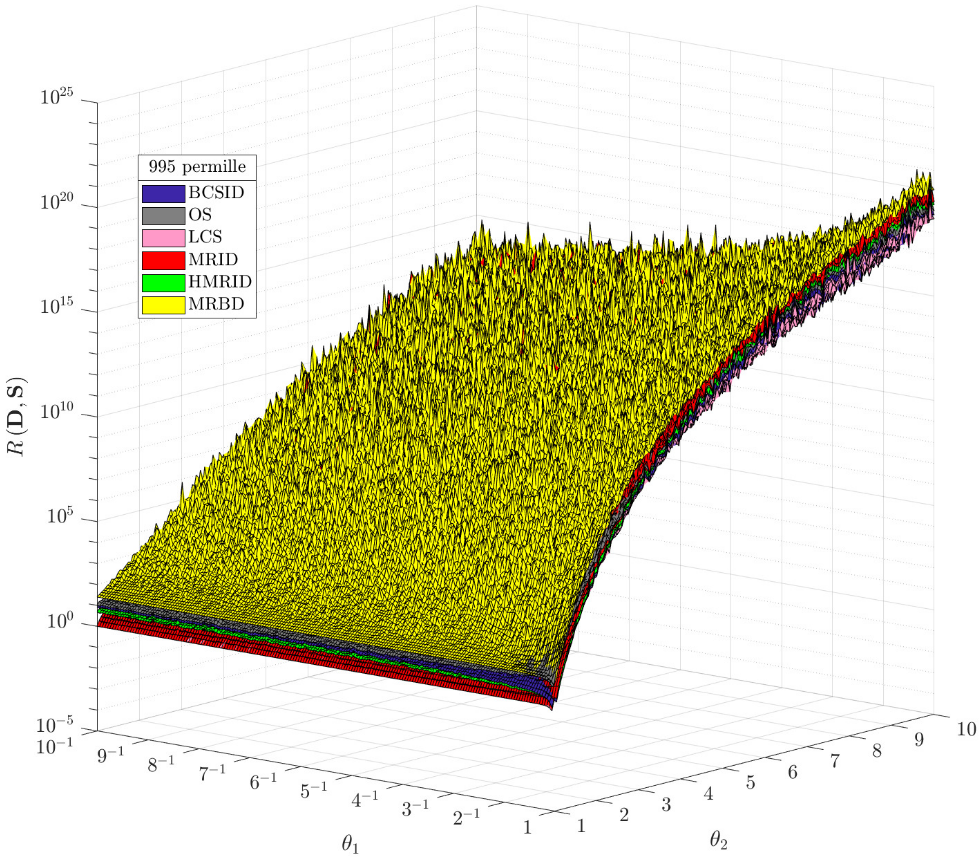

Figure 3: Influence of θ_1 and θ_2 on the 995 permille performance of the algorithms using the standard parameter setting with 1,000 experiments and $T = 50$

- OS is worst whenever $\theta_2 \approx 1$. MRBD is worst for all other parameter combinations.
- Again, MRID is better than all other algorithms (except for LCS) when $\theta_2 \to 1$.
- The observations made in Figure 3 also apply to this figure.

We now consider the performance of all six algorithms for various values of h as depicted in Figure 5. We fix $j = 1, \theta_1 = 0.5$ and $\theta_2 = 2$. The following

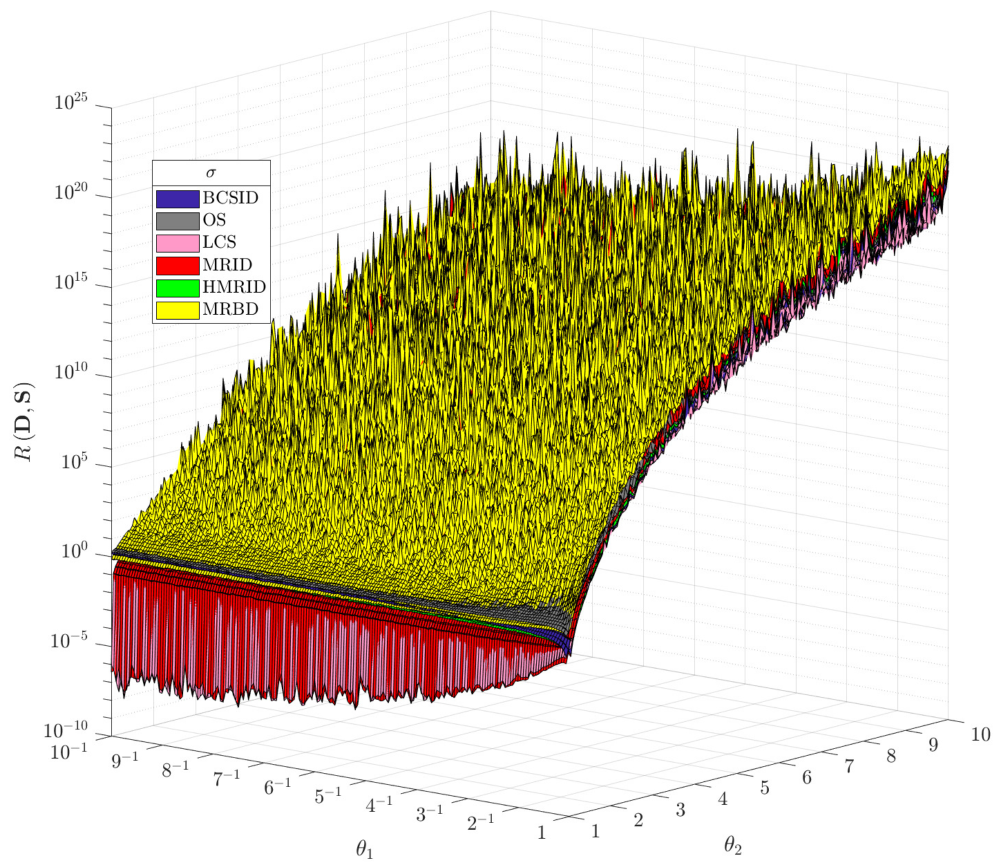

Figure 4: Influence of θ_1 and θ_2 on the σ performance of the algorithms using the standard parameter setting with 1,000 experiments and $T = 50$

observations are made:

- MRID is worst when $h > j$ (median and 5 permille performance)
- HMRID outperforms OS, MRID and MRBD (all four performance measures). Except for the 5 permille performance measure, HMRID tends to be close to BCSID.
- BCSID outperforms all other algorithms if $h > j$ or $h < 4j$ (all four performance measures)

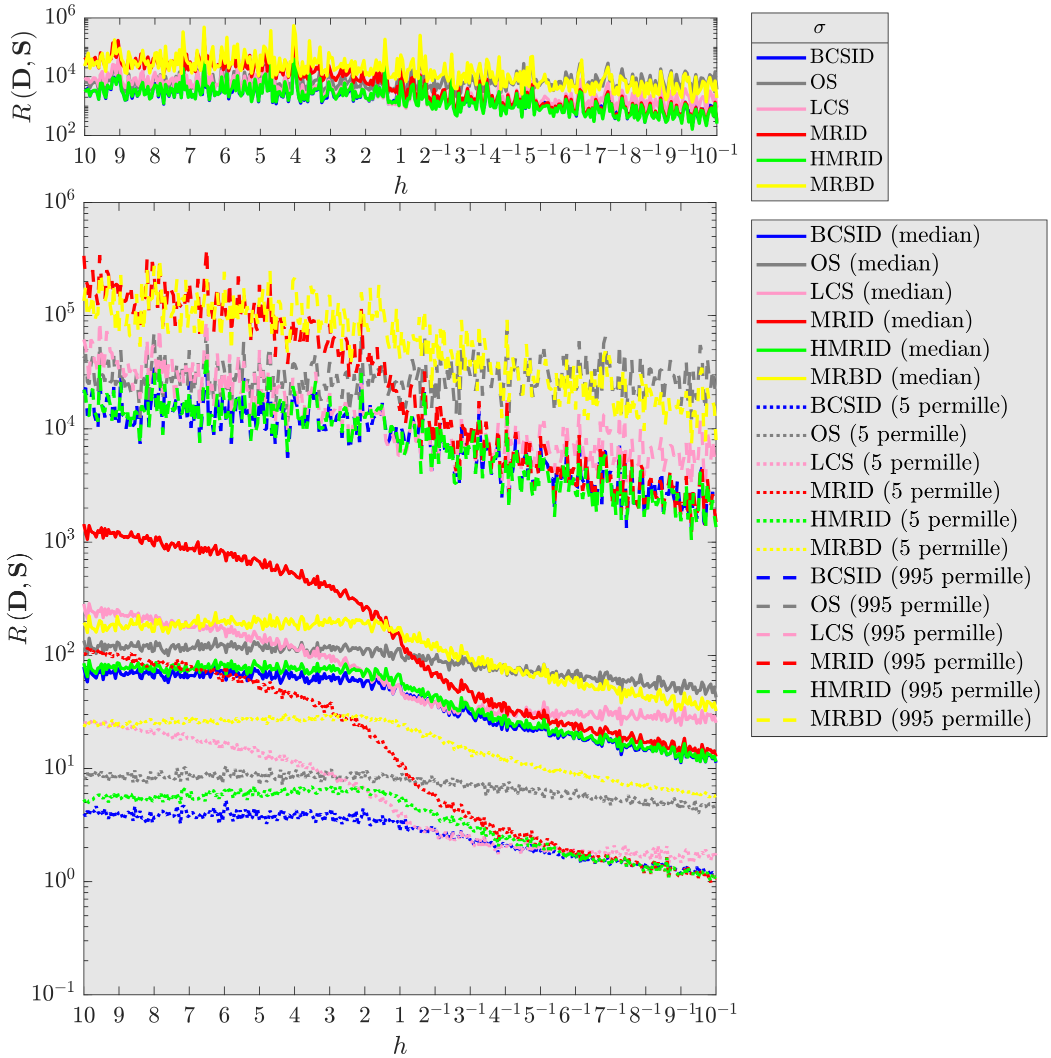

Figure 5: Influence of h on the performance of the algorithms using the standard parameter setting with 1,000 experiments, and $T = 50$

- MRBD is worst whenever $h < 2j$ (5 permille performance)

5. Conclusion

In this paper we solve the cash management problem with uncertain demands that are globally bounded or interrelated. We consider algorithms for this problem that minimize the competitive difference and thus the min-max regret. We first derive the algorithm MRBD when demands are globally bounded. Furthermore, we derive a heuristic algorithm HMRID which solves the problem for interrelated demands. We also derive the algorithm MRID which minimizes maximal regret for interrelated demands. We then test these three new algorithms against the established OS, LCS and BCSID. When knowledge about θ_1, θ_2, D_0 and T is given, then BCSID is clearly outperforming MRID and MRBD; furthermore, if m and M are given, then OS is clearly better than MRBD. The selection of the algorithm depends on the given information and the desired quality of the solution, i.e. whether we want to minimize competitive ratio and tend to perform well on non-worst case sequences or whether we want to minimize competitive difference and risk of performing poorly on non-worst-case sequences. In addition, HMRID is performing well on non-worst case sequences and its competitive difference is low (albeit not as low as the one of MRID).
It remains an open question whether the combined knowledge of m, M, θ_1, θ_2, T and D_0 would further lower the minimal competitive difference and how such an algorithm performs against all tested algorithms. Ultimately, one must evaluate the used model and implement further aspects considered in cash management literature. For instance, we might also consider cash inflows. A cash inflow C_t of period t can be modelled as $C_t \in [C_t\alpha_1, C_t\alpha_2]$; α_1 (α_2) is the maximal relative decrease (increase) in received cash. This will improve the applicability to real life scenarios.

Acknowledgements

The authors thank the three anonymous reviewers for their comments and suggestions which greatly improved this paper. This work has been supported by FP7-EYE.

[1] Aissi, H., Bazgan, C., Vanderpooten, D. (2009). Min-max and min-max regret versions of combinatorial optimization problems: A survey, *European Journal of Operational Research*, *197*(2), 427–438.

[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] Choi, B., Chung, K. (2016). Min-max regret version of a scheduling problem with outsourcing decisions under processing time uncertainty. *European Journal of Operational Research*, *252*(2), 367–375.

[6] Cover, T. M. (1991). Universal Portfolios, *Mathematical Finance*, *1*(1), 1–29.

[7] Dai, W., Dong, Y., Zhang, X. (2016). Competitive analysis of the online financial lease problem, *European Journal of Operational Research*, *250*, 865–873.

[8] El-Yaniv, R., Fiat, A., Karp, R., Turpin, G. (1992). Competitive Analysis of Financial Games. In IEEE Symposium on Foundations of Computer Science (pp. 327-333).

[9] Ferstl, R., Weissensteiner, A. (2010). Cash Management Using Multi-Stage Stochastic Programming, *Quantitative Finance*, *10*(2), 209–219.

[10] Gilbert, H., Spanjaard, O. (2017). A double oracle approach to minmax regret optimization problems with interval data, *European Journal of Operation Research*, *262*, 929–943.

[11] 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.

[12] Kacem, I., Kellerer, H. (2019). Complexity Results for Common Due Date Scheduling Problems with Interval Data and Minmax Regret Criterion, *Discrete Applied Mathematics*, *264*, 76–89.

[13] Krumrey, L., Moeini, M., Wendt, O. (2018). A Cash-Flow-Based Optimization Model for Corporate Cash Management: A Monte-Carlo Simulation Approach. In: Le NT., van Do T., Nguyen N., Thi H. (eds), *Advanced Computational Methods for Knowledge Engineering*, Advances in Intelligent Systems and Computing, 629, 34–46.

[14] Liu, B., Xin, C. (2008). An Online Model for Managing Cash: An Alternative Approach to the Miller-Orr Model. In Proceedings of 2008 Fourth International Conference on Natural Computation (pp. 314–317).

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

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

[17] 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, Decision Engineering*, Springer, 11–28.

[18] Rahimian, H., Bayraksan, G., Homem-de-Mello, T. (2019). Controlling Risk and Demand Ambiguity in Newsvendor Models, *European Journal of Operational Research*, doi: https://doi.org/10.1016/j.ejor.2019.06.036.

[19] Righetto, G. M., Morabito, R., Alem, D. (2016). A Robust Optimization Approach for Cash Flow Management in Stationary Companies, *Computer and Industrial Engineering*, *99*, 137–152.

[20] Salas-Molina, F. (2017). An empirical method to select cash management models, *3C Tecnología: glosas de innovación aplicadas a la pyme*, *6*(3), 1–13.

[21] Salas-Molina, F., Martin, F., M. Rodríguez-Aguilar, J., A., Serrà, J., Arcos, J., Ll. (2017), Empowering cash managers to achieve cost savings by improving predictive accuracy, *International Journal of Forecasting*, *33*, 403–415.

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

[23] Salas-Molina, F., Rodríguez-Aguilar, J., A. (2018). Data-driven multiobjective decision-making in cash management, *EURO Journal on Decision Processes*, *6*(1-2), 77–91.

[24] Salas-Molina, F., Rodríguez-Aguilar, J., A., Díaz-García, P. (2018). Selecting cash management models from a multiobjective perspective, *Annals of Operations Research*, *261*(12), 275–288.

[25] Salas-Molina, F., Rodríguez-Aguilar, J., A., Serrà, J., Guillen, M., Martin, F., M. (2018). Empirical analysis of daily cash flow time-series and its implications for forecasting, *SORT - Statistics and Operations Research Transactions*, *42*(1), 73–98.

[26] Savage, L. J. (1951). The theory of statistical decision, *Journal of the American Statistical Association*, *28*, 202–208.

[27] 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.

[28] Schroeder, P., Kacem, I. (2017). Optimal cash management with uncertain and interrelated demands. In Proceedings of the 47th International Conference on Computer and Industrial Engineering 2017 (pp. 502–508).

[29] Schroeder, P., Kacem, I. (2019). Optimal cash management with uncertain, interrelated and bounded demands, *Computers & Industrial Engineering*, *133*, 195-206.

[30] Schroeder, P., Schmidt, G., Kacem, I. (2016). Optimal on-line algorithms for bi-directional non-preemptive conversion with interrelated conversion rates. In Proceedings of 2016 International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 28–33).

[31] Sleator, D. D., Tarjan, R. E. (1985). Amortized Efficiency of List Update and Paging Rules, *Communications of the ACM*, *28*, 202–208.

[32] Wang, W., Wang, L., Lan, Y., Zhang, J. X. (2016). Competitive difference analysis of the one-way trading problem with limited information, *European Journal of Operational Research*, *252*, 879–887.

[33] Zhang, W., Xu, Y. Zheng, F., Dong, Y. (2016). Optimal Algorithms for Online Time Series Search and One-way Trading with Interrelated Prices, *Journal of Combinatorial Optimization*, *23*, 159–166.