Multidimensional monopolist pricing problem with uncertain valuations

Maciej Drwal
Department of Computer Science, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

Abstract
We consider the problem of revenue maximization by a monopolist selling a set of substitutable goods to an unit-demand buyer of uncertain type. The Bayesian variant of this problem has received much interest in the computer science and economics communities in the past years. Here we assume that no distribution of buyer’s type is known. We consider the problem of finding robust solutions in the min-max regret sense. We give polynomial time algorithms for optimal robust pricing when uncertainty is modeled by intervals or discrete scenario sets.

Keywords: robustness, revenue maximization, min-max regret, mechanism design

1. Introduction
We consider the problem of revenue maximization by a monopolist selling \( n \) heterogenous substitutable non-divisible goods to an unit-demand buyer. It is assumed that the buyer can be described by an \( n \)-dimensional type vector \( \mathbf{x} \in \Omega \subset \mathbb{R}^n_+ \), where \( x_i \) is the buyer’s valuation of \( i \)th item. The utility from buying goods is modeled by a quasi-linear function, i.e., buyer gets utility \( u_i = x_i - p_i \) if he buys \( i \)th item for the price \( p_i \). The buyer is incentive compatible, which means that he would buy an item that gives the highest utility. Ties are broken first in favor for a lower price \( p_i \), then lexicographically with respect to the item indexing. We additionally assume that the buyer is individually rational, and thus would buy only if the utility is non-negative.

The monopolist’s task is to determine the set of prices for the items that would result in the highest revenue, subject to a limited knowledge of buyer’s
type. The revenue is defined as:

\[ r(p) = \max \{ \{ p_i : \exists_i x_i - p_i \geq 0, \forall_j x_i - p_i \geq x_j - p_j \} \cup \{0\} \}. \]

The problem is easy to solve if the buyer’s type is known: the monopolist would set the prices to make sure that the highest-valued item would be sold; the revenue is then equal to the highest valuation of an item. In practical applications however, the monopolist has only a limited knowledge of the buyer’s type. The problem is interesting in Bayesian settings, when a probability distribution of buyer’s type is given. It is known as Bayesian Unit-demand Item Pricing Problem (see Section 2), and has been a subject to intensive research, both in computer science and economics communities.

Here we consider this problem in a more restrictive setting. We assume that no probability distribution is known, and the only available information regarding the buyer’s type is a set of its possible values \( \Omega \). This variant of the problem is of interest especially in the context of introducing a new line of products to the market, when no prior information regarding the customers’ reception of the products is available.

Two special cases are of interest. First case is called interval uncertainty, \( \Omega = \{ \mathbf{x} : x_i \in [x_i^-, x_i^+], i = 1, \ldots, n \} \), i.e., we are given interval \([x_i^-, x_i^+]\) for each item’s valuation. Second case is called discrete uncertainty, \( \Omega = \{ \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(K) \} \), i.e., all possible type vectors are explicitly enumerated and there is a finite number \( K \) of them.

Any type vector \( \mathbf{x} \in \Omega \) is called a scenario in the robust optimization framework, to stress that it is a realization of uncertain quantity. Let \( r(p, \mathbf{x}) \) be the revenue obtained by offering the prices \( p \) for the items valued by the buyer according to the type vector \( \mathbf{x} \).

Given a solution (a price vector) \( p \) and a scenario \( \mathbf{x} \), we define the regret as the difference between the value \( r^*(\mathbf{x}) \) of an optimal solution for the scenario \( \mathbf{x} \) and the value of a solution \( p \) in that scenario:

\[ R(p, \mathbf{x}) = r^*(\mathbf{x}) - r(p, \mathbf{x}). \]  

(1)

A worst-case scenario \( \mathbf{x}(p) \) for a solution (a price vector) \( p \) is the one that maximizes the regret \( R(p, \mathbf{x}) \). We sometimes simply write \( \mathbf{x} \), when it is clear from the context that \( \mathbf{x} \) depends on \( p \). Let us denote:

\[ Z(p) = \sup_{\mathbf{x} \in \Omega} R(p, \mathbf{x}). \]

A solution \( p^* \) is said to be robust if it minimizes the maximum regret \( Z \) among all solutions.
2. Related Works

We present a brief summary of related works. The monopolist pricing problem with stochastic types dates back to the works of Mussa and Rosen [1] (single item pricing) and Armstrong [2] (multiple items pricing). Many techniques have been developed for tackling selected variants of this problem, using techniques from functional analysis and variational calculus [3], [4], [5]. More recently, optimal pricing problems were brought to the attention of the computer science community [6], [7], [8], where this problem is usually called the Bayesian Unit-demand Item Pricing Problem. Many algorithmic and complexity theoretic results were established recently.

The min-max regret approach to robustness has been applied to the analysis of optimization problems in many application areas [9]. Robust monopolist pricing problem has been formulated with the use of min-max regret criterion in [10] and [11]. In the former paper it is assumed that the type distribution is in an appropriately defined neighborhood of a given distribution, while in the latter it is assumed that only the support of type distribution is known. Works [12] and [13] consider a time-dependent variant of the problem and show how to compute robust price schedules. All these papers contain results only for the pricing of a single item. Moreover, they allow the seller to use mixed strategies. In contrast, this paper considers pricing of more than one item by a monopolist, and additionally restricts the solutions to pure (deterministic) pricing strategies. The representations of parameter uncertainty given as intervals or discrete scenario sets constitute two simplest and most commonly encountered cases.

3. Interval Uncertainty

If the vector \( \mathbf{x} \in \Omega \) is known to the seller, then the revenue equal to the maximum valuation \( x_j = \max_i x_i \) can be always extracted by setting prices \( p_j = x_j \) and \( p_i = x_i + \alpha \) for \( i \neq j \), \( \alpha > 0 \). The following theorem characterizes all robust solutions of multidimensional monopolist pricing problem when uncertain valuations are given as compact intervals.

**Theorem 1.** Let \( J = \arg\max \{x_i^+ : i = 1, \ldots, n\} \) and \( j^* \in J \). Denote by \( \mathbf{\tilde{p}} \) a solution (a price vector) such that \( \tilde{p}_{j^*} = \max \{x_{j^*}^-, \frac{1}{2}x_{j^*}^+\} \), and for all \( i \notin J \), let \( \tilde{p}_i \in (\tilde{p}_{j^*} - x_{j^*}^+, x_{j^*}^+, \tilde{p}_{j^*}] \). Solution \( \mathbf{\tilde{p}} \) is robust. Moreover, the value of optimal solution is \( Z^*(\mathbf{\tilde{p}}) = x_{j^*}^+ - \tilde{p}_{j^*} \).

**Proof.** Consider a given vector of prices \( \mathbf{p} \geq 0 \) and the buyer’s valuation vector \( \mathbf{x} \). The value of maximum regret can be expressed explicitly by distinguishing two mutually exclusive cases.
If \( \forall_i x_i < p_i \), then the maximum regret is \( Z_1(p) = \max_i x_i \), since no item is sold at prices \( p \), but the seller can extract \( r^*(x) = \max_i x_i - \epsilon \) of revenue from valuations \( x \), for \( \epsilon \to 0 \). Note that in the worst-case scenario the valuation \( x_j = \max_i x_i \) would be set to the upper bound of the respective interval. Then the worst-case prices would contain \( p_j = x_j + \epsilon \), which results in \( Z_1(p) = p_j \).

If at least one item \( i \) is sold at given price vector \( p \), then the maximum regret is \( Z_2(p) = x_j^+ - p_j^* \), where \( j^* \) is the item with the highest upper-bound \( x_j^+ \) of valuation interval, among all items \( j \) with higher valuation \( x_j \) than price \( p_j \).

To see this, first note that \( Z_2(p) = r^*(x) - r(p, x) \), where \( r^*(x) = \max_i x_i := x_i^* \), and \( r(p, x) = p_j \), where \( j \) is an item such that \( x_j - p_j > x_k - p_k \) for all \( k \neq j \). Assume that \( p_j > p_i^* \). Then \( x_j - p_j > x_i^* - p_i^* \geq x_j - p_i^* \), which implies \( p_j < p_i^* \), a contradiction. Consequently \( Z_2(p) \geq x_i^* - p_i^* \). This value is maximized by selecting the worst-case valuation \( x_i^*(p) = x_i^+ \), for \( i^* \) being the index of an item with the highest upper-bound value. The equality occurs if for the given price vector \( p \) the condition \( x_k - p_k < x_i^* - p_i^* \) is satisfied for all \( k \neq i^* \).

Consequently, the maximum regret can be expressed as:

\[
Z(p) = \max\{Z_1(p), Z_2(p)\} = \max\{p_1, \ldots, p_{j^*}, \ldots, p_n, x_j^+ - p_j^*\},
\]

where \( j^* \) is the index of an item with the highest upper-bound valuation, and provided that for each \( k \neq j^* \), either:

\[
x_k - p_k < x_j^* - p_j^*,
\]

or

\[
x_k - p_k = x_j^* - p_j^* \quad \text{and} \quad p_k \geq p_j^*.
\]

From the expression (2), we get that the condition on \( \tilde{p} = (\tilde{p}_1, \ldots, \tilde{p}_n) \) to minimize the maximum regret is that no price \( \tilde{p}_j \) may exceed \( x_j^+ - p_j^* \). This is achieved when for all \( k \neq j^* \), \( \tilde{p}_k \leq \tilde{p}_j^* \) and the value \( \tilde{p}_j^* \) minimizes the function \( f(p) = \max\{p, x_j^+, p\} \) for \( p \geq x_j^* \). The unique minimum is \( \tilde{p}_j^* = \max\{x_j^*, \frac{1}{2} x_j^+\} \).

Finally, to make sure that (3)–(4) holds for \( \tilde{p} \), item \( j^* \) must be the one that has the greatest utility. Thus we must have for all \( k \neq j^* \), \( x_k(\tilde{p}) - \tilde{p}_k < x_j^* (\tilde{p}) - \tilde{p}_j^* \), or \( \tilde{p}_k \geq \tilde{p}_j^* \) in case when both items \( k \) and \( j^* \) give the same utility. But since \( x_j^* (\tilde{p}) = x_j^+ \), the inequality is always satisfied, if \( x_k^+ - \tilde{p}_k < x_j^+ - \tilde{p}_j^* \).

Consequently, \( \tilde{p}_k > x_k^+ - x_j^* + \tilde{p}_j^* \) for such items \( k \) that \( x_k^+ < x_j^+ \), and \( \tilde{p}_k = \tilde{p}_j^* \), if \( x_k^+ = x_j^+ \).

\[\square\]
4. Discrete Uncertainty

In case of discrete set $\Omega$, $|\Omega| = K$, observe that for all $k = 1, \ldots, K$, $r^*(x^{(k)}) = x_{j^*}^{(k)} = \max_j x_j^{(k)}$. The problem is to minimize the maximum regret:

$$Z(p) = \max_{k \in \{1, \ldots, K\}} \left( x_{j^*}^{(k)} - r(p, x^{(k)}) \right). \quad (5)$$

Note that in a trivial case of a single scenario, an optimal prices would be the ones matching the valuations (adjusted appropriately to exploit the tie-breaking rule in favor for the most valuable item). Given multiple scenarios, intuitively, a robust solution would consist of prices that are close to the valuations in all scenarios simultaneously.

We start by defining this problem as a mixed-integer program. The problem of minimizing $Z(p)$ is equivalent to the problem of minimizing an objective function equal to a new variable $\eta \geq 0$, subject to the following constraints:

$$\forall i = 1, \ldots, K \quad x_j^{(i)} - \sum_{j=1}^{n} y_{ij} p_j \leq \eta,$$

$$\forall i = 1, \ldots, K \quad \sum_{j=1}^{n} y_{ij} \leq 1$$

$$\forall i = 1, \ldots, K \quad \sum_{j=1}^{n} y_{ij} (x_j^{(i)} - p_j) \geq 0,$$

$$\forall i = 1, \ldots, K \forall k = 1, \ldots, n \quad \sum_{j=1}^{n} y_{ij} (x_j^{(i)} - p_j) - \left( 1 - \sum_{j=1}^{n} y_{ij} \right) \epsilon \geq x_k^{(i)} - p_k, \quad (6)$$

$$\forall i = 1, \ldots, K \forall j = 1, \ldots, n \quad y_{ij} \in \{0, 1\}.$$  

In the above set of constraints, the auxiliary variables $y_{ij}$ can be interpreted as indicator variables for item $j$ being sold in scenario $i$ (note that if no item is sold in scenario $i$ then variables with that index sum up to zero). Constant $\epsilon > 0$ that appears in constraint (6) is the smallest possible valuation increment, introduced here in order for the program to be consistent with the assumed tie-breaking rule. The resulting program is a mixed-integer quadratic constrained problem with binary variables $y = [y_{ij}]$ and continuous variables $\eta, p$. Unfortunately, the problem is not convex, and most likely cannot be solved efficiently using standard mathematical programming methods. For a (small) fixed number of scenarios $K$ the problem can be solved by fixing subsequent feasible values of $y$ and then solving the obtained linear program (LP), keeping track of $y$ that results in LP
Algorithm 1 Min-max regret pricing for discrete uncertainty.

```plaintext
1: \( p_j \leftarrow \max \{ x_j^{(i)} : i = 1, \ldots, K \} \) for all \( j = 1, \ldots, n \) (initialize price vector \( p \))
2: \( z_0 \leftarrow 0; z_{max} \leftarrow \infty; s^* \leftarrow 0 \); \( p^* \leftarrow p; Z^* \leftarrow \infty \)
3: \( z_i \leftarrow R(p, x^{(i)}), \) for all \( i = 1, \ldots, K \) (main loop starts here)
4: \( s \leftarrow \arg \max_i z_i \) (determine a scenario giving the maximum regret)
5: if \( z_s^* \geq z_{max} \) then
6: \( \) halt and return \( p^* \) (terminate if the highest regret has not decreased)
7: else
8: \( z_{max} \leftarrow z_s; s^* \leftarrow s \) (keep track of the highest regret scenario)
9: end if
10: if \( z_s < Z^* \) then
11: \( Z^* \leftarrow z_s; p^* \leftarrow p \) (update the best solution found so far)
12: end if
13: if \( z_s = \max_j x_j^{(s)} \) then
14: \( u_j \leftarrow p_j - x_j^{(s)}, \) for all \( j = 1, \ldots, n \) (compute excess in price for each item)
15: \( j^* \leftarrow \arg \min_j \{ u_j \} \) (find an item with the smallest excess)
16: \( p_{j^*} \leftarrow p_{j^*} - u_{j^*} \) (decrease the price to make item \( j^* \) being sold)
17: else
18: \( u_j \leftarrow x_j^{(s)} - p_j, \) for all \( j = 1, \ldots, n \) (compute utilities)
19: \( j^* \leftarrow \arg \max_j \{ u_j \} \) (find an item with the highest utility)
20: \( k^* \leftarrow \arg \max_{j \neq j^*} \{ u_j \} \) (find the second-highest utility item)
21: \( p_{k^*} \leftarrow p_{k^*} - u_{k^*} + u_{j^*} - \epsilon \) (decrease the price to make item \( k^* \) being sold)
22: end if
23: Go to Step 3.
```

with the smallest optimal solution value. This method requires solving up to \( (n + 1)^K \) LPs in the worst case.

We now propose a much more efficient constructive method. This method is presented in pseudocode as Algorithm 1. The algorithm maintains a price vector and updates it iteratively, in order to decrease the regret generated by the current worst-case scenario. It terminates when any further changes to the prices would not decrease the regret in the worst-case scenario.

The algorithm starts from an initial price vector, consisting of the highest possible prices that allow for at least one item to be sold in some scenario. Then it successively updates the price vector by decreasing one selected price at a time. In each iteration of the algorithm, a price is selected and decreased. The choice of price is such that if the regret \( R(p, x^{(s)}) \) of a worst-case scenario \( x^{(s)} \) decreases, then at the same time the regret \( R(p, x^{(s')}) \) of a new worst-case scenario \( x^{(s')} \) is guaranteed to increase by the smallest possible amount.

**Theorem 2.** Algorithm 1 computes an optimal robust solution for min-max regret multidimensional monopolist pricing problem with discrete uncertainty.

**Proof.** We call the price \( p_j \) winning, if \( j \)th item is sold given price vector \( p =
From (5) we have that the maximum regret is a difference between the highest valuation and the winning price, in some scenario. We call such scenario a regret-maximizing scenario. The algorithm proceeds in iterations. At the beginning of each iteration, a regret-maximizing scenario is determined. Denote this scenario by $x^{(s)}$. First, let us assume that an item $j^*$ is the winning item in that scenario (we consider the case when there is no winning item later). The algorithm modifies the price vector $p$ in order to decrease the regret $R(p, x^{(s)})$. Observe that this regret can only be decreased by forcing an item with a higher price than $p_{j^*}$ to be winning.

Due to the choice of initial prices, an increase of $p_{j^*}$ would cause that no item is sold, and this always increases the regret $R(p, x^{(s)})$ to the maximum level $\max_j x_j^{(s)}$. Consequently, we chose to decrease a price $p_k$ of another item, until the utility $x_k^{(s)} - p_k$ becomes greater than the current winning $x_j^{(s)} - p_{j^*}$. We select such item $k$ that requires the smallest amount of change in $p_k$ in order to force the item $k$ to be winning. Due to this, we guarantee that, in case when this change makes (possibly another) scenario $s'$ regret-maximizing, the increase in $R(p, x^{(s')})$ is the minimal possible that simultaneously allows for reducing $R(p, x^{(s)})$. If after decreasing $p_{j^*}$ the new winning price $p_k$ is not higher than $p_{j^*}$, then $\max_j x_j^{(s)} - p_k \geq \max_j x_j^{(s)} - p_{j^*}$, and the maximum regret cannot be further decreased – the algorithm terminates. Otherwise we proceed to the next iteration, and repeat the process.

Note that in the above reasoning, the argument for not increasing $p_{j^*}$ was backed by the choice of initial prices, which no longer holds from the second iteration of the algorithm onwards. Observe, however, that from that point an increase of the current winning price $p_{j^*}$ brings us back to the previous state, with higher maximum regret in scenario $x^{(s)}$, considered in one of the previous iterations. We conclude that only decreasing the current prices may reduce the maximum regret.

It remains to consider the case when no item is winning in a regret-maximizing scenario $x^{(s)}$, that is, for all $j = 1, \ldots, n$, $p_j > x_j^{(s)}$. In such case the only way to reduce the regret is to force some item to be sold. We chose to decrease a price $p_k$ that requires the smallest amount of change in order to allow for an item sale. Due to this, if item $k$ happens to be winning in another scenario $s'$, then the regret $R(p, x^{(s')})$ is increased by the smallest possible amount that allows for regret $R(p, x^{(s')})$ to decrease. Such decrease of $R(p, x^{(s)})$ is not necessarily the largest possible, however the problem is now reduced to the previous case, when there exists a winning item, which would be checked in the next iteration.

It is easy to see that this algorithm runs in polynomial time. Consider a regret-maximizing scenario and the winning price $p_{j^*}$; without the loss of generality, let
\( j^* = 1 \). In a single iteration of the algorithm we select an item, say with index \( k = 2 \), to have its price decreased to the value \( p_2 \), such that \( p_2 > p_1 \). As long as this price is higher than the previous winning price, the new value of regret in that scenario is also decreased. For any scenario, this process may continue only up until the price \( p_n \) is set, so that \( p_n > p_{n-1} \), and the corresponding regret achieved so far is minimal. Any further price decrease of any item \( k \) would bring us to a winning price \( p_k' \) of the item \( k \), such that \( p_k' < p_n \). But this gives a greater regret than the one calculated for a price vector with the winning price \( p_n \). Consequently, the number of iterations of the algorithm is bounded by \( O(Kn) \).

Concluding our results, we obtain:

**Theorem 3.** *Min-max regret multidimensional monopolist pricing problem with interval or discrete uncertainty can be solved in polynomial time.*

Table 1 contains a summary of an experimental study performed using an implementation of Algorithm 1 on random input data. Scenarios were generated by drawing valuations as uniformly distributed vectors of integers, each between 10 and 10000. These results indicate that for a fixed number of scenarios, the number of iterations of the algorithm before termination decreases with an increasing number of items, but a single iteration becomes more time consuming, and the running time is longer. Generally, the more scenarios are in the uncertainty set, the greater the number of iterations is required. However, when the number of items is also very large, the iteration count drops so significantly, than the algorithm actually terminates faster. This interesting feature of the algorithm can be observed, for instance, when \( K = 2000 \) in our experimental results. Moreover, there were always significantly less iterations than scenarios, and the more items or scenarios, the lower the optimal maximum regret tends to be.

5. **Concluding Remarks and Further Work**

When designing revenue maximizing mechanisms we are often faced with the lack of meaningful, reliable probabilistic information. This motivates the need for the development of robust solution methods for multidimensional optimal mechanism design, that would complement the Bayesian approach. In this paper we showed how to solve two of the simplest robust multi-item pricing problems, which was previously known only for a single item. An interesting venue for further research is to consider more general pricing problems within the min-max regret framework, with such features as: arbitrary demands, variable item costs, non-monopoly, multiple buyers, time-dependent valuations. Moreover, it is interesting to consider more general sets of uncertain valuations, which would allow for more flexibility in modeling.
Table 1: Experimental results. Parameter $K$ is the number of scenarios, $n$ is the number of items, and $Z^*$ is the optimal solution value. The last two columns contain the number of iterations and the running time, respectively.

| $K$ | $n$ | $Z^*$ | iter. | time (sec.) | $K$ | $n$ | $Z^*$ | iter. | time (sec.) |
|-----|-----|-------|-------|-------------|-----|-----|-------|-------|-------------|
| 10  | 10  | 1239  | 5     | 0.05        | 1000| 10 | 5164  | 953   | 3.78        |
| 10  | 100 | 1047  | 5     | 0.07        | 1000| 100| 828   | 843   | 20.69       |
| 10  | 1000| 898   | 4     | 0.08        | 1000| 1000| 59    | 354   | 82.36       |
| 10  | 10000|2057 | 2     | 0.26        | 1000| 10000|24    | 85    | 259.36      |
| 100 | 10  | 4981  | 90    | 0.09        | 2000| 10 | 5311  | 1866  | 14.32       |
| 100 | 100 | 484   | 36    | 0.16        | 2000| 100 | 830   | 1749  | 84.07       |
| 100 | 1000| 172   | 18    | 0.61        | 2000| 1000| 75    | 1108  | 502.45      |
| 100 | 10000|230 | 10    | 4.35        | 2000| 10000|20    | 37    | 254.33      |
| 500 | 10  | 5445  | 471   | 1.06        | 4000| 10 | 4923  | 3608  | 55.79       |
| 500 | 100 | 826   | 390   | 4.88        | 4000| 100 | 1089  | 3460  | 330.08      |
| 500 | 1000| 52    | 94    | 11.2        | 4000| 1000| 109   | 2637  | 2336.81     |
| 500 | 10000|76  | 9     | 19.80       | 4000| 5000| 20    | 900   | 4892.24     |

References

[1] M. Mussa, S. Rosen, Monopoly and product quality, Journal of Economic Theory 18 (2) (1978) 301–317.
[2] M. Armstrong, Multiproduct nonlinear pricing, Econometrica 64 (1) (1996) 51–75.
[3] J. C. Rochet, P. Chone, Ironing, sweeping, and multidimensional screening, Econometrica 66 (4) (1998) 783–826.
[4] S. Basov, Hamiltonian approach to multi-dimensional screening, Journal of Mathematical Economics 36 (1) (2001) 77–94.
[5] A. M. Manelli, D. R. Vincent, Multidimensional mechanism design: Revenue maximization and the multiple-good monopoly, Journal of Economic Theory 137 (1) (2007) 153–185.
[6] Y. Cai, C. Daskalakis, Extreme-value theorems for optimal multidimensional pricing, in: Proceedings of the 52nd Annual Symposium on Foundations of Computer Science, IEEE, 2011, pp. 522–531.
[7] C. Daskalakis, A. Deckelbaum, C. Tzamos, Strong duality for a multiple-good monopolist, in: Proceedings of the 16th ACM Conference on Economics and Computation, ACM, 2015, pp. 449–450.
[8] P. Briest, S. Chawla, R. Kleinberg, S. M. Weinberg, Pricing lotteries, Journal of Economic Theory 156 (2015) 144–174.
[9] H. Aissi, C. Bazgan, D. Vanderpooten, Min–max and min–max regret versions of combinatorial optimization problems: A survey, European Journal of Operational Research 197 (2) (2009) 427–438.
[10] D. Bergemann, K. Schlag, Robust monopoly pricing, Journal of Economic Theory 146 (6) (2011) 2527–2543.
[11] D. Bergemann, K. H. Schlag, Pricing without priors, Journal of the European Economic Association 6 (2-3) (2008) 560–569.
[12] R. Caldentey, Y. Liu, I. Lobel, Intertemporal pricing under minimax regret, Available at SSRN 2357083.
[13] B. R. Handel, K. Misra, Robust new product pricing, Marketing Science 34 (6) (2015) 864–881.