BMI Paper

The Effects of Start Prices on the Performance of the Certainty Equivalent Pricing Policy

Faculty of Sciences

VU University Amsterdam

De Boelelaan 1081

1081 HV Amsterdam

Netherlands

Author:

R.D.R. van de Kamer

Supervisor:

A. Haensel

July 2011

Preface

This paper is written as part of the Business Mathematics and Informatics master program at the VU University in Amsterdam. The paper should emphasize the business related aspects of the study program, next to the mathematical and/or computer science aspects.

During the research, the student is supervised by member of the faculty who specializes in the subject chosen by the student. My thanks goes to A. Haensel, who was the supervisor for this paper.

Table of Contents

1. Introduction and Literature

Revenue management is no longer only applied in the aviation industry, but it is finding its way to other industries as well, for example the retail industry. A revenue management technique that is well applicable in the retail industry is dynamic pricing. This technique aims to maximize revenue by controlling the prices of a product over time. By experimenting with different prices, a firm can find the optimal selling price for their product. This experimenting with prices is of course costly, because every time the firm chooses to sell their product for a non-optimal price it loses potential revenue. There are several dynamic pricing techniques that can help to find the optimal selling price of a product, each having their own advantages and disadvantages. Some techniques only require a few time periods to give a semi-optimal price, while others do find the optimal price, but in a larger amount of time periods.
One of the simplest techniques is the certainty equivalent pricing policy, also known as myopic learning. This pricing policy is applicable to dynamic pricing problems with stochastic demand and no inventory considerations. The demand is modelled as a random variable, which depends linearly on the price of the product. The distribution of the demand is mostly assumed to be normal or log-normal. The price of the product can be changed at the beginning of each time period. During each time period the demand for the product is measured and with this information the price for the next time period is determined. Under the certainty equivalent pricing policy, the new price it set equal to the price that would be optimal if the data obtained up until this point was exactly correct. For this technique, the first two prices have to be decided otherwise. This technique is very fast, but as proven in [den Boer and Zwart, 2011], there is a positive probability that the newly obtained price does not converge to the optimal price. Because the certainty equivalent pricing policy is so fast, it is still applicable to products with a very short life span, in terms of how often the price can be changed.

The goal of this research is to find the optimal way of using this technique, or in other words, the optimal starting prices. These prices should minimise the expected potential revenue loss, also known as the regret. Besides the optimal starting prices, this paper also discusses the effects the different starting prices might have on the probability of converging to the optimal price and the convergence speed. These results were obtained numerically, by simulating the certainty equivalent pricing policy for different price sets.

Other methods to solve this dynamic pricing problem are given by [Broder and Rusmevichienong, 2009], [Carvalho and Puterman, 2005], [Lobo and Boyd, 2003] and [den Boer and Zwart, 2011]. [Carvalho and Puterman, 2005] propose one-step ahead pricing to maximize the cumulative expected revenue. This pricing policy aims to maximize the sum of the revenues over the next two periods, instead of over only one period. For this they use a Taylor expansion of the expected revenue for the following period. Their paper shows that the one-step ahead pricing policy performs better than myopic learning.
A solution for a logit demand model, proposed by [Broder and Rusmevichienong, 2009], is the MLE cycle. In their paper they show that when only the demand function is unknown, the optimal strategy is a greedy one. The MLE cycle is also extended to a setting where the market share is unknown as well. In this case the optimal strategy is to differentiate between exploration and exploitation. In both cases, the MLE cycle performs well.

An extension that is applicable to certainty equivalent pricing, the MLE cycle as well as one-step ahead pricing is controlled variance pricing by [den Boer and Zwart, 2011]. They propose to add a taboo interval around the newly chosen prices at each interval, to make sure the price does not converge too fast. They show that this method will eventually provide the correct value of the optimal price.
[Lobo and Boyd, 2003] evaluate the performance of certainty equivalent pricing policies and instead propose a dynamic program solution, which can be extended to include a stochastic demand that changes over time and multiple products. Solving this dynamic program is intractable for a large number of time intervals, so they give an approximate solution by solving a convex optimization problem.

This paper is organised as follows. In Section 2 the model for certainty equivalent pricing policy will be explained, as well as the simulation techniques that were used. Section 3 contains the results of this research, while Section 4 concludes.

2. Methods

In this section the model for the certainty equivalent pricing policy will be described, as well as the simulation methods that were used to examine the behaviour of this model.

2.1 Model

In this setting we assume that the firm is a price setter and produces only a single product. There are no capacity constraints on the amount of products that can be produced and the marginal costs of a product are zero. The selling price p_t for this product is determined at the beginning of each time interval t. This price lies between p_{min} and p_{max}, which are the minimum and maximum price that the firm wants to sell this product for. The demand D_t is a linear function of the price p_t, with an extra random component ε_t.

(1)

The random components ε_t are independent identically distributed with mean zero and variance σ^2. The parameters a = (a_0, a_1) are the ones that have to be estimated and are thus unknown to the firm. It is assumed that α_0 is a positive number and α_1 a negative number, such that the demand decreases with an increase in the price. Before each new time interval these parameters are estimated using a least-squares linear regression on the data collected up until this time interval. The least-squares estimates $\hat{a}$ = ($\hat{a}_{0t}$, $\hat{a}_{1t}$) minimize the mean square error:

(2)

The revenue R_t that is obtained during a time interval t is equal to $p_t * D_t$. The revenue is also denoted as R(p, a) to indicate the dependence on the price and the demand. Using the least-squares estimates for the regression parameters, the price that maximizes the expected revenue for the next period is equal to:

(3)

If p_{new} does not lie between p_{min} and p_{max}, then either p_{min} or p_{max} is chosen as p_{new}, depending on which one gives the higher expected revenue under the current parameters $\hat{a}$ = ($\hat{a}_{0t}$, $\hat{a}_{1t}$). This process is repeated at each time interval until p_{new} converges.

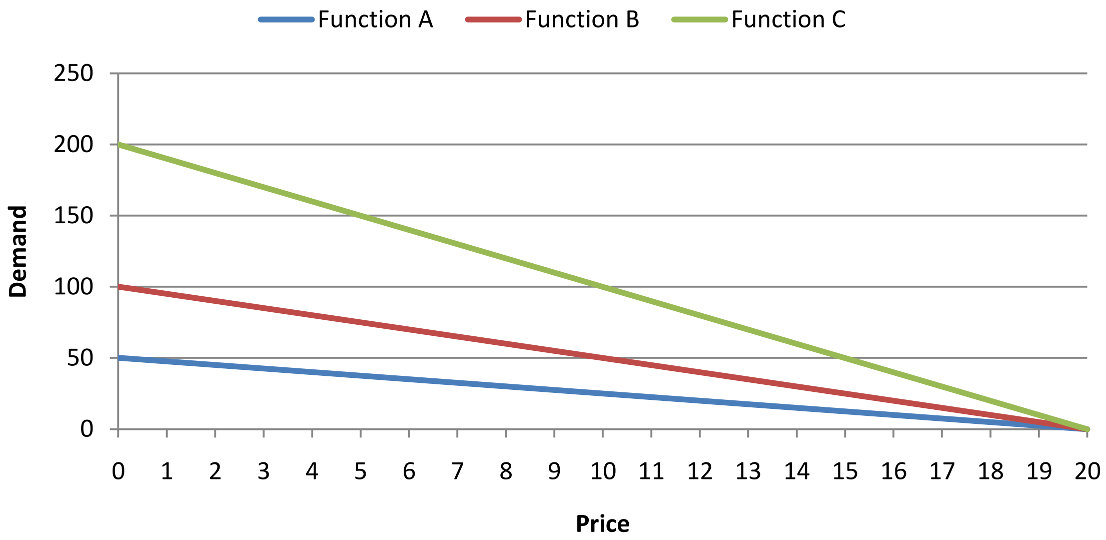
2.2 Simulation and evaluation

To examine the behaviour of the certainty equivalent pricing policy under different starting prices, several simulations were done. For each combination of the available start prices, the pricing policy was executed a B number of times, in this case 100. Over these 100 measurement the average and standard deviations were taken and used to evaluate the performance of the pricing policy for different settings. As discussed in the next section, simulations were done for different demand functions and levels of noise in the demand, as well as one case were the demand is continuous and two cases were the demand is discrete. The different measurements that are needed to evaluate the performance of this pricing policy are discussed in section 2.2.2.

2.2.1 Simulation parameters

Simulations were done in three different settings, one were demand was continuous and two were demand was discreet. In all these three settings, simulations were done for different combinations of demand functions and levels of noise in the demand. The noise in the demand was normally distributed with a sigma of either 0.5, 1, 2 or 5. With each level of noise, three different demand functions were simulated, giving a total number of twelve simulations per setting. Figure 1 shows the three demand functions that were used. Each demand function has the same price range, but a different slope. The slopes are -2.5, -5 and -10, respectively. These demand functions were chosen such that in all cases the optimal price is 10. This corresponds to a demand of 25, 50 or 100, respectively. In all simulations, the start prices range from 1 to 19, with a 0.5 interval.

Figure 1: The three different demand functions.



In the first simulation setting, the demand is assumed to be continuous, as show in Figure 1. For the simulation this means that the demand is used as it was calculated using Equation (1), except for when the demand is negative. In this case the demand is set to zero. In the second and third setting, demand is discrete. Here it is still calculated using Equation (1), but then rounded to the nearest integer. Again, if the demand is negative it is set to zero.

The third setting differs from the second in how it chooses the price for the next iteration. It tries to take more advantage of the knowledge that demand is discrete, by only choosing prices for which the demand is at that time assumed to be discrete. In the first and second setting, the new price is obtained via Equation (3). In the third setting this price is calculated as well, but is then changed to one that is almost equal, but assumed to yield a discrete demand. This is done by first calculating the demand that matches the price given by Equation (3), using the current estimates $\hat{a} = (\hat{a}_{0t}, \hat{a}_{1t})$:

(4)

This demand is then rounded up as well as down, after which the price that corresponds to those demands is calculated. For both of these prices the revenue is calculated and the price that gives the higher revenue is chosen.

2.2.2 Evaluation

To determine the performance of the pricing policy under different start prices, the three main aspects that are evaluated are regret, convergence speed and the probability of finding the optimal solution. For each of these aspects, several variables are calculated and represented in different ways. For each combination of start prices, the average over the 100 runs is taken en represented in a 3-dimensional graph. For some variables, a separate graph shows the standard deviation over these 100 runs. To find the best starting prices, the price combination that gives the best performance per variable is determined, together with the standard deviation were it is available. To evaluate the performance of the pricing policy under different settings, the average over all combinations of start prices is calculated as well.

The first aspect of the performance is regret. The regret is defined as the sum of the differences between the revenue that can be obtained selling the product at the optimal price and selling the product for the current price, for each time interval t:

, (5)

where $a = (a_0, a_1)$ are the parameters of the real demand function. During the simulation, two regret variables are calculated, namely the average regret over all iterations of the simulation and the average regret after convergence. The first variable is obtained by dividing the total regret over all iterations by the number of iterations. The second variable is obtained by calculating the regret in the last iteration and multiplying that with a fixed number, in this case 100. This indicates the revenue loss over 100 time intervals, if the found price is to be used as the optimal price. A pricing policy is considered to perform better if the regret is low. Related to the regret after convergence are the absolute difference between the found and the optimal price, as well as the average, converged price. The regret variables and absolute difference between the found price and the optimal price were calculated for each combination of start prices, as well as a total average. The average, converged price is only calculated as the average price over all combinations of start prices.

The convergence speed is the number of iterations needed until the price converges. During these simulations, the price was considered to be converged when the difference between the current price and the previous price was smaller than one cent. To make sure the price did not converge to a local optimum, the minimum amount of iterations was set to 50. If the price converged before reaching those 50 iterations and it did not change more than one cent in the following iterations, the number of iterations is equal to the number needed until the last convergence. Otherwise, the number of iterations is equal to the first point of convergence after those 50 iterations. The number of iterations until convergence was calculated for each combination of start prices and also an overall average was given.
The number of iterations used to calculate the average regret is the total number of iterations that were done, as the regret is summed up over all those iterations. This means that the number of iterations used for this calculation is always greater of equal to 50.

The last aspect is the probability of finding the optimal solution. At the end of each execution of the pricing policy, the last found price is compared to the optimal price. The number of executions for which they are equal is counted and then divided by B to give the probability of finding the optimal solution. Because prices are expressed with an accuracy no larger than cents, the same was done for

found prices that were first rounded to the nearest cent and then compared.
Another interesting variable is how close the found demand line is to the actual demand line on the point of convergence. At this point, the difference between the two lines is calculated as:

(6)

This difference is calculated for each combination of start prices and represented in a graph. The number of times the difference is exactly zero was also obtained. In the settings were demand is discrete, this last variable is also calculated using the rounded values.

3. Results

This sections discussed the results of the various simulations. It is divided into three parts, corresponding to the three settings for which the simulations were done. The first section describes the results of the setting with continuous demand, the second section that for discreet demand and finally the third section holds the results for the setting with discreet demand and the alternative pricing method.

3.1 Continuous demand

In this setting the demand function is continuous. Twelve simulations were done for each combination of the different demand functions and levels of noise. First the effects of these differences will be discussed, followed by the effects of the different start prices.

3.1.1 The effects of the slope of the demand function and levels of noise

In this section the effects of the different demand functions and different levels of noise will be discussed. There are three different demand functions, A, B and C, where C is the steepest. The level of noise is normally distributed, with sigma either 0.5, 1, 2 or 5. The performance of the pricing policy under these different parameter settings is evaluated by looking at the overall average values of the performance variables.

Table 1 shows the overall average regret per iteration. As can be seen from this table, the pricing policy performs worse when the level of noise is higher. Where the regret was only 4.08 for demand function A and a noise level with a sigma of 0.5, it goes up to 44.48 for the same demand function but with the noise distributed with a sigma of 5. For a steeper demand function like C, the difference between a high and a low level of noise is somewhat smaller. When the noise is distributed with sigma equal to 0.5, the regret is higher, up to 11.65, than for a flatter demand slope, but sigma equals 5, the regret is lower, down to 42.85. This difference in behaviour can also be seen in the graphs of the regret per iterations set out against the start prices. Figure 2 shows the regret per iteration for different start prices for the steepest demand function and the lowest level of noise. This graph looks very smooth and the highest regret per iteration is obtained with start prices furthest away from the optimal price. Figure 3 shows the same for the flattest demand function and the highest level of noise. Here the highest regret per iteration is obtained at those start prices that are close together.

The overall average number of iterations needed until convergence, as shown in Table 2, is also affected by the slope of the demand function as well as the level of noise. For the flattest demand curve and lowest level of noise, the average number of iterations needed until convergence is 25.62, while it is 35.30 when the level of noise is distributed with a sigma of 5. For the steepest demand curve the difference only gets bigger. When the level of noise is very low, it only takes on average 9.27 iterations until convergence is reached, while this number still goes up to 35.87 for the highest level of noise.

Table 3 shows the overall average converged price, while Table 4 shows the average, absolute difference between the converged price and the optimal price. The overall average regret after convergence is shown in Table 5. From Table 3 it can be seen that on average, the pricing policy converges to a price that is higher than the optimal price and it only gets higher when the level of noise in the demand increases. On average, the converged price is closest to the optimal price for a steep demand function, like C, and a low level of noise. This gives an average converged price of 10.004. For the flattest demand slope and a level of noise distributed with a sigma equal to 5, this price goes up to 11.33. The average, absolute difference between the converged price and the optimal price follows the same behaviour. The difference is only 0.08 for the steepest demand curve and lowest level of noise, while it goes up to 2.90 for the flattest demand curve and highest level of noise. Table 5 shows the overall, average regret after convergence. This is the regret over 100 time periods if the converged price is used in all those 100 time periods. Naturally, it follows the same pattern as the average converged price, but here the regret ranges from 23.14 to 4408.20. The absolute difference between the converged price and the optimal price is always higher than the difference between the average converged price and the optimal price, meaning that there are still cases when the converged price is lower than the optimal price.

Table 1: Average regret per iteration, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	4,08	8,14	18,44	44,48
Demand B	6,18	8,16	16,32	47,04
Demand C	11,65	12,37	16,35	42,85

Table 2: Average number of iteration until convergence, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	25,62	33,99	37,83	35,30
Demand B	15,71	25,64	34,03	37,90
Demand C	9,27	15,68	25,65	35,87

Table 3: Average converged price, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	10,06	10,22	10,58	11,33
Demand B	10,02	10,06	10,22	10,73
Demand C	10,004	10,02	10,07	10,32

Table 4: Average absolute difference between the converged price and the optimal price, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	0,35	0,74	1,45	2,90
Demand B	0,17	0,35	0,74	1,76
Demand C	0,08	0,17	0,35	0,93

Table 5: Average regret after convergence, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	128,89	557,91	1647,80	4408,20
Demand B	49,24	257,27	1117,80	4368,40
Demand C	23,14	98,82	518,12	3316,90

Figure 2: Regret per iteration for different start prices, for demand function C and sigma 0.5. Continuous demand.

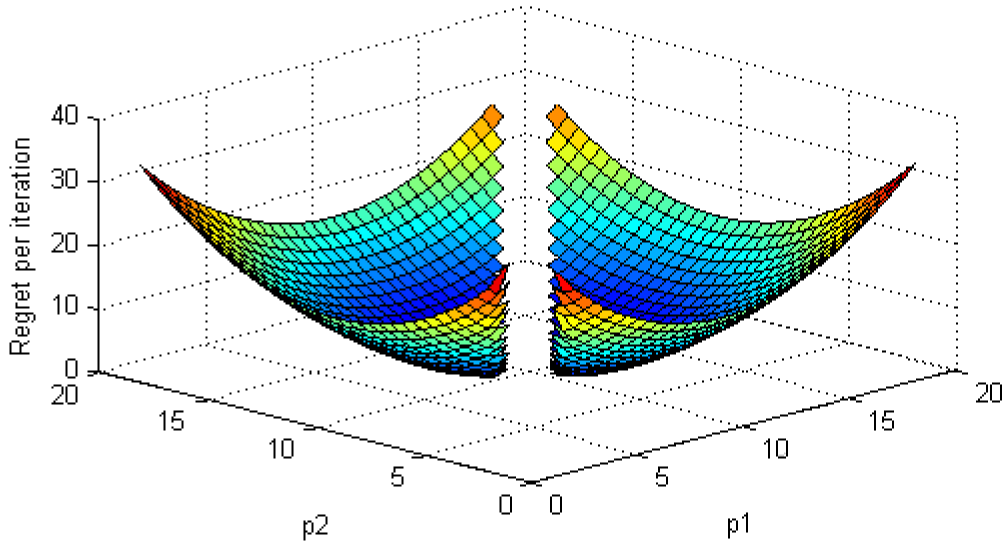
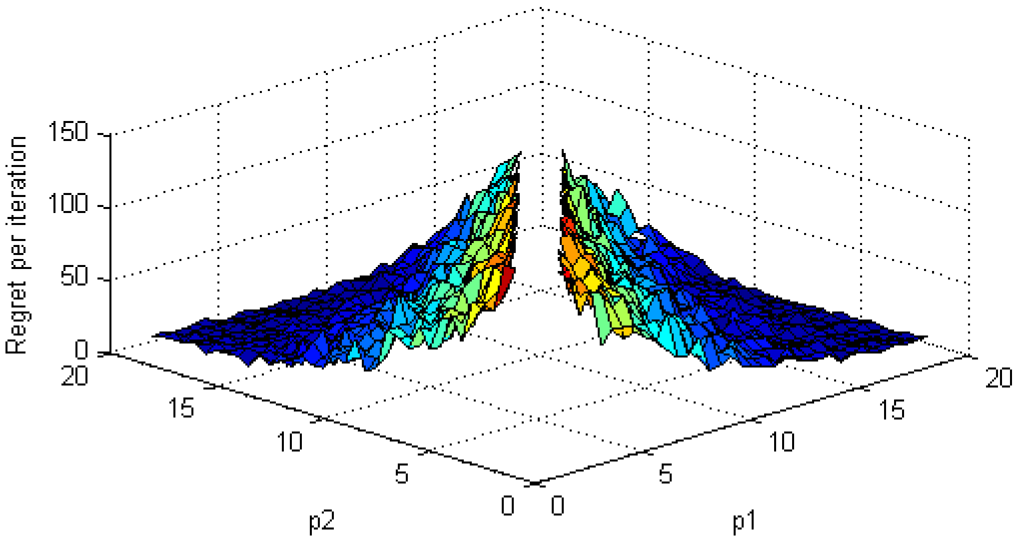


Figure 3: Regret per iteration for different start prices, for demand function A and sigma 5. Continuous demand.



3.1.2 The effect of different starting prices

To obtain the best start prices for the certainty equivalent pricing policy, the price combinations that gave the best performance measurements were recorded. The start prices range from 1 to 19, with a 0.5 interval.

The most important aspect of the performance is the regret. As this pricing policy converts rather fast, the most important variable is the regret after convergence. Table 6 to Table 8 show the minimum regret that was obtained, at which price combination they were obtained and what the standard deviation of the regret after convergence at that point is. The minimum values of the regret after convergence as shown in Table 6 are considerably lower than the average values shown in Table 5. As can be seen in Table 7, the best combinations of start prices differ a somewhat between the different simulation parameters, without a very clear pattern. It is clear that one price should be very high, but the choice for the second price is debatable. That the choice for the second price is unclear, is nicely illustrated in the graphs that show the regret after convergence set out against the price. Figure 4 shows this graph for the steepest demand line and the lowest level of noise. On the diagonal between the start prices the regret after convergence is very high, but the rest of the graph shows an almost flat surface where the regret is low. Only when the noise in the demand increases, as is shown in Figure 5, does the graph take more shape. Here a good choice of start prices is more effective.
Similar results are found when considering the difference between the converged and the optimal price. Table 9 shows the minimum difference between the converged and the optimal price, Table 10 the corresponding prices and Table 11 the standard deviations. In some cases the converged price comes very close to the optimal price. The prices at which this happens, given in Table 10, are similar to those that give the minimum regret after convergence, as seen in table Table 7.

Another variable that is highly influenced by the choice of start prices is the average number of iterations needed until convergence. Table 12 shows the minimum number of iterations needed for each combination of demand function and level of noise. For those combinations where the level of noise is still rather low, the minimum number of iterations needed lies around 5. Only when the level of noise is quite high, like in the case of demand function A and the noise distributed with a sigma equal to 5, does the number of iteration until convergence increase more drastically. In this case it goes up to 9. That the number of iterations needed really depends on the start prices can be seen from Figure 6. This graph shows the number of iterations needed until convergence, set out against the start prices, for demand function B and a sigma equal to 1. For all other combinations of demand functions and levels of noise the graphs have a similar shape, only they differ in scale. It can be seen from these graphs that the minimum number of iterations needed is obtained for a combination of start prices that have one very high price and once price that is slightly below the optimal price. Figure 7 shows the standard deviations of these minimum values. These standard deviations are very low for those prices where the number of iterations needed is very high and in this case they even reach zero for those places where the number of iterations needed is very low.

The final measurements are about how often the pricing policy converges to the optimal point. In the case of continuous demand the converged price was never exactly equal to the optimal price. However, when the converged price is rounded to the nearest cent and then compared to the optimal price, the values are sometime equal. Table 13 shows the maximum percentage of equal values that was obtained. For demand function C and a sigma equal to 1, the maximum percentage that was obtained is 13%. Figure 8 shows the percentages for all combinations of start prices for this parameter setting. This graph shows the highest percentages are obtained in those regions where one of the start prices is relatively high and the other price is around the optimal price or lower. The difference between the found demand line and the actual demand line, on the point of convergence, is also never exactly zero. However, as can be seen from Figure 9, it is very close to zero.

Table 6: Minimum regret after convergence, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	2,44	9,56	36,27	213,49
Demand B	1,15	4,27	17,59	91,98
Demand C	0,55	2,05	9,87	55,76

Table 7: Start prices corresponding to the minimum regret after convergence, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	(18.5, 8)	(19, 8.5)	(19, 7)	(19, 6)
Demand B	(19, 7.5)	(18.5, 7.5)	(19, 8)	(19, 2)
Demand C	(19, 11)	(18.5, 6)	(19, 7)	(19, 9)

Table 8: Standard deviations of the minimum regret after convergence, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	3,38	14,48	54,92	514,21
Demand B	1,42	6,45	25,79	143,57
Demand C	0,78	3,21	11,36	64,05

Figure 4: Regret after convergence, for demand function C, sigma 0.5 and continuous demand

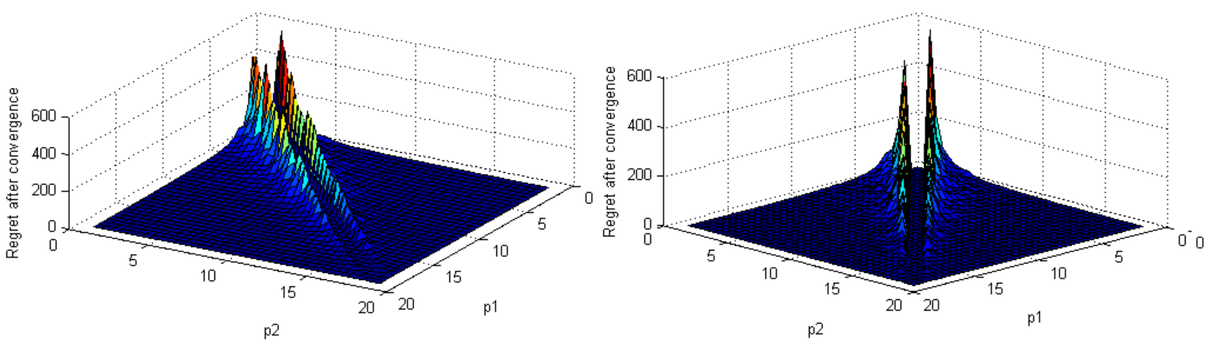


Figure 5: Regret after convergence, for demand function A, sigma 5 and continuous demand.

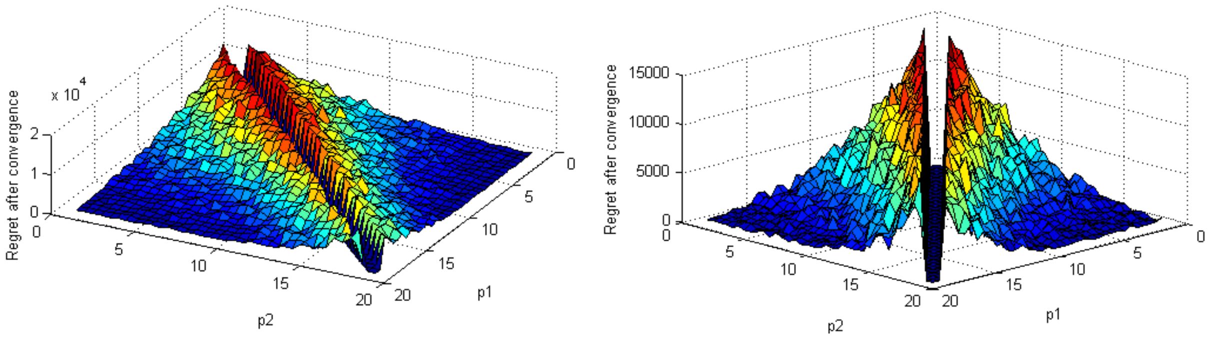


Table 9: Minimum difference between the converged and the optimal price, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	0,08	0,15	0,30	0,69
Demand B	0,04	0,07	0,15	0,36
Demand C	0,02	0,03	0,08	0,19

Table 10: Start prices corresponding to the minimum difference between the converged and the optimal price, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	(18.5, 4.5)	(19, 8.5)	(19, 7)	(19, 6)
Demand B	(19, 9)	(18.5, 7.5)	(19, 8)	(19, 6)
Demand C	(19, 11)	(18.5, 6)	(18.5, 6.5)	(18.5, 4)

Table 11: Standard deviations of the minimum difference between the converged and the optimal price, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	0,07	0,12	0,24	0,62
Demand B	0,03	0,06	0,12	0,27
Demand C	0,01	0,03	0,07	0,15

Table 12: Minimum number of iterations needed until convergence, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	5,00	5,00	5,02	9,00
Demand B	4,87	5,00	5,00	5,23
Demand C	4,75	4,81	5,00	5,00

Figure 6: Average number of iterations needed until convergence, for different starting prices. Demand function B, sigma equal to 1 and continuous demand.

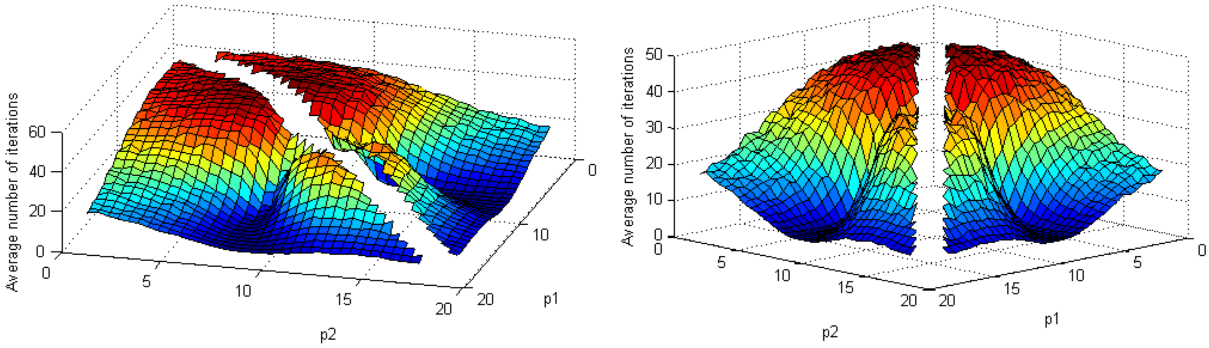


Figure 7: Standard deviations of the average number of iterations needed until convergence, for different starting prices. Demand function B, sigma equal to 1 and continuous demand.

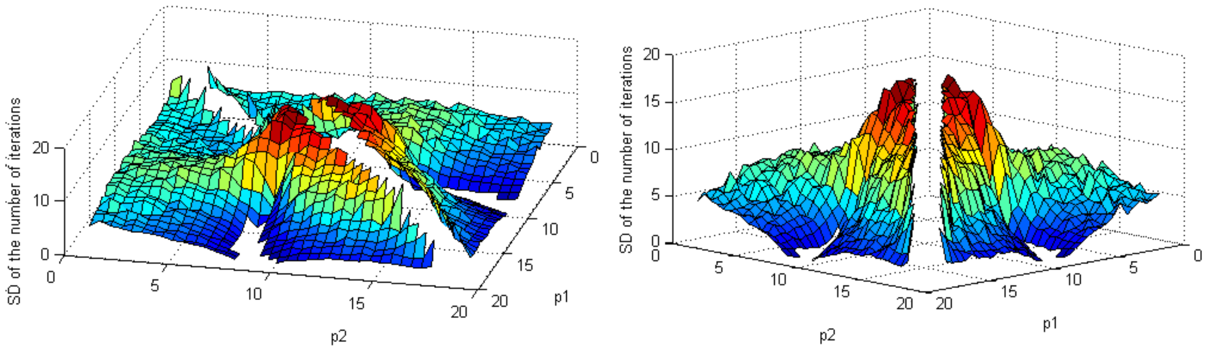


Table 13: Maximum percentage of rounded, converged prices equal to the optimal price, for continuous demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	8%	7%	5%	3%
Demand B	13%	8%	6%	4%
Demand C	22%	13%	9%	6%

Figure 8: Percentage of rounded, converged prices equal to the optimal price, for different starting prices. Demand function C, sigma equal to 1 and continuous demand.

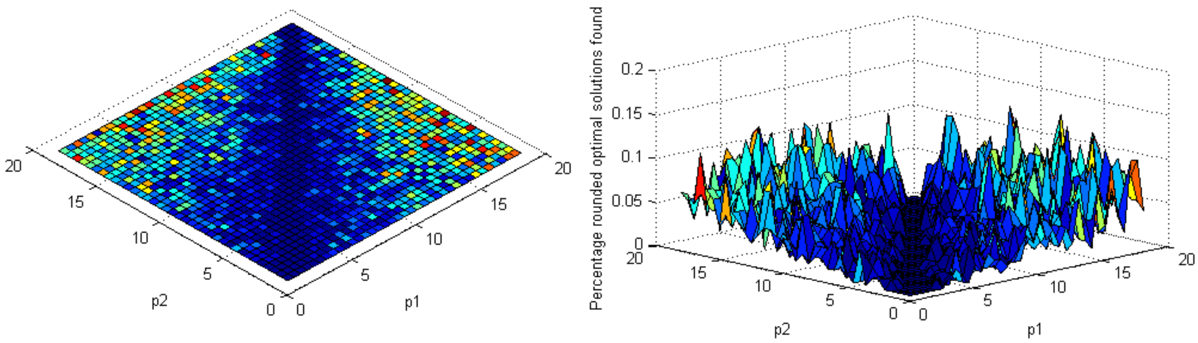
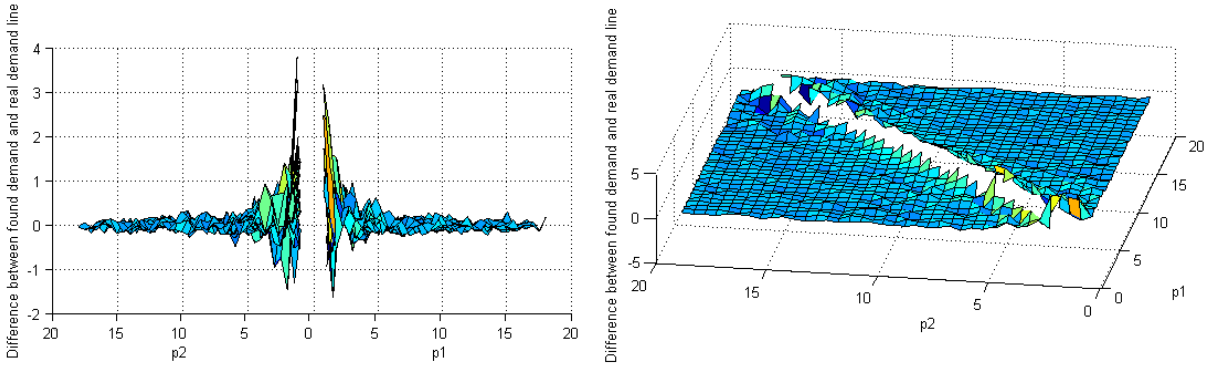


Figure 9: Difference between found demand line and actual demand line, for different starting prices. Demand function C, sigma equal to 1 and continuous demand.



3.2 Discrete demand

For most types of products, the demand will be discrete instead of continuous. This is also the case in the second and third setting discussed in this paper. The next section will describe the results for the second setting, were demand is discrete, but prices are still chosen the same way as in the continuous case. This section will only show the effects of the start prices, not the effects of the different demand functions and levels of noise. After that, the effects of the start prices on the performance of the pricing policy under the third setting will be discussed. In this setting only those prices that are assumed to give a discreet demand are used. The effects of the different demand functions and levels of noise are similar to those in the case of the continuous demand.

3.2.1 Continuous pricing method

In this setting the demand is discrete, but new prices are chosen using a continuous demand function. The goal is to find the best start prices, so for each measurement, the price combination that gives the best performance is obtained. The measurements are discussed in the same order as for the continuous case.

First up is the regret after convergence. Figure 10 shows the regret after convergence for demand function C and the noise in the demand distributed with a sigma equal to 0.5. In the case of discrete demand, this is calculated with the demands rounded to the nearest integer. Unfortunately, this sometimes leads to an negative regret after convergence, which makes it hard to compare which combinations of start prices actually give the best results. Table 14 shows the minimum, positive regret after convergence, Table 15 gives the prices that correspond to these minimum regrets and finally, Table 16 shows the standard deviations at these points. While the minimum, positive regrets are very low, their standard deviations are extremely high. This means that positive as well as negative results occur for the same price combination.

Something that should give a more reliable measure is the absolute difference between the converged and the optimal price. Table 17 shows the minimum difference between the converged and the optimal price. For demand function C and a sigma of 0.5, the average difference between the converged and the optimal price is as low as 0.01. For the flattest demand function, A, and a sigma of 5, this difference goes up to 0.57. These values are very similar to those of the continuous case, as shown in Table 9, and sometimes even a little lower. The same goes for the standard deviations, shown in Table 19, and the price combinations that give the lowest difference, as shown in Table 18. This does mean, however, that there is no clear price combination that gives the best result. The highest price is always between 18 and 19, but the lowest price ranges from 6.5 to 11.5 and once even goes as low as 1.

The minimum number of iterations needed until convergence is shown in Table 20. As in the continuous case, the minimum number of iteration needed lies around 5. Only in the case of the flat demand function A, with the highest level of noise does the value really differ. In that case it goes up to 7.99. Figure 11 shows the minimum number of iterations needed for each combination of start prices, for demand function B and a sigma equal to 1. This graph has the same shape as in the continuous setting. This also goes for the standard deviations, as shown in Figure 12. The price combinations that give the lowest number of iterations needed until convergence consist again of one very high price and one price that is around the optimal price or lower. These prices also give the lowest standard deviation.

Table 21 shows the percentage of simulations that resulted in the exact optimal price. In this setting, the exact optimal price was reached in a maximum of 5% of the cases, for the lowest level of noise. As the level of noise increases, the exact optimal price is obtained less often. When the converged price is first rounded to the nearest cent and then compared to the optimal price, the percentage of simulations that resulted in the optimal price is a lot higher. As can be seen from Table 22, the highest percentage was reached for demand function C, with the lowest level of noise. Here the percentage of rounded, converged prices equal to the optimal price is as high as 65%. When the level of noise increases, the percentage of simulations that resulted in the rounded, converged price being equal to the optimal price, goes down. Figure 13 shows this percentage for demand function C and the noise in the demand distributed with sigma equal to 1. The maximum percentage reached in this setting is 27%. As can be seen from this graph, the highest percentages are reached for price combinations where one price lies around 8, while the other price is 14 or higher.

Finding the optimal price does not guarantee that the found demand line is also equal to the actual demand line. Table 23 shows that the difference between the found demand line and the actual, continuous demand line, at the point of convergence, is only equal to zero in 1% of the cases and only for demand function C, with the lowest level of noise. Figure 14 shows the difference between the found and the actual demand line. This difference is indeed not always zero, but for most of the price combinations it is very close to zero. Because the demand is taking to be discrete, the found demand line should also be compared to the actual, discrete demand line. The maximum percentage of simulations where this difference is equal to zero is given in Table 24. For the steepest demand line and the lowest level of noise, this percentage is as high as 94%. For the flattest demand line and the highest level of noise, the difference between the two rounded demand lines is equal to zero in 16% of the simulations. Figure 15 shows the percentage of the simulation for which the difference between the found demand line and the actual, discrete demand line, at the point of convergence, is equal to zero, for demand function C and a sigma equal to 1. The price combinations for which this percentage is highest seem to have one price slightly above the optimal price, around 12.5, and one price slightly below the optimal price, around 7.5.

Table 14: Minimum, positive regret after convergence, for discrete demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	0,03	0,29	1,31	196,16
Demand B	0,09	0,07	0,27	47,25
Demand C	0,03	0,02	0,26	10,23

Table 15: Prices corresponding to the minimum, positive regret after convergence, for discrete demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	(14.5, 3)	(17.5, 5)	(17, 10)	(19, 1)
Demand B	(16.5, 1)	(18.5, 14.5)	(18, 9)	(19, 9.5)
Demand C	(19, 17)	(13, 9.5)	(16.5, 7)	(19, 9.5)

Table 16: Standard deviations for the minimum, positive regret after convergence, for discrete demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	282,98	288,28	267,73	456,81
Demand B	310,80	291,83	279,97	431,76
Demand C	211,16	302,43	273,07	292,27

Figure 10: Regret after convergence, for demand function C, sigma 0.5 and discrete demand

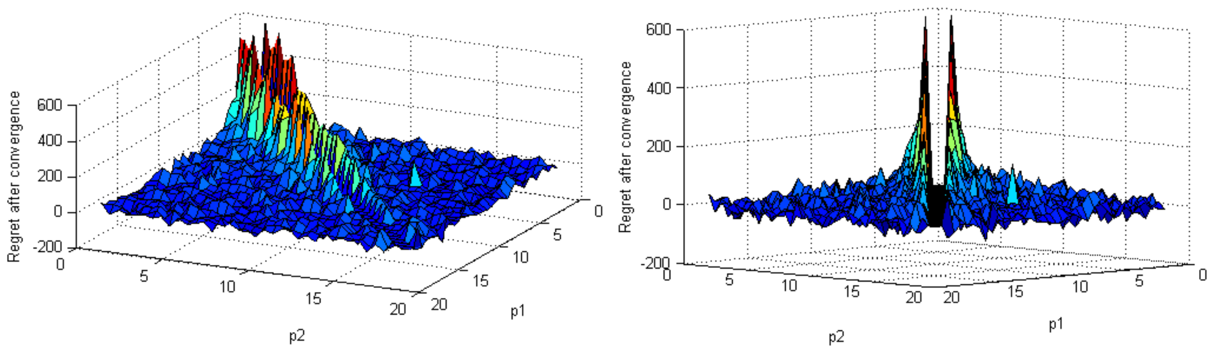


Table 17: Minimum difference between the converged and the optimal price, for discrete demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	0,08	0,17	0,28	0,57
Demand B	0,03	0,08	0,16	0,36
Demand C	0,01	0,04	0,08	0,19

Table 18: Prices corresponding to the minimum difference between the converged and the optimal price, for discrete demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	(18, 7)	(18.5, 7,5)	(19, 10)	(19, 9)
Demand B	(19, 10.5)	(19, 6.5)	(18.5, 8)	(19, 11.5)
Demand C	(19, 9.5)	(19, 10.5)	(18.5, 8.5)	(19, 1)

Table 19: Standard deviations of the minimum difference between the converged and the optimal price, for discrete demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	0,10	0,15	0,21	0,49
Demand B	0,05	0,06	0,13	0,29
Demand C	0,02	0,04	0,06	0,15

Table 20: Minimum number of iterations needed until convergence, for discrete demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	5,00	5,00	5,06	7,99
Demand B	4,62	4,98	5,00	5,12
Demand C	4,31	4,62	4,97	5,00

Figure 11: Average number of iterations needed until convergence, for different starting prices. Demand function B, sigma equal to 1 and discrete demand.

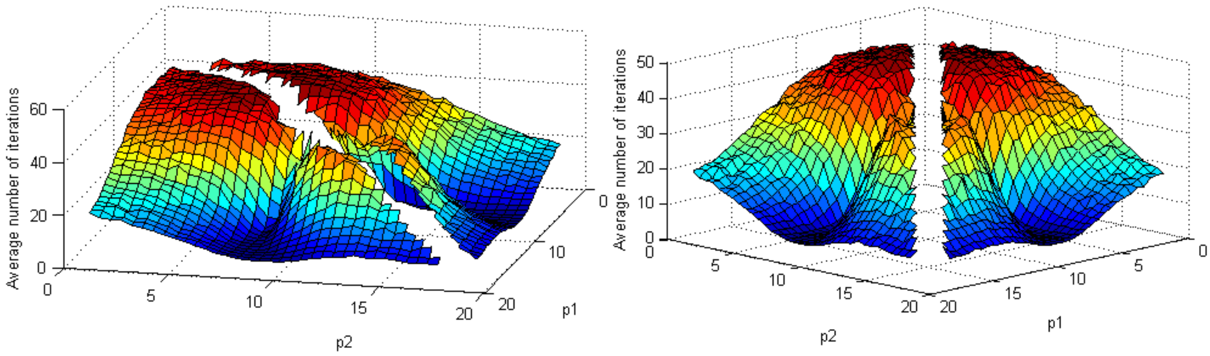


Figure 12: Standard deviations of the average number of iterations needed until convergence, for different starting prices. Demand function B, sigma equal to 1 and discrete demand.

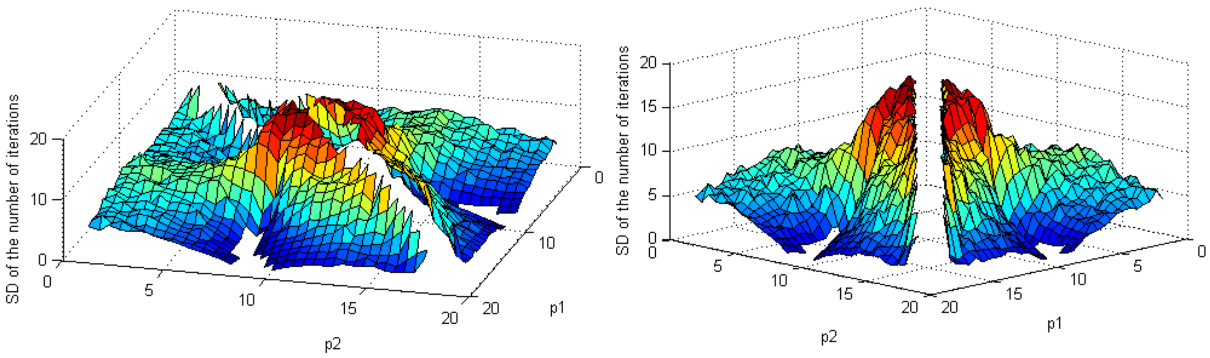


Table 21: Maximum percentage of converged prices equal to the optimal price, for discrete demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	4%	1%	1%	0%
Demand B	5%	2%	0%	1%
Demand C	4%	1%	0%	0%

Table 22: Maximum percentage of rounded, converged prices equal to the optimal price, for discrete demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	51%	22%	10%	4%
Demand B	54%	21%	8%	4%
Demand C	65%	27%	15%	5%

Figure 13: Percentage of rounded, converged prices equal to the optimal price, for different starting prices. Demand function C, sigma equal to 1 and discrete demand.

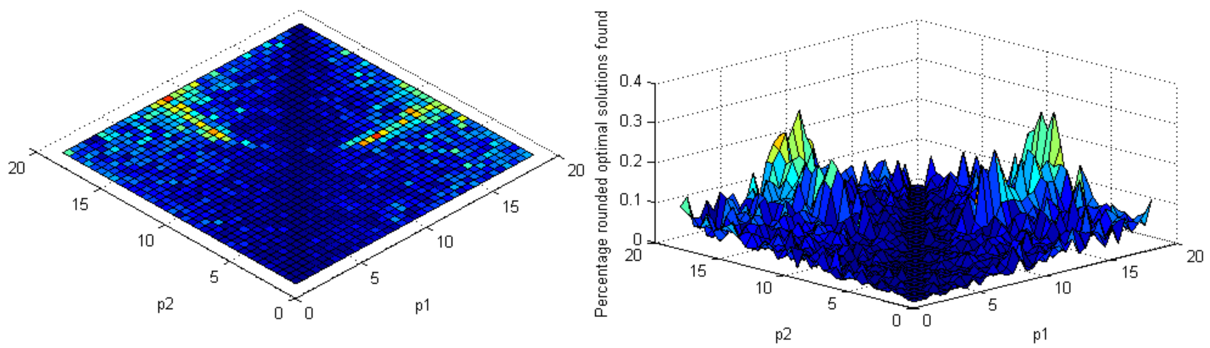


Table 23: Maximum percentage of simulations where the difference between the found and the actual, continuous demand line is equal to zero, for discrete demand.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	0%	0%	0%	0%
Demand B	0%	0%	0%	0%
Demand C	1%	0%	0%	0%

Table 24: Maximum percentage of simulations where the difference between the rounded, found demand line and the actual, discrete demand line is equal to zero.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	86%	54%	32%	16%
Demand B	90%	62%	33%	18%
Demand C	94%	67%	39%	18%

Figure 14: Difference between found demand line and actual demand line, for different starting prices. Demand function C, sigma equal to 1 and discrete demand.

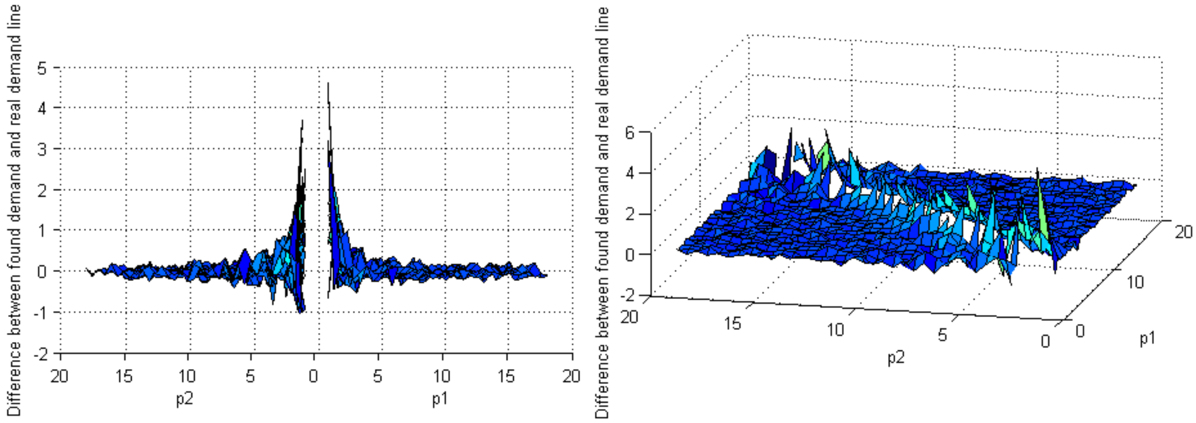
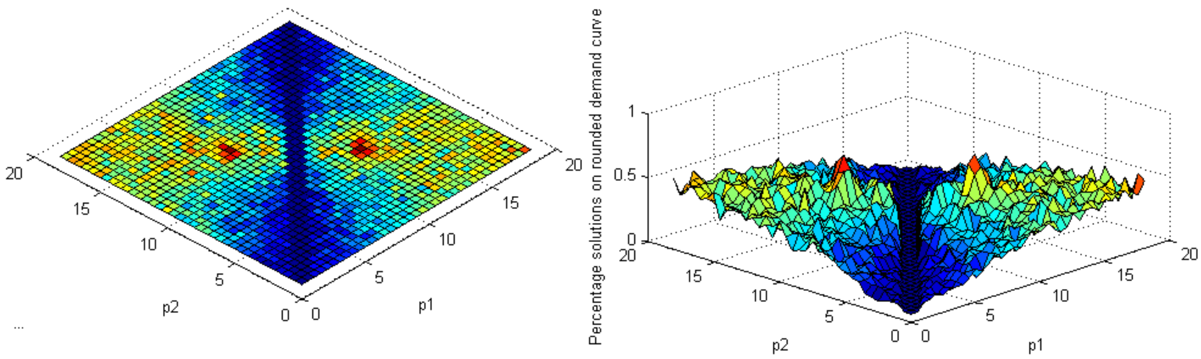


Figure 15: Percentage of rounded, found demand lines equal to the actual, discrete demand line, for different starting prices. Demand function C, sigma equal to 1 and discrete demand.



3.2.2 Alternative pricing method

In this setting an alternative pricing method is used. Instead of accepting all prices in the available price range, only those prices that are assumed to yield a discrete demand are used.

As with the normal pricing method, the regret after convergence is calculated in such a way that it sometimes becomes negative. This is best seen in Figure 16, which shows the regret after convergence for demand function C and a sigma equal to 0.5. Table 25 shows the minimum, positive regret after convergence, with values very close to zero. The standard deviations given in Table 26, however, are extremely high. On average they lie around 300, which means that these results are very unreliable.

Table 27 shows the minimum difference between the converged price and the optimal price. Their standard deviations are given in Table 29, while Table 28 shows the prices at which the minimum values were obtained. With values between 0.03 and 0.70, the minimum difference between the converged and optimal price in this setting is very similar to those in the previous setting and thus also similar to those in the continuous setting. The biggest difference is that the price combination that gives the minimum value is even a little harder to pin down. There is more fluctuation in the higher price, as it now ranges from 17.5 to 19 and the lower price still has a long range, now from 5.5 up to 10.

The biggest difference between this setting and the others is in the number of iterations needed until convergence. Table 30 shows the minimum number of iterations needed for different parameter settings. While the average number of iterations needed was around 5 for the previous settings, the average now lies around 30. The least number of iterations needed is still obtained with demand function C and the lowest level of noise, but even here the average number of iterations needed is already 10.41. There is also no clear pattern in how the average number of iterations needed until convergence changes with an increase in the level of noise in the demand. As the level of noise increases, the average number of iterations fluctuates for demand function B and C and even goes down for demand function A. Fluctuating is also a good term to describe the graph of the average number of iterations needed until convergence, set out against the start prices. This graph is shown in Figure 17, Figure 18 shows the standard deviations. The general shape of the graph as it had in the previous settings is still there, but it is no longer a nice and smooth graph. It is also flattened quite a lot, so there is no longer much difference between the high and the low values.

In this setting, the converged price is never exactly equal to the optimal price. The results for when the converged price is first rounded and then compared to the optimal price are found in Table 31. In the best case, with demand function C and the lowest level of noise in the demand, the percentage of simulations for which the rounded, converged price is equal to the optimal price is 17%. In the case of demand function A and the highest level of noise, this percentage is 3%. These percentages are a lot lower than in the previous, discrete setting and even a little lower than in the continuous setting, as in Table 13. Figure 19 shows the percentage of rounded, converged prices equal to the optimal price, set out against the starting prices, for demand function C and sigma equal to 1. From this graph it can be seen that the best performance is obtained with those combinations of start prices with one high price and one price that is around the optimal price or lower.

The difference between the found demand line and the actual demand line, at the point of convergence, is never equal to zero for this setting. But Figure 20 shows that the difference between these two points is very small for most combinations of start prices. When the found demand line at the point of convergence is rounded to the nearest integer and then compared to the actual, discrete demand line at this point, the values are sometimes found to be equal, as shown in Table 32. The highest percentage of simulations for which these values are equal is found with demand function C and the lowest level of noise. Here the maximum percentage is 82%. Surprisingly, the lowest values are found with demand function B, the minimum of them being 16%. Figure 21 shows the percentage of the simulation for which the difference between the found demand line and the actual, discrete demand line, at the point of convergence, is equal to zero, for demand function C and a sigma equal to 1. There is quite a large area of combinations of start prices that outperforms the rest of the combinations, but it is unclear exactly where the best performance is achieved. In this case, all price combinations that have one price above the optimal price and one price below the optimal price seem to work well.

Table 25: Minimum, positive regret after convergence, for discrete demand and alternative pricing method.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	0,00	0,01	7,26	213,96
Demand B	0,06	0,03	0,30	69,78
Demand C	0,15	0,12	0,60	8,35

Table 26: Standard deviations of the minimum, positive regret after convergence, for discrete demand and alternative pricing method.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	295,51	325,34	302,08	599,28
Demand B	293,46	295,08	287,56	380,04
Demand C	302,38	306,73	305,96	283,07

Figure 16: Regret after convergence, for demand function C, sigma 0.5, discrete demand and alternative pricing method.

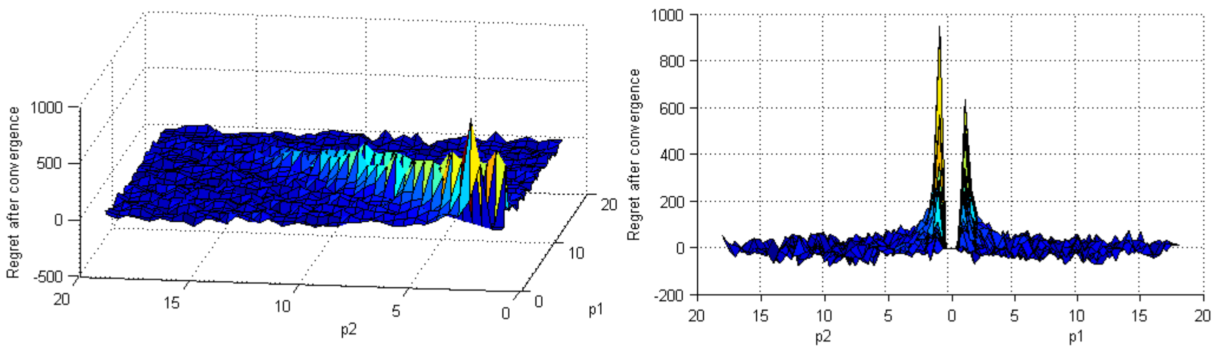


Table 27: Minimum difference between the converged and the optimal price, for discrete demand and alternative pricing method.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	0,12	0,18	0,32	0,70
Demand B	0,06	0,09	0,16	0,37
Demand C	0,03	0,04	0,08	0,18

Table 28: Prices corresponding to the minimum difference between the converged and the optimal price, for discrete demand and alternative pricing method.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	(17.5, 9.5)	(19, 6)	(19, 10)	(19, 8.5)
Demand B	(19, 10)	(19, 5.5)	(18.5, 5)	(19, 8.5)
Demand C	(19, 9.5)	(19, 7)	(19, 8)	(19, 8)

Table 29: Standard deviations for the minimum difference between the converged and the optimal price, for discrete demand and alternative pricing method.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	0,10	0,13	0,25	0,71
Demand B	0,06	0,07	0,14	0,26
Demand C	0,02	0,03	0,07	0,15

Table 30: Minimum number of iterations needed until convergence, for discrete demand and alternative pricing method.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	35,84	34,16	32,11	26,75
Demand B	20,13	36,02	36,68	29,55
Demand C	10,41	18,64	34,00	33,54

Figure 17: Average number of iterations needed until convergence, for different starting prices. Demand function B, sigma equal to 1, discrete demand and alternative pricing method.

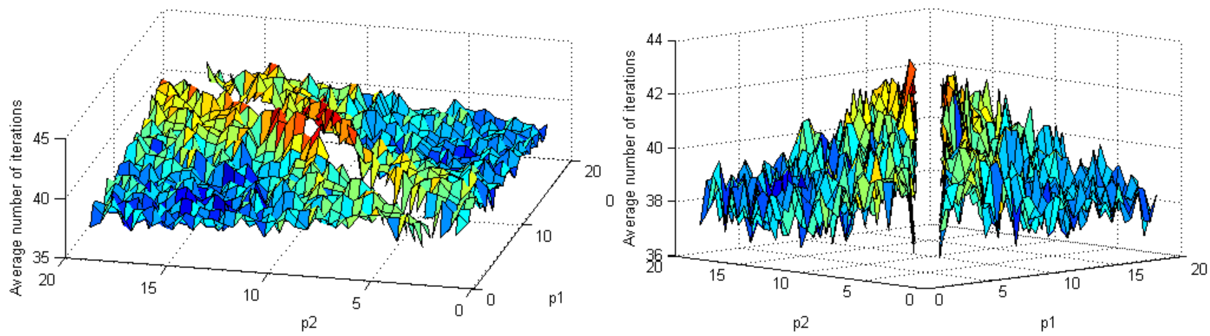


Figure 18: Standard deviations of the average number of iterations needed until convergence, for different starting prices. Demand function B, sigma equal to 1, discrete demand and alternative pricing method.

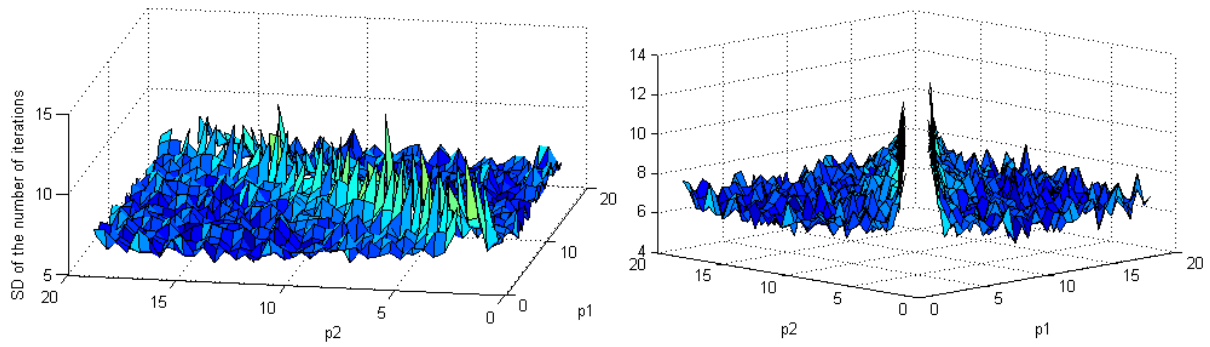


Table 31: Maximum percentage of rounded, converged prices equal to the optimal price, for discrete demand and alternative pricing method.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	8%	5%	4%	3%
Demand B	12%	8%	9%	5%
Demand C	17%	13%	9%	5%

Figure 19: Percentage of rounded, converged prices equal to the optimal price, for different starting prices. Demand function C, sigma equal to 1, discrete demand and alternative pricing method.

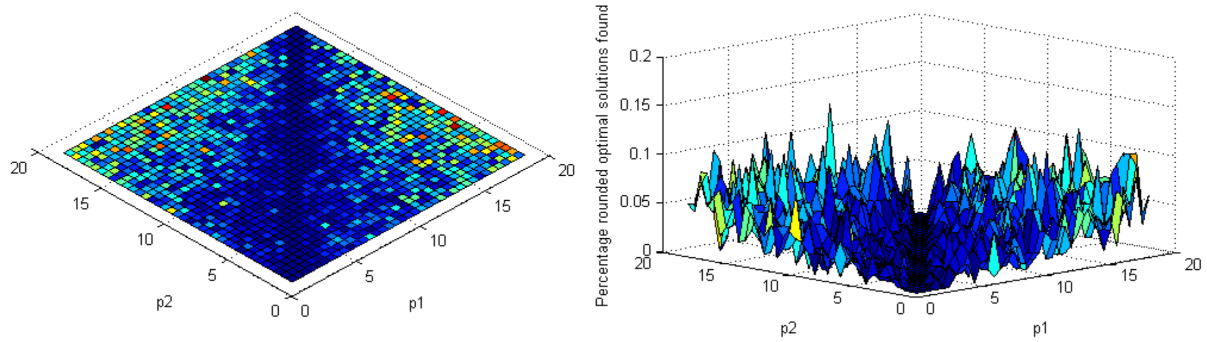


Table 32: Maximum percentage of simulations where the difference between the rounded, found demand line and the actual, discrete demand line is equal to zero, for discrete demand an alternative pricing method.

	Sigma 0.5	Sigma 1	Sigma 2	Sigma 5
Demand A	80%	55%	36%	17%
Demand B	78%	52%	32%	16%
Demand C	82%	54%	34%	18%

Figure 20: Difference between found demand line and actual demand line, for different starting prices. Demand function C, sigma equal to 1, discrete demand and alternative pricing method.

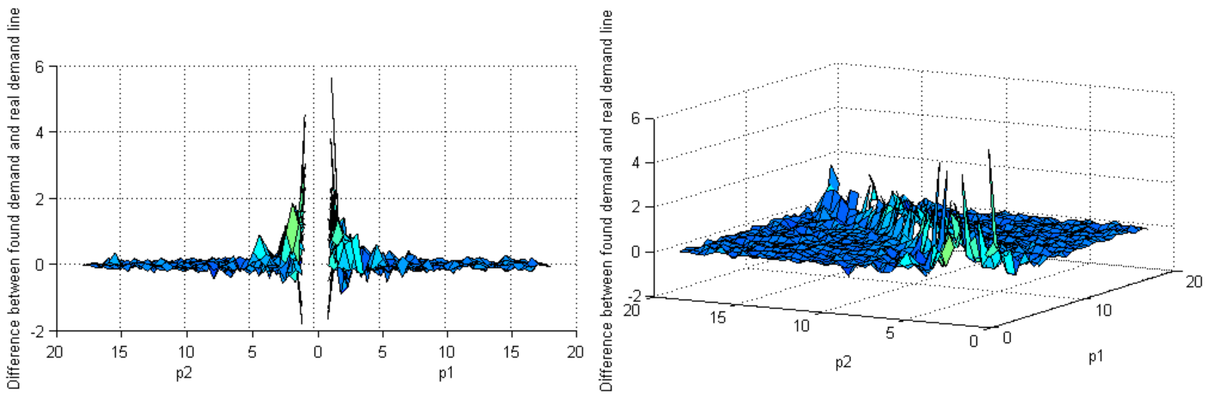
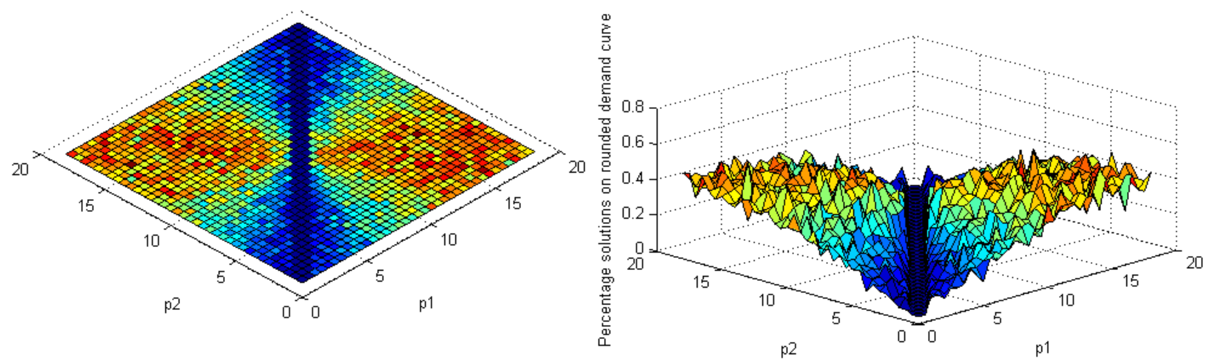


Figure 21: Percentage of rounded, found demand lines equal to the actual, discrete demand line, for different starting prices. Demand function C, sigma equal to 1, discrete demand and alternative pricing method.



4 Conclusion and Discussion

Finding an optimal price to sell your product for is very important for running a successful firm. One of the easiest ways to do this is using a certainty equivalent pricing policy. This pricing policy converges rather fast, but it does not always find the best solution. The fast convergence does make it highly applicable to products with a shorter life-span, in terms of how often the price can be changed.

The uncertainty about the demand and the shape of the demand curve play a role in the performance of this pricing policy. A high uncertainty in the demand is of course bad for the performance. What is considered as high uncertainty depends on the shape of the demand curve. A steeper demand curve can handle a little more uncertainty than a flatter one. This means that the steeper demand curve usually performs better.
The exception to this is when the average regret per iteration is considered. Table 1 showed that the average regret per iteration was higher for low levels of noise and a steep demand curve. This difference in behaviour could also be seen in the graphs showing the regret per iteration for different combinations of start prices. Figure 2 shows this graph for the steepest demand function and the lowest level of noise, while Figure 3 shows this for the flattest demand function and highest level of noise. As can be seen from Figure 2, the low level of noise almost has no effect on the performance of the pricing policy. The graph is very smooth and has exactly the shape that would expected. The regret per iteration is high for very low or very high start prices, because there the regret per iteration largely depends on the regret of the start prices. As the level of noise increases, so do the number of iterations needed until convergence. When the number of iterations increases, the regret per iteration depends less on the regret of the start prices and the shape of the graph changes.

Overall, the certainty equivalent pricing policy performs best under continuous demand. Of course, the type of demand depends on the product and most of the time it will be discrete. When the demand is discrete, it is best to assume a continuous demand function while finding the optimal price. When the demand is assumed to be discrete and the prices are matched to that, the performance of the pricing policy will go down. In this setting a lot of price possibilities are never considered, because, under the current knowledge about the demand curve, they correspond to a non-discrete demand. As long as the pricing policy has not yet yielded a converged price, the knowledge about the demand line is not accurate enough to base such decisions on.

This paper has shown that when using this pricing policy, it is important to carefully choose the start prices. The start prices should be chosen such that the regret is as low as possible and the probability of converting to the optimal price as high as possible. The number of iterations needed until convergence might be of less importance, but it is always beneficial if the optimal price is reached sooner than later. While it is not possible to give two exact start prices for which the certainty equivalent pricing policy performs the best, there is a certain pattern to which combinations of start prices perform better than others. First of all, in all graphs shown in this paper it can be seen that this pricing policy performs badly for start prices that are close together. But there are also prices that always perform rather well. In the continuous setting, the combination of start prices that gave the lowest regret after convergence, as seen in Figure 5, consist of one high price and price that is somewhere between one and the optimal price. The lowest number of iterations needed until convergence, for the continuous setting (Figure 6) as well as for the discrete setting (Figure 11), were achieved with combinations of start prices that consist of one very high price and one price that is at or slightly below the optimal value. In the continuous case, the highest percentage of simulations for which the rounded, converted price was equal to the optimal price (Figure 8) was given by start prices that included one high price and one price that was equal to the optimal price or lower. The slight exception to this pattern is given by the discrete setting. Here the highest percentages of rounded, converted prices that were equal to the optimal price (Figure 13) were achieved with one price slight below the optimal value and the other price slightly above the optimal value, instead of a very high value. The same goes for the percentages of the simulations for which the found demand line at the point of convergence was equal to the actual, discrete demand line at this point, as can be seen in Figure 15. To be certain of a good performance of the certainty equivalent pricing policy, it is recommended to use one high price and one price slightly below what is assumed to be the optimal value.

To gain a more accurate price, one might try to execute the certainty equivalent pricing policy twice and use the results to give a better estimate of the demand line. As shown in Figure 9 for the continuous case and Figure 14 for the discrete case, the difference between the found demand line and the actual demand line, at the point of the converged price, is very small. When two of these points are known, a new demand line can be constructed, which hopefully is a more accurate representation of the actual demand line. It can also be the case, however, that these two points are always close together and thus very sensitive to small changes in the demand. Further research will have to be done to see if this is a useable application for the certainty equivalent pricing policy.

5 References

A. den Boer and B. Zwart. Simultaneously Learning and Optimizing using Controlled Variance Pricing. 2010

J. Broder and P. Rusmevichientong. Dynamic pricing under a logit choice model. Technical report, Cornell University, 2009. URL http://www.cam.cornell.edu/~jbroder/logit-pricing.pdf.

A.X. Carvalho and M.L. Puterman. Dynamic optimization and learning: How should a manager set prices when the demand function is unknown? Technical report, Instituto de Pesquisa Econmica Aplicada - IPEA, Discussion Papers 1117., 2005b.

M.S. Lobo and S. Boyd. Pricing and learning with uncertain demand. Technical report, Fuqua School of Business, Duke University, 2003.

Appendix: Matlab code

Bold code shows changes made for the discrete setting. ***Bold, italic*** code shows extensions made for the alternative pricing method.

```
intercept = 200;
slope = -10
p1vector = 1:0.5:19;
p2vector = 1:0.5:19;
B = 100;
sigma = 1
CONVERGENCELIMIT = 0.01;
MINITERATIONS = 50
pmin = p1vector(1);
pmax = p1vector(end);
popt = intercept / (-2 * slope);
dopt = round(intercept + slope * popt); %in expectation, random element is zero, demand discrete
ropt = popt * dopt;

regretPerIterationMatrix = zeros(length(p1vector),length(p1vector));
sdRegretPerIterationMatrix = zeros(length(p1vector),length(p1vector));
iterationsMatrix = zeros(length(p1vector),length(p1vector));
sdIterationsMatrix = zeros(length(p1vector),length(p1vector));
optimalMatrix = zeros(length(p1vector),length(p1vector));
optimalRoundedPriceMatrix = zeros(length(p1vector),length(p1vector));
percentageOptimalLineMatrix = zeros(length(p1vector),length(p1vector));
percentageOptimalLineRoundedMatrix = zeros(length(p1vector),length(p1vector));
differenceMatrix = zeros(length(p1vector),length(p1vector));
sddifferenceMatrix = zeros(length(p1vector),length(p1vector));
meanPriceMatrix = zeros(length(p1vector),length(p1vector));
sdPriceMatrix = zeros(length(p1vector),length(p1vector));
regretAfterConvergenceMatrix = zeros(length(p1vector),length(p1vector));
sdregretAfterConvergenceMatrix = zeros(length(p1vector),length(p1vector));
differenceFromDemandLineMatrix = zeros(length(p1vector),length(p1vector));
sddifferenceFromDemandLineMatrix = zeros(length(p1vector),length(p1vector));


%for each p1 and p2, do B runs
for i = 1:length(p1vector);
   p1 = p1vector(i);
   for j = 1:length(p2vector);
      if j < (i - 1) || j > (i + 1)      %prev: j~=i
         p2 = p2vector(j);
         regretPerIterationvec = [];
         regretAfterConvergencevec = [];
         iterationsvec = [];
         differenceConvergedAndOptimalvec = [];
         pricevec = [];
         percentageOptimal = 0;
         percentageOptimalLine = 0;
         optimalRoundedPrice = 0;
         percentageOptimalLineRounded = 0;
         differenceFromDemandLineVector = [];

         for k=1:B;
            %initialise
            pvector = [p1, p2];
            demand1 = round(intercept + slope * p1 + random('norm', 0,sigma));
            if demand1 < 0
               demand1 = 0;
            end
            demand2 = round(intercept + slope * p2 + random('norm', 0,sigma));
            if demand2 < 0
               demand2 = 0;
            end
            dvector = [demand1, demand2];
            iterations = 2;
            regret = (ropt - p1*round(intercept + slope * p1))+(ropt-p2*round(intercept + slope * p2));

            %do linear regression
```

```
a = polyfit(pvector, dvector, 1);
a1 = a(1);
a0 = a(2);

if a1 == 0
   pnew = pmax;
else

   %calculate new price
   pnew = a0 / (-2 * a1);

   %calculate discrete price
   demand = a0 + a1 * pnew;
   demandlow = floor(demand);
   demandhigh = ceil(demand);
   plow = (demandlow - a0) / a1;
   phigh = (demandhigh - a0) / a1;
   rlow = plow * (a0 + a1 *plow);
   rhigh = phigh * (a0 + a1 * phigh);
   if rlow > rhigh
      pnew = plow;
   else
      pnew = phigh;
   end

   %check if price is allowed
   if (pnew < pmin) || (pnew > pmax)
      rmin = pmin * (a0 + a1 * pmin);
      rmax = pmax * (a0 + a1 * pmax);
      if rmin > rmax
         pnew = pmin;
      else
         pnew = pmax;
      end
   end
end

pvector = [pvector, pnew];
dnew = round(a0 + a1 * pnew + random('norm', 0,sigma));
if dnew < 0
   dnew = 0;
end
dvector = [dvector, dnew];
iterations = iterations + 1;
dreal = round(intercept + slope * pnew);
rreal = pnew * dreal;
regret = regret + (ropt - rreal);

test1=0;
it_aux=iterations;
while (abs(pnew - pvector(end - 1)) > CONVERGENCELIMIT) || iterations < MINITERATIONS

   if (abs(pnew - pvector(end - 1)) <= CONVERGENCELIMIT) && iterations < MINITERATIONS
      if test1 == 0
         test1 = 1;
         it_aux = it_aux + 1;
      end
   else
      it_aux=it_aux+1;
   end
   if test1==1 && (abs(pnew - pvector(end - 1)) > CONVERGENCELIMIT)
      test1=0;
      it_aux=iterations + 1;
   end

   %do linear regression
   a = polyfit(pvector, dvector, 1);
   a1 = a(1);
   a0 = a(2);

   if a1 == 0
```

```
            pnew = pmax;
        else

            %calculate new price
            pnew = a0 / (-2 * a1);

            %calculate discrete price
            demand = a0 + a1 * pnew;
            demandlow = floor(demand);
            demandhigh = ceil(demand);
            plow = (demandlow - a0) / a1;
            phigh = (demandhigh - a0) / a1;
            rlow = plow * (a0 + a1 *plow);
            rhigh = phigh * (a0 + a1 * phigh);
            if rlow > rhigh
                pnew = plow;
            else
                pnew = phigh;
            end

            %check if price is allowed
            if (pnew < pmin) || (pnew > pmax)
                rmin = pmin * (a0 + a1 * pmin);
                rmax = pmax * (a0 + a1 * pmax);
                if rmin > rmax
                    pnew = pmin;
                else
                    pnew = pmax;
                end
            end
        end
        pvector = [pvector, pnew];
        dnew = round(a0 + a1 * pnew + random('norm', 0,sigma));
        if dnew < 0
            dnew = 0;
        end
        dvector = [dvector, dnew];
        iterations = iterations + 1;
        dreal = round(intercept + slope * pnew);
        rreal = pnew * dreal;
        regret = regret + (ropt - rreal);
    end

    differenceFromDemandLine = (a0 + pnew * a1) - (intercept + slope * pnew);
    differenceFromDemandLineVector = [differenceFromDemandLineVector, differenceFromDemandLine];
    %solution on demand line?
    if (a0 + pnew * a1) == (intercept + slope * pnew)
        percentageOptimalLine = percentageOptimalLine + 1;
    end
    %solution on rounded line?
    if round(a0 + pnew * a1) == round(intercept + slope * pnew)
        percentageOptimalLineRounded = percentageOptimalLineRounded + 1;
    end
    %optimal solution?
    if pnew == popt
        percentageOptimal = percentageOptimal + 1;
    end
    if (round(pnew * 100) / 100) == popt
        optimalRoundedPrice = optimalRoundedPrice + 1;
    end
    differenceConvergedAndOptimalvec = [differenceConvergedAndOptimalvec, (abs(pnew - popt))];
    iterationsvec = [iterationsvec, it_aux];
    regretPerIteration = regret / iterations;
    regretPerIterationvec =[regretPerIterationvec, regretPerIteration];
    regretAfterConvergencevec = [regretAfterConvergencevec, ((ropt - rreal) * 100)];
    pricevec = [pricevec, pnew];
end

regretPerIterationMatrix(i,j) = mean(regretPerIterationvec);
sdRegretPerIterationMatrix(i,j) = std(regretPerIterationvec);
iterationsMatrix(i,j) = mean(iterationsvec);
```

```
        sdIterationsMatrix(i,j) = std(iterationsvec);
        optimalMatrix(i,j) = percentageOptimal / B;
        percentageOptimalLineMatrix(i,j) = percentageOptimalLine / B;
        percentageOptimalLineRoundedMatrix(i,j) = percentageOptimalLineRounded / B;
        optimalRoundedPriceMatrix(i,j) = optimalRoundedPrice / B;
        differenceMatrix(i,j) = mean(differenceConvergedAndOptimalvec);
        sddifferenceMatrix(i,j) = std(differenceConvergedAndOptimalvec);
        meanPriceMatrix(i,j) = mean(pricevec);
        sdPriceMatrix(i,j) = std(pricevec);
        regretAfterConvergenceMatrix(i,j) = mean(regretAfterConvergencevec);
        sdregretAfterConvergenceMatrix(i,j) = std(regretAfterConvergencevec);
        differenceFromDemandLineMatrix(i,j) = mean(differenceFromDemandLineVector);
        sddifferenceFromDemandLineMatrix(i,j) = std(differenceFromDemandLineVector);
      end
    end
end

%values
averageRegretPerIteration = mean(regretPerIterationMatrix(regretPerIterationMatrix~=0))
best_value_regret = min(regretPerIterationMatrix(regretPerIterationMatrix>0))
[r_regret c_regret] = find(regretPerIterationMatrix==best_value_regret);
SD_best_regret = sdRegretPerIterationMatrix(r_regret, c_regret)
r_regret = (r_regret - 1) * 0.5 + 1
c_regret = (c_regret - 1) * 0.5 + 1

averageIterations = mean(iterationsMatrix(iterationsMatrix~=0))
best_value_iterations = min(iterationsMatrix(iterationsMatrix>0))
[r_iterations c_iterations] = find(iterationsMatrix==best_value_iterations);
if length(r_iterations) == 1
   SD_best_iterations = sdIterationsMatrix(r_iterations, c_iterations)
   r_iterations = (r_iterations - 1) * 0.5 + 1
   c_iterations = (c_iterations - 1) * 0.5 + 1
end

averageDifference = mean(differenceMatrix(differenceMatrix~=0))
best_value_diff = min(differenceMatrix(differenceMatrix>0))
[r_diff c_diff] = find(differenceMatrix==best_value_diff);
SD_best_diff = sddifferenceMatrix(r_diff, c_diff)
r_diff = (r_diff - 1) * 0.5 + 1
c_diff = (c_diff - 1) * 0.5 + 1

averagePrice = mean(meanPriceMatrix(meanPriceMatrix~=0))

averageRegretAfterConvergence = mean(regretAfterConvergenceMatrix(regretAfterConvergenceMatrix~=0))
best_value_regret_ac = min(regretAfterConvergenceMatrix(regretAfterConvergenceMatrix>0))
[r_regretac c_regretac] = find(regretAfterConvergenceMatrix==best_value_regret_ac);
SD_best_regret_ac = sdregretAfterConvergenceMatrix(r_regretac, c_regretac)
r_regretac = (r_regretac - 1) * 0.5 + 1
c_regretac = (c_regretac - 1) * 0.5 + 1

percentage_opt = max(optimalMatrix(:))
percentage_opt_line = max(percentageOptimalLineMatrix(:))
percentage_rounded_opt = max(optimalRoundedPriceMatrix(:))
[r_opt c_opt] = find(optimalRoundedPriceMatrix==percentage_rounded_opt);
if length(r_opt) == 1
   r_opt = (r_opt - 1) * 0.5 + 1
   c_opt = (c_opt - 1) * 0.5 + 1
end

percentage_opt_line_rounded = max(percentageOptimalLineRoundedMatrix(:))
[r_optline_rounded c_optline_rounded] = find(percentageOptimalLineRoundedMatrix==percentage_opt_line_rounded);
if length(r_optline_rounded) == 1
   r_optline_rounded = (r_optline_rounded - 1) * 0.5 + 1
   c_optline_rounded = (c_optline_rounded - 1) * 0.5 + 1
end

%plots
regretPerIterationMatrix(regretPerIterationMatrix==0) = NaN;
subplot(2,2,1);
surf(p1vector,p2vector,regretPerIterationMatrix);
xlabel('p1');
```

```
ylabel('p2');
zlabel('Regret per iteration');
title('Regret per iteration for different starting prices');

sdRegretPerIterationMatrix(sdRegretPerIterationMatrix==0) = NaN;
subplot(2,2,2);
surf(p1vector,p2vector,sdRegretPerIterationMatrix);
xlabel('p1');
ylabel('p2');
zlabel('SD of the regret per iteration');
title('SD of the regret per iteration for different starting prices');

iterationsMatrix(iterationsMatrix==0) = NaN;
subplot(2,2,3);
surf(p1vector,p2vector,abs(iterationsMatrix));
xlabel('p1');
ylabel('p2');
zlabel('Average number of iterations');
title('Average number of iterations for different starting prices');

sdIterationsMatrix(sdIterationsMatrix==0) = NaN;
subplot(2,2,4);
surf(p1vector,p2vector,sdIterationsMatrix);
xlabel('p1');
ylabel('p2');
zlabel('SD of the number of iterations');
title('SD of the number of iterations for different starting prices');

saveas(gcf, 'CEPP_discrete_norm_sigma1_slope10_part1.fig');

clf;
meanPriceMatrix(meanPriceMatrix==0) = NaN;
subplot(2,2,1);
surf(p1vector,p2vector,meanPriceMatrix);
xlabel('p1');
ylabel('p2');
zlabel('Average found price');
title('Average found price for different starting prices');

sdPriceMatrix(sdPriceMatrix==0) = NaN;
subplot(2,2,2);
surf(p1vector,p2vector,sdPriceMatrix);
xlabel('p1');
ylabel('p2');
zlabel('SD of the found price');
title('SD of the found price per iteration for different starting prices');

subplot(2,2,3);
surf(p1vector,p2vector,abs(differenceMatrix));
xlabel('p1');
ylabel('p2');
zlabel('Absolute difference found and optimal price');
title('Absolute difference found and optimal price for different starting prices');

subplot(2,2,4);
surf(p1vector,p2vector,abs(sddifferenceMatrix));
xlabel('p1');
ylabel('p2');
zlabel('SD of the absolute difference found and optimal price');
title('SD of the absolute difference found and optimal price for different starting prices');

saveas(gcf, 'CEPP_discrete_norm_sigma1_slope10_part2.fig');

clf;
subplot(2,2,1);
surf(p1vector,p2vector,optimalRoundedPriceMatrix);
xlabel('p1');
ylabel('p2');
zlabel('Percentage rounded optimal solutions found');
title('Percentage rounded optimal solutions found for different starting prices');
```

```
subplot(2,2,2);
surf(p1vector,p2vector,percentageOptimalLineRoundedMatrix);
xlabel('p1');
ylabel('p2');
zlabel('Percentage solutions on rounded demand curve');
title('Percentage solutions on rounded demand curve for different starting prices');

regretPerIterationMatrix(regretAfterConvergenceMatrix==0) = NaN;
subplot(2,2,3);
surf(p1vector,p2vector,regretAfterConvergenceMatrix);
xlabel('p1');
ylabel('p2');
zlabel('Regret after convergence');
title('Regret after convergence for different starting prices');

sdRegretPerIterationMatrix(sdregretAfterConvergenceMatrix==0) = NaN;
subplot(2,2,4);
surf(p1vector,p2vector,sdregretAfterConvergenceMatrix);
xlabel('p1');
ylabel('p2');
zlabel('SD of the regret after convergence');
title('SD of the regret after convergence for different starting prices');

saveas(gcf, 'CEPP_discrete_norm_sigma1_slope10_part3.fig');

clf;
differenceFromDemandLineMatrix(differenceFromDemandLineMatrix==0) = NaN;
subplot(2,1,1);
surf(p1vector,p2vector,differenceFromDemandLineMatrix);
xlabel('p1');
ylabel('p2');
zlabel('Difference between found demand and real demand line');
title('Difference between found demand and real demand line for different starting prices');

subplot(2,1,2);
surf(p1vector,p2vector, sddifferenceFromDemandLineMatrix);
xlabel('p1');
ylabel('p2');
zlabel('SD of the difference between found demand and real demand line');
title('SD of the difference between found demand and real demand line for different starting prices');

saveas(gcf, 'CEPP_discrete_norm_sigma1_slope10_part4.fig');
```