SHAKYA, S., OLIVEIRA, F. and OWUSU, G. 2009. Analysing the effect of demand uncertainty in dynamic pricing with EAs. In Bramer, M., Petridis, M. and Coenen, F. (eds.) *Research and development in intelligent systems XXV: proceedings of the 28th British Computer Society's Specialist Group on Artificial Intelligence (SGAI) international conference on innovative techniques and applications of artificial intelligence (AI-2008)*, 9-11 December 2008, Cambridge, UK. London: Springer [online], pages 77-90. Available from: https://doi.org/10.1007/978-1-84882-171-2_6

Analysing the effect of demand uncertainty in dynamic pricing with EAs.

SHAKYA, S., OLIVEIRA, F. and OWUSU, G.

2009

Analysing the Effect of Demand Uncertainty in Dynamic Pricing with EAs

Siddhartha Shakya, Fernando Oliveira, and Gilbert Owusu

Abstract Dynamic pricing is a pricing strategy where a firm adjust the price for their products and services as a function of its perceived demand at different times. In this paper, we show how Evolutionary algorithms (EA) can be used to analyse the effect of demand uncertainty in dynamic pricing. The experiments are conducted in a range of dynamic pricing problems considering a number of different stochastic scenarios with a number of different EAs. The results are analysed, which suggest that higher demand fluctuation may not have adverse effect to the profit in comparison to the lower demand fluctuation, and that the reliability of EA for finding accurate policy could be higher when there is higher fluctuation then when there is lower fluctuation.

1 Introduction

Pricing is one of the most important decisions that a firm needs to make in order to survive in a competitive marketplace. If done carefully, it can be a valuable tool for the firm to achieve a number of different business goals, such as profit maximisation, demand management, value creation, etc. Conversely, a poor pricing policy could lead to a loss, and consequently extinction of the firm. Dynamic pricing [27] [15][3] is a pricing strategy where a firm adjust the price for their products and services as a function of its perceived demand at different times. Traditionally, it has been

Siddhartha Shakya
Intelligent Systems Research Centre, BT Group Chief Technology Office, Adastral Park, Ipswich, IP5 3RE, UK,e-mail: sid.shakya@bt.com

Fernando Oliveira
Warwick Business School, University of Warwick, Coventry, CV4 7AL, UK, e-mail: fernando.oliveira@wbs.ac.uk

Gilbert Owusu
Intelligent Systems Research Centre, BT Group Chief Technology Office, Adastral Park, Ipswich, IP5 3RE, UK,e-mail: gilbert.owusu@bt.com

applied in service industries, such as airlines, hotels and rentals [16]. For example, in airlines, the price for a seat changes according to the time remaining prior to the flight and number of available seats. Recent developments in information technology and eCommerce have led the dynamic pricing to spread over a wide range of other industries such as retail [11][6][1], wholesale [23] and auctions [24].

In this paper we show how the evolutionary models can be used for dynamic pricing in a stochastic setting where the demand is uncertain. Our motivation is to use such pricing approach to control demand and manage resources in a service industry [28][18]. In the type of problems we are analyzing, resource management is the effective workforce utilization for a given calendarised work demand profile, while meeting a set of constraints such as quality of service targets, conflict resolution schemes, such as overtime and borrowing additional workforce [17]. The system described in [18] integrates various Artificial Intelligence and Operational Research techniques in order to forecast demand for specific products and services at regional level, and to optimize the allocation of resources to each one of the region. Our aim is to use evolutionary algorithms as an alternative technique to manage resources by means of effective pricing. In particular, we extend the model presented in [26] and implement several evolutionary algorithms (EA) [7] for solving them. We analyse the performance of these algorithms in finding optimal profit, and also analyse the effect of demand uncertainty have on total profit and on the reliability of these algorithms.

EAs have been successfully applied in wide range of search and optimization problems. They are inspired by Darwin's theory of evolution where *selection* and *variation* work together to evolve a better solution. Different EA has been proposed using different approaches to selection and variation. In this paper we investigate two EAs to solve dynamic pricing problems, namely the genetic algorithm (GA) [7][12] and the estimation of distribution algorithms (EDA) [10],[14]. GA and EDA differ in the way they implement the variation operator. In particular, GA uses *crossover* and *mutation* approach to variation. In contrast, EDA uses probabilistic approach to variation, where a probabilistic model is built and sampled to generate new solutions. EDA is a relatively new area in evolutionary computation field and are being increasingly applied to real-world optimization problems. They are often reported to perform better than the traditional GAs [8] [22]. It is, therefore, interesting to see the performance of both EDA and GA with regards to dynamic pricing.

The objectives of this paper are to: a) analyze the performance of evolutionary algorithms as tools to approximate optimal behaviour in dynamic pricing; b) compare the algorithms under different scenarios with different levels of demand uncertainty; and c) analyse the effect of demand uncertainty have on the profit.

The paper is structured as follows. Section 2 presents the mathematical model of uncertainty for dynamic pricing. Section 3 describes a way to representing the dynamic pricing problem for solving them using EA, and also gives an overview of the implemented EAs. Section 4 describes the experimental results and presents the analysis of the results. Finally, section 5 concludes the paper.

2 A Mathematical Model of Dynamic Pricing

Depending on the nature of the product (or service), and the expected demand behaviour, a company has to choose between short-term or long-term profits. Short-term profit is to take advantage of the dynamics of demand throughout a week, or even during a day. Long-term profit is to model the long-term implications of short-term pricing and investment policies with the goal of maximising the long-term, for example months or years, profit. This section describes a dynamic pricing model that can be used for analyzing both short-term and long-term profits.

The total profit, (Π), earned from a product during the planning horizon can be modelled as

$$\Pi = \sum_{t=1}^{N} (P_t Q_t - C_t Q_t) \tag{1}$$

Here, N is the number of periods in planning horizon, Q_t is the total sales (or the production) of the product (which is equal to, or less than, the demand for the product), P_t is the average price of the product, and C_t is the cost of producing one extra product in period t, $P_t Q_t$ is the total revenue at period t, and $C_t Q_t$ is the variable cost at period t. In the situations where the demand is uncertain, the sales can be given by the sum of expected sales $E(Q_t)$ and a stochastic term modelling the fluctuation in demand as

$$Q_t = E(Q_t) + \varepsilon_t \tag{2}$$

Here, ε_t represents the fluctuation in demand. We give it as a normal random variable, upper and lower bounded by the positive and negative expected sales respectively. This can be written as

$$\varepsilon_t = \begin{cases} max\{-E(Q_t), N(0, \sigma E(Q_t))\} & \text{if} \quad N(0, \sigma E(Q_t)) < 0 \\ min\{E(Q_t), N(0, \sigma E(Q_t))\} & \text{if} \quad N(0, \sigma E(Q_t)) > 0 \end{cases} \tag{3}$$

where, $\sigma \geq 0$ is the fraction of $E(Q_t)$ representing the strength of the fluctuation. Higher σ represents high fluctuation in demand and lower σ represents lower fluctuation in demand. From (1) and (2), the total profit can be written as

$$\Pi = \sum_{t=1}^{N} (E(Q_t) + \varepsilon_t)(P_t - C_t) \tag{4}$$

Therefore, the expected profit can be written as

$$E(\Pi) = \sum_{t=1}^{N} E(Q_t)(P_t - C_t) \tag{5}$$

Also, the expected sales $E(Q_t)$ in a period t depends on the price for the product in that period and the price for the product in other periods in the planning horizon. For example, in airlines (or hotels), sales for seats (or rooms) in a given day depend on their price on that day and on other days within the planning horizon, which are

visible to customers. We represent this price-demand relationship linearly [1] as

$$E(Q_t) = b_{0t} + b_{1t}P_1 + b_{2t}P_2 + ... + b_{tt}P_t + ... + b_{Nt}P_N \tag{6}$$

where, b_{0t} is the intercept of the linear model representing the customer base (total customers willing to buy the product in period t), and b_{jt} are the parameters known as slopes which represent the impact of price at time j have on the demand at time t. Note that, in general, the parameter b_{tt} is negative, since higher price for the product in a period is likely to decrease the demand for that product in that period.

Inversely, the price for the product P_t in a period can be written in terms of the expected sales for the product in that period and the expected sales in other periods in the planning horizon as

$$P_t = a_{0t} + a_{1t}E(Q_1) + a_{2t}E(Q_2) + ... + a_{tt}E(Q_t) + ... + a_{Nt}E(Q_N) \tag{7}$$

where, a_{0t} is the intercept and a_{jt} are the parameters representing the impact of sales at time j have on the price at time t. Note that, in general, the parameter a_{tt} is negative, since higher sales for the product in a period is likely to be due to the lower price for that product in that period.

From (4) and (7) we get the general model for the total profit with stochastic demand as

$$\Pi = \sum_{t=1}^{N} (E(Q_t) + \varepsilon_t) \left(a_{0t} + \sum_{j=1}^{N} a_{jt}E(Q_j) - C_t \right) \tag{8}$$

The model of stochastic dynamic pricing presented here is different than the model presented in [26]. Here, rather than applying the single stochastic term to total profit, we apply stochastic term to each individual periods in order to accurately model individual demand fluctuations in different selling periods.

Now let us define some additional constraints a firm needs to impose when defining its policy for pricing a given product.

a. Capacity constraints - These are the number of products available in a given period, and have the lower and upper bounds, represented for all $t = 1..N$ as

$$\begin{aligned} M_t &\leq Q_t - \text{Lower bound for the capacity constraint} \\ K_t &\geq Q_t - \text{Upper bound for the capacity constraint} \end{aligned} \tag{9}$$

b. Price caps - These are the selling price of a product in a given period, and also have lower and upper bounds, represented for all $t = 1..N$ as

$$\begin{aligned} \overline{P}_t &\leq P_t - \text{Lower bound for the price cap} \\ \underline{P}_t &\geq P_t - \text{Upper bound for the price cap} \end{aligned} \tag{10}$$

[1] Linear models are widely used for representing the price and demand relationship, both in research and in practice. There are, however, other models, such as *exponential* and *multinomial logit* [27][23], that could be similarly used to represent this relationship. Testing these models is out of the scope of this paper and could be the part of the future work.

Given the parameters a_{jt}, the upper bound and lower bound to both the capacity constraint and the price cap, and also the σ for ε representing the fluctuation in demand, our goal is to find a policy $E(Q_t)$ (from which, using 7, we get the P_t) for all $t = \{1..N\}$ that maximize the total profit, i.e. maximize Π in the equation (8). Since equation (8) is nonlinear and requires to satisfy constraints defined in (9) and (10), this problem is a nonlinear constrained optimization problem.

3 Optimising Stochastic DP models using EAs

A general constrained optimization problem can be defined as $max_x f(x), \quad x \in S \subset \mathbb{R}^n$ subject to the linear or nonlinear constraints $g_i(x) \leq 0, \quad i = 1,...,m$. Here m is the total number of constraints. One of the most popular ways to solving constrained optimization problems with EAs is by using a *penalty function*. The idea is to construct a function that penalizes the original objective function for violating the constraints in the model. In order to avoid the penalty, the algorithm tries to focus its search on the feasible part of the search space. Here we use one such technique adopted from [19] and also implemented by [26], and define the penalty function as

$$F(x) = f(x) - h(k)H(x), \quad x \in S \subset \mathbb{R}^n \tag{11}$$

where, $f(x)$ is the original objective function (in our case it is defined by Π in equation (8)). $h(k)H(x)$ is the penalising part of the function, where $H(x)$ is the main penalty factor (equals to 0 when no constraints are violated) and $h(k)$ is known as the dynamically modified penalty value that intensifies the level of penalty according to the algorithm's current iteration k. Due to the limited space, we do not describe these factors in detail, interested readers are referred to [19] [26].

Solution representation for EA: A solution, x, is represented as a set $E(Q) = \{E(Q_1), E(Q_2), ..., E(Q_N)\}$, where each $E(Q_t)$ is represented by a bit-string of length l. The total length of a bit-string solution, $x = \{x_1, x_2, ..., x_n\}$, where $x_i \in \{0,1\}$, is therefore, equal to $n = l \times N$. The goal of an algorithm is to maximize the modified objective function defined in (11).

Overview of the used EAs: We adopt the approach presented in [26] and implement two EDAs and a GA for solving this problem. They include Population Based Incremental Learning (PBIL) algorithm [2]), Distribution Estimation using Markov Random Field with direct sampling (DEUM$_d$) algorithm [25] and a GA [7]. We also find it interesting to use a non-population based algorithm known as Simulated Annealing (SA) [9] for this problem. Let us describe the workflow of these algorithms.

PBIL

---

1. Initialize a probability vector $p = \{p_1, p_2, ..., p_n\}$ with each $p_i = 0.5$. Here, p_i represents the probability of x_i taking value 1 in the solution
2. Generate a population P consisting of M solutions by sampling probabilities in p

3. Select set D from P consisting of N best solutions
4. Estimate probabilities of $x_i = 1$, for each x_i, as

$$p(x_i = 1) = \frac{\sum_{x \in D, x_i = 1} x_i}{N}$$

5. Update each p_i in p using $p_i = p_i + \lambda(p(x_i = 1) - p_i)$. Here, $0 \leq \lambda \leq 1$ is a parameter of the algorithm known as the learning rate
6. Go to step 2 until termination criteria are meet

DEUM$_d$

---

1. Generate a population, P, consisting of M solutions
2. Select a set D from P consisting of N best solutions, where $N \leq M$.
3. For each solution, x, in D, build a linear equation of the form

$$\eta(F(x)) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n$$

   Where, function $\eta(F(x)) < 0$ is set to $-ln(F(x))$, for which $F(x)$, the fitness of the solution x, should be ≥ 1; $\alpha = \{\alpha_0, \alpha_1, \alpha_2, ..., \alpha_n\}$ are equation parameters.
4. Solve the build system of N equations to estimate α
5. Use α to estimate the distribution $p(x) = \prod_{i=1}^{n} p(x_i)$, where

$$p(x_i = 1) = \frac{1}{1 + e^{\beta \alpha_i}}, \;\; p(x_i = -1) = \frac{1}{1 + e^{-\beta \alpha_i}}$$

   Here, β (inverse temperature coefficient) is set to $\beta = g \cdot \tau$; g is current iteration of the algorithm and τ is the parameter known as the cooling rate
6. Generate M new solution by sampling $p(x)$ to replace P and go to step 2 until termination criteria are meet

---

GA

---

1. Generate a population P consisting of M solutions
2. Build a breeding pool by selecting N promising solutions from P using a selection strategy
3. Perform crossover on the breeding pool to generate the population of new solutions
4. Perform mutation on new solutions
5. Replace P by new solutions and go to step 2 until termination criteria are meet

SA

1. Randomly generate a solutions $x = \{x_1, x_2, ..., x_n\}$
2. For $i = 1$ to r do
   a. Randomly mutate a variable in x to get x'
   b. Set $\Delta F = F(x') - F(x)$
   c. Set $x = x'$ with probability

$$p(x') = \begin{cases} 1 & \text{if} \quad \Delta F \leq 0 \\ e^{-\Delta F/T} & \text{if} \quad \Delta F > 0 \end{cases}$$

   Where, temperature coefficient T was set to $T = 1/i \cdot \tau$; here, i is the current iteration and τ is the parameter of the algorithm called the cooling rate

3. Terminate with answer x.

The two implemented EDAs, PBIL and DEUM$_d$, both fall in the category of univariate EDAs, and assume that the variables in the solutions are independent. Other categories of EDA include, bivariate EDA [4][21], assuming at most pair-wise interaction between variables, and multivariate EDA [20][13][5], assuming interaction between multiple variables. Our motivation behind using univariate EDAs is two fold. Firstly, they are simple, and, therefore, often quickly converge to the optima, resulting in higher efficiency. This is particularly important in dynamic environment, where the pricing decisions have to be frequently changed. Secondly, the number of problems that has been shown to be solved by them is surprisingly large.

4 Experiments and Results

We perform three sets of experiments for both sort-term and long-term analysis, where each set modelled different scenarios.

For short-term analysis, we assume that the production for a given day is a negative function of the price on that day and a positive function of the prices on other days of the week. More specifically, we assume that at any given time t: a) Production decreases by one unit for each unit increase in price; b) an increase in sales in a given day reduces the sales during other days of the week. Further, the cost of an additional unit of production was assumed to be zero (all costs are fixed, i.e. no incremental cost) and the minimum production for each day was also assumed to be zero. Moreover, it was assumed that demand is higher during the first few days of

the week [2]. These are reflected in table 1 showing the setup for all a_{jt}. Also, for experiment 1, maximum production capacity, M_t, was set to 1000 units and maximum price, $\overline{P_t}$, was set to 250/unit (i.e., higher production flexibility and lower pricing flexibility), for experiment 2, M_t was set to 1000 units and $\overline{P_t}$ was set to 1000/unit (i.e., higher production flexibility and higher pricing flexibility), and for experiment 3, M_t was set to 300 units and $\overline{P_t}$ was set to 1000/unit during all seven days of the week (i.e., lower production flexibility and higher pricing flexibility).

Table 1 a_{jt} for all three short-term experiments

t	a_{0t}	a_{1t}	a_{2t}	a_{3t}	a_{4t}	a_{5t}	a_{6t}	a_{7t}
1	900	-1.0	0.1	0.1	0.1	0.1	0.1	0.1
2	800	0.0	-1.0	0.1	0.1	0.1	0.1	0.1
3	800	0.0	0.0	-1.0	0.1	0.1	0.1	0.1
4	700	0.0	0.0	0.0	-1.0	0.1	0.1	0.1
5	600	0.0	0.0	0.0	0.0	-1.0	0.1	0.1
6	500	0.0	0.0	0.0	0.0	0.0	-1.0	0.1
7	400	0.0	0.0	0.0	0.0	0.0	0.0	-1.0

Table 2 a_{jt} for long-term experiments no 1

t	a_{0t}	a_{1t}	a_{2t}	a_{3t}	a_{4t}	a_{5t}	a_{6t}	a_{7t}
1	3000	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
2	3000	0.5	-1.0	0.0	0.0	0.0	0.0	0.0
3	3000	0.0	0.5	-1.0	0.0	0.0	0.0	0.0
4	3000	0.0	0.0	0.5	-1.0	0.0	0.0	0.0
5	3000	0.0	0.0	0.0	0.5	-1.0	0.0	0.0
6	3000	0.0	0.0	0.0	0.0	0.5	-1.0	0.0
7	3000	0.0	0.0	0.0	0.0	0.0	0.5	-1.0

Table 3 a_{jt} for long-term experiments no 2 and 3

t	a_{0t}	a_{1t}	a_{2t}	a_{3t}	a_{4t}	a_{5t}	a_{6t}	a_{7t}
1	3000	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
2	3000	0.9	-1.0	0.0	0.0	0.0	0.0	0.0
3	3000	0.0	0.9	-1.0	0.0	0.0	0.0	0.0
4	3000	0.0	0.0	0.9	-1.0	0.0	0.0	0.0
5	3000	0.0	0.0	0.0	0.9	-1.0	0.0	0.0
6	3000	0.0	0.0	0.0	0.0	0.9	-1.0	0.0
7	3000	0.0	0.0	0.0	0.0	0.0	0.9	-1.0

For long-term analysis, we assumed that the production in a year is a negative function of the average price in that year and positive function of the production during previous year. More specifically, we assumed that at any given time t: a) production decreases by one unit for each pound increase in price; b) the company keeps a given proportion of its customers from the previous year. These are reflected in table 2 and 3 showing the setup for all a_{jt}. Further, in all three experiments, the cost of an additional unit production was assumed to be zero. For experiment 1, the maximum production capacity, M_t, was set to 3000 units during the first 4 years and set to 6000 units during the last 3 years, no maximum price, $\overline{P_t}$, was set and it was assumed that the company keeps 50% of its customers from the previous year. For experiment 2, the M_t was set to 3000 units during the first 4 years and set to 6000 units during the last 3 years, no maximum price was set and it was assumed that the

[2] This is a typical scenario for products, or services, whose demand is higher during working days of the week, such as airline seats, mobile phone use, and even restaurant lunch hour sales

company keeps 90% of its customers from the previous year. For experiment 3, the M_t was set to 3000 units during all 7 years, no maximum price was set and it was assumed that the company keeps 90% of its customers from the previous year.

Also, for each of these three scenarios in both short-term and long-term environment, we experiment with eleven different setups for demand fluctuation ranging from $\sigma = 0$ (i.e. no fluctuation) to $\sigma = 1$ (i.e. very high fluctuation).

In terms of the algorithms, we set the size of the bit string representing each $E(Q_t)$ to $l = 12$. Therefore the solution length, n, was equal to $l \times N = 84$. Since, there were a very high number of simulations involved, using moderate size of $l = 12$ significantly reduced the simulation time. Also, in order to parameterise the algorithms, we conduct a range of experiments using wide range of parameter setups for each algorithm and choose the setup that had the best performance. Such an empirical approach to parameterisation is typical in EA research. Following were the setups used for each of the algorithm: In each execution, the algorithm was allowed to do a fixed number of fitness evaluations. This was equal to 400000 for PBIL, DEUM$_d$ and GA, and 600000 for SA [3]. The number of fitness evaluation for PBIL, DEUM$_d$ and GA was calculated as the product of their population size, $PS = 400$, and the maximum number of generations, $MG = 1000$. For all experiments, the learning rate for PBIL was set to 0.02 and cooling rate for DEUM$_d$ was set to 0.02. For SA a very small cooling rate of 0.00001 was used. 10 best solutions were selected in PBIL and DEUM$_d$ for estimating the marginal probabilities. For GA, one-point crossover was used with crossover probability set to 0.7 and the mutation probability set to 0.01.

4.1 Results

A total of 100 executions of each algorithm were done for each experiment and the best policy together with the total profit found in each execution was recorded. The average total profit (Mean), the standard deviation of total profit (Stdev), and the best total profit (Max) out of all 100 executions for each of the algorithms are shown in Table 4 for short-term experiment 1, in Table 5 for short-term experiment 2 and in Table 6 for short-term experiment 3 for five different setups of σ [4]. Similarly, Table 7, 8 and 9 shows the results for long-term experiments 1, 2 and 3 respectively.

Also, a reliability factor (*RL*) measuring the reliability of the policy found by the algorithm is shown in the tables along with the other three metrics. For each algorithm, the reliability factor is the total percentage of runs where the *final population* of the algorithm converged to a feasible policy, i.e. the one satisfying all the constraints. If the final population in all of the runs converged to the feasible solution, the *RL* of the suggested policy is very high, since this indicates the high probability

[3] Since SA was not performing well in comparison to other algorithms, we allowed it to do more evaluation, in order to see whether its performance will be improved

[4] Although we perform experiment with 11 different setups of σ, due to the space limitation, we only present the tables with 5 different setups

Table 4 Short-term 1

σ	metric	SA	PBIL	DEUM$_d$	GA
0.0	Mean	N/A	924938	923564	**942937**
	Stdev	N/A	3188	4176	**5998**
	Max	N/A	927935	938937	**949922**
	RL	0%	100%	100%	**100%**
0.2	Mean	N/A	907226	904476	**914536**
	Stdev	N/A	8454	10579	**10167**
	Max	N/A	928237	934521	**937266**
	RL	0%	46%	46%	**48%**
0.5	Mean	N/A	899672	**898091**	N/A
	Stdev	N/A	13946	**14223**	N/A
	Max	N/A	925939	**931484**	N/A
	RL	0%	26%	**28%**	2%
0.8	Mean	N/A	**886883**	890867	N/A
	Stdev	N/A	**1452**	16913	N/A
	Max	N/A	**917616**	934883	N/A
	RL	0%	**19%**	15%	0%
1.0	Mean	N/A	**878393**	884754	N/A
	Stdev	N/A	**17150**	17642	N/A
	Max	N/A	**911962**	926150	N/A
	RL	0%	**12%**	11%	0%

Table 5 Short-term 2

σ	metric	SA	PBIL	DEUM$_d$	GA
0.0	Mean	1160467	**1173280**	1173251	1173243
	Stdev	7524	**165**	233	283
	Max	1173253	**1173299**	1173299	1173299
	RL	100%	**100%**	100%	100%
0.2	Mean	1129388	**1166115**	1161622	1163458
	Stdev	22396	**4313**	6453	5829
	Max	1167782	**1172747**	1172320	1171941
	RL	91%	**100%**	100%	100%
0.5	Mean	1086776	**1149881**	1138054	1136587
	Stdev	40797	**5965**	12182	12827
	Max	1155558	**1159827**	1163172	1156967
	RL	69%	**100%**	98%	100%
0.8	Mean	1085741	**1151703**	1140924	1139230
	Stdev	31044	**5526**	9257	10383
	Max	1144442	**1160956**	1157105	1159019
	RL	55%	**100%**	98%	100%
1.0	Mean	1084185	**1153184**	1141891	1140133
	Stdev	28600	**4160**	8779	8746
	Max	1154238	**1160043**	1158329	1156692
	RL	60%	**100%**	98%	100%

Table 6 Short-term 3

σ	metric	SA	PBIL	DEUM$_d$	GA
0.0	Mean	968970	968970	968970	**968970**
	Stdev	0.00	0.01	0.00	**0.21**
	Max	968970	968970	968970	**968970**
	RL	100%	100%	100%	**100%**
0.2	Mean	803924	847691	849495	**786725**
	Stdev	33489	10418	17122	**14848**
	Max	883369	869569	888038	**840326**
	RL	66%	47%	27%	**84%**
0.5	Mean	690419	695022	710566	**593640**
	Stdev	51689	23680	31968	**26811**
	Max	811061	764219	793566	**674953**
	RL	38%	49%	39%	**86%**
0.8	Mean	627084	**582653**	597541	571215
	Stdev	53243	**7547**	25650	13810
	Max	774356	**610820**	713464	646189
	RL	36%	**99%**	63%	98%
1.0	Mean	607560	**582401**	577594	570692
	Stdev	60597	**3837**	18248	13769
	Max	807870	**591077**	661086	655611
	RL	37%	**100%**	72%	95%

of achieving at least the suggested profit with such demand. Obviously, this is the case when there is no demand fluctuation ($\sigma = 0$). However, when uncertainty is introduced, for some tightly constrained problems, most of the policy could be out of the boundary of the constraints due to the higher fluctuation in demand. The algorithm would then converge to a non feasible region of the solution space. However, even in these scenarios, few random policies could be sampled with lower fluctuation that satisfies the constraints, simply due to the random error. Although, such a policy is not likely to achieve suggested profit, the algorithm would keep it as the best policy, even though its final population would converge to a non feasible policy. In these scenarios, *RL* allows us to identify the probability of obtaining such a false result by examining whether both the best policy found during entire iteration of the algorithm and the final population of the algorithm converged to a feasible solution. We, therefore, take *RL* as the main measure to test the performance of the

Table 7 Long-term 1

σ	metric	SA	PBIL	DEUM$_d$	GA
0.0	Mean	27821948	**28206184**	28205996	28205951
	Stdev	345156	**7**	672	814
	Max	28205221	**28206185**	28206185	28206185
	RL	100%	**100%**	100%	100%
0.2	Mean	26372470	27198936	27193683	**26876389**
	Stdev	706535	229150	274612	**208255**
	Max	27516273	27880747	27814142	**27486350**
	RL	79%	78%	78%	**91%**
0.5	Mean	24173310	25526098	25593119	**23866124**
	Stdev	1261215	407533	570856	**501397**
	Max	26969345	26788111	26964174	**25128428**
	RL	52%	77%	72%	**95%**
0.8	Mean	23404411	**24098148**	24216003	23585299
	Stdev	1059765	**396234**	553567	356613
	Max	25950583	**25215353**	26031802	24663332
	RL	42%	**99%**	74%	99%
1.0	Mean	23018437	**24010452**	23889324	23639083
	Stdev	1257847	**140554**	438312	268953
	Max	25709590	**24281190**	25256155	24170419
	RL	35%	**100%**	77%	99%

Table 8 Long-term 2

σ	metric	SA	PBIL	DEUM$_d$	GA
0.0	Mean	52203789	**54779449**	54779309	54779247
	Stdev	2201309	**34**	237	1130
	Max	54777957	54779471	54779471	54779470
	RL	100%	100%	100%	100%
0.2	Mean	46971642	50236548	50426208	**46345442**
	Stdev	1928960	686963	1099312	**1026420**
	Max	51136540	52412106	52589577	**49887769**
	RL	61%	43%	29%	**86%**
0.5	Mean	40631344	44166229	43926709	**35766901**
	Stdev	2900701	1512555	1842733	**1595034**
	Max	50084429	48960669	48251958	**41688039**
	RL	50%	29%	34%	**85%**
0.8	Mean	37252169	38197104	38735770	**34775462**
	Stdev	3133699	399310	1640259	**1399360**
	Max	46234840	38886057	43009539	**38414691**
	RL	45%	47%	43%	**95%**
1.0	Mean	36380548	38315131	37495900	**34631043**
	Stdev	3026899	380666	1579632	**1495657**
	Max	46097274	40576934	41897947	**40827152**
	RL	46%	57%	48%	**97%**

Table 9 Long-term 3

σ	metric	SA	PBIL	DEUM$_d$	GA
0.0	Mean	48635382	**48636408**	48636394	48636406
	Stdev	843	**0.00**	144	2.45
	Max	48636408	48636408	48636408	48636408
	RL	100%	100%	100%	100%
0.2	Mean	40051377	42674286	42917395	**38312603**
	Stdev	1808341	843943	1028819	**891778**
	Max	44852209	45048582	45973335	**40744748**
	RL	60%	40%	26%	**79%**
0.5	Mean	33785719	34590258	35346913	**28040236**
	Stdev	3308233	1530366	1755677	**1076736**
	Max	43427472	40296346	39909935	**30777018**
	RL	46%	43%	39%	**86%**
0.8	Mean	29884971	27536913	29387154	**26901891**
	Stdev	2994800	706814	1763762	**803103**
	Max	41867831	30598589	35041308	**30141010**
	RL	48%	96%	45%	**96%**
1.0	Mean	28830505	27362045	27753364	**26741629**
	Stdev	3187659	168031	1216581	**553008**
	Max	37370691	27687168	31542620	**29126642**
	RL	44%	100%	57%	**98%**

algorithms. If two algorithms had similar *RL*, the mean profit was taken as the next measurer of the performance. Also note that, the profits presented in the table represent the expected profit, $E(\Pi)$ (5) i.e., the one without the error term included. The value for the best performing algorithm is plotted in bold. Following are the analysis of the results.

a. Performance of the algorithms in different scenarios: As we can see from the tables, for all of the experiments, SA has the worst performance compared to other three algorithms with lowest value for *RL*. The performance of DEUM$_d$ and PBIL was somewhat comparable for all four metrics, with occasionally one outperforming another. In terms of mean profit, PBIL and DEUM$_d$ has the best performance. Finally, GA has the highest value for *RL* and therefore has the overall best performance. It can be noticed that, when there is no uncertainty in demand,

the reliability of all three EAs are similar, though in terms of mean total profit two EDAs, PBIL and DEUM$_d$, were better than GA (similar to [26]). However, when demand uncertainty is introduced our results show that GA, in most cases, retained its reliability, whereas the reliability of two EDAs decreased.

b. Impact of demand uncertainty on total profit: It can be observed from the tables that, in general, demand uncertainty reduces the total profit, and is true for all the tested algorithms. In order to show this, a graph showing the best average total profit found by the algorithms for long-term experiment 1 with 11 different setups of σ is plotted in Figure (1) [5]. Note that the total profit is mapped to the ratio between 1 and 0. It can be seen that the total profit reduces as demand fluctuation gets higher, till it reaches to the mark of 50% ($\sigma = 0.5$). However, the decrement in

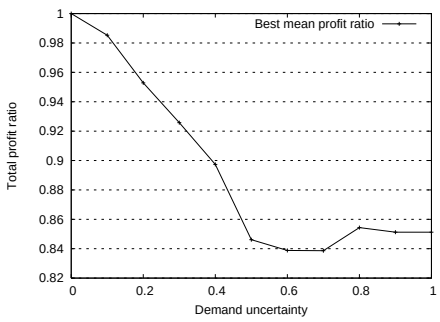

Fig. 1 Fluctuation Vs Profit

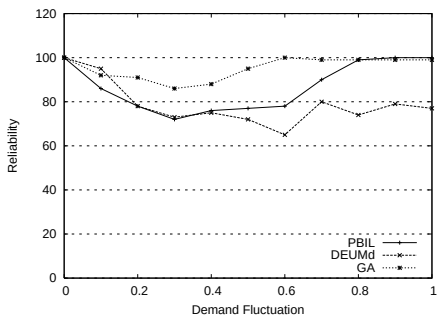

Fig. 2 Fluctuation Vs Reliability

profit slows as the fluctuation in demand gets very high, from 0.5 onwards till 1.0. Interestingly, the high fluctuation in demand results in slight increment in the total profit. This indicates that, in some cases, higher fluctuation in demand may not have extremely adverse effect to the profit in comparison to the low fluctuation.

c. Algorithm reliability vs. uncertainty: Another interesting observation that we make is the effect of fluctuation in demand have on the reliability of the results found by each algorithms. A typical illustration is given in Figure (2), which plots the *RL* found by each algorithm for long-term experiment 1 (the curves were similar for rest of the instances). It can be noticed that the *RL*. for the algorithms are very high (equal to 100%) when there is no fluctuation in the demand. Once the fluctuation is introduced the reliability decreases. However, interestingly, once the fluctuation gets very high, the reliability of the results starts increasing back to 100%. As can be seen from the figure, this is true for all three tested algorithms. This suggests that, when there is some fluctuation in demand, it is difficult to find a reliable policy that guarantees the profit, however if there is very high fluctuation in demand the found profit could be guaranteed. These results also supplement the

[5] Due to limited space we do not plot this graph for rest of the instances, though we note that the shape of the curves were similar

results on the impact of uncertainty on total profit where the profit slightly increases once demand fluctuation is very high. The explanation for this is as follows: when there is extreme fluctuation in demand with σ closer to 1, the fluctuating term Q_t in equation (2) is either twice the expected sales $E(Q_t)$ or equal to zero (from equation (3)).It is therefore easier for the algorithm to solve the problem with only two values for Q_t than when there are higher number of possible values for Q_t (in case of lower σ). This therefore increases the reliability of the algorithm.

5 Conclusion

In this paper, we used evolutionary algorithms for solving dynamic pricing problem in a stochastic setup. The model used in this paper is more realistic than the one used in [26] since different uncertainty is imposed to each individual periods. Our results show that GA is the most reliable algorithm for solving dynamic pricing in stochastic setup with implemented model, although similar to [26] we found that EDAs were better when there was no demand fluctuation. We also found that higher fluctuation in demand may not have adverse effect in comparison to lower fluctuation, and could result in increased reliability of the found pricing policy.

We note that the results found in this paper apply to the binary EAs. It would be interesting to see the performance of the real valued version of these algorithms on this problem. Also, further work should be done to theoretically justify these empirical results. This work is under way and interesting results are expected in the near future.

References

1. Baker, W., Marn, M.V., Zawada, C.: Price smarter on the net. Harvard Business Review **79** (2001)
2. Baluja, S.: Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning,. Tech. Rep. CMU-CS-94-163, Pittsburgh, PA (1994). URL citeseer.nj.nec.com/baluja94population.html
3. Bichler, M., Kalagnanam, J., Katircioglu, K., King, A.J., Lawrence, R.D., Lee, H.S., Lin, G.Y., Lu., Y.: Applications of flexible pricing in business-to-business electronic commerce. IBM Systems Journal **41(2)**, 287–302 (2002)
4. de Bonet, J.S., Isbell Jr., C.L., Viola, P.: MIMIC: Finding optima by estimating probability densities. In: M.C. Mozer, M.I. Jordan, T. Petsche (eds.) Advances in Neural Information Processing Systems, vol. 9. The MIT Press (1997). URL citeseer.nj.nec.com/debonet96mimic.html
5. Etxeberria, R., Larrañaga, P.: Global optimization using Bayesian networks. In: A. Ochoa, M.R. Soto, R. Santana (eds.) Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99), pp. 151–173. Havana, Cuba (1999)
6. Ferdows, K., Lewis, M.A., Machura, J.A.M.: Rapid-fire fulfilment. Harvard Business Review **82**, 104–110 (2004)
7. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley (1989)

8. Inza, I., Merino, M., Larrañaga, P., Quiroga, J., Sierra, B., Girala, M.: Feature subset selection by population-based incremental learning. A case study in the survival of cirrhotic patients with TIPS. Artificial Intelligence in Medicine (2001)
9. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science, Number 4598, 13 May 1983 **220, 4598**, 671–680 (1983). URL citeseer.ist.psu.edu/kirkpatrick83optimization.html
10. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers (2002)
11. McWilliams, G.: Lean machine: How dell fine-tunes its pc pricing to gain edge in slow market. Wall Street Journal (June 8, 2001)
12. Mitchell, M.: An Introduction To Genetic Algorithms. MIT Press, Cambridge, Massachusetts (1997)
13. Mühlenbein, H., Mahnig, T.: FDA - A scalable evolutionary algorithm for the optimization of additively decomposed functions. Evolutionary Computation **7**(4), 353–376 (1999). URL citeseer.nj.nec.com/uhlenbein99fda.html
14. Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions: I. binary parameters. In: H.M. Voigt, W. Ebeling, I. Rechenberg, H.P. Schwefel (eds.) Parallel Problem Solving from Nature – PPSN IV, pp. 178–187. Springer, Berlin (1996). URL citeseer.nj.nec.com/uehlenbein96from.html
15. Narahari, Y., Raju, C.V., Ravikumar, K., Shah, S.: Dynamic pricing models for electronic business. Sadhana **30**(part 2,3), 231–256 (April/June 2005)
16. Netessine, S., Shumsky, R.: Introduction to the theory and practice of yield management. INFORMS Transactions on Education **3**(1) (2002)
17. Owusu, G., Dorne, R., Voudouris, C., Lesaint, D.: Dynamic planner: A decision support tool for resource planning, applications and innovations in intelligent systems x. In: Proceedings of ES 2002, pp. 19–31 (2002)
18. Owusu, G., Voudouris, C., Kern, M., Garyfalos, A., Anim-Ansah, G., Virginas, B.: On Optimising Resource Planning in BT with FOS. In: Proceedings International Conference on Service Systems and Service Management (2006)
19. Parsopoulos, K., Vrahatis, M.: Particle swarm optimization method for constrained optimization problems. Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies, volume 76 of Frontiers in Artificial Intelligence and Applications pp. 214–220 (2002)
20. Pelikan, M., Goldberg, D.E., Cantú-Paz, E.: BOA: The Bayesian Optimization Algorithm. In: W. Banzhaf et al. (ed.) Proceedings of the Genetic and Evolutionary Computation Conference GECCO99, vol. I, pp. 525–532. Morgan Kaufmann Publishers, San Fransisco, CA (1999)
21. Pelikan, M., Mühlenbein, H.: The bivariate marginal distribution algorithm. In: R. Roy, T. Furuhashi, P.K. Chawdhry (eds.) Advances in Soft Computing - Engineering Design and Manufacturing, pp. 521–535. Springer-Verlag, London (1999)
22. Petrovski, A., Shakya, S., McCall, J.: Optimising cancer chemotherapy using an estimation of distribution algorithm and genetic algorithms. In: proceedings of Genetic and Evolutionary Computation COnference (GECCO 2006). ACM, seattle, USA (2006)
23. Phillips, R.: Pricing and revenue optimization. Stanford University Press (2005)
24. Sahay, A.: How to reap higher profits with dynamic pricing. MIT Sloan management review **48**, 53–60 (2007)
25. Shakya, S.: Deum: A framework for an estimation of distribution algorithm based on markov random fields. Ph.D. thesis, The Robert Gordon University, Aberdeen, UK (April 2006)
26. Shakya, S., Oliveira, F., Owusu, G.: An Application of EDA and GA to Dynamic Pricing. In: proceedings of Genetic and Evolutionary Computation COnference (GECCO2007), pp. 585–592. ACM, London, UK (2007)
27. Talluri, K., van Ryzin, G.: The Theory and Practice of Revenue Management. Springer, Berlin Heidelberg, New York (2004)
28. Voudouris, C., Owusu, G., Dorne, R., Ladde, C., Virginas, B.: Arms: An automated resource management system for british telecommunications plc. European Journal for Operational Research **171**, 951–961 (2006)