www.elsevier.com/locate/caor

A Non-Parametric Approach to Demand Forecasting in Revenue Management

Sh. Sharif Azadeh, P. Marcotte, G. Savard

PII: S0305-0548(15)00079-9
DOI: http://dx.doi.org/10.1016/j.cor.2015.03.015
Reference: CAOR3755

To appear in: *Computers & Operations Research*

A Non-Parametric Approach to Demand Forecasting in Revenue Management

Sh. Sharif Azadeh[a,*], P. Marcotte[b], G.Savard[a]

[a] *Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal Québec, Canada H3C 3A7*
[b] *Department of Computer Science and Operation Research, University of Montréal, Montréal Québec, Canada H3T 1J4*

Abstract

In revenue management, the profitability of the inventory and pricing decisions rests on the accuracy of demand forecasts. However, whenever a product is no longer available, true demand may differ from registered bookings, thus inducing a negative bias in the estimation figures, as well as an artificial increase in demand for substitute products. In order to address these issues, we propose an original Mixed Integer Nonlinear Program (MINLP) to estimate product utilities as well as capturing seasonal effects. This behavioral model solely rests on daily registered bookings and product availabilities. Its outputs are the product utilities and daily potential demands, together with the expected demand of each product within any given time interval. Those are obtained via a tailored algorithm that outperforms two well-known generic software for global optimization.

Keywords: Revenue management, Forecasting, Integer programming, Branch-and-bound, Heuristics.

1. Introduction

According to Cross (1997), Revenue Management (RM) is the research area that focuses on the study of disciplined tactics for making product availability and pricing decisions, with the aim of maximizing revenue growth. In

*Corresponding author
Email addresses: `shadi.sharifazadeh@polymtl.ca` (Sh. Sharif Azadeh), `marcotte@iro.umontreal.ca` (P. Marcotte), `gilles.Savard@poltml.ca` (G.Savard)

the service industry, this goal can only be achieved through accurate demand forecasting, which must take into account the volatility of product availabilities over the booking horizon. Clearly, registered bookings alone are not sufficient to depict the true demand. Indeed, as soon as a product reaches its capacity (*booking limit*), true demand is constrained (*censored*) and cannot be observed. Upcoming customers can then either switch to a higher fare product (*buy-up*), switch to a lower fare product (*buy-down*), or renege (*spill*). According to Weatherford and Belobaba (2002), ignoring the data censorship phenomenon can lead to demand underestimation ranging from 12.5% to 25%, and negatively affect revenue by 1% to 3%, a significant amount for major rail or airline operators.

Although unconstraining techniques may have a big impact on the success of revenue management systems, this topic has not been paid much attention in the literature. In general, two frameworks have been considered to deal with the issue: statistics and optimization. In this paper, we tackle the problem of demand modeling by the use of optimization techniques. However, first, in this section, we present the challenges and rewards of using statistical methods for uncensoring demand.

Based on the literature, Statistical techniques such as time series, exponential smoothing, or linear regression have mostly been considered in this context. All of these are able to include seasonal effects within their demand forecasts. Zeni (2001) and Queenan et al. (2009) have provided a comprehensive study of these methods, and have compared their respective impact on revenue. Their main drawback is that they cannot respond to sudden changes in customer behavior when a product becomes unavailable (see Sa (1987); Littlewood (2005); Pölt (2000); Weatherford (2000); Lee (1990)). On the other hand, there are many reseachers who have addressed the problem from optimization point of view (Escudero et al. (2013); Kunnumkal and Topaloglu (2010); Möller et al. (2008); Vulcano et al. (2012)). Actually, authors such as van Ryzin (2005) have claimed that revenue management systems should focus on customer behavior and choice probabilities, rather than blindly estimating demand from historical booking data.

Choice-based models were introduced by Andersson (1998), and analyzed by Talluri and van Ryzin (2004) and Vulcano et al. (2010) within the framework of discrete choice theory. In the latter two works, the parameters of the model have been estimated by maximum-likelihood techniques. In another research, Ratliff et al. (2008) have integrated historical demand data within a multi-flight heuristic procedure. Also, Vulcano et al. (2012) have applied customer choice models to the estimation of product primary demand (first-choice demand). In all the abovementioned optimization models, a parametric method of estimation (Expectation-Maximization, or EM in short) is used to estimate the parameters of the choice model, under demand independence assumptions. Although the approaches have been used for many years with some success, several issues still need to be addressed:

- Demand across fare products is not independent. Dealing with dependency yields a complex parameter estimation process that has been considered and tested by (Stefanescu (2009)) on small instances.
- As the proportion of censored demand in historical data grows, the accuracy of the standard estimation methods decreases (see Talluri and van Ryzin (2004); Vulcano et al. (2012); Haensel and Koole (2010)).
- Several statistical methods fail to accurately capture seasonal effects.
- Choice probabilities should enter the optimization process as variables, not as parameters to be estimated. Indeed, these probabilities depend on the set of products available within each time period.

All these issues have motivated us to develop a non-parametric and distribution-free estimation procedure that, based upon historical bookings, takes explicitly into account the set of available products. The contribution of this work is twofold. First, we formulate a model for minimizing the difference between estimated and registered bookings. In order to obtain a realistic representation of customer behaviour, cross-temporal utilities enter the model as variables, and seasonal effects are captured by classifying daily demand flows into a predefined number of clusters. Next, we formulate the problem as a MINLP (mixed-integer nonlinear program), for which we develop a semi-global optimization algorithm.

We close this introductory section with an outline of the paper's structure. Following the description of the problem, together with its underlying assumptions and mathematical formulation (Section 2), we provide a detailed description of the solution algorithm, including the node selection strategy and the valid inequalities used for enhancing the branch-and-bound framework (Section 3). Computational results on synthetic data are analyzed in Section 4, while the concluding Section 5 opens avenues for future research.

2. Problem formulation

To illustrate demand censorship, let us consider the two-product example involving the data displayed in Table 1. As soon as demand for product A exceeds its booking limit 35, which it does since true demand is equal to 40, the data collection system stops counting the number of upcoming customers. As a result, the real demand for A is censored and may exceed 35. In the present case, one A-customer switched to B, while the other 4 reneged.

Table 1: Demand censorship is placed here.

The main objective of our mathematical model is to minimize the difference between temporal registered bookings and their estimates. Let us introduce its main elements: a **product** i corresponds to a fare class offered at a given period[1], and is endowed with a utility u_i. The set of products available at a given period j is the **choice set** S_j. A **cluster** c denotes the set of periods that share common features based on the demand flow, such as weekdays, weekends, holidays, etc. Each **daily potential demand** d_j is associated with a unique cluster. For given utilities u_i and choice sets S_j the **choice probability** p_{ij} of selecting product i on day j is computed according to the multinomial logit (MNL) formula (Liu and van Ryzin (2008)):

$$p_{ij}(S_j, u_i) = \begin{cases} \exp(u_i)/(\sum\limits_{k \in S_j} \exp(u_k) + \exp(u_0)) & \text{if } i \in S_j \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where u_0 represents the utility of the no-choice option.

For a given time horizon, $d_1, d_2, \ldots, d_{|J|}$ and a set of products I, we wish to minimize the discrepancy e_{ij} between the expected bookings w_{ij} of each available product i at a given day j and its associated observed registered booking O_{ij}, thus simultaneously capturing seasonal effects and customer behavior. We will therefore have achieved the three following goals:

- external segmentation (classification of days within clusters);
- estimation of daily potential demand;
- estimation of product utilities.

[1] Throughout, the terms 'time period' and 'day' are used interchangeably.

A summary of the notation used in the model is displayed in Table 2 and Table 3.

Table 2 and Table 3: Summary of notations for MINLP and RELAX model are placed here.

The objective of the model is to minimize the difference between estimated and observed reservations, through the estimation of potential demand, product utilities, and cluster membership. This is achieved by solving the following mathematical model:

$$\text{MINLP:} \quad \min_{\delta_c,u,z,d_j} \quad \sum_{j\in J}\sum_{i\in S_j}\left(p_{ij}(S_j,u_i)d_jA_{ij}-O_{ij}\right)^2 \tag{2}$$

$$\text{subject to} \quad p_{ij}(S_j,u_i)=\frac{\exp(u_i)}{\sum_{k\in S_j}\exp(u_k)+\exp(u_0)} \qquad i\in S_j \tag{3}$$

$$d_j=\sum_c \delta_c z_{jc} \qquad j\in J \tag{4}$$

$$\sum_{c\in C} z_{jc}=1 \qquad j\in J \tag{5}$$

$$z_{jc}\in\{0,1\} \qquad j\in J, c\in C. \tag{6}$$

In the objective function of the model, predicted booking for product i is set to the product of the relevant choice probability, potential demand of day j (In constraint (4), for algorithmic purposes, d_j is defined as an integer variable) and products availability status.

Solving the above mentioned problem is challenging as there are several sources of nonconvexity and nonlinearity in this model: First, the product of choice probabilities and daily potential demand, second the fractional presentation of choice probabilities and finally the existence of class membership binary variables. In the following section, we will address all these challenges and we will introduce a linearized model.

3. Algorithmic framework

While the mathematical formulation of the problem is concise, its numerical resolution is challenging, due to both its combinatorial and nonlinear (fractional or multiplicative) nature. As we will observe later in the paper, it is not amenable to solution by global optimization software, its continuous relaxation being itself a difficult nonconvex program.

The line of attack that we have pursued is based on an approximate mixed integer *linear* reformulation, which was strengthened by valid inequalities. Provided with an appropriate initial solution, and through the application of efficient branching rules, quasi-optimal solutions could be obtained from an off-the-shelf MIP software such as CPLEX. The key elements of the algorithmic framework are presented in more details as follows:

General Solution Framework

Input: Registered bookings O_{ij}, set of available products, S_j

Output: Daily demand flows d_j, cluster memberships z_{jc}, utilities u_i

(1) Transformation into a MIP
- i: Linearization
- ii: Relaxation
- iii: Convexification

(2) Preprocessing at root node
- iv: Valid inequalities
- v: Initial solution
- vi: Domain reduction

(3) Branch-and-bound
- vii: Branching strategy
- viii: Adjustment of bounds at branching nodes

3.1. Linearization

Let us assume that the potential demand for cluster c lies within predetermined bounds, i.e., $\delta_c \in [R_c^L, R_c^U]$, and let us introduce the normalized variable

$$r_c^N = \delta_c / R_c^U \in [R_c^L / R_c^U, 1]. \tag{7}$$

Equation (4) then takes the form

$$d_j = \sum_c (r_c^N R_c^U) z_{jc} \tag{8}$$

and, upon the change of variable

$$d_{jc}^N = r_c^N z_{jc}, \tag{9}$$

one derives the linear equation

$$d_j = \sum_c d_{jc}^N R_c^U. \tag{10}$$

Using the fact that z_{jc} is binary-valued, Equation (9) can be linearized. Indeed, if $z_{jc} = 1$, then $d_{jc}^N = r_c^N$. Since $z_{jc} \leq 1$, we have that

$$z_{jc} + d_{jc}^N \leq r_c^N + 1 \tag{11}$$

and

$$z_{jc} + r_c^N \leq d_{jc}^N + 1 \tag{12}$$

If $z_{jc} = 0$, then $d_{jc}^N = 0$, i.e.,

$$d_{jc}^N \leq z_{jc} \qquad i \in I, j \in J. \tag{13}$$

To tighten the feasible domain, the lower bound R_c^L on cluster demand d_j has been set to the minimum value of daily cumulative registered reservations.

3.2. Relaxation and convexification

Three sources of nonconvexity occur in the MINLP mathematical model:

- the choice probabilities, p_{ij} derived from MNL model involve a fractional term;
- the estimated demand for a given product on a given day involves the bilinear term ($w_{ij} = p_{ij} d_j$);
- variables z_{jc} are binary-valued.

To deal with the first source of nonconvexity, we base our relaxation on choice probabilities p_{ij} (versus utilities), we approximate the bilinear terms w_{ij} by their convex envelopes, and resort to classical continuous relaxations for the binary variables z_{jc}.

Note that substituting the independent choice probability variables p_{ij} to the utilities u_i in the original problem may induce infeasibilities. To address the issue we check that, for each product i and day j, the inequality

$$P_{ij}^L \leq p_{ij} = \frac{\exp(u_i)}{\sum\limits_{k \in S_j} \exp(u_k) + \exp(u_0)} \leq P_{ij}^U \qquad i \in I, j \in J \tag{14}$$

holds in the MINLP model solved by a nonlinear solver.

There are several ways to linearize products. Whenever bilinear terms involve integer variables, this can be achieved by performing a sequence of Fortet inequalities to obtain a Mixed Integer Linear program (MILP) (Costa and Liberti (2012); Belotti et al. (2009)). Alternatively, if variables are continuous, one must resort to solution techniques for nonconvex programs, such as McCormick inequalities, where a linear approximation of the bilinear terms involves additional variables and constraints (McCormick (1976)). Therefore, in our case, to deal with the bilinear term $w_{ij} = p_{ij}d_j$, we contrast the concave and convex envelopes of these functions against the relaxations. Each bilinear term is relaxed independently. Making use of the bounds

$$\mathcal{E} = \left\{ w_{ij} = p_{ij}d_j \in [P_{ij}^L, P_{ij}^U] \times [D_j^L, D_j^U] \times R \right\}, \tag{15}$$

we locally convexify the bilinear term based on the following inequalities, which are only valid if product i is available on day j, i.e., $A_{ij} = 1$:

$$\begin{aligned}
w_{ij} &\geq D_j^U p_{ij} + P_{ij}^L d_j - P_{ij}^L D_j^U && i \in I, j \in J \\
w_{ij} &\geq D_j^L p_{ij} + P_{ij}^L d_j - D_{ij}^L P_{ij}^L && i \in I, j \in J \\
w_{ij} &\leq D_j^U p_{ij} + P_{ij}^L d_j - P_{ij}^L D_j^U && i \in I, j \in J \\
w_{ij} &\leq D_j^L p_{ij} + P_{ij}^U d_j - P_{ij}^U D_j^L && i \in I, j \in J.
\end{aligned}$$

We then iteratively update, for each subproblem and at each node of the enumeration tree, the upper and lower bounds of the choice probabilities of available products. This yields the convex quadratic program

$$
\begin{aligned}
\text{RELAX:} \quad & \min_{p,d,z} \sum_{i \in I} \sum_{j \in J} e_{ij}^2 && && (16)\\
& w_{ij} A_{ij} - O_{ij} = e_{ij} && i \in I, j \in J \\
& \\
& P_{ij}^U d_j + D_j^U p_{ij} - P_{ij}^U D_j^U \leq w_{ij} && i \in I, j \in J \\
& P_{ij}^L d_j + D_j^L p_{ij} - P_{ij}^L D_j^L \leq w_{ij} && i \in I, j \in J \\
& P_{ij}^U d_j + D_j^L p_{ij} - P_{ij}^U D_j^L \geq w_{ij} && i \in I, j \in J \\
& P_{ij}^L d_j + D_j^U p_{ij} - P_{ij}^L D_j^U \geq w_{ij} && i \in I, j \in J \\
& \\
& d_{jc}^N \leq z_{jc} && j \in J, c \in C \\
& z_{jc} + r_c^N \leq d_{jc}^N + 1 && j \in J, c \in C \\
& z_{jc} + d_{jc}^N \leq r_c^N + 1 && j \in J, c \in C \\
& d_j = \sum_{c \in C} d_{jc}^N R_c^U && j \in J \\
& R_c^L / R_c^U \leq r_c^N && c \in C \\
& \sum_{c \in C} z_{jc} = 1 && j \in J \\
& 0 \leq z_{jc} \leq 1 && j \in J, c \in C
\end{aligned}
$$

where the second to fifth constraints express the McCormick inequalities of the bilinear terms, while the next six constraints assign each day to a specific cluster.

4. Solution algorithm

The algorithm for globally solving the original problem is a branch-and-bound based on RELAX, and where branching is performed with respect to the integer variables δ_c, the integer variables d_j, as well as the continuous variables p_{ij}. While a linear solver is put to contribution for the first two sets of variables, a nonlinear solver is required for computing the choice probabilities. We now provide a detailed description of the main elements of the algorithm.

4.1. Preprocessing

The performance of the enumeration scheme can be greatly enhanced through three procedures: introduction of valid inequalities at the root node, warm-starting the algorithm with a feasible solution provided by a heuristic algorithm, and tightening the feasible domain at each node of the branch-and-bound tree.

4.1.1. Valid inequalities

Based on the set of available products, one can derive logical relations that must be satisfied by any optimal solution. In general, considering the choice sets of two separate days, S_j and $S_{j'}$, three cases may happen: (i) the choice probabilities of two products are equal, $p_{ij} = p_{ij'}$ (ii) one of them is less than the other one, $p_{ij} > p_{ij'}$ (iii) we cannot establish a logical relation between two probabilities.

Valid inequality 1. It is a property of the multinomial logit that, if the choice set of day j is a subset of the choice set of day j', that is $S_j \subseteq S_{j'}$, then we have

$$p_{ij} = \frac{\exp(u_i)}{\sum\limits_{k \in S_j} \exp(u_k) a_{kj} + \exp(u_0)} \geq \frac{\exp(u_i)}{\sum\limits_{k \in S_{j'}} \exp(u_k) a_{kj'} + \exp(u_0)} = p_{ij'} \quad (17)$$

Valid inequality 2. In order to discard symmetric and equivalent solutions we order, without loss of generality, the demands associated with the cluster indices, i.e.,

$$\delta_1 < \delta_2 \leq \ldots \leq \delta_k \leq \ldots \leq \delta_{|C|}. \quad (18)$$

Equivalently:

$$r_1^N R_1^U \leq r_2^N R_2^U \leq \ldots \leq r_k^N R_k^U \leq \ldots \leq r_{|C|}^N R_{|C|}^U. \quad (19)$$

4.1.2. Initial solution

At the root node, initially, we find estimated daily potential demand, d_j, by solving RELAX problem. Then, an integer initial solution is obtained via a K-nearest neighbor algorithm to fix class membership variables, z_{jc} (Cover and Hart (1967);Dudani (1976)).

First, one matches each day to its own cluster. Then, one iteratively merges the two clusters having the closest averages, until the required number of clusters is attained. Since ties are broken arbitrarily, different choices could yield different partitions of the set of days into clusters. Table 4 shows the progression of the algorithm corresponding to the vector of daily potential demands $\{36, 6, 30, 14, 42\}$, and a number of final clusters set to two. In this example, the same solution would have been achieved if 36 and 42 had been merged at the first iteration. Of course, this result does not hold in general, as can be readily verified on the demand vector $\{1, 3, 5\}$ with two clusters yielding either the partition $\{1, 3\}$ $\{5\}$ or $\{1\}$ $\{3, 5\}$.

By using fixed z_{jc}s, we again solve the RELAX model. As soon as the solution of RELAX assumes integer values for δ_c , i.e., all days are assigned to one of the clusters, we switch to MINLP, which is solved by the nonlinear solver to find product utilities u_i.

Table 4: Clustering algorithm for determining initial solution is placed here.

4.1.3. Domain reduction

Prior to branching, the respective ranges of the variables w_{ij}, p_{ij} and d_j can be tightened. For example, the sum of registered bookings on a given day d_j provides the lower bound

$$d_j \geq D_j^L = \sum_i O_{ij} \qquad j \in J. \tag{20}$$

When $\mathcal{F}^*$ shows the best integer solution, an upper bound $D_j^U(0)$ on d_j can be set to the optimum of the convex optimization problem

$$\begin{aligned} \max_d \quad & d_j \\ \text{subject to} \quad & \sum_i \sum_j e_{ij}^2 \leq \mathcal{F}^* \qquad i \in I, j \in J \\ & \text{constraints of RELAX.} \end{aligned}$$

In a similar fashion, upper bounds $P_{ij}^U(0)$ on the choice probabilities p_{ij}

are obtained by solving the convex program

$$\begin{aligned} \max_{p} \quad & p_{ij} \\ \text{subject to} \quad & \sum_{i}\sum_{j} e_{ij}^2 \leq \mathcal{F}^* \qquad i \in I, j \in J \\ & \text{constraints of RELAX.} \end{aligned}$$

A total of $2|I|+2|I||J|$ optimization problems are solved to derive the above upper bounds. Finally, it follows from the inequality

$$\sum_{i\in I}\sum_{j\in J} e_{ij}^2 \leq \mathcal{F}^*$$

that w_{ij} can be upper bounded by $\sqrt{\mathcal{F}^*} + O_{ij}$.

4.2. Branch-and-bound

The optimum of the relaxed program provides a lower bound on the true optimal value, while the corresponding solution can be used to construct a feasible solution that yields an upper bound on the optimum. Note that the performance of the partial enumeration process rests in large part on the quality of the upper bounds on the variables, hence the importance of tightening these.

At each node of the enumeration tree, we implement a series of range reductions with respect to daily potential demand, choice probabilities, potential demand of each cluster $R_c^L < \delta_c < R_c^U$, and the bilinear term w_{ij}. Several techniques, such as interval arithmetic, have been implemented. For instance one can fix the value of z_{jc} without branching. Indeed, if for a given node n, the set $[D_j^L(n), D_j^U(n)] \cap [R_c^L(n), R_c^U(n)]$ is empty, then z_{jc} must be zero.

For node n, the bounds on cluster demand δ_c can be set to

$$\begin{aligned} R_c^L(n) &= \max\left\{R_c^L(n), \min_j D_j^L(n)\right\} \\ R_c^U(n) &= \min\left\{R_c^U(n), \max_j D_j^U(n)\right\}. \end{aligned}$$

The lower bound can be updated according to the formula

$$D_j^L(n) = \max\left\{ D_j^L(n), \min\left\{ \frac{w_{ij}^U(n)}{P_{ij}^L(n)}, \frac{w_{ij}^U(n)}{P_{ij}^U(n)}, \frac{w_{ij}^L(n)}{P_{ij}^L(n)}, \frac{w_{ij}^L(n)}{P_{ij}^U(n)} \right\}\right\} \quad i \in I, j \in J \tag{21}$$

The upper bounds $D_j^U(n)$ on daily potential demand, as well as the bounds on choice probabilities are updated in a similar fashion. Next, we adjust the upper and lower bounds of daily potential demand d_j, and potential demand of each cluster δ_c. Meanwhile, we fix the value of assignment variables z_{jc}, whenever the ranges of d_j and δ_c intersect.

For a given node, if the range of a variable d_j obtained from the relaxation model overlaps with the ranges of two or more clusters, then its lower bound is updated to

$$D_j^L(n) = \max\left\{R_c^L(n), D_j^L(n)\right\}. \tag{22}$$

Similarly, the upper bound is set to

$$D_j^U(n) = \min\left\{R_c^U(n), D_j^U(n)\right\}. \tag{23}$$

Finally, feasibility conditions are verified by using (14) and (19). In addition, the solution of RELAX and MINLP problems are used to prune the partial enumeration tree.

We close this section with a description of the branching strategy. At node n, the binary variables z_{jc} are relaxed, and d_j can therefore 'partially' belong to more than one class. Let $\hat{d}_j(n)$ be the estimated potential demand of day j obtained from optimal solution of the RELAX problem. Let

$$I_c(\hat{d}_j(n)) = \begin{cases} 1 & \text{if} \quad \hat{d}_j(n) \in [R_c^L(n), R_c^U(n)] \qquad c \in C \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

$$I(n) = \sum_{j \in J} I_c(\hat{d}_j(n)) \qquad c \in C. \tag{25}$$

In the branching scheme, **node selection** follows these rules:

- Branch on the node from which the relaxed optimum is minimal.
- In case of a tie, branch on the deepest node.

- In case of yet another tie (this rarely occurs in practice), branch on any node having the maximum number of overlapping intervals with respect to variables d_j and δ_c, i.e., $[D_j^L(n), D_j^U(n)] \cap [R_c^L(n), R_c^U(n)] \neq \emptyset$.

As far as **variable selection** is concerned, we prioritize the cluster demand variables δ_c for branching, but switch to daily potential demand d_j when all clusters are disjoint. A variable δ_c is selected if it achieves maximum interval length, ties being broken in favor of clusters with large $I(n)$-values in RELAX. The assignment of each day to a single disjunctive cluster is achieved by branching on d_j. To reduce each cluster to a singleton, we branch again on δ_c. As mentioned above, we branch on d_j to fix z_{jc}, and select variables for which the difference between lower and upper bounds is the largest.

Once all days have been assigned to clusters and δ_c is integer-valued, we solve MINLP to find product utilities u_i. Finally, if the gap between the RELAX and MINLP solutions is larger than a predefined threshold, we branch on variables p_{ij}, with the aim of either fathoming the current node or obtain a better feasible solution.

5. Computational results

The algorithm has been tested on a number of synthetic instances, and its performance assessed with respect to three criteria:

- **Calibration**: this criterion is used to verify whether the algorithm is able to recover exactly the data used to generate the synthetic instances, i.e., achieves a zero objective for MINLP.

- **Classification:** this criterion is used to determine the error level achieved by the algorithm on perturbed instances, and also to compare the performance of the algorithm against two well-known global and nonlinear solvers.

- **Generalization:** this criterion is used to assess the robustness of the estimation process, i.e., verifying how well the parameters calibrated on a set of controlled instances can generalize to distinct perturbed datasets, thus constituting a reliable tool for decision making.

In this paper, the main focus is on the algorithmic comparison with existing solvers, however, the impact of the choice of parametric or non-parametric demand models on revenue has been addressed in a separate article by the authors (Azadeh et al. (2014)).

5.1. Data generation

Each instance is characterized by a triple (C, J, I) where C denotes the number of clusters (2, 3, or 4), J the number of days (7, 14, 21, or 28) and I the number of products (4, 6, or 8). Observed bookings (O_{ij}) have been generated according to the formula

$$O_{ij} = A_{ij}p_{ij}dj, \tag{26}$$

which requires knowledge of the set of available products, as well as the utilities u_i from which the probabilities p_{ij} are derived. In this process, the product utilities and the potential demand δ_c of each cluster are exogenous. The availability parameters A_{ij} associated with product i on a given day j are generated according to a Bernoulli random variable. Finally, each day j has been randomly assigned to one of the clusters.

A first set of 33 unperturbed instances allowed to check whether the algorithm could actually replicate the original values z_{jc}, d_j and u_i. Next, a second set of 33 perturbed samples were created to test the generalization ability of the model. Keeping the other parameters (choice set, potential cluster demand, class membership, product utilities) fixed, the daily demand was modified according to the formula

$$d_j = d_j(1 + \gamma(2\epsilon_j - 1)) \qquad j \in J, \tag{27}$$

where the perturbation parameter γ was fixed to 0.1, and ϵ_j was uniformly distributed between 0 and 1.

The outcomes of our proposed model have been compared with those of two of the most acknowledged softwares: Knitro 8 (Ziena (2013)), a nonlinear solver (IPOPT (2013)), and Baron 11.0, a global optimization solver (BARON (2013)). Our algorithm has been halted whenever no improvement occurred within 60 minutes of CPU time. The computational experiments have been carried out on a Quad-core computer with 2.4 GHz CPU and 8 GB of RAM. The branching algorithm was implemented in C++, and we resorted to the Quadratic Solver of CPLEX 12.3 (sequential quadratic programming) and IPOPT 3.11 (interior point method) as nonlinear solvers.

The software Baron was accessed through the NEOS server (see IBM (2013), NEOS (2013)).

5.2. Numerical experiments

The non-perturbed instances used for calibration purposes are displayed in Table 5. For all instances, the global optimum with zero value was reached. Moreover, the algorithm was able to reproduce the exact original product utilities u_i and cluster potential demand δ_c from which the data was initially generated.

Table 5: Non-Perturbed instances is placed here.

The numerical results corresponding to the 33 perturbed instances are summarized in Table 6. The first three columns describe instances and their characteristics: number of clusters, number of days and number of products. The four 'Time' columns contain execution time (CPU time in seconds) of different parts of the algorithm: 'Total' (sum of the next two columns plus the time spent to implement branching strategy), 'Relax' (time spent solving the RELAX model), 'NLP' (time spent solving MINLP using IPOPT), 'Pre-Proc.' (time spent implementing the pre-processing at root node). Cases where the run time is significantly less than 3 600 seconds attested to the efficiency of the branching strategy.

The next four columns under 'Node' provide statistics related to the branch-and-bound tree. The first column 'Gen.' represents the total number of nodes generated during the branch-and-bound procedure. Although reasonable for small instances, it increases quickly with the number of products and clusters. Column 'Br.' represents the number of branched nodes, which is significantly lowered by implementing the feasibility conditions and valid inequalities. The caption 'Dis.' refers to the number of nodes that have been discarded during the branching process, through the violation of the feasibility conditions (14) and (19). The heading 'Domin.' refers to the number of nodes dominated by the current best solution. Data in these two columns attest to the efficiency of the algorithm, more precisely to the large number of subtrees that could be pruned.

Column '# NLP' refers to the number of times the algorithm resorted to IPOPT for solving MINLP, i.e., the number of times the relaxation problem reached an integer feasible solution. Numbers under the heading 'VI#1' correspond to the number of valid inequalities appended to the model, and thus are a measure of the contribution of constraint (17).

The next three columns show the initial solution 'Ini Sol' obtained from a variant of K-nearest neighbor algorithm, the best feasible solution 'Best' and the best bound 'Bound'. They illustrate the sharp improvement of the initial solution. The iterative process halts if one of two conditions holds: either the gap between the best solution (MINLP) and the best bound (RELAX) is less than 1%, or the algorithm makes no improvement for a period of 3 600 seconds. The 'Gap', set to the value (Best bound-Best solution)/100, is displayed in the last column.

Figure 1: Algorithm effectiveness in reducing the gap is placed here.

In Figure 1, for an instance involving three classes, 28 days and four products, we illustrate the effectiveness of the algorithm by plotting the best bound and the best solution against CPU time. The Figure is logarithmically scaled for the ease of presentation. The final gap between MINLP and RELAX models for this example is equal to 0.98%.

In Table 7, we contrast the performance of our algorithm against those of Knitro and Baron, on the 33 perturbed instances of Table 6. The first three columns specify the dimensions of the instances. The next three under 'BB Sol' show the outcomes of our demand model, initialized with 'Ini Sol', the prediction error (objective function or calibration error) 'MSE', and the classification error 'Class.%'. The latter indicates whether each day has been properly assigned to one of the clusters, based on its demand flow. This information helps us to accurately predict demand seasonal features. We observe that the algorithm has been successful in correctly assigning each day to one of the clusters. From a theoretical point of view, small gaps between the solutions of MINLP and RELAX, as well as null classification errors, testify to the effectiveness of the algorithm.

Under the heading 'Baron', the next two columns show the prediction ('MSE') and classification ('Class') errors associated with the global optimizer Baron. While Baron is efficient on small instances, it is highly sensitive to the size of the problem. It actually fails to solve problems involving more than 14 days and four clusters.

Likewise, the two columns under 'Knitro' correspond to prediction and classification errors for that solver. Once again, this solver successfully classifies days into clusters for small instances, with prediction errors similar to those of Baron and our algorithm. Besides size (based on the number of clusters, days and products), Knitro is also sensitive to product availabilities. When the number of available products for a given day decreases, the values of MSE and classification error sharply increase.

The figures in the last column ('Generalization') of Table 7 illustrate the good performance of our algorithm on perturbed data. This stability actually depends on the availability of the products. In some cases, the error is slightly lower than the calibration error (MSE), due to the number of available products being small compared to the number used for calibrating the model. In all cases, the proposed algorithm outperformed by a large margin both Knitro and Baron.

Table 6: Perturbed instances is placed here.

Table 7: Comparison framework is placed here.

6. Conclusion

In revenue management, dealing with censored demand is a complex issue, especially when a large proportion of products are not available to customers, a situation that occurs in practice. To address this problem, we have proposed a choice-based non-parametric method whose output is the demand for any given product within any given time period, and proposed for its solution an efficient algorithm that has been validated on synthetic data. Simultaneously, we obtained a variety of solutions for class potential demands, product utilities and daily assignments, depending on the features of historical data. The size of the real-life problems is of course context dependent. For example, in high speed rail and airlines networks, the number of alternatives could reach 15 to 20. As for clusters, their number and respective widths should be set such that data scarcity is avoided.

Through the introduction of exogenous customer segmentation into the mathematical model, or through the implementation of a revenue maximization model, these sets could be reduced, and a small number of scenarios retained. In a future work, we intend to investigate situations involving more sophisticated segmentation models, with the aim of accurately estimating the probabilities of buy-ups or buy-downs.

7. References

Andersson, S., 1998. Passenger choice analysis for seat capacity control: A pilot project in Scandinavian Airlines. International Transactions in Operational Research 5, 471–486.

Azadeh, S.S., Hosseinalifam, M., Savard, G., 2014. The impact of customer behavior models on revenue management systems. Computational Management Science , 1–11.

BARON, 2013. Baron. http://www.gams.com/dd/docs/solvers/baron.pdf.

Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A., 2009. Branching and bounds tighteningtechniques for non-convex minlp. Optimization Methods & Software 24, 597–634.

Costa, A., Liberti, L., 2012. Relaxations of multilinear convex envelopes: dual is better than primal, in: Experimental Algorithms. Springer, pp. 87–98.

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. Information Theory, IEEE Transactions on 13, 21–27.

Cross, R.G., 1997. Revenue management: hard-core tactics for market domination. Universities Press.

Dudani, S.A., 1976. The distance-weighted k-nearest-neighbor rule. Systems, Man and Cybernetics, IEEE Transactions on , 325–327.

Escudero, L.F., Monge, J.F., Morales, D.R., Wang, J., 2013. Expected future value decomposition based bid price generation for large-scale network revenue management. Transportation Science 47, 181–197.

Haensel, A., Koole, G., 2010. Estimating unconstrained demand rate functions using customer choice sets. Journal of Revenue & Pricing Management 10, 438–454.

IBM, 2013. Cplex. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

IPOPT, 2013. https://projects.coin-or.org/Ipopt.

Kunnumkal, S., Topaloglu, H., 2010. A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior. Production and Operations Management 19, 575–590.

Lee, A., 1990. Airline reservations forecasting: Probabilistic and statistical models of the booking process. Technical Report. Flight Transportation Laboratory, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology.

Littlewood, K., 2005. Special issue papers: Forecasting and control of passenger bookings. Journal of Revenue & Pricing Management 4, 111–123.

Liu, Q., van Ryzin, G., 2008. On the choice-based linear programming model for network revenue management. Manufacturing & Service Operations Management 10, 288–310.

McCormick, G., 1976. Computability of global solutions to factorable nonconvex programs Convex underestimating problems. Mathematical Programming 10, 147–175.

Möller, A., Römisch, W., Weber, K., 2008. Airline network revenue management by multistage stochastic programming. Computational Management Science 5, 355–377.

NEOS, 2013. http://www.neos-server.org/neos/.

Pölt, S., 2000. From bookings to demand: the process of unconstraining. AGIFORS Reservations and Yield Management Study Group, New York .

Queenan, C., Ferguson, M., Higbie, J., Kapoor, R., 2009. A comparison of unconstraining methods to improve revenue management systems. Production and Operations Management 16, 729–746.

Ratliff, R., Rao, B., Narayan, C., Yellepeddi, K., 2008. A multi-flight recapture heuristic for estimating unconstrained demand from airline bookings. Journal of Revenue & Pricing Management 7, 153–171.

van Ryzin, G., 2005. Future of revenue management: models of demand. Journal of Revenue & Pricing Management 4, 204–210.

Sa, J., 1987. Reservations forecasting in airline yield management. Technical Report. Cambridge, MA: Flight Transportation Laboratory, Massachusetts Institute of Technology.

Stefanescu, C., 2009. Multivariate customer demand: modeling and estimation from censored sales. Technical Report. London Business School.

Talluri, K., van Ryzin, G., 2004. Revenue management under a general discrete choice model of consumer behavior. Management Science 50, 15–33.

Vulcano, G., van Ryzin, G., Chaar, W., 2010. Om practice-choice-based revenue management: An empirical study of estimation and optimization. Manufacturing & Service Operations Management 12, 371–392.

Vulcano, G., Van Ryzin, G., Ratliff, R., 2012. Estimating primary demand for substitutable products from sales transaction data. Operations Research 60, 313–334.

Weatherford, L., 2000. Unconstraining methods, in: AGIFORS Reservations and Yield Management Study Group.

Weatherford, L., Belobaba, P., 2002. Revenue impacts of fare input and demand forecast accuracy in airline yield management. Journal of the Operational Research Society 52, 811–821.

Zeni, R., 2001. Improved forecast accuracy in airline revenue management by unconstraining demand estimates from censored data. Ph.D. thesis. MIT.

Ziena, 2013. Knitro. `http://www.ziena.com/docs/Knitro70_UserManual.pdf`.

Table 1: Demand censorship

Product	A	B
Availability status	yes	yes
Observed demand	35	5
Booking limit	35	6
Real demand	40	4

Table 2: Summary of notations (MINLP)

sets	
Product	$i \in I = \{1, \ldots, \lvert I \rvert\}$
Day	$j \in J = \{1, \ldots, \lvert J \rvert\}$
Cluster	$c \in C = \{1, \ldots, \lvert C \rvert\}$
Choice set	S_j , set of products available on day j
parameters	
O_{ij}	observed bookings for product $i \in I$ on day $j \in J$
A_{ij}	availability status of product $i \in I$ on day $j \in J$
variables	
d_j	daily potential demand (integer)
p_{ij}	probability of selecting product $i \in I$ on day $j \in J$
z_{jc}	cluster membership variable (binary)
u_i	utility of product i
δ_c	potential demand of cluster c

Table 3: Additional summary of notations (RELAX)

parameters	
R_c^U	upper bound on potential demand for cluster $c \in C$
R_c^L	lower bound on potential demand for cluster $c \in C$
D_j^U	upper bound on potential demand on day $j \in J$
D_j^L	lower bound on potential demand on day $j \in J$
P_{ij}^U	upper bound on choice probability for product $i \in I$ on day $j \in J$
P_{ij}^L	lower bound on choice probability of product $i \in I$ on day $j \in J$
variables	
e_{ij}	difference between estimated demand w_{ij} and observed bookings O_{ij}
w_{ij}	expected demand for product $i \in I$ on day $j \in J$
d_{jc}^N	normalized daily potential demand $\in [0, 1]$
r_c^N	normalized potential demand for each cluster $\in [0, 1]$

Table 4: Clustering algorithm for determining initial solution

iteration						
0	clusters	$\{36\}$	$\{6\}$	$\{30\}$	$\{14\}$	$\{42\}$
	averages	36	6	30	14	42
1	clusters	$\{36, 30\}$	$\{6\}$	$\{14\}$	$\{42\}$	
	averages	33	6	14	42	
2	clusters	$\{36, 30, 42\}$	$\{6\}$	$\{14\}$		
	averages	36	6	14		
3	clusters	$\{36, 30, 42\}$	$\{6, 14\}$			
	averages	36	10			

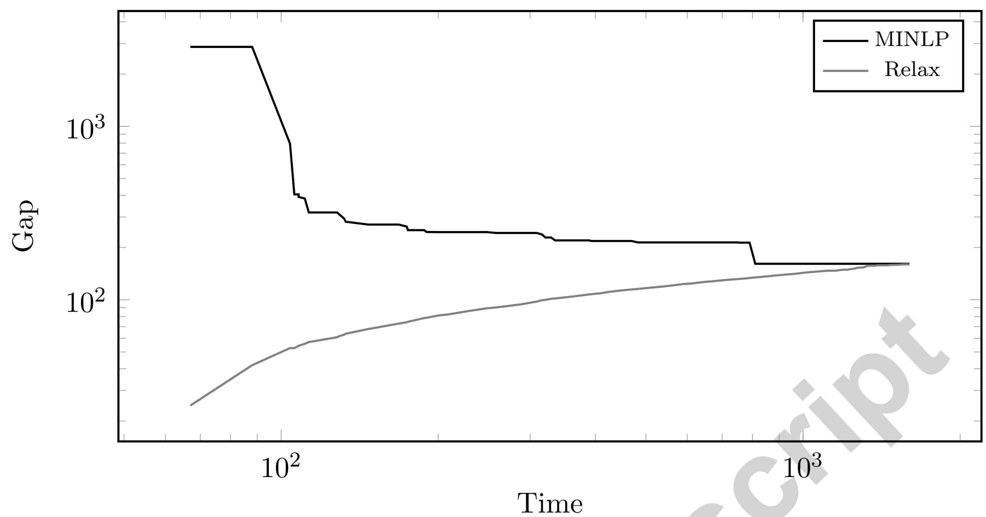

Figure 1: Algorithm effectiveness in reducing the gap

Table 5: Non-Perturbed instances

Class	Days	Product	Time				Node				#IPOPT	VI#1	Ini Sol.
			Total	Relax	NLP	Pre-Proc.	Gen.	Br.	Dis.	Dom.			
		4	1.38	0.94	0.02	0.73	221	81	14	4	2	12	172.87
	7	6	0.28	0.28	0.00	1.98	76	30	46	0	2	32	419.99
		8	19.30	10.84	0.00	1.69	2367	1034	73	53	54	5	341.65
		4	3.00	1.72	0.00	3.27	339	135	12	32	4	67	276.54
	14	6	3.84	2.52	0.00	6.39	301	117	9	4	4	75	369.85
		8	9.80	5.09	0.06	9.47	657	457	6	178	179	54	1684.91
2		4	0.70	0.25	0.02	7.88	55	21	8	5	1	138	71.74
	21	6	32.27	17.17	0.00	12.75	1461	655	12	12	13	134	449.93
		8	5.92	2.98	0.00	21.72	242	109	19	3	4	116	1364.69
		4	4.19	2.31	0.00	13.69	179	79	6	27	3	218	314.34
	28	6	11.50	5.88	0.00	27.05	329	138	6	37	3	227	178.34
		8	45.27	20.44	0.00	136.34	1022	420	21	0	1	283	2720.01
		4	3.94	2.42	0.03	0.66	625	227	12	13	13	14	136.63
	7	6	82.34	1.72	0.13	2.56	1100	800	213	32	126	23	534.00
		8	17.42	9.88	0.05	2.48	1620	708	34	58	59	19	794.63
		4	2.45	1.44	0.00	3.55	189	77	6	0	1	80	484.42
	14	6	37.36	17.33	0.45	2.20	4026	2284	24	633	634	16	830.42
3		8	10.14	5.33	0.06	2.48	1313	615	50	106	107	7	896.69
		4	5.50	3.05	0.00	8.97	338	134	17	4	1	152	350.19
	21	6	14.00	7.27	0.00	14.55	594	236	19	0	1	139	912.96
		8	155.17	84.11	0.00	34.39	4385	1667	240	5	6	163	2956.55
		4	28.05	12.86	0.02	14.52	1102	453	24	1	2	233	2666.03
	28	6	149.88	72.53	0.02	31.11	3656	1463	84	26	27	277	929.66
		8	3396.59	1505.86	0.00	203.69	51052	21811	1981	0	0	302	2675.14
		4	1234.98	408.80	1.11	3.92	72210	31117	1339	2050	1951	76	358.54
	14	6	18.94	10.52	0.00	9.48	1172	433	55	0	1	106	1827.04
		8	179.41	81.27	0.05	17.39	8835	3640	39	56	57	141	2171.08
		4	136.69	67.05	0.02	11.72	6651	2044	113	0	1	143	1093.43
4	21	6	7.08	3.58	0.00	23.34	236	74	8	0	1	242	1555.92
		8	1669.09	733.95	0.02	151.38	40997	13503	1213	50	27	252	2479.78
		4	169.84	70.72	0.00	19.63	5367	1728	66	1	2	260	1307.35
	28	6	38.17	17.91	0.00	41.47	785	269	10	0	1	340	1937.09
		8	212.00	106.64	0.00	217.91	2813	1032	6	2	3	342	3822.17

Table 6: Perturbed instances

			Time			Node										
Class	**Days**	**Product**	Total	Relax	NLP	Pre-Proc.	Gen.	Br.	Dis.	Dom.	#NLP	VI#1	Ini Sol.	Best	Bound	Gap(%)
		4	0.70	0.48	0.00	1.14	125	54	26	24	6	32	154.55	5.70	5.68	0.02
	7	6	4.06	2.31	0.11	2.64	539	438	30	34	228	38	761.24	7.53	6.96	0.57
		8	4.23	2.59	0.17	4.19	468	362	36	51	173	44	561.02	6.66	6.30	0.36
		4	4.33	2.30	0.13	4.59	436	339	40	55	168	94	536.44	30.75	30.01	0.74
	14	6	15.69	7.53	0.36	8.77	1089	902	40	137	467	115	1589.76	28.70	27.90	0.80
2		8	9.89	4.80	0.14	70.59	590	476	61	46	234	109	1212.62	16.56	15.72	0.84
		4	19.02	9.05	0.36	8.41	1162	904	47	116	426	190	2095.22	38.80	38.20	0.60
	21	6	107.64	51.44	1.17	14.72	4687	4019	135	518	2052	136	2853.39	32.46	31.80	0.66
		8	133.75	55.98	1.50	65.63	4492	3988	133	313	2156	181	3046.96	48.83	47.94	0.89
		4	13.63	6.22	0.11	15.31	557	376	14	48	144	320	2521.28	48.69	48.46	0.23
	28	6	55.95	24.56	0.17	33.97	1593	1142	54	378	446	326	476.92	47.02	46.60	0.42
		8	124.27	62.31	0.38	179.95	2177	1599	61	413	631	413	2941.36	27.30	26.31	0.99
		4	5.16	3.23	0.08	0.89	728	359	83	54	83	16	363.41	11.27	10.28	0.99
	7	6	370.47	174.22	9.06	2.23	36309	29545	469	1530	13914	35	756.64	4.98	3.99	0.99
		8	31.39	16.63	0.92	4.59	3045	2446	188	129	1150	50	959.91	4.06	3.24	0.83
		4	127.33	62.52	2.03	4.59	10965	8029	282	1634	3485	91	1890.64	17.31	16.50	0.81
	14	6	3619.84	903.67	28.56	7.69	128032	112342	4118	1711	45030	81	216.42	15.07	10.89	4.18
		8	3709.91	1098.17	31.66	13.34	117360	102422	2920	2262	44936	87	1566.21	20.46	15.94	4.52
3		4	2362.35	962.11	4.19	20.81	48931	27561	1493	19653	6349	293	1768.88	71.78	70.99	0.79
	21	6	3119.17	1026.05	15.92	22.28	95000	66618	3366	17583	24113	252	103.92	25.10	24.37	0.72
		8	3954.83	1144.75	14.64	28.23	84114	64588	4043	2885	20754	137	1214.47	28.51	23.69	4.82
		4	1473.64	601.23	3.41	19.50	45650	24643	1361	18775	5151	300	2974.46	75.63	74.70	0.93
	28	6	3691.58	1225.94	13.97	40.33	70195	53171	3501	5069	18088	351	1837.06	36.76	32.73	4.03
		8	3670.41	1515.92	3.00	250.94	51928	34237	1822	5114	4210	260	2430.05	29.89	25.59	4.30
		4	4238.61	1243.47	0.55	3.94	121336	71764	5178	40387	916	90	1361.45	11.87	11.03	0.84
	14	6	4247.70	1500.89	0.84	7.80	110504	62968	3582	31329	1104	78	1688.36	12.60	9.43	3.17
		8	7302.53	1864.11	11.11	12.41	155860	93126	4153	41149	13749	50	859.01	12.13	7.45	4.68
		4	8083.58	2306.41	11.50	11.52	180764	102470	5470	60386	15246	169	248.52	39.63	37.23	2.40
4	21	6	6189.22	1774.72	9.89	17.52	118253	72650	4220	23518	13046	165	1328.48	34.23	29.72	4.51
		8	5645.56	2245.20	10.13	29.92	140178	94594	4552	27345	14177	102	1696.79	38.47	33.85	4.62
		4	4064.64	1112.59	15.48	23.00	94901	60890	2288	13886	19503	386	416.61	53.68	49.64	4.04
	28	6	9252.73	3292.19	7.67	43.23	132019	87375	5832	29557	13077	343	953.01	56.95	52.10	4.85
		8	11689.00	3746.00	9.83	296.54	147664	90428	6373	32346	15746	368	1234.02	45.63	40.93	4.70

Table 7: Comparison framework

			BB Sol			Baron		Knitro		Generalization
Class	**Days**	**Product**	Ini Sol	MSE	Class.(%)	MSE	Class.(%)	MSE	Class.(%)	MSE
2	7	4	154.55	5.70	0.00	5.70	0.00	5.63	0.00	32.95
		6	761.24	7.53	0.00	7.53	0.00	7.53	0.00	22.17
		8	561.02	6.66	0.00	8.08	14.29	8.12	14.29	14.01
	14	4	536.44	30.75	0.00	3682.29	28.57	157.58	21.43	54.58
		6	1589.76	28.70	0.00	3441.98	50.00	147.56	42.86	37.94
		8	1212.62	16.56	0.00	456.94	14.29	232.32	21.43	53.14
	21	4	1768.88	71.78	0.00	17983.55	76.19	239.75	38.10	66.28
		6	2853.39	32.46	0.00	880.03	23.81	165.07	38.10	40.39
		8	3046.96	48.83	0.00	5830.66	66.67	180.49	47.62	83.47
	28	4	2521.28	48.69	0.00	499.01	14.29	548.69	25.00	43.28
		6	476.92	47.02	0.00	463.12	21.43	347.69	14.29	38.84
		8	2941.36	27.30	0.00	827.22	17.86	247.26	46.43	36.51
3	7	4	363.41	11.27	0.00	161.32	57.14	132.84	28.57	23.33
		6	756.64	4.98	0.00	149.63	42.86	171.26	42.86	11.72
		8	959.91	4.06	0.00	18.64	14.29	19.15	28.57	17.32
	14	4	1890.64	17.31	0.00	604.04	14.29	207.60	42.86	49.09
		6	216.42	15.07	0.00	1954.43	28.57	135.67	28.57	34.02
		8	1566.20	20.46	0.00	117.07	42.86	157.09	28.57	18.25
	21	4	2095.22	38.80	0.00	2314.24	28.57	91.38	14.29	36.77
		6	103.92	25.10	0.00	9283.33	71.43	124.95	52.38	33.79
		8	1214.47	28.51	0.00	9308.44	76.19	279.36	47.62	41.72
	28	4	2974.46	75.63	0.00	21006.68	67.86	198.67	32.14	93.28
		6	1837.06	36.76	0.00	15188.60	67.86	856.47	57.14	63.89
		8	2430.05	29.89	0.00	2054.74	25.00	1146.00	32.14	45.76
4	14	4	1361.45	11.87	0.00	796.87	64.29	456.36	42.86	23.67
		6	1688.36	12.60	0.00	5620.43	57.14	2345.00	57.14	69.58
		8	859.01	12.13	0.00	4235.06	71.43	4235.06	57.14	14.44
	21	4	248.52	39.63	0.00	n\a	n\a	n\a	n\a	41.12
		6	1328.48	34.23	0.00	n\a	n\a	n\a	n\a	44.48
		8	1696.79	38.47	0.00	n\a	n\a	n\a	n\a	33.67
	28	4	416.60	53.68	0.00	n\a	n\a	n\a	n\a	58.51
		6	953.01	56.95	0.00	n\a	n\a	n\a	n\a	69.58
		8	1234.02	45.63	0.00	n\a	n\a	n\a	n\a	58.63