

6-2015

An Evaluation of Integer Programming Models for Restaurant Reservations

Gary Thompson

Cornell University School of Hotel Administration, gmt1@cornell.edu

Follow this and additional works at: <http://scholarship.sha.cornell.edu/articles>



Part of the [Hospitality Administration and Management Commons](#)

Recommended Citation

Thompson, G. M. (2015). *An evaluation of integer programming models for restaurant reservations* [Electronic version]. Retrieved [insert date], from Cornell University, SHA School site:
<http://scholarship.sha.cornell.edu/articles/837>

This Article or Chapter is brought to you for free and open access by the School of Hotel Administration Collection at The Scholarly Commons. It has been accepted for inclusion in Articles and Chapters by an authorized administrator of The Scholarly Commons. For more information, please contact hlmdigital@cornell.edu.

An Evaluation of Integer Programming Models for Restaurant Reservations

Abstract

A notable difference in rooms (hotel) revenue management reservations versus table (restaurant) revenue management (TRM) reservations is the variation that occurs in duration. In the hotel setting, durations are explicit in the reservation itself: a stay of a specified number of nights. In restaurants, by contrast, there is a natural variation in the amount of time parties are at the table. This duration variation presents interesting challenges to TRM. Dealing with these challenges is our goal in the article. Specifically, we introduce and evaluate 10 different models for restaurant capacity and reservations, five each of two different types. In one type of model, tables are pooled and parties are not explicitly matched to tables; in the other parties are matched to specific tables. The objective is to maximize revenue (or contribution) from known reservation demand. Variables are both the mix of tables in the restaurant and the reservations accepted. An important ancillary goal we have is to evaluate the effectiveness of the models from the perspective of customers, specifically examining whether a table is ready for them at the time of the reservation, an issue of high importance to restaurant patrons. Of the 10 models, seven define a pareto frontier between revenue and service; of those seven, five are pooling models. We use this frontier to offer advice to restaurateurs looking to better manage reservations.

Keywords

revenue management, services, restaurants, integer programming applications

Disciplines

Hospitality Administration and Management

Comments

Required Publisher Statement

© Palgrave Macmillan. This is a post-peer-review, pre-copyedit version of an article published in *Journal of Revenue and Pricing Management*. The definitive publisher-authenticated version: Thompson, G. M. (2015). An evaluation of integer programming models for restaurant reservations. *Journal of Revenue and Pricing Management*, 14, 305-320. doi:10.1057/rpm.2015.17 is available online at: <http://www.palgrave-journals.com/rpm/journal/v14/n5/abs/rpm201517a.html>

AN EVALUATION OF INTEGER PROGRAMMING MODELS FOR RESTAURANT RESERVATIONS

Gary M. Thompson

School of Hotel Administration

Cornell University

Ithaca, NY 14853

ABSTRACT

A notable difference in rooms (hotel) revenue management reservations versus table (restaurant) revenue management (TRM) reservations is the variation that occurs in duration. In the hotel setting, durations are explicit in the reservation itself: a stay of a specified number of nights. In restaurants, by contrast, there is a natural variation in the amount of time parties are at the table. This duration variation presents interesting challenges to TRM. Dealing with these challenges is our goal in the article. Specifically, we introduce and evaluate 10 different models for restaurant capacity and reservations, five each of two different types. In one type of model, tables are pooled and parties are not explicitly matched to tables; in the other parties are matched to specific tables. The objective is to maximize revenue (or contribution) from known reservation demand. Variables are both the mix of tables in the restaurant and the reservations accepted. An important ancillary goal we have is to evaluate the effectiveness of the models from the perspective of customers, specifically examining whether a table is ready for them at the time of the reservation, an issue of high importance to restaurant patrons. Of the 10 models, seven define a pareto frontier between revenue and service; of those seven, five are pooling models. We use this frontier to offer advice to restaurateurs looking to better manage reservations.

KEYWORDS: revenue management; services; restaurants; integer programming applications

INTRODUCTION

Variation in service duration variation is markedly different in rooms (hotel) revenue management reservations versus table (restaurant) revenue management (TRM) reservations. In the hotel setting, durations are explicit in the reservation itself: a stay of a specified number of nights. In restaurants, by contrast, there is a natural variation in the amount of time parties are at the table. Some of this is predictable, in that durations tend to increase as the party size increases (Bell and Pliner, 2003; Kimes and Robson, 2004; Kimes and Thompson, 2004). Some of the variation is unpredictable, however, and simply a function of the speed with which customers make food and beverage choices, how quickly they eat, and the number of courses they order. While customers like to control the pace of dinner (Noone *et al*, 2007), restaurants can manage this to some extent by offering such things as *prix fixe* menus, or upselling or underselling during portions of the meal.

Duration variation presents interesting challenges to TRM, as first noted by Kimes *et al* (1998). Dealing with these challenges is our goal in the article, specifically introducing and evaluating a number of integer programming models for restaurant capacity and reservations. The objective of the models is to maximize revenue (or contribution) from known reservation demand. Variables are both the mix of tables in the restaurant and the reservations accepted. An important ancillary goal of ours is to evaluate the effectiveness of the models from the perspective of customers, by examining whether a table is ready for them at the time of the reservation, an issue of high importance to restaurant patrons (McDougall and Levesque, 1999; Kimes, 2008). We evaluate a total of 10 models – five models of each of two types, one where tables are pooled and parties are not explicitly matched to tables and another where parties are

matched to specific tables. We find that the table-specific models are larger and much harder to solve optimally. We also find that using mean service duration leads to problematic customer service levels – in some instances over 25 per cent of parties would not have a table ready at the specified reservation time. However, by evaluating a range of controllable parameters in the models, we are able to develop a pareto frontier for revenue and customer service that is defined by the models' solutions. With restaurant sales in the USA forecast to be over \$471 billion in 2015 (Anonymous, 2015), our advice to restaurateurs looking to better manage reservations is likely to be of broad interest.

LITERTURE REVIEW

The term 'restaurant revenue management' was coined by Kimes *et al* (1998). They contended that by reducing variation in dining duration, restaurateurs will be better able to determine 'whether and for what time reservations should be taken'. In our studies, one of the issues we examine is the effect of variation in dining duration with regard to reservations.

Since Kimes *et al*'s (1998) formative paper, there has been a large body of work on the broad topic of restaurant revenue management, much of it related to restaurants taking walk-in customers (for a recent summary, see Thompson, 2010). Interestingly, and despite Thompson's (2010) identifying reservations as one of the hanging research areas in restaurants, little has been done on the topic. In the context of restaurants, of particular relevance to our work are one experiment-based study, one survey-oriented study and three modeling studies.

McDougall and Levesque (1999) conducted an experiment to attempt to mitigate the effects of a table not being ready at the time of a reservation. They found that none of the tested mitigation efforts – apologies, assistance, compensation and assistance plus compensation – lead to positive future intentions towards the restaurant. Kimes (2008), in a convenience survey of restaurant customers, found that customers had high expectations that tables will be ready at the time of the reservation. The implications of these two studies is that reservations need to be honored, at the specified time. That is a driving factor in our work.

Bertsimas and Shioda (2003) presented the first investigation into restaurant reservations. Their context was that of rolling decisions, where the choice was to accept or not accept customers with reservations and walk-in customers. Notably, their models could turn down customers with reservations. Our investigation does not allow this to happen.

Thompson and Kwortnik (2008) looked at whether reservations should be matched to specific tables or pooled by table size. In a survey on which they reported, 80 per cent of restaurants matched reservations to specific tables. However, they observed higher service levels (that is, fewer customers waiting for table at the time of the reservation) when reservations were pooled into like-size tables. The context for their study was a dining situation, such as in a cruise ship, where the parties are seated *en-masse* at two or more ‘seatings’ where the capacity mix of tables had been previously identified. In our models, we consider the more common situation of overlapping and staggered reservations. We also allow the capacity mix to vary. Both differences matter, because in our context the pooled models, while ensuring sufficient aggregate capacity at all times, could require that parties switch tables during their meal. Because we do not allow that, we expect that our non-pooled models will yield superior service levels. This would be counter

to the general observations regarding the benefits of pooling resources; for example, safety stock in parts production (Collier, 1982), lead times (Sheu and Wacker, 1997), labor costs (Thompson, 1999) and equipment costs (Pasin *et al*, 2002).

Recently, Alexandrov and Lariviere (2012) looked at the issue of whether a restaurant should or should not take reservations. A key issue in their results is the no-show rate, and they evaluated various strategies to mitigate the effects of no-shows. Their models consider capacity in aggregate over time, rather than as time-dependent and their models do not distinguish between party sizes. In our study, we are assuming that the decision to take reservations has been made. We also consider capacity as time-dependent and affected by party sizes, by looking for the optimal mix of capacity and reservations accepted.

Restaurant reservations may seem akin to health-care appointment scheduling, for which there is an extensive literature (for comprehensive coverage, see Cayirli and Veral, 2003; Gupta and Denton, 2008). However, in the TRM environment, customer waiting is problematic as noted above, compared to health-care settings where waiting is often a foregone conclusion. Moreover, the ability to have a flexible capacity mix like we consider is much less common in health-care settings. Finally, in the TRM setting, no ethical issues arise by denying service by accepting fewer reservations than customers demand. Perhaps this explains why little has been done on revenue management in health-care appointment scheduling contexts.

IP MODELS FOR RESTAURANT CAPACITY AND RESERVATIONS

The context we examine is where the decision has been made to take reservations and the issues at hand are how to best allocate the restaurant capacity to different size tables and which

reservations, from a set of known demand, should be accepted. One can think of the outputs as inward-oriented – the mix of tables – and outward-oriented – the set of reservation slots that would be open to customers via the reservation book. We distinguish between planning – when pooling is model-dependent – and control, when all like-sized tables are pooled. We are applying pooling in all cases in real-time because the actual dining duration for a party cannot be known in advance and so sticking with the planned table assignments would lead to more parties waiting for tables.

We define two model types, which differ in how parties are assigned to tables. In the pooled-table model type, parties are not matched to specific tables, but all the pooled-table models ensure that sufficient tables are available at all times for the parties with reservations. This is somewhat akin to the pooling approach to reservations taken by Thompson and Kwortnik (2008). The second model type explicitly matches reservations to tables, a process used by 80 per cent of the restaurants in Thompson and Kwortnik's (2008) survey. Before describing the models, we define assumptions and then identify model components.

Modeling assumptions

We have four main assumptions in the models we present. First we assume that the restaurant is reservations-only and does not take walk-in customers, a matter to which we return in the discussion. We do this to maintain focus on the issue at hand: how to best model reservations. Second, we assume there are no no-shows. Restaurateurs take many actions to limit no-shows, including charging customers' credit cards, requiring customers to call to reconfirm the reservation, or calling customers to reconfirm the reservation (see, for example, Kimes *et al*, 1998). We note that if the number of no-shows is not negligible, then the restaurant could

consider overbooking or also taking walk-in customers, though both are beyond the scope of the current investigation.

Third, we assume that all parties arrive at the designated reservation time. However, this assumption could easily be relaxed, as late shows would just increase the mean duration and variability of dining times. Finally, we assume that parties do not share tables. Although communal tables are not uncommon in walk-in restaurants, anecdotal evidence suggests that they are uncommon in restaurants taking reservations.

Model components

Below we define sets and indexes, parameters, and decision variables.

Indices and sets:

c

index for party sizes;

i

index for table seats;

j

index for specific tables;

l, l'

index for dining length, in periods;

p, p'

index for periods;

C

set of customer (party) sizes;

P

set of dining periods in which reservations are accepted;

S

set of table sizes;

T

set of specific tables; and

L_c

set of possible meal durations (in periods) for a party of c people.

Parameters are:

dmd_{cp}
demand for reservations (in number of parties) of size c at period p ;
 $lenmin_c$
minimum meal duration (in periods) for a party of c people;
 $lenmax_c$
maximum meal duration (in periods) for a party of c people;
 $lenprop_{cl}$
probability that a party of c people will take longer than l periods to dine;
 $mntblsz$
minimum number of seats allowed in any table;
 $mxtbls_i$
maximum number of individual tables of i seats= $\lfloor space_{avl}/space_i \rfloor$;
 $ovrlmxtb$
overall maximum number of individual tables (this is the number of elements in set T);
 $space_{avl}$
space available for seating in the restaurant;
 $space_i$
space required for a table of i seats; and
 $value_c$
value of a party of c people.

Decision variables are:

n_i
number of tables with i seats;
 g_{cpl}
number of reservations accepted for a party of size c , length (duration) l , at period p , placed in a generic (pooled) table with i seats;
 t_{cplj}
1, if a reservation is accepted for a party of size c , length (duration) l , at period p , in specific table j ; 0, otherwise; and
 s_{ij}
1, if specific table j has i seats; 0, otherwise.

Pooled-table capacity and reservation models

We present four models, two for each of the pooled-table and table-specific cases. By varying the parameters in the four basic models we arrive at the 10 we evaluate, as we describe below. The basic pooled-table model (TP1) is:

$$\text{MAX} \sum_{c \in C} \text{value}_c \left(\sum_{l \in L_c} \sum_{p \in P} \sum_{i \in S} g_{clpi} \right) \quad (1)$$

$$\sum_{i \in S} \text{space}_i \star n_i \leq \text{spaceavl} \quad (2)$$

$$\sum_{c \in C} \sum_{l \in L_c} \sum_{\{p' \in P | p-1+1 \leq p' \leq p\}} g_{clp} \leq n_i \quad \forall p \in P, i \in S \quad (3)$$

$$\sum_{l \in L_c} \sum_{i \in S} g_{clpi} \leq \text{dmd}_{cp} \quad \forall c \in C, p \in P \quad (4)$$

$$n_i = \{0, 1, 2, \dots, \text{mxtbls}_i\} \quad \forall i \in S \quad (5)$$

$$g_{clpi} = \{0, 1, 2, \dots, \text{dmd}_{cp}\} \\ \forall c \in C, l \in L_c, p \in P, i \in S \quad (6)$$

The objective (1) maximizes the value of the reservations accepted. Constraint set (2) ensures that the tables selected fit within the restaurant's available space for seating. Constraint set (3) limits the number of parties seated at each size table, in each period, to the number of tables of the designated size. Demand limitations on the number of reservations accepted are imposed in constraint set (4). Finally, constraint sets (5) and (6) impose the non-negative and integrality restrictions on the number of each size table and number of reservations accepted, respectively.

We evaluate three forms of TP1, based on the extent to which dining times are inflated. In TP1(0), we use the mean dining duration for each size party, rounded up to the nearest number of periods. This results in the sets L_c each having only a single member. For example, given our

use of 15-min periods, a mean dining duration of 48 min for a party of two would be represented as four periods, so $L_2=\{4\}$.

While TP1(0) is the simplest model, it ignores potential problems resulting from natural variation in party dining times. Moreover, by rounding up to the nearest number of periods, the closer the mean duration is to the rounded up number of periods, the less likely it would be that a party would finish dining within the allocated time. For example, parties with a mean duration of 58 min are much less likely to complete dining within 60 min (four periods) than are parties with a mean duration of 47 min. To incorporate a hedge against stays that are longer than average, we also evaluate TP1(1) and TP1(2), where the dining durations used in TP1(0) are all inflated by one and two periods, respectively. Continuing the example from the previous paragraph, $L_2=\{5\}$ for TP1(1) and $L_2=\{6\}$ for TP1(2).

We also use another way to capture the variation of within-party dining durations, in the pooled-table model with party length adjustments, TP2. TP2 allows the sets L_c to each contain more than a single member. These set members are also related to the $lenmin_c$, $lenmax_c$ and $lenprop_{cl}$ parameters, as we describe here. For each party size c , we set $lenmin_c$ to one fewer than the rounded up number of periods given by the mean dining duration, providing that the probability of the dining duration falling below that number of periods equals or exceeds 0.00005, otherwise we set it to the rounded up number of periods from the mean dining duration. For each party size c we set $lenmax_c$ to the smaller of two (TP2(2)) or five (TP2(5)) periods higher than the rounded up number of periods given by the mean dining duration and the number of periods where the probability of the dining duration falling above that number of periods is less than 0.00005. Table 1 presents a numerical example, taken from our studies, for a party size

of five people, with a mean dining duration of 67.67 min and a coefficient of variation of 0.30 in dining times. As we use a lognormal distribution for dining times (for reasons we discuss later), this gives a mean and standard deviation of the lognormal distribution of 4.1754 and 0.2961, respectively. The mean dining duration of 67.67 min would translate to five periods, and so the lower limit on $lenmin_c$ is $5-1=4$ periods and the upper limit on $lenmax_c$ is $5+2=7$ periods in TP2(2) and $5+5=10$ periods in TP2(5). TP2 is:

(1), (2), (3), (4), (5), (6) and

$$\sum_{\{l' \in L_c | l' \geq l\}} g_{cl'pi} \geq lenprop_{cl} \sum_{l' \in L_c} g_{d'pi}$$

$$\forall c \in C, \{l \in L_c | l > lenmin_c\}, p \in P, i \in S \quad (7)$$

Constraint set (7) ensures that a sufficient proportion of the accepted reservations are for longer dining durations. For example, based on the data in Table 1, at least 5.33 per cent of the reservations accepted for parties of five (seated in a particular table size in a particular period) would require durations of seven periods (105 min) or longer. We note that while one does not know what the dining duration of any party will be when they are seated, TP2 ensures that capacity is allocated to cover longer-than-average stays.

We do not make any *a priori* predictions as to which will be better, TP1 or TP2, other than that TP1(0) runs the risk of poor service, because it has the least compensation for duration variation.

Specific-table capacity and reservations models

While the pooled-table models include constraints that ensure no more parties are in the restaurant than there is capacity to serve them, because these models do not explicitly match parties to specific tables it is possible that solutions would require parties to switch tables during their meal. While room changes happen in hotels, table changes are impractical in restaurants (unless, of course, they occur based on customer requests early after being seated). As we do not allow parties to change tables in our study, the result is that parties might wait for a table to become available before they can be seated. This motivated the table-specific models, which explicitly match parties to tables. As noted earlier, 80 per cent of the restaurants in Thompson and Kwortnik's (2008) survey reported matching reservations to specific tables. Also as noted earlier, even with the table-specific models, we use pooling in real-time assignments, because the actual dining duration cannot be known in advance. This means, then, that the actual table assigned to a party may well be different than that in the solution to the table-specific model.

An important parameter in the table-specific models is $ovrlmxtb$, the maximum number of possible tables. We set this as follows:

$$ovrlmxtb = \max \left(\frac{spaceavl}{space_{mmtblsz}} \star 1.2 \star \frac{\sum_{\{c \in C, c \leq mmtblsz\}} \sum_{p \in P} dmd_p}{\sum_{c \in C} \sum_{p \in P} dmd_p}, \max_{\{i \in S, i > mmtblsz\}} mxtbls_i \right) \quad (8)$$

The first component of (8) is the maximum number of the smallest tables that could be used, based on the demand for that table size as a proportion of overall demand, inflated by 20 per cent. The second component of (8) finds, for all other table sizes, the maximum number of that size table that would fit in the restaurant. The largest of any of those values gives $ovrlmxtb$.

The basic table-specific (TS1) model is:

$$\text{MAX} \sum_{c \in C} \text{value}_c \left(\sum_{l \in L_c} \sum_{p \in P} \sum_{j \in T} t_{clpj} \right) \quad (9)$$

$$\sum_{i \in S} \text{space}_i \left(\sum_{j \in T} s_{ij} \right) \leq \text{spaceavl} \quad (10)$$

$$\sum_{i \in S} s_{ij} \leq 1 \quad \forall j \in T \quad (11)$$

$$\sum_{i \in S} i^* (s_{ij} - s_{i,j+1}) \geq 0 \quad (12)$$

$$\forall \{j \in T \mid j < \text{ovrlmxtb}\}$$

$$c^* t_{dpj} \leq i^* \sum_{i \in S} s_{ij} \quad (13)$$

$$\forall c \in C, l \in L_c, p \in P, j \in T$$

$$\sum_{c \in C} \sum_{l \in L_c} \sum_{\{p' \mid p-l+1 \leq p' \leq p\}} t_{clp'j} \leq 1 \quad (14)$$

$$\forall p \in P, j \in T$$

$$\sum_{l \in L_c} \sum_{j \in T} t_{dpj} \leq \text{dmd}_{cp} \quad \forall c \in C, p \in P \quad (15)$$

$$s_{ij} = \{0, 1\} \quad \forall i \in S, j \in T \quad (16)$$

$$t_{clpj} = \{0, 1\} \quad (17)$$

$$\forall c \in C, l \in L_c, p \in P, j \in T$$

Like constraint set (7), constraint set (18) ensures that a sufficient proportion of the reservations accepted are for longer dining durations. As we did with TP2, we also evaluate two variants of TS2 – TS2(2) and TS2(5) – where the value of $lenmax_c$ is capped at two and five periods higher than the rounded-up mean dining duration. A summary of the 10 models we evaluated is provided in Table 2. Later we show a comparison of model sizes.

STUDY

To examine the effectiveness of the models with respect to revenue and customer service, we created 768 test environments, using a full-factorial design with seven factors, as summarized in Table 3. We included these factors either because we thought they could affect model performance, or to ensure we represent the range of situations seen in restaurants. While the models are all designed to maximize revenue (or contribution), there is no direct way, other than implementation, to measure the service level their solutions yield. As implementation is impractical, we resort to the methodology of simulation to evaluate service levels.

Study factors

We used three levels of restaurant size: 40, 80 and 160 seats. The largest size falls within the median and mean restaurant sizes of 150 and 211 seats in a survey conducted by Thompson (2011a). We included the smaller restaurant sizes, however, because we suspected that dining duration variation would be more problematic when there were few tables in the restaurant, simply because the smaller pool of available tables would reduce flexibility.

Demand Load Factor had four levels, ranging from 90 to 120 per cent of full capacity, in increments of 10 per cent, where full capacity would be the level of customer demand that would use the full capacity of the restaurant. Obviously, as the demand load factor increases, the restaurant could be more selective in the reservations it accepts and so tend to fill the restaurant more fully, perhaps leading to reduced service. Previous studies, all for restaurants taking only walk-in parties, have used demand load factors ranging from 95 per cent of full capacity (Thompson and Sohn, 2009) to 300 per cent of full capacity (Thompson, 2011b), with most falling around or slightly over full capacity.

Day Length had two levels – 2 and 4 hours – broken into 15 periods. Given our assumption of a 60-min mean dining duration in all scenarios, the day lengths mean that a restaurant should be able to turn its tables about two and four times for the levels of this factor. Anecdotal evidence suggests that four table turns in any meal period is at the upper limit for most full-service restaurants. We expect that it is harder to achieve stellar customer service under the longer day length. This is because the longer day length, by giving more timing flexibility for reservations, is likely to result in fuller restaurants and so lower customer service.

The two levels of *Mean Party Size* had 2.5 and 3.0 people per party. The only two studies of which we are aware that explicitly report mean party sizes are Kimes and Robson (2004) and Thompson (2011b), who report 2.5 and 2.55 people, respectively. However, mean party sizes can be inferred from the material presented by Kimes and Thompson (2005) for the restaurant they studied, which ranged from 2.5 to 2.9 people across the days of the week, with Friday, Saturday and Sunday all having mean party sizes of 2.8 people or higher. The party size probabilities we used are reported in Table 4. While larger mean party sizes will skew the optimal mix of tables

in the restaurant to larger sizes, we do not have any expectations regarding a relationship between party size and customer service.

Across Party Duration Variation factor captured the phenomenon of dining duration increasing with party size observed by Bell and Pliner (2003), Kimes and Robson (2004), and Kimes and Thompson (2004). The two levels had ratios of 1.5 and 2.0 for the mean dining time for a party of 10 compared to the mean dining time for a party of one, which fall within the range of reported values – a low of 1.14 (Kimes and Thompson, 2004) to a high of 3.18 (Bell and Pliner, 2003). For both factor levels we assumed that duration increased linearly across party sizes. Coupled with our party size probabilities, our mean dining times (Table 4) maintained the weighted average duration of 60 min. We included this factor for completeness, as we do not have any *a priori* expectations of different service levels across the factor levels.

Within Party Duration Variation had two levels: coefficients of variation in dining duration of 0.15 and 0.30. These values fall largely within the range of coefficients of variation reported in the literature of 0.16–0.50 (Bell and Pliner, 2003; Kimes and Robson, 2004). Following Kimes *et al* (1998), our expectation is that greater variation in the at-the-table time would reduce the likelihood of tables being ready when parties arrived for their reservation.

Average Check Variation had two levels, each having a weighted average check of \$20.00 per person. This factor was motivated by observations that larger parties spend less per person than do smaller parties (Kimes and Robson, 2004; Kimes and Thompson, 2004). The two levels have a spend per person for parties of 10 equal to 0.8 and 0.9 times the spend per person of parties of one, with linear changes across party sizes. Table 4 reports the values we used. This is another factor included solely to represent a range of restaurant situations.

The combinations of the levels of the seven factors reported in Table 3 gave 384 scenarios. For each of these, we created two sets of reservation demand patterns, giving a total of 768 test environments.

Process

While many previous restaurant table mix problems have been solved with heuristics (see, for example, Kimes and Thompson, 2005), the process we followed was to create each problem environment and then (attempt to) solve the models optimally using Gurobi (Gurobi, 2013), run from the command line, with a time limit of 10 min per problem, on an Intel i5-based personal computer, with four cores. This was implemented with an MPS file generator in Excel® and via running Gurobi from the command line. For reference, the computer had a Linpack (Dongarra and Luszczek, 2011) benchmark of 1,737.7 MFLOPS. Table 5 reports model sizes across the levels of restaurant size and day length.

After solving a problem, we loaded the solution and simulated 100 days of that reservation mix using a Visual Basic for Applications macro in Excel®. The randomness in the simulation was the dining duration for the parties. We used a lognormal distribution for service times, as it maps well onto dining durations in restaurants – the longer right tail of the distribution captures the phenomenon of some parties lingering over their meal. This distribution has also been used in previous studies (including Thompson and Kwornik, 2008; Thompson, 2011a).

In the simulation, we assumed that parties arrived on time (an assumption we noted earlier). Since, as noted earlier, pooling is applied in real-time, arriving parties were assigned to

an open table of the size specified by the model's solution. If there were no open tables, the party would be assigned to the next available table of the assigned size. Once assigned to a table, parties remained there for the duration of the meal, regardless of model being evaluated; no table switching occurs.

We collected performance metrics on: the average revenue; average percentage of parties that waited upon arrival for a table; the average percentage of customers waiting more than 1, 2, 5, 10, 15, 20, 25 and 30 min; and the average wait time of those who waited for a table. Our primary metric of customer service – the *service level* – is the proportion of parties who wait, which means that lower values of this metric indicate better levels of customer service.

RESULTS

Table 6 presents results across the 768 test environments, ordered by increasing service levels. Seven of the 10 models fell on the pareto frontier, with all three versions of TS2 (that is, TS2(1), TS2(2) and TS2(5)) falling in the interior. Solution times for the five TP models were all under 1 second, on average, and all of these solved optimally within the 10-min time limit. The five TS models all averaged solution times over 200 seconds and all had a notable number of instances where optimal solutions were not obtained within the 10-min limit.

Both TP1(0) and TS1(0) delivered very poor levels of customer service, with tables not being available for over 15 and 12 per cent of parties, respectively. To better explore the relationship between model revenue and actual service levels, we plotted the results for the

pareto optimal models, in Figure 1. Note the pronounced inflection point around the 0.01 service level (that is, at the TP1(1) model).

To further examine the service-level performance, Figure 2 shows, for the five pareto optimal pooled-table models, the proportion of all parties that waited more than 1, 2, 5, 10, 15, 20, 25 and 30 min. Moving from left to right in the figure (that is, from the lower-revenue, higher-service models to the higher-revenue, lower-service models), one sees a small deterioration in service through TP2(2). However, service declines markedly with TP1(0).

Table 7 presents a comparison of the service level and profitability results for like models. Recall our contention that service levels would be higher in the table-specific models, because the pooled-table models could require parties to switch between tables. As the table shows, the results were mixed. In two of the like-model pairs, the service levels were higher in the table-specific models, while service levels were higher in the pooled-table models in the three other like-model pairs.

To examine the robustness of the models, we identified the experimental conditions under which each pareto optimal model had the poorest service levels, as reported in Table 8. Generalizing the patterns in worst case conditions, we see that the models tended to deliver lower levels of service with smaller restaurants, higher load factors and greater variation in dining times within party sizes, which are all consistent with our *a priori* expectations. The worst case service levels ranged from a low of 1.71 times the average service level (for TP1(0)) to a high of 5.62 (for TP2(2)). Two of the models – TP1(0) and TS1(0) – had worst case service levels over 0.25, indicating that more than one quarter of parties would wait for a table. At the other extreme, two models – TP1(2) and TP2(5) – had worst case service levels under 0.01. For most

of the models, the average wait for a table in the worst case conditions, for those who waited, was in the range of 8–10 min, with the exception of TP2(2), which had an average wait of under 4 min.

Table 9 reports the service levels for the pareto optimal models, for the two levels of variation in within-party dining times. In all cases, the service level was poorer with higher variation, which is consistent with Kimes *et al*'s (1998) contention.

DISCUSSION AND CONCLUSIONS

Restaurateurs are faced with the act of balancing the objectives of maximizing revenue and ensuring tables are available at the time of the customers' reservations. McDougall and Levesque's (1999) findings of the ineffectiveness of service recovery mechanisms when tables are not available at the time of the reservation would suggest that high revenue, under poor service levels, is false revenue, in that it would likely be unsustainable. We saw that seven of the 10 reservations models we evaluated yielded solutions on the pareto frontier of revenue and service. In general, the pooled-table type models proved superior, with all five of them falling on the pareto frontier of service levels and revenue, while only two of the five table-specific models fell on the frontier. An added benefit of the pooled-table models is that their solution times were under 1 second, on average, compared to several minutes for the table-specific models.

Our finding regarding the superiority of the pooled-table models echoes that of Thompson and Kwortnik (2008) who found superior results when reservations were pooled for same-sized tables, as opposed to matching reservations to specific tables. We note, however, that

they were examining the situation of multiple, simultaneous seatings of the entire restaurant, while we examined the more typical situation of overlapping reservations. Also, they assumed capacity was predefined, instead of optimizing the mix of capacity as we do. Our result further brings into question the appropriateness of the restaurateurs matching reservations to specific tables, of which 80 per cent of restaurateurs did in Thompson and Kwornik's (2008) survey.

Our expectation that the table-specific models would yield better service levels than the pooled-table, because the table-specific models prohibit table switching, was not supported. One might be tempted to attribute this to the fact that many of the table-specific models did not solve optimality. However, the objective in all the models was to maximize revenue. If a problem is not solved optimally, less-than-optimal revenue is achieved. Lower revenue in general means fewer parties in the restaurant, which should translate into better service levels than under optimal revenue. Therefore, it may simply be that the rounding-up of mean durations and adding a sufficient cushion of two periods in the fixed-durations model (that is, TP1(2)) may incorporate sufficient capacity to ensure that table switching is unnecessary. The evidence supporting this is that when mean duration is inflated insufficiently, the service levels were higher for the table-specific models (that is, in TP1(0) versus TS1(0) and TP1(1) versus TS1(1)). Moreover, the fact that all three TS2 models did not fall on the pareto frontier suggests that these models over-adjusted for duration variation – using both table-specificity and length flexibility proved to be unnecessary.

Given the shape of the pareto frontier, we are comfortable making recommendations to restaurateurs regarding the choice of a model. Our primary recommendation is to use TP1(1), the model at the pronounced inflection point in the pareto frontier shown in Figures 1 and 2. Across

the 768 test environments, this model yielded solutions where only about 1 per cent of parties had to wait for a table. We note that TP1(1) inflates the rounded-up mean dining duration by one period for all parties sizes, as a buffer against variation in dining times. While the models with the unadjusted durations – TP1(0) and TS1(0) – both fall on the pareto frontier, they delivered very poor service.

An additional compelling reason to use TP1(1) is that TP1 is the simplest of the models we evaluated. Is the additional complexity of the other models warranted? Perhaps. If a restaurateur was comfortable with a reduced level of service compared to that delivered by TP1(1), our recommendation would be TP2(2). Compared to TP1(1), TP2(2) increased revenue by 3 per cent, but doubled the number of parties that needed to wait for a table. Further increases in revenue come at the expense of significantly poorer service, however. If a restaurateur desired better service level than provided by TP1(1), perhaps because its worst case service level was slightly over 3 per cent, then TP1(2) is the recommended choice. TP1(2)'s revenues are about 13 per cent lower than TP1(1), yet the proportion of parties waiting for a table drops to under 0.2 per cent and the worst case service level is under 1 per cent and can be solved optimally in a comparable time (see Table 6).

Our results with respect to the effect of variation in within-party-size dining times supports Kimes *et al's* (1998) contention that reducing duration variation will improve the likelihood of a party getting a table. Reducing the variation to half its value improved service levels by at most 2 per cent of all parties. While 2 per cent does not sound by much in an absolute sense, consider that for our broadly recommended model, TP1(1), at the high level of duration variation, 1.8 per cent of parties would experience a wait for a table, compared to 0.3

per cent at the low level of variation. Therefore, efforts to reduce duration variation have the potential to largely eliminate instances of parties waiting for their tables.

In terms of limitations that might be relevant to restaurateurs, it is important to remember that our investigation used 15-min planning intervals. We do not expect that the same models would define the pareto frontier if reservations are made using other planning durations. Moreover, one would want to verify that dining times followed a lognormal distribution, as other distributions would likely affect the set of pareto models and the pareto frontier. We also note that we did not include any constraints on the number of parties that could be seated simultaneously. In practice, restaurateurs may wish to impose such constraints so as to better balance waitstaff and kitchen workloads, though these constraints could well change the pareto optimal models and affect the pareto frontier. While our assumption of known reservation demand should little trouble popular restaurants, less popular restaurants may wish to use both walk-ins and reservations. Finally, our recommendation for using TP1(1) – a pooling model – is not consistent with situations where one must match a party to a specific table (for example, a VIP guest matched to his/her regular table). One seemingly simple approach would be to just remove a table from the pool, but the challenge would be in knowing how long to make the table unavailable. How best to deal with such exceptions is left to future research.

We see three main extensions to our work, other than evaluating the models under the different conditions as noted above. First, it would be possible to create hybrid versions of TP1, where the rounding up was handled differently. For example, a hybrid could increase the rounded-up mean by one period in some cases and either zero periods or two periods in other cases. Second, another notable difference between restaurant and hotel reservations is that

customers typically have flexibility in their actual arrival time, as we are talking about time intervals of 15 min, rather than the whole day differences seen in hotels. Thus, one could think of each potential party as having an ideal reservation time, but a window within which arrival is possible. All of the models we evaluated could be extended to cover the case of arrival flexibility. We expect that doing so will increase revenue, because demand is no longer treated as being rigid. But, by increasing the volume of business, service levels could well decline. Third, one could attempt to devise a solution methodology that integrated the stochastic nature of the problem. Doing so could enable better exploration of the pareto frontier.

In conclusion, we saw while that seven of the 10 models we evaluated fell on the pareto frontier of revenue and service levels, the service level was substandard in two of the models. Given the size of the restaurant industry (Anonymous, 2014), our recommendation of a simple, easy to solve, yet effective model, is likely to be of broad interest to restaurateurs.

Table 1. Example of parameter values in TP2 and TS2, for a mean dining duration of 67.67 min and a coefficient of variation of 0.30 for dining times.

<i>Periods</i>	3	4	5	6	7	8	9	10
End of period in minutes	45	60	75	90	105	120	135	150
Probability that dining duration ≤ periods	0.1070	0.3920	0.6840	0.8630	0.9470	0.9810	0.9930	0.998
<i>lenprop_{cl}</i>	n/a	0.6080	0.3160	0.1370	0.0530	0.0190	0.0070	0.002

Table 2. Summary of the parameter values in the ten models we examined.

<i>Model</i>	<i>Pooled tables or specific tables</i>	<i>lenmin_c</i>	<i>lenmax_c</i>
TP1(0)	Pooled	RUM ^a	RUM
TP1(1)	Pooled	RUM+1	RUM+1
TP1(2)	Pooled	RUM+2	RUM+2
TP2(2)	Pooled	RUM-1	RUM+2
TP2(5)	Pooled	RUM-1	RUM+5
TS1(0)	Specific	RUM	RUM
TS1(1)	Specific	RUM+1	RUM+1
TS1(2)	Specific	RUM+2	RUM+2
TS2(2)	Specific	RUM-1	RUM+2
TS2(5)	Specific	RUM-1	RUM+5

^a RUM=rounded-up mean (=mean rounded up to the next higher number of periods).

Table 3. Factors and factor levels for the test environments.

<i>Factor</i>	<i># Levels: Levels</i>
Restaurant size (seats)	3: 40, 80, 160
Demand Load Factor (% of full capacity)	4: 90, 100, 110, 120
Day Length (hours)	2: 2, 4
Mean Party Size (people)	2: 2.5, 3.0
Across Party Duration Variation (ratio of mean duration of a party of 10 people to mean duration for a party of one person)	2: 1.5, 2.0
Within Party Duration Variation (coefficient of variation)	2: 0.15, 0.30
Average Check Variation (ratio of spend per person for parties of 10 to the spend per person of parties of one)	2: 0.9, 0.8

Table 4. Party-size related parameter values in the test scenarios.

<i>MPS</i>	<i>Item</i>	<i>Party size</i>									
		<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>
2.5	Probability	0.2	0.47	0.15	0.1	0.03	0.02	0.01	0.01	0.005	0.005
3	Probability	0.12	0.32	0.26	0.18	0.05	0.03	0.02	0.01	0.005	0.005
2.5	Avg Duration (low APDV)	55.37	58.45	61.52	64.6	67.67	70.75	73.83	76.90	79.98	83.06
2.5	Avg Duration (high APDV)	51.40	57.12	62.83	68.54	74.25	79.96	85.67	91.39	97.1	102.81
3	Avg Duration (low APDV)	53.99	56.99	59.99	62.98	65.98	68.98	71.98	74.98	77.98	80.98
3	Avg Duration (high APFV)	49.07	54.52	59.97	65.42	70.88	76.33	81.78	87.23	92.69	98.14
2.5	Avg PP Spend (low ACV)	\$21.09	\$20.62	\$20.15	\$19.68	\$19.21	\$18.75	\$18.28	\$17.81	\$17.34	\$16.871
2.5	Avg PP Spend (high ACV)	\$20.51	20.28	\$20.05	\$19.82	\$19.60	\$19.37	\$19.14	\$18.91	\$18.69	\$18.46
3.0	Avg PP Spend (low ACV)	\$21.30	\$20.83	\$20.35	\$19.88	\$19.41	\$18.93	\$18.46	\$17.99	\$17.51	\$17.04
3.0	Avg PP Spend (high ACV)	\$20.61	\$20.38	\$20.15	\$19.93	\$19.7	\$19.47	\$19.24	\$19.01	\$18.78	\$18.55

MPS=mean party size (people); APDV=across party dining variation; PP = per-person; ACV = average check variation.

Table 5. Average IP model sizes, by restaurant size and day length.

<i>Seats</i>	<i>Day length (hours)</i>	<i>TG1</i>	<i>TG2(2)</i>	<i>TG2(5)</i>	<i>TS1</i>	<i>TS2(2)</i>	<i>TS2(5)</i>
<i>(a) No of variables</i>							
40	2	245.0	965.0	1496.0	1188.7	4544.9	7127.7
40	4	485.0	1925.0	2987.0	2273.9	8888.9	14 178.1
80	2	245.0	965.0	1496.0	2372.0	9069.5	14 233.0
80	4	485.0	1925.0	2987.0	4622.6	18 070.1	28 671.6
160	2	245.0	965.0	1496.0	4774.6	18 255.9	28 734.4
160	4	485.0	1925.0	2987.0	9252.9	36 211.4	57 314.0
<i>(b) No of constraints</i>							
40	2	121.0	841.0	1372.0	1338.6	4934.8	7705.2
40	4	241.0	1681.0	2743.0	2613.1	9708.1	15 374.5
80	2	121.0	841.0	1372.0	2591.6	9529.1	14 879.6
80	4	241.0	1681.0	2743.0	5146.8	19 074.3	30 052.4
160	2	121.0	841.0	1372.0	5135.5	18 856.7	29 522.3
160	4	241.0	1681.0	2743.0	10 141.9	37 581.3	59 059.2
<i>(c) No of non-zeros</i>							
40	2	1398.0	8466.0	17 010.8	13 473.4	66 995.5	124 403.0
40	4	3082.0	18 442.0	37 764.6	27 899.8	139 946.3	268 714.4
80	2	1398.0	8466.0	17 010.8	26 898.5	133 707.5	248 479.3
80	4	3082.0	18 442.0	37 764.6	56 732.2	284 522.8	543 380.5
160	2	1398.0	8466.0	17 010.8	54 152.9	269 145.3	501 577.4
160	4	3082.0	18 442.0	37 764.6	113 568.6	570 155.2	1 086 226

Table 6. Service levels, revenue, solution times and number of optimal solutions for the models.

<i>Model</i>	<i>Daily revenue</i>	<i>Service level</i>	<i>Solution time (seconds)</i>	<i># Optimal solutions</i>	<i>Pareto optimal</i>
TP1(0)	\$5130.54	0.154538	0.16	768	Yes
TS1(0)	\$5046.26	0.124731	283.05	493	Yes
TP2(2)	\$4651.31	0.021241	0.19	768	Yes
TS2(2)	\$4518.49	0.021805	347.66	389	No
TP1(1)	\$4513.54	0.010567	0.09	768	Yes
TS1(1)	\$4447.35	0.009611	286.37	505	Yes
TP1(2)	\$4005.44	0.001754	0.11	768	Yes
TS1(2)	\$3962.17	0.002226	302.36	473	No
TP2(5)	\$3254.72	2.31E-05	0.10	768	Yes
TS2(5)	\$3241.10	2.47E-05	218.81	603	No

Table 7. A comparison of the pooled-table versus table-specific results.

<i>Like-model pair</i>	<i>Model</i>	<i>Daily revenue / Service level</i>	<i>Specific/Pooled^a</i>
1	TP1(0)	\$5130.54 0.154538	
	TS1(0)	\$5046.26 0.124731	0.807
2	TP1(1)	\$4513.54 0.010567	
	TS1(1)	\$4447.35 0.009611	0.910
3	TP1(2)	\$4005.44 0.001754	
	TS1(2)	\$3962.17 0.002226	1.269
4	TP2(2)	\$4651.31 0.021241	
	TS2(2)	\$4518.49 0.021805	1.027
5	TP2(5)	\$3254.72 2.31E-05	
	TS2(5)	\$3241.10 2.47E-05	1.069

^a Specific/Pooled=Service level of the pooled-table model divided by the service level of the table-specific model.

Table 8. Worst case performance for the models.

	<i>TP1(0)</i>	<i>TS1(0)</i>	<i>TP2(2)</i>	<i>TP1(1)</i>	<i>TS1(1)</i>	<i>TP1(2)</i>	<i>TP2(5)</i>
Restaurant size (seats)	40	40	160	40	40	40	40
Demand Load Factor (% of full capacity)	120	120	120	120	120	120	100
Day length (hours)	4	4	4	—	—	—	4
Mean Party Size (people)	—	3.0	2.5	3.0	3.0	3.0	3.0
Across Party Duration Variation (ratio of mean duration of a party of 10 people to mean duration for a party of one person)	1.5	—	1.5	1.5	—	1.5	2.0
Within Party Duration Variation (coefficient of variation)	0.30	0.30	0.30	0.30	0.30	0.30	0.30
Average Check Variation	—	—	—	—	—	—	—
Average Service Level	0.155	0.125	0.021	0.011	0.010	0.0018	0.00002
Worst Case Service Level	0.265	0.257	0.068	0.033	0.041	0.0098	0.00006
Average Delay in Seating (min)	9.07	10.20	3.71	8.20	9.63	9.74	9.70

Empty cells indicate little difference in results across the levels of that experimental factor.

Table 9. Service levels of the pareto optimal models, by level of variation in the within-party-size dining durations.

	<i>TP1(0)</i>	<i>TS1(0)</i>	<i>TP2(2)</i>	<i>TP1(1)</i>	<i>TS1(1)</i>	<i>TP1(2)</i>	<i>TP2(5)</i>
Low variation	0.14478	0.11675	0.01341	0.00316	0.00280	0.00010	0.00000
High variation	0.16430	0.13271	0.02907	0.01797	0.01642	0.00341	0.00005

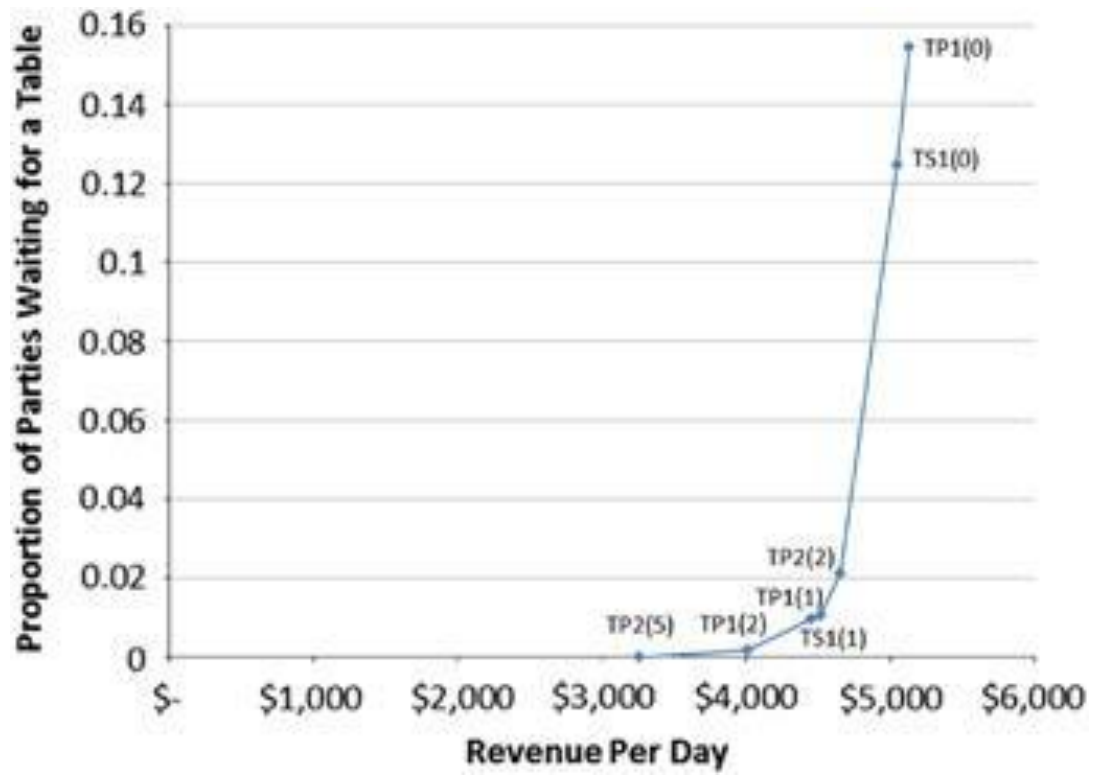


Figure 1. Pareto frontier of profitability and service level for the models.

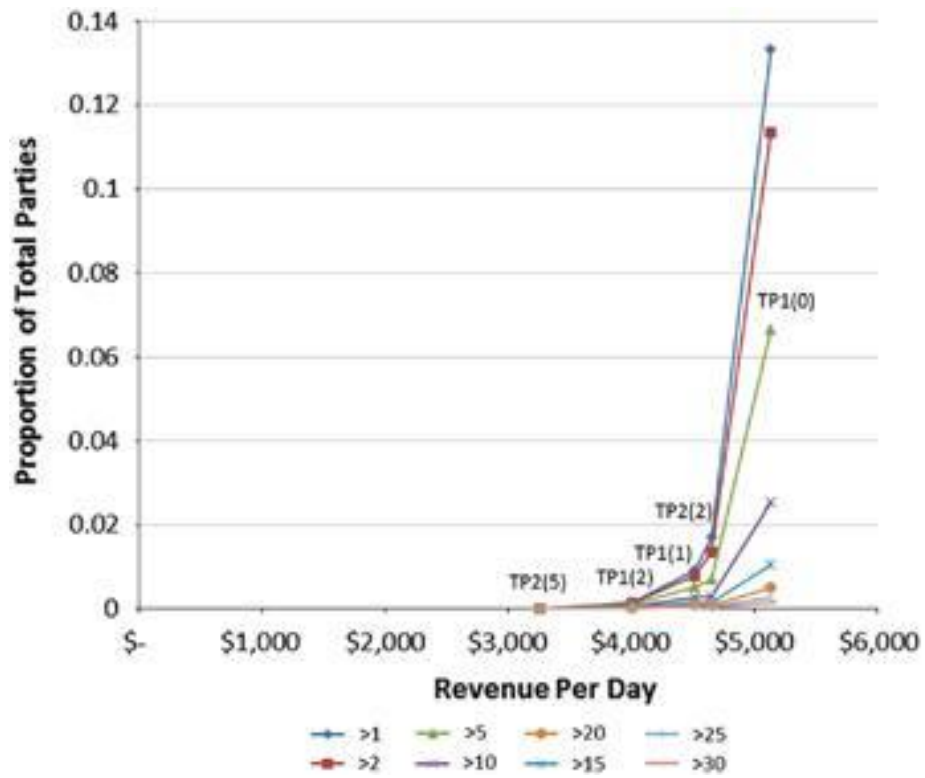


Figure 2. Detailed service-level results for the five Pareto optimal pooled-table models.

References

- Alexandrov, A. and Lariviere, M.A. (2012) Are reservations recommended? *Manufacturing and Service Operations Management* 14(2): 218–230.
- Anonymous (2015) 2015 Restaurant Industry Pocket Factbook. Washington DC: National Restaurant Association.
- Bell, R. and Pliner, P.L. (2003) Time to eat: The relationship between the number of people eating and meal duration in three lunch settings. *Appetite* 41(2): 215–218.
- Bertsimas, D. and Shioda, R. (2003) Restaurant revenue management. *Operations Research* 51(3): 472–486.
- Cayirli, T. and Veral, E. (2003) Outpatient scheduling in health care: A review of literature. *Production and Operations Management* 12(4): 519–549.
- Collier, D. (1982) Aggregate safety stock levels and component part commonality. *Management Science* 28(11): 1296–1303.
- Dongarra, J. and Luszczek, P. (2011) LINPACK benchmark. In: D. Padua (ed.) *Encyclopedia of Parallel Computing*. New York: Springer, pp. 1033–1036.
- Gupta, D. and Denton, B. (2008) Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions* 40(9): 800–919.
- Gurobi (2013) Houston, TX: Gurobi Optimization, www.gurobi.com.
- Kimes, S.E. (2008) A Consumer's View of Restaurant Reservation Policies. Cornell University: Ithaca NY USA. Cornell Hospitality Report, 8.
- Kimes, S.E., Chase, R.B., Choi, S., Lee, P.Y. and Ngonzi, E.N. (1998) Restaurant revenue management: Applying yield management to the restaurant industry. *Cornell Hotel and Restaurant Administration Quarterly* 39(3): 32–39.
- Kimes, S.E. and Robson, S.K.A. (2004) The impact of restaurant table characteristics on meal duration and spending. *Cornell Hotel and Restaurant Administration Quarterly* 45(4): 333–346.

- Kimes, S.E. and Thompson, G.M. (2004) Restaurant revenue management at Chevys: Determining the best table mix. *Decision Sciences* 35(3): 371–391.
- Kimes, S.E. and Thompson, G.M. (2005) An evaluation of heuristic methods for determining the best table mix in full-service restaurants. *Journal of Operations Management* 23(2): 197–208.
- McDougall, G.H.G. and Levesque, T.J. (1999) Waiting for service: The effectiveness of recovery strategies. *International Journal of Contemporary Hospitality Management* 11(1): 6–15.
- Noone, B.M., Kimes, S.E., Mattila, A.S. and Wirtz, J. (2007) The effect of meal pace on customer satisfaction. *Cornell Hotel and Restaurant Administration Quarterly* 48(3): 231–244.
- Pasin, F., Jobin, M.H. and Cordeau, J.F. (2002) An application of simulation to analyse resource sharing among health-care organisations. *International Journal of Operations and Production Management* 22(4): 381–393.
- Sheu, C. and Wacker, J.G. (1997) The effects of purchased parts commonality on manufacturing lead time. *International Journal of Operations and Production Management* 17(8): 725–745.
- Thompson, G.M. (1999) Labor scheduling part 4: Controlling workforce schedules in real time. *Cornell Hotel and Restaurant Administration Quarterly* 40(3): 85–96.
- Thompson, G.M. (2010) Restaurant profitability management: The evolution of restaurant revenue management. *Cornell Hotel and Restaurant Administration Quarterly* 51(3): 308–322.
- Thompson, G.M. (2011a) Cherry-picking customers by party size in restaurants. *Journal of Service Research* 14(2): 210–213.
- Thompson, G.M. (2011b) Inaccuracy of the ‘naïve table mix’ calculations. *Cornell Hotel and Restaurant Administration Quarterly* 52(3): 241–252.
- Thompson, G.M. and Kwortnik Jr, R.J. (2008) Pooling restaurant reservations to increase service efficiency. *Journal of Service Research* 10(4): 335–346.

Thompson, G.M. and Sohn, H.J. (2009) Time- and capacity-based measurement of restaurant revenue. *Cornell Hospitality Quarterly* 50(4): 520–539.