Pesquisa Operacional (2018) 38(1): 73-86
© 2018 Brazilian Operations Research Society
Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
doi: 10.1590/0101-7438.2018.038.01.0073

RESTAURANT RESERVATION MANAGEMENT CONSIDERING TABLE COMBINATION

Qing Miao[1*], Yihua Li[2] and Xiubin B. Wang[3]

Received April 29, 2017 / Accepted January 10, 2018

ABSTRACT. This paper presents a case study of table reservation practice for restaurant business within Walt Disney World. A unique feature here is to consider table combination to capture revenue potentials from different party sizes and at different time periods. For example, a party of large size can be served by combining two or more small tables. A mixed integer programming (MIP) model is developed to make the reservation recommendation. We propose a rolling horizon reservation policy such that the value of a particular table is periodically evaluated and updated. This is a typical revenue management method in the airlines and other industries, the essence of which is to compare the future expected revenue with a currently offered price. Using historical data, numerical test shows a significant revenue improvement potential from our proposed model.

Keywords: Restaurant Revenue Management, Table Reservation Optimization, Linear Programming.

1 INTRODUCTION

This paper conducts a study for table reservation in the restaurant business based on the practices at Walt Disney World. The Disney theme park area restaurants have seen a significant increase of demand in recent years. Restaurants are often booked out well before the dining hours. Often parties of high revenue potentials are declined upon arrival because the seating capacity has been occupied by earlier arriving customers. Managing the increasing demand, pricing and table utilization is critical to the restaurant business profitability.

The key here is to determine the value for each seat of each table at each time period so that the reservation requests of different size can be decided to be either accepted or denied to maximize not only the table utilization but also the seat utilization. For instance, if a table size of four is

*Corresponding author.

[1] The Home Depot, 2455 Paces Ferry Rd SE, Atlanta, GA 30339-1834, United States, 979-450-8779.
E-mails: qing_miao@homedepot.com; miao.terran@gmail.com

[2] United Airlines, Willis Tower, 233 South. Wacker Drive, Chicago, IL 60606, United States, 781-801-2396.
E-mail: yihua.li@united.com

[3] Texas A&M University, College Station, TX 77843, United States, 979-845-9901. E-mail: bwang@tamu.edu

available tomorrow and if our forecast shows a strong demand for party of size four, we can make the decision whether to book this table out to a request from a part of size two.

The restaurant table reservation is a well-defined problem. Specifically, in this problem, a restaurant has a set of service tables. The tables each have a number of seats available for dining. There are a number of fixed time periods of a certain interval, thirty minutes in this study, in each of which there is a number of requests for reservations. The size of a reservation is represented by the number of seats requested. Each reservation brings deterministic revenue to the restaurant often relevant to the party size. In addition, the dining duration of a reservation could span several time periods, largely determined by the party size. Tables and seats are allowed to combine to satisfy demand of larger groups. The objective is to make assignment of reservations to tables in order to maximize the total expected revenue from the accommodated demand. The request that cannot find a table assignment is declined.

This represents a deterministic version of the restaurant reservation management problem. The average revenue and dining duration of each party size are obtained from estimates using historical data. We assume that a dining duration belongs to a normal distribution. According to the time of day (e.g., noon, afternoon, evening and night), we calculate the mean and the variance of dining duration for each party size, then add one sigma (σ) on the average of party dining duration to determine the duration period k to reduce party waiting time, where σ is the standard deviation.

This is a combinatorial problem to match parties of sizes to tables of different seating capacity. A special feature in this problem is to allow table combination. For example, a group of six could be assigned to a combination of tables with four and two seats respectively, or two tables of four seats each, or tables of other combinations according to the actual situation. In addition, we assume that demand starts at the beginning of every half hour from 16:00 to 22:00. In this paper, a total of 12 different dining periods are considered each day according to market segmentation, and party sizes range from one to eight. Since party sizes above eight are rare, we combine them into the size eight through special arrangements in the data.

We develop a *mixed integer programming* (MIP) model to make reservation assignment to tables that considers table combinations. We propose a reservation policy in such a way that the value of a particular table is periodically evaluated via the MIP algorithm. Each time new reservations are requested, the assignment model is applied. The *linear programming* (LP) relaxation of the table assignment model gives dual values of tables. The dual values are often referred to as bid price in the revenue management literature. A reservation is accepted if its expected revenue out pays the currently estimated bid price; and the reservation is rejected, otherwise. Equivalently, if the assignment model assigns tables to a reservation, it means that the potential revenue from this reservation is higher than the bid price of the tables assigned and that the reservation is accepted. The LP relaxation of this MIP model is repeatedly applied to periodically update the bid prices on a rolling time horizon as demand is being unfolded. This revenue management problem bears much similarity essentially to the literature in the airlines and other industries.

The paper is organized as follows. Section 2 provides a literature review. In section 3, we present a MIP program for table assignment without table combination followed by a MIP and an efficient heuristic that consider table combination. Section 4 presents numerical tests. We conclude this paper in Section 5.

2 LITERATURE REVIEW

According to Liozu et al. (2012), of the three main approaches to pricing in industrial markets – cost-based, competition-based and valued-based – the last is considered superior by most marketing scholars (Anderson & Narus, 1998; Ingenbleek et al., 2003; Hinterhuber, 2004) and pricing practitioners (Forbis & Mehta, 1981; Dolan & Simon, 1996; Nagle & Holden, 2002; Fox & Gregory, 2004). This statement is particular true in this restaurant case since the cost-based pricing would lower the price to almost zero (without considering the opportunity cost) and competition-based pricing is impossible to evaluate due to the lack of similar product (Disney theme park restaurant). Among the major obstacles to implement value-based pricing (Hinterhuber, 2008), the difficulties in making value assessments are solved by the mathematical model presented in the next section, and the difficulty in customer communication is not a problem because the restaurant simply accepts or rejects reservation request. Sales force is trained and regulated by the software built upon our model, which help reducing operational complexity. Most importantly, most the senior management trusts the value-based pricing approach.

The pricing and revenue management can be dated back to as early as the 1970s (Littlewood, 1972). Today, revenue management has been extensively applied to media, hospital, cruise line, car rental and hotel room reservation management (Geragphty & Johnson, 1997; Ladany & Arbel, 1991; Lieberman, 2004; McGill & van Ryzin, 1999; Li et al., 2014, 2016). It is considered critical to the profitability of an industry.

Compared to the widespread applications mentioned above, the applications of revenue management to restaurants attract less attention until recent years. The study of Kimes et al. (1998) was the first attempt to use revenue management concepts. They introduced the revenue per available seat hour as a performance metric and as a means to compare different party sizes. Later, Bertsimas & Shioda (2003) proposed two classes of rigorous dynamic seating models to address trade-offs between expected revenue, average waiting time, and perceived fairness. These studies indicated that the forecasting of the customers and the related size of parties is important to restaurant revenue management. Along this direction, Kimes (1999), Kimes et al. (1997), Tsai & Kimes (2009), and Weatherford & Kimes (2003) discussed several methods and the accuracy of forecasting. Another important factor in restaurant revenue management is the perception of customers, ranging from the table spacing (Robson et al., 2011) to food price (Yang et al., 2009). In addition, (McGuire et al., 2010) proposed a framework to evaluate the waiting tolerance of customers.

It is observed that restaurants may combine small tables to serve large parties during the peak hour. Some literature focuses on the table mix to maintain the capacity of a restaurant (Kimes & Thompson, 2004, 2005; Thompson, 2003, 2011b) and they tried to develop some high level strategies to determine the best table mix for a restaurant. Since small parties typically spend more per person compared to larger parties, and they take shorter time to occupy the table, the restaurant might not be willing to serve large parties, especially for a small size restaurant. Interestingly, Thompson (2002) found that for restaurants bigger than 50 seats, combining tables may not always be profitable. The issue is that tables to be combined usually don't free up at the same time, necessitating that a table be placed on hold, until the adjacent table becomes free. A recent study (Thompson, 2011a) examined a phenomenon of so called cherry-picking customers – serving the high value customers and denying service to low value customers. Size of tables is found to be one of the important factors influencing this phenomenon.

Recent study of Wang et al. (2017) identified the mismatch between size and value when a restaurant that receives booking requests from customer parties elects to cherry-pick among the intending parties, and proposed the average spending per person per minute is a better indicator for evaluating any intending party's real value. Thompson (2016) developed a strategic planning tool to determine the best mix of tables in a restaurant. Based on the user inputs, the toll, which uses the Solver add-in for Microsoft Excel, returns the recommended table mix for greatest revenue.

This paper presents a novel valued-based pricing approach and a MIP model to assign table tops to reservations to maximize the total expected revenue. The essence here is to evaluate customer characteristics and table capacity so that each table is priced optimally. Then the table reservations are treated as products and are sold to the parties having higher revenue potentials only. This problem has a unique feature that allows table combination to satisfy requests from large groups when profitable.

We adopt SAS OR$^{\text{TM}}$ (version 9.22) to solve the mathematical model. Table combination implies production of a new product. The elementary tables can be considered as resources. These resources yield products. Another unique feature of this problem is that the dining duration of each party spans multiple periods, and is of varying length.

3 PROBLEM FORMULATIONS AND ALGORITHM

In this section, we first present an LP model that does not allow combining tables.

3.1 Table Assignment without Combination

The assignment model without table combination provides a baseline to the revenue management.

$$\text{[Program 0]:} \qquad \text{Maximize} \sum_{i \in P} \sum_{j \in T} r_i x_{ij} \tag{1.1}$$

Subject to:

$$\sum_{j \in T} x_{ij} \leq 1 \qquad \forall i \in P \tag{1.2}$$

$$x_{ij} \leq b_{ij} \qquad \forall j \in T, \quad \forall i \in P \tag{1.3}$$

$$\sum_{i \in P} \delta_{ik} x_{ij} \leq 1 \qquad \forall j \in T, \quad \forall k \in I \tag{1.4}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in P, \quad \forall j \in T \tag{1.5}$$

where:

I set of all 30 minute intervals along a day.

P set of all parties.

T set of all tables.

r_i expected revenue from party i.

δ_{ik} indicator variable which equals to 1 if party i stays over period k; 0 otherwise.

x_{ij} indicator variable which equals to 1 if party i is assigned to table j; 0 otherwise.

b_{ij} predefined incidence variable which equals to 0 if party i has a size larger than the seating capacity of table type j; 1, otherwise.

Constraint (1.2) means that each party can be assigned to a table at most once. Constraint (1.3) limits the assignment of party i to table j through the use of b_{ij}. For instance, if the size of party i is bigger than that of table j, b_{ij} is preset to 0 to avoid the assignment. Constraint (1.4) ensures that each table, at a moment k, can only take at most one party.

In the LP relaxation model, the dual value associated with constraint (1.4) presents the bid-price for table j over moment k. This is defined as bid-price and it is the most important indicator to guide the decision when the customer requests a reservation. The reservation is accepted if its expected revenue out pays the bid price; and the reservation is rejected, otherwise. The model is repeatedly applied to give update of bid prices as demand is being unfolded.

3.2 Table Assignment with Combination

Table combination normally results in an increased capacity to serve the larger parties and an increased utilization of, and therefore an increased value of, smaller tables.

Our next model, Program 1, explicitly considers table combination, and it expands the model in Program 0. For example, a party of size six can be accommodated with three tables of size two, or two tables of sizes two and four respectively, or two tables of both sizes four. In our model below, we specifically propose a way to model table combination by using the ratio of table size over party size. Denote by θ_{ij} the ratio of the jth table size over the ith party size. We require that the sum of ratios from tables assigned to a party be no less than 1.0. Take an example of a party of six just mentioned, the total sum of ratios according to the three assignment methods is as follows, where the numerators are for seating capacities and denominator the party size.

$$\text{Assignment 1: } 2/6 + 2/6 + 2/6 \geq 1,$$

$$\text{Assignment 2: } 2/6 + 4/6 \geq 1,$$

$$\text{Assignment 3: } 4/6 + 4/6 \geq 1.$$

Before we present the formulation, we introduce the notation below.

T set of regular and dummy tables.

$\widehat{T}$ set of dummy table, where $\widehat{T} \in T$.

I set of 30 minute intervals along a day.

P set of all parties.

b_{ij} parameter which equals to 0 if party i has a size larger than the seating capacity of table type j; 1, otherwise.

δ_{ik} parameter which equals to 1 if party i stay over period k.

r_i revenue associated with party i.

θ_{ij} ratio of size of table j over size of party i.

x_{ij} variable which equals to 1 if party i is assigned to table j; 0 otherwise.

y_i binary variable and set to 1 if party i is assigned to at least one table (one party could be assigned to a number of combined tables, e.g. a party with 4 people can be assigned to 2 tables with size 2); 0 otherwise. y_i indicates reservation accepted when it is 1; rejected otherwise.

c_{ij} cost associated with assigning party i table j.

[Program 1]:

$$\text{Minimize} \sum_{i\in P} r_i \sum_{j\in\tau} x_{ij} + \sum_{i\in P} \sum_{j\in T\backslash\tau} c_{ij}x_{ij} \tag{2.1}$$

Subject to:

$$\sum_{j\in T} \theta_{ij}x_{ij} \geq 1 \qquad \forall i \in P \tag{2.2}$$

$$\sum_{i\in P} \delta_{ik}x_{ij} \leq 1 \qquad \forall j \in T, \quad \forall k \in I \tag{2.3}$$

$$x_{ij} \leq b_{ij} \qquad \forall j \in T, \quad \forall i \in P \tag{2.4}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in P, \quad \forall j \in T \tag{2.5}$$

In Program 1, every reservation request is assigned to a table or tables, either to regular or dummy ones. We assume that enough dummy tables are available to accommodate all reservations. The objective function minimizes the loss of revenue due to service by dummy tables and due to table combination. c_{ij} corresponds to extra labor to construct a table from resources; c_{ij} is empirically set to \$5 in our numerical test. Obviously, when providing service without table combination is feasible, then tables should not be combined.

We allow one party to be assigned to multiple tables if the total seating capacity of the tables assigned to this party is equal to or greater than the party size, which is represented by constraint (2.2). Constraint (2.3) mandates each table always be assigned to no more than one party at any time. Constraint (2.4) means the assignment is limited by the table's availability or size. Constraint (2.5) is a binary one. *Similar with constraint (1.4), the dual values associated with constraint (2.3) represent the bid-price to guide reservation decisions.* If dummy tables are not explicitly considered, an alternative model is proposed as follows.

[Program 2]:

$$\text{Maximize} \sum_{i\in P} r_i y_i - \sum_{i\in P}\sum_{j\in T} c_{ij}x_{ij} \tag{3.1}$$

Subject to:

$$\sum_{j\in T} \theta_{ij}x_{ij} \geq y_i \qquad \forall i \in P \tag{3.2}$$

$$\sum_{i\in P} \delta_{ik}x_{ij} \leq 1 \qquad \forall j \in T, \quad \forall k \in I \tag{3.3}$$

$$x_{ij} \leq b_{ij} \qquad \forall j \in T, \quad \forall i \in P \tag{3.4}$$

$$x_{ij} = 0, 1 \qquad \forall i \in P, \quad \forall j \in T \tag{3.5}$$

$$y_i = 0, 1 \qquad \forall i \in P \tag{3.6}$$

The objective function in (3.1) maximizes the total revenue from assigning the parties to the tables while it minimizes the table combination since combining tables always comes with some labor cost. The minimization of the labor cost is a secondary priority for this objective function. Therefore, the value of parameter c_{ij} should be very small compared to r_j. This cost can be an arbitrarily small value associated with variable x_{ij}.

Constraint (3.2) means each party must be covered with sufficient seats if it is assigned to table(s). We illustrate through an example. If a party of six is assigned to tables of two and four seats, then $2/6 + 4/6 = 1$; if a party of six is assigned to two tables of four seats each, we have accordingly $4/6 + 4/6 = 4/3$: both satisfy constraint (3.2).

Constraint (3.3) prescribes that in each interval, a table can have at most one party at time. *Similar with constraint (1.4) and (2.3), the dual values associated with constraint (3.3) represent the bid-price to guide reservation decisions.*

Constraint (3.4) limits the assignment to table availability and size. Constraints (3.5) and (3.6) are binary constraints.

Proposition 1. Program 1 and Program 2 have the same set of table assignments and an identical total revenue from their respective optimal assignments.

Proposition 1 is clear by considering a special situation as follows. Set aside one and only one dummy table exclusively for each reservation. Suppose that dummy table j is set aside for reservation i, which means $x_{ij} \leq 1$. Additionally, make $\theta_{ij} = 1.0$ and $\delta_{ij} = 1.0$. Slightly changing constraint (3.2) into the form $\sum_{k\in T\setminus\{j\}} \theta_{ik}x_{ik} \geq 1.0 - x_{ij}, \forall i \in P$, one can see that both programs are equivalent by making $y_i = 1 - x_{ij}$. In addition, setting more or less dummy tables of varying sizes would not improve Program 1. One can also find the objective functions of Program 2 is the opposite of that of program 1 plus a constant term $\sum_{i\in P}\sum_{j\in T} r_i x_{ij}$.

3.3 Heuristic Method for Table Assignments

In the following, we always refer to Program 2 as our new model. Program 2 can be directly solved by the *branch-and-bound (B&B)* algorithm for almost all the cases as in our subsequent tests through SAS OR$^{\text{TM}}$ version 9.22. The direct adoption of B&B method does not guarantee an optimal solution within a reasonable amount of time. We propose a heuristic to find an initial feasible solution to Program 2 to accomplish two goals: to have a high quality initial solution to read in the B&B process to expedite the solution (Geoffrion et al., 1972), and the other is to always ensure that a solution is found quickly. Our numerical experiments show that the heuristic provides an acceptable solution. The heuristic can be implemented in an environment in which optimization software is not available.

Below, we introduce the proposed heuristic to which we refer to as Algorithm 1.

Algorithm 1 – Table assignment.

Initialization

Sort parties according to the order of their arrivals (the primary criterion) and in the decreasing order of revenue (their secondary criterion). Next, tables are sorted in the decreasing order according to the table size. Initialize all $x_{ij} = 0$. (Note that $x_{ij} = 1$ if party i is assigned to table j; 0 otherwise).

Iterations

Step 1. From the sorted party list, find the top unassigned party and compare with the top unassigned table in the sorted table list. If the top unassigned table does not fit, go down the sorted table list until a feasible table is found. If no single table can satisfy the party, try combine tables from the top unassigned tables in the same sorted table list. If no combined tables can satisfy the party, assign party to a dummy table, which has a very high cost. Find a feasible assignment by allowing table combination to the reservation with the maximum revenue, go to Step 2; if there is no feasible assignment identified, check off this reservation from the reservation list and go to Step 3.

Step 2. Set the according $x_{ij} = 1$. Determine the available table time occupied by the assigned party. Select the assigned party from the sorted party list. Go back to Step 1.

Step 3. Stop if the reservations are completed, i.e., a feasible assignment of parties to tables is identified. Otherwise, go to step 1.

The feasible solution from Algorithm 1 serves as a lower bound to Program 2. Program 2 can be set with a time limit or the optimality gap to further improve the solution by the method branch and bound. Note that by slightly modifying the procedure accordingly, this heuristic also applies to the case without table combination.

4 NUMERICAL TESTS

The numerical test uses a DELL OPTIPLEX 760 machine with an Intel Core (TM) 2 Quad processor Q8400 both with 2.66 GHz and 4.00 GB of RAM. The operating system is Windows 7. The optimization software SAS OR^{TM} version 9.22 is used to solve the Program 2. We set a time limit on the maximum solution time to be 3600 seconds for each instance.

The demand is provided based on the restaurant's historical data. A sample demand data around dinner time is shown on Table 1, where the arrival and departure times indicate the dining periods of the parties. The departure time is estimated by our algorithm according to the historical dining deuration. A sample party data is illustrated on Table 2. We compare the performances between direct application of Program 2 with table combination, Program 0 without table combination (PRGM 0), heuristic Algorithm 1 (HEU) and a simple first-come-first-serve method (FCFS) that allows table combination. Even though the program without table combination is not realistic, it provides a baseline to benchmark the performances of the other methods.

Table 1 – An Example of the Demand Forecast.

Index No.	Party Size	Arrival Time	Estimated End Time	Forecasted Demand
1	1	16:00	16:30	6
2	2	16:00	16:30	8
3	3	16:00	16:30	2
4	4	16:00	16:30	8
5	5	16:00	16:30	1
6	6	16:00	16:30	1
7	7	16:00	16:30	1
8	8	16:00	16:30	4
9	9	16:00	16:30	0
10	10	16:00	16:30	0
11	1	16:30	17:00	6
12	2	16:30	17:00	8
⋮	⋮	⋮	⋮	⋮
98	8	21:00	21:30	4
99	9	21:00	21:30	0
100	10	21:00	21:30	0

Table 2 shows an example expected revenue and time duration of each party size.

The comparison among four methods is conducted with a comprehensive set of instances. Each instance is characterized by respective mix of table sizes and party sizes. Table 3 presents 6 cases, each having a different mix of party sizes. Table 4 shows the mixed table sizes. Combination of Table 3 and Table 4 provides an extensive range of realistic situations with which the models are evaluated.

Table 2 – Characteristics of the Dining Parties.

Party Size	Expected Revenue	Duration in Minute
1	$23.93	61.2
2	$63.47	70.3
3	$86.14	71.4
4	$109.46	73.3
5	$134.31	74.7
6	$166.54	77.2
7	$189.22	78.4
8	$224.65	80.7
9	$258.63	82.6
10	$280.53	83.7

Table 3 – Instances of Parties (total parties for each case = 352).

	Size 1	Size 2	Size 3	Size 4	Size 5	Size 6	Size 7	Size 8
Case1	22	66	44	33	11	33	55	88
Case2	22	46	44	33	44	55	40	68
Case3	22	48	34	23	54	65	60	46
Case4	0	0	88	0	88	0	88	88
Case5	0	0	88	0	132	0	88	44
Case6	0	0	88	0	132	44	44	44

Table 4 – Number of Tables.

	1 seat	2 seats	3 seats	4 seats	5 seats	6 seats	7 seats
Mix1	0	5	0	58	0	0	0
Mix2	0	5	0	53	0	5	0
Mix3	0	5	0	48	0	5	5

(Note: Total number of tables for each mix is 63)

Table 5 shows the performances of four methods that we study in this paper. The table specifies the computational time (CPU measured in second) followed by the number of the parties, the total number of guests served (guest), and the total revenue (revenue). Due to the nature of the MIP program, some instances, as marked with an asterisk (*), are not solved optimally within one hour limit. In that case, a percentage gap between the best solution found and the best upper bound to the optimal solution through the B&B algorithm is provided in parenthesis. The solution time in those cases is the time the best solution was first found. For example, in Case 1 and Mix 1, Program 2 does not solve optimally within an hour. However, the best solution by the end of an hour was first found in 664 seconds. Program 2 solves a case optimally if no optimality gap is specifically labeled for a case. Note that Program 0 solves all the cases optimally almost instantly.

Table 5 – Results from Models.

	Models	Table: Mix 1				Table: Mix 2				Table: Mix 3			
		CPU	party	guest	Revenue	CPU	party	guest	Revenue	CPU	party	guest	revenue
Case 1	FCFS	0.14	161	790	20885	0.13	173	847	22477	0.18	184	925	24661
	PRGM 0	0.00	165	418	11242	0.00	185	538	14482	0.00	205	678	18162
	HEU	0.22	112	872	23156	0.21	112	872	23156	0.22	120	928	24668
	PRGM 2	664*	140	936	24728 (0.15%)	454	156	980	26064 (Opt)	2730*	165	1037	27606 (1.24%)
Case 2	FCFS	0.13	156	761	19954	0.14	168	819	21566	0.14	179	885	23403
	PRGM 0	0.00	145	378	10082	0.00	165	498	13322	0.00	185	637	16980
	HEU	0.22	110	834	21990	0.22	121	843	22354	0.23	119	890	23527
	PRGM 2	778	144	908	23984 (Opt)	222	154	968	25610 (Opt)	425	169	1027	27304 (Opt)
Case 3	FCFS	0.13	149	744	19370	0.13	161	797	20855	0.17	174	871	22892
	PRGM 0	0.00	127	312	8348	0.00	147	432	11588	0.00	167	572	15268
	HEU	0.21	110	809	21121	0.20	110	809	21121	0.22	118	859	22491
	PRGM 2	549*	135	883	23202 (0.01%)	699*	150	941	24776 (0.15%)	2906*	159	1002	26507 (0.07%)
Case 4	FCFS	0.14	138	783	20315	0.16	148	848	22100	0.18	161	913	23918
	PRGM 0	0.00	88	264	7128	0.00	108	364	9708	0.00	128	504	13388
	HEU	0.21	112	872	23156	0.22	112	872	23156	0.22	120	928	24668
	PRGM 2	237*	120	888	23284 (3.48%)	1410*	126	936	24593 (1.11%)	2336*	139	1001	26353 (1.17%)
Case 5	FCFS	0.16	140	740	18988	0.15	149	794	20457	0.13	161	852	22055
	PRGM 0	0.00	88	264	7128	0.00	108	364	9708	0.00	128	504	13388
	HEU	0.20	114	842	21930	0.20	114	842	21930	0.22	120	884	23064
	PRGM 2	2720*	136	858	22290 (1.76%)	642*	136	886	22964 (2.90%)	1801*	156	962	25068 (0.27%)
Case 6	FCFS	0.16	140	724	18636	0.16	150	780	20169	0.17	161	835	21681
	PRGM 0	0.00	88	264	7128	0.00	108	384	10368	0.00	128	524	14048
	HEU	0.22	112	804	21044	0.21	120	852	22380	0.22	129	906	23898
	PRGM 2	973	153	852	22543 (Opt)	1802*	163	902	23857 (0.63%)	528*	173	958	25475 (0.95%)

The CPU time for Program 0 is recorded by SAS OR$^{\text{TM}}$ which does not count the time for reading in data. The CPU time for both FCFS and HEU includes the data reading time, and therefore is larger than in Program 0. If Program 0 considers data reading time, the CPU is almost identical among four Models.

The comparison shows table combination has a significant revenue implication. The heuristic method proposed is generally within 90 percent of the optimal solution, and in many cases, 95 percent of optimality. When using SAS OR$^{\text{TM}}$ version 12.0, the initial solution from Algorithm 1 does not significantly help expedite the B&B solution to Program 2. However, when using an earlier version of SAS OR$^{\text{TM}}$, the initial solution from Algorithm 1 helps the B&B process in Program 2 significantly. Compared with the FCFS method, our proposed Program 2 increases revenue by up to 20 percent.

5 CONCLUSION

As a research project at Walt Disney World, we study a table reservation problem for restaurant revenue management. This is a special revenue management problem with unique features compared with the airline and hotel industries. In this problem, resources are each provided in blocks (e.g. tables with multiple seats), and that products (e.g. a combination of one or more tables for parties of varying sizes) consume multiple resources. A block of resource cannot be split to satisfy two different requests. In addition, each request has a varying dining time. We propose a value-based approach to price each table as a resource via linear programming, which is then used to guide real time reservation decisions. A MIP model is solved in both optimal and heuristic methods to achieve assignment when planning.

For the sake of simplicity, we do not consider cancellations and no no-shows. But this study provides a good basis from which to address the cancellations in future studies. In addition, there is no consideration on the loss of productive time when tables need to be physically combined. Ideally, the tables should be combined with adjacent ones. These physical constraints and moving inconvenience are not considered in this study.

Worthy of a mention, the MIP model can be implemented directly and solved with the B&B algorithm in commercial software such as SAS^{TM} optimizer or Cplex^{TM}. To solve the problem with guaranteed computational time, we propose a heuristic algorithm. This heuristic solution performs satisfactorily, reaching about 90% of optimality on average.

The implementation includes two applications. The first application simply compares the current bid price for each table with the expected revenue from the ongoing reservation. This is a real time process and it takes no time to either accept or reject the reservation party. Customers do not have to wait. The second application provides update to the current bid price. It periodically runs one of the table assignment models in section 3 based on the current booking situation and remaining forecast. Depending on serve's computing power, the model may take seconds to minutes to complete. This means the bid price in application 1 is updated at least every a few minutes.

Our extensive numerical tests based on the operations data show a significant revenue improvement potential from our proposed model that allows table combination. Given the daily revenue increases per restaurant in the testing cases, it is expected to have a significant revenue increase over the 43 locations in a year. We believe this proposed optimization method has general application potential in the restaurant industry.

REFERENCES

[1] Anderson JC & Narus JA. 1998. Business marketing: Understand what customers value. *Harvard Business Review*, **76**(6): 53–65.

[2] Bertsimas D & Shioda R. 2003. Restaurant revenue management. *Operations Research*, **51**: 472–486.

[3] Dolan R & Simon H. 1996. Power Pricing: How Managing Price Transforms the Bottom Line. Boston, MA: The Free Press.

[4] Forbis J & Mehta N. 1981. Value-based strategies for industrial products. *Business Horizons*, **24**(3): 32–42.

[5] Fox J & Gregory R. 2004. The Dollarization Discipline: How Smart Companies Create Customer Value – and Profit From It. Hoboken, NJ: John Wiley & Sons.

[6] Geragphty MK & Johnson E. 1997. Revenue management saves National Car Rental. *Interfaces*, **27**: 107–127.

[7] HINTERHUBER A. 2004. Towards value-based pricing – An integrative framework for decision making. *Industrial Marketing Management*, **33**(8): 765–778.

[8] HINTERHUBER A. 2008. Customer value-based pricing strategies: why companies resist. *Journal of Business Strategy*, **29**(4): 41–50.

[9] INGENBLEEK P, DEBRUYNE M, FRAMBACH R & VERHALLEN R. 2003. Successful new product pricing practices: A contingency approach. *Marketing Letters*, **14**(4): 289–305.

[10] KIMES SE. 1999. Group forecasting accuracy in hotels. *Journal of the Operational Research Society*, **50**: 1104–1110.

[11] KIMES SE, CHASE RB, CHOI S, LEE PY & NGONZI EN. 1998. Restaurant revenue management: Applying yield management to the restaurant industry. *Cornell Hotel and Restaurant Administration Quarterly*, **39**: 32–39.

[12] KIMES SE, KIM YS, PARK J & TERRY CL. 1997. Development and application of hotel group forecasting. *Decision Sciences Institute, 1997 Annual Meeting, Proceedings*, **1-3**: 1560–1562.

[13] KIMES SE & THOMPSON GA. 2005. An evaluation of heuristic methods for determining the best table mix in full-service restaurants. *Journal of Operations Management*, **23**: 599–617.

[14] KIMES SE & THOMPSON GM. 2004. Restaurant revenue management at Chevys: Determining the best table mix. *Decision Sciences*, **35**: 371–392.

[15] LADANY S & ARBEL A. 1991. Optimal cruise-linear passenger cabin pricing policy. *European Journal of Operational Research*, **55**: 136–147.

[16] LI Y, MIAO Q & WANG B. 2016. Stochastic model for hotel room pricing and upgrading. *Journal of Revenue and Pricing Management*, **15**: 500–508.

[17] LI Y, MIAO Q & WANG B. 2014. Modeling A Cruise Line Revenue Management Problem. *Journal of Revenue and Pricing Management*, **13**: 247–260.

[18] LIEBERMAN WH. 2004. Revenue management in the health care industry. Thomson, London.

[19] LIOZU SM, HINTERHUBER A, BOLAND R & PERELLI S. 2012. The conceptualization of value-based pricing in industrial firms. *Journal of Revenue & Pricing Management*, **11**(1): 12–34.

[20] LITTLEWOOD K. 1972. Forecasting and control of passengers bookings. AGIFORS Symposium Proc.

[21] MCGILL JI & VAN RYZIN G. 1999. Revenue management: Research overview and prospects. *Transportation Science*, **33**: 233–256.

[22] MCGUIRE KA, KIMES SE, LYNN M, PULLMAN ME & LLOYD RC. 2010. A framework for evaluating the customer wait experience. *Journal of Service Management*, **21**: 269–290.

[23] NAGLE T & HOLDEN R. 2002. The Strategy and Tactics of Pricing: A Guide to Profitable Decision Making. Upper Saddle River, NJ: Prentice Hall.

[24] ROBSON SKA, KIMES SE, BECKER FD & EVANS GW. 2011. Consumers' Responses to Table Spacing in Restaurants. *Cornell Hospitality Quarterly*, **52**: 253–264.

[25] THOMPSON GM. 2002. Optimizing a restaurant's seating capacity: Use dedicated or combinable tables? *The Cornell Hotel and Restaurant Administration Quarterly*, **43**(4): 48–57.

[26] THOMPSON GM. 2003. Optimizing restaurant-table configurations: Specifying combinable tables. *The Cornell Hotel and Restaurant Administratiion Quarterly*, **44**(1): 53–60.

[27] THOMPSON GM. 2011a. Cherry-Picking Customers by Party Size in Restaurants. *Journal of Service Research*, **14**: 201–213.

[28] THOMPSON GM. 2011b. Inaccuracy of the "Naive Table Mix" Calculations. *Cornell Hospitality Quarterly*, **52**: 241–252.

[29] THOMPSON GM. 2015. Restaurant Reservations Optimization Tool. *Cornell Hospitality Tool*, **6**(4): 3–10.

[30] TSAI TH & KIMES SE. 2009. A Time Series Case-Based Predicting Model for Reservation Forecasting. *Opportunities and Challenges for Next-Generation Applied Intelligence*, **214**: 53–58.

[31] WANG J, LIN Y, KUO C & WENG S. 2017. Cheery-picking restaurant reservation customers. *Asia Pacific Management Review*, **22**: 113–121.

[32] WEATHERFORD LR & KIMES SE. 2003. A comparison of forecasting methods for hotel revenue management. *International Journal of Forecasting*, **19**: 401–415.

[33] YANG SS, KIMES SE & SESSAREGO MM. 2009. Menu price presentation influences on consumer purchase behavior in restaurants. *International Journal of Hospitality Management*, **28**: 157–160.