Optimizing the Mix and Locations of Slot Machines on the Casino Floor

Jason Fiege and Anastasia Baran
nQube Data Science Inc.
Winnipeg, MB, CANADA

ABSTRACT: We present a mathematical framework and computational approach that aims to optimize the mix and locations of slot machine types and denominations, to maximize the overall performance of the gaming floor. We introduce a powerful multi-objective evolutionary optimization and data-modelling platform, developed by the presenter since 2002, and show how this software can be used for casino floor optimization. We develop a generalized non-linear parameterized model for casino floor performance that takes into account the mixture of machine types; their individual performances; their location on the gaming floor relative to other machines, table games, and attributes such as walls, walkways, and entrances; and their location within a bank of machines. Our model also fits, optimizes, and predicts the duty cycle of each slot machine on the gaming floor, defined as the fraction of time that a machine is played, which serves as a proxy for where players spend their time. Thus, the model presented combines a spatial model of the slot floor with a model of this aspect of player behavior. We show that machine-level performance data for individual machines can uniquely determine all model parameters, free of mathematical degeneracy, and we demonstrate that our model is robust against statistical noise. Once model parameters have been fit from data, we optimize the model to generate realistic, maximally performing casino floor configurations. These optimized slot floors are visualized using heat maps showing the spatial distribution of duty cycle, coin-in, and win on the floor.

1. Introduction

Slot machines in Nevada have continuously proven to be the largest revenue generators on the casino floor (Nevada Gaming Control Board 2016). This success has prompted research into methods to optimize the mix and location of machines in order to maximize their performance.

Studies have been conducted that analyze the effects of changing casino property variables on slot performance. For example, Lucas, A. F., & Roehl, W. S. (2002), analyze the effect of floor location on video poker machine performance and find that high traffic areas result in higher machine win. In another study, Lucas, A. F. (2014) examines the relationship between sports book wagering and performance of the slot floor. Variables such as a machines placement in a bank, ceiling height and proximity of aisles have also been studied in order to understand their effect on machine performance (Lucas, A. F., & Dunn, W. T. (2005)). These studies show that individual machine and overall casino attributes have some measurable effect on individual slot machine performance.

A particularly relevant study for the present work was conducted by Ghaharian, K. C. (2010), who used linear programming to perform an optimization of the mix of slot machines at a Las Vegas locals casino. His study used coin-in and win data from 1812 slot machines, divided into 19 types and denominations. Constraints were specified, corresponding to a 10% maximum change to the number of machines that the management might be willing to implement. This work presented results in which the coin-in and win performance metrics were separately maximized to find the ideal mix of machines on the floor.

We extend this study in two important directions. Firstly, we recast the problem within a multi-objective optimization framework, using our company's multi-objective optimization software to *simultaneously* optimize both the coin-in and win objectives, finding a trade-off curve (Pareto front) characterized by the property that win cannot be further improved without sacrificing coin-in, and vice-versa. Secondly, we develop a non-linear spatial model of the slot floor that takes into account effects on performance due to a machine's placement on the floor.

Our model self-consistently takes into account the effects of different rooms or regions on the floor, the position of a slot machine within a bank of machines, and the distance to other casino attributes such as major walkways, table games, walls, entrances, etc. Perhaps most importantly, we also model and predict the *duty cycle* of each machine on the floor, which we define as the fraction of the total time available that a particular machine is in use by a player. Duty cycle can be viewed as a proxy for where players spend time on the floor, and which regions are more or less crowded. Heat maps of this model output are especially informative (see Section 4.3). Some players may prefer exciting, crowded areas, while others may seek out more isolated areas. Thus, player crowding is determined by the spatial effects on performance noted above, but also directly impacts the performance of each machine in a highly non-linear fashion, which must be solved for self-consistently. The model developed below is therefore one that combines a spatial slot floor model with a model that captures some aspects of player preferences and behaviour, which we believe to be unique in this field.

We present numerical experiments below that use artificial data to demonstrate that all parameters of our model can be determined by modeling measurable machine-level performance data. We further show that once a model has been fit to the data, the mix and spatial locations of machines can be optimized to design a slot floor that maximizes both coin-in and win within our multi-objective optimization framework.

2. A Non-Linear Spatial Model of the Slot Floor

2.1. <u>The Efficient Slot Floor</u>

We begin with an idealized non-spatial model that describes player behavior on the casino floor. On this simplified efficient slot floor the players

1. are thoroughly mixed throughout the slot floor;
2. can easily find their preferred machine type;
3. are indifferent to spatial location on the floor;
4. have no preference for crowded or isolated regions;

5. and are indifferent to the placement of the machine on the bank.

These simplifications are relaxed in Sections 2.4 – 2.6, where we also add effects due to region, bank position, spatial location, and player crowding.

We define a player preference probability function ψ_i, which specifies the odds that a given player would prefer a slot machine of type i. The probability function is normalized such that $\sum_{i=1}^{M}\psi_i = 1$, where M is the number of machine *types* considered. There are N_0 machines in total on the floor, and N_i of each type *i*.

In general, there is a mapping

$$i = MT(n), \tag{1}$$

which maps a particular machine's unique identifier n to its type i. For the remainder of this paper, we will use indices n and m when referring to individual machine identifiers, and i and j when referring to the corresponding machine types. *Indices i and j will always be assumed to be functions of n and m.* We note that n and m can always be mapped to i and j respectively by equation (1), but the mapping is not one-to-one, since there are generally many more machines on the floor than machine types. Equation (1) is therefore not invertible.

On an efficient floor, the probability that a given player will choose machine n is given by

$$p_n^{EF} := \frac{\psi_i}{N_i}. \tag{2}$$

It is easy to show that p_n^{EF} is normalized. Consider a floor where the total number of machines N_0 is given by the sum over machine types:

$$N_0 = \sum_{i=1}^{M} N_i. \tag{3}$$

Then, noting that p_n^{EF} depends only on machine type i,

$$\sum_{n=1}^{N_0} p_n^{EF} = N_1 p_1^{EF} + N_2 p_2^{EF} + \cdots + N_M p_M^{EF} = \sum_{i=1}^{M}\psi_i = 1. \tag{4}$$

On a realistic gaming floor, none of the conditions of the efficient floor are generally satisfied. However, p_n^{EF} still provides an important benchmark probability that is useful to describe the overall dynamics of the floor. We express the deviation of the probability function p_n from p_n^{EF} by the equation

$$p_n := \frac{p_n^{EF}(1+\Delta_n')}{\sum_{n=1}^{N_0} p_n^{EF}(1+\Delta_n')}, \tag{5}$$

where $\Delta_n' \in [-1,\infty)$, for machine n. Note that equation (5) is normalized so that $\sum_{n=1}^{N_0} p_n = 1$.

Later in this paper, we propose a specific model for Δ_n' that takes into account spatial and player clustering effects, where the reason for the prime notation will also become clear. We note that $p_n = p_n^{EF}$ for the efficient model. Therefore, we will use p_n when an equation applies to both the efficient and non-efficient probability distribution, reserving p_n^{EF} for cases where the efficient distribution is strictly required.

2.2. Machine Duty Cycle and Non-Linear Effects of Gaming Floor Capacity

Consider a gaming floor containing N_0 machines in total, where at some particular time, there are P_0 players on the floor. We can then define the "busyness" of the floor as

$$Q := \frac{P_0}{N_0}, \tag{6}$$

which parameterizes the occupation of the gaming floor relative to its capacity. Considering the ebb and flow of the casino floor over an extended period of time, this busyness parameter follows a probability distribution $\theta(Q)$, which we assume to be normalized such that

$$\int_0^{Q_{max}} \theta(Q)dQ = 1. \tag{7}$$

Any function $f(Q)$ can be averaged over Q by

$$\langle f(Q)\rangle_Q := \int_0^{Q_{max}} \theta(Q)f(Q)dQ. \tag{8}$$

In practice, $\theta(Q)$ is only likely to be measured at N_q discretely binned values of Q. In this case, the average of function f over Q can be obtained by

$$\langle f(Q)\rangle = \sum_{q=1}^{N_q} \theta_q\, f_{q,} \tag{9}$$

where we assume the normalization

$$\sum_{q=1}^{N_0} \theta_q = 1. \quad \textbf{(10)}$$

We assume for the present model that $\theta(Q)$ is stationary; however, it would be straightforward to extend the model to the case where Q follows a non-stationary distribution that varies seasonally or over longer periods of time.

The duty cycle $f_n(Q)$ of a particular slot machine n, defined above as the fraction of time that it is used, is given by:

$$f_{n,q} \coloneqq f_n(Q) = \min\big(1, p_n P_0(Q)\big), \quad \textbf{(11)}$$

where the *min* function is necessary to ensure a physicality constraint requiring that no machine is utilized by more than one player at a time, regardless of the number of players on the floor, thus requiring that $f_{n,q} \in [0,1]$. Note that we have used p_n rather than p_n^{EF} here, since this result does not require the conditions of the efficient floor. It is easy to see, from the definition of Q in equation (6), that equation (11) can be written as

$$f_{n,q} = \min(1, p_n Q N_0). \quad \textbf{(12)}$$

We note that for an efficient slot floor, where $p_n = p_n^{EF}$, this equation reduces to

$$f_{n,q} = \min\left(1, \frac{\psi_i Q}{v_i}\right), \quad \textbf{(13)}$$

where

$$v_i \coloneqq \frac{N_i}{N_0}. \quad \textbf{(14)}$$

The *min* function in equations (12) and (13) introduces non-linearity due to the finite capacity of the gaming floor, which is apparent in the results shown in Section 4.1.

2.3. Coin-In and Win

Consider that a particular machine n of type i has a baseline coin-in performance CI_i^0, which would *hypothetically* be obtained if the machine were played continuously, with a duty cycle equal to one. This machine would have an actual coin-in performance, for an arbitrary duty cycle, of

$$CI_{n,q} := CI_n(Q) = CI_i^0 f_{n,q} \,, \qquad \textbf{(15)}$$

when the floor is operating at busyness Q. The total coin-in for the floor, averaged over Q, is given by:

$$CI := \sum_{n=1}^{N_0} CI_n = \sum_{q=1}^{N_q} \left(\theta_q \sum_{n=1}^{N_0} CI_i^0 \, f_{n,q} \right). \qquad \textbf{(16)}$$

For now, we assume that the win metric for a given machine is given by

$$WIN_i = WF_i \, CI_i \,, \qquad \textbf{(17)}$$

where WF_i is the time-averaged "win fraction", or the casino's average take as a fraction of the coin-in, for a machine of type i. In general, the win fraction can be calculated from the machine's pay table, but is subject to considerable statistical noise. This is especially true for games that feature very large but rare jackpots. Therefore, the coin-in and duty cycle are better quantities to use to constrain our model parameters from data.

2.4. The Δ Model of the Non-Efficient Floor

In this section, we build a more realistic, non-efficient, spatial model of the gaming floor, on the foundations of the efficient model. The Δ_n' parameter in equation (5) expresses the deviation from the efficient floor model for each machine n, such that the efficient model is recovered when $\Delta_n' = 0$ for all machines. This section proposes one possible formulation for Δ_n' that takes into account several attributes of the gaming floor that may affect player preferences. However, before embarking on a specific formulation, let us first consider the domain of this variable, which is given by $\Delta_n' \in [-1, \infty)$. The Δ_n' parameter has a finite lower bound, which expresses that the player preference probability function must be non-negative. The Δ_n' parameter does not require an upper bound, because equation (5) is normalized.

It is mathematically convenient to replace Δ_n' with an *unbounded* variable $\Delta_n \in (-\infty, \infty)$, where Δ_n and Δ_n' are related by the non-linear transformation:

$$\Delta_n' = T(\Delta_n), \qquad \textbf{(18)}$$

where $T(\cdot)$ is a non-linear function that maps $(-\infty, \infty) \longrightarrow [-1, \infty)$. Thus, equation (5) becomes

$$p_n \coloneqq \frac{p_n^{EF}\left(1 + T(\Delta_n)\right)}{\sum_{n=1}^{N_0} p_n^{EF}\left(1 + T(\Delta_n)\right)}. \tag{19}$$

The physicality constraint on the non-negativity of probabilities does not provide any further hints regarding the mathematical structure of T. However, it is only useful to consider monotonic, increasing functions. The simplest function with the required properties is given by

$$T(\Delta_n) = \max(-1, \Delta_n), \tag{20}$$

which we will use to obtain our numerical results below. Note however that $T(\Delta_n)$ encodes a fundamental non-linearity describing how player preferences respond to Δ_n, which encodes individual machine attributes, their placement within the casino environment, and effects of crowding by other players. The next section uses equation (20) to model and explore these preferences. However, it should be possible, with sufficiently detailed data, to further constrain the mathematical structure of T itself, thereby obtaining new insights into the non-linear response of players to their environment.

2.5. Framework for Δ_n

Further progress requires a specific model for Δ_n in equation (19). There are many possible formulations for such a model, so we choose one particularly simple version that should model some of the main effects that have been identified as contributing to the performance of machines on the slot floor. This specific model can be readily tested with data, and altered if necessary. Such testing will be the focus of future work.

We split $\Delta_{n,q}$ into four main components that add linearly, by the equation

$$\Delta_{n,q} = \Delta_n^R + \Delta_n^B + \Delta_n^S + \Delta_{n,q}^C, \tag{21}$$

where $\Delta_{n,q}^R$ represents the effect of the region of the slot floor that machine n is located in; $\Delta_{n,q}^B$ represents the effect of the position of the machine within a *bank* of machines; $\Delta_{n,q}^S$ represents detailed *spatial* effects due to the proximity of other machines, table games, walkways, walls, entrances, etc.; and $\Delta_{n,q}^C$ represents the effects of player *clustering,* recognizing that some players prefer more exciting, crowed regions, while other players prefer more isolated surroundings. Note that we have added a subscript q to Δ_n because any reasonable formulation of the clustering term would be dependent on the overall busyness Q of the casino, and therefore so would be the sum $\Delta_{n,q}$.

We discuss each of the terms of equation (21) in detail in this section.

Regional Preferences Δ_n^R:

Most slot floors can be divided coarsely into regions defined by different rooms, large groups of machines separated by broad walkways, alcoves, etc. These regions feature differences in architecture, atmosphere, and view. Thus, some regions may inherently perform better or worse due to the effect of these attributes on player preferences (Lucas, A. F., and Dunn, W. T. (2005)). This Δ can be parameterized simply by

$$\Delta_n^R = \delta_r^R, \tag{22}$$

where subscript $r = r(n)$ determines the region where machine n resides, and δ_r^R represents a set of free parameters of the model, requiring one free parameter for each of N_R regions.

Bank Position Preferences Δ_n^B:

It is widely believed that a machine's performance may be affected by the type of bank in which it resides, and its position within the bank (Dunn, W.T. (2014), Lucas, A. F., & Dunn, W. T. (2005)). It would not be unreasonable to assume that machines located in other types of banks, such as round banks for example, might also perform differently. Therefore, we parameterize Δ_n^B by the equation

$$\Delta_n^B = \delta_b^B, \tag{23}$$

where $b = b(n)$ is an index defining the *bank attribute* of the machine. Bank attributes follow an indexing scheme, where each b corresponds to a particular defined bank attribute. Therefore, δ_b^B represents a set of free model parameters, with one free parameter for each of N_B bank attributes considered.

Spatial Preferences Δ_n^S:

We assume that player preferences for machine n can be affected by the proximity of other attributes on the floor, such as other machines, table games, walkways, walls, washrooms, etc. We further assume that the influence of other attributes decreases with distance. The simplest way to parameterize Δ_n^S is with a power law:

$$\Delta_n^S = \sum_{m=1}^{N_A} \delta_{i,j}^S \, d_{n,m}^{-\gamma^S}, \tag{24}$$

where $\gamma^S > 0$, $d_{m,n}$ is the distance between machine n (of type i) and attribute m (of attribute type j), and $\delta_{i,j}$ determines the magnitude and sign of the effect of the attribute on the machine performance. Here, the attribute type index j is determined by the mapping $j = AT(m)$, in analogy to equation (1), and the total number of attributes affecting machine n is given by N_A. We assume that the proximity of all slot machine types has the same impact and assign $j = 1$ to all other slot machines, while other attributes are assigned higher values of j. The choice of a power law formulation is motivated by the fact that power laws are uniquely scale-free. Any other choice of function would require a distance scale parameter in addition to the $\delta_{m,n}$ strength parameter, thus doubling the number of parameters required. The total number of free parameters required by Δ_n^S is therefore given by $M * A$, where A is the number of attributes defined, including one attribute type to represent other slot machines.

Player Crowding Preferences $\Delta_{n,q}^C$:

It is reasonable to assume that players of different machine types vary, on average, in their preferences for busy areas of the gaming floor, or more isolated areas (Lucas, A. F., & Roehl, W. S. (2002)). Therefore, we parameterize this crowding preference by the equation

$$\Delta_{n,q}^C := \delta_i^C \sum_{m=1}^{N_0} f_{m,q}\, d_{n,m}^{-\gamma^C}, \qquad \textbf{(25)}$$

where δ_i^C describes the strength and sign of this clustering effect on machine performance, $f_{m,q}$ is the duty cycle of machine m when the casino is operating at busyness Q_q, and $d_{n,m}$ is the distance between machines n and m. We have chosen a power law formulation, with index γ^C, for the same reasons described in the previous section. We observe that $\Delta_{n,q}^C$ introduces additional non-linearity, since it explicitly depends on $f_{m,q}$, but is also required to calculate $f_{m,q}$ via equations (12) and (19) – (21).

2.6. A Non-Linear Model for Slot Performance and Player Clustering

Our model for machine duty cycle $f_{n,q}$ (and hence player clustering) is now fully defined. We note that the right hand side of equation (25) contains a sum over $f_{m,q}$. The inclusion of player clustering behavior in the model necessitates a self-consistent solution to the set of $N_0 * N_q$ simultaneous non-linear equations embodied in equation (12). This set of equations would typically be quite large, since $N_0 \sim 10^3$ for a typical gaming floor, and we envision at least a coarse sampling of $N_q \sim 10$ levels of busyness. An efficient, rapidly converging, reliable numerical algorithm is required to self-consistently solve this large, non-linear set of equations.

Once $f_{n,q}$ is known, the coin-in $CI_{n,q}$ of each machine is determined by equation (15), and the win metric can be calculated from equation (17).

3. Finding Model Parameters from Data

The non-linear model presented in the previous sections allows one to calculate $f_{n,q}$ and $CI_{n,q}$ given the spatial structure of a casino, the type of every machine on the floor, and all model parameters. The following table summarizes each type of model parameter, and the number required of each type:

Parameter symbol	Description	Number of free parameters
ψ_i	Player preference probability	M (number of machine types);
δ_r^R	Strength and sign of Δ^R (regional delta)	N_R (number of regions)
δ_b^B	Strength and sign of Δ^B (bank delta)	N_B (number of identified bank position attributes)
$\delta_{m,n}^S$	Strength and sign of Δ^S (spatial delta)	$M * A$ (number of machine types times number of attribute types)
δ_r^C	Strength and sign of Δ^C (clustering delta)	M (number of machine types);
γ^S	Spatial power law index	1
γ^C	Clustering power law index	1

Table 1. Summary of free model parameters.

We define $\boldsymbol{P}$ as a column vector that contains the entire set of parameters listed here. The length of $\boldsymbol{P}$ is given by the total number of free parameters:

$$\text{length}(\boldsymbol{P}) = (A+2)M + N_R + N_B + 2. \quad \textbf{(26)}$$

We assume that, by tracking machine-level transactions, the actual duty cycle $f_{n,q}^{data}$ and coin-in $CI_{n,q}^{data}$ can be measured for each machine on the slot floor at various Q values. Thus our task is to determine $\boldsymbol{P}$ given $f_{n,q}^{data}$ and $CI_{n,q}^{data}$. This can be accomplished by searching for the vector $\boldsymbol{P}$ that minimizes the statistic

$$F = \frac{(f_{n,q}^{model} - f_{n,q}^{data})^2}{\langle f_{n,q}^{data} \rangle^2} + \frac{(CI_{n,q}^{model} - CI_{n,q}^{data})^2}{\langle CI_{n,q}^{data} \rangle^2}, \quad \textbf{(27)}$$

where each term is normalized by the mean of $f_{n,q}^{data}$ and $CI_{n,q}^{data}$ respectively. This function bears some similarity to the usual χ^2 from statistics, but is not the same, due to the normalization in the denominator and the lack of an error term. The statistic could be modified to include an error term, if sufficient data were collected to determine the uncertainty of all $f_{n,q}^{data}$ and $CI_{n,q}^{data}$ in the data set. We note that $F = 0$ if the model fits the data exactly. This would never occur in practice, due to statistical noise and the certainty that our model does not capture all of the complicated effects that actually determine machine performance.

Searching for the optimal $\boldsymbol{P}$ is challenging due to the large number of parameters involved. For example, a simple test presented in the Section 4, with only three machine types, has 29 parameters, and the number of parameters increases linearly with M by equation (26). Large, non-linear optimization problems are often very challenging and require sophisticated computational methods. Local, gradient-based optimization methods are not usually useful for problems as difficult as this one, since they are easily trapped by local minima. For this problem, is not practical to calculate numerical gradients regardless, due to numerical noise introduced by the iterative numerical method required to solve equation (12). It is also not feasible to sample all possible parameter sets for such large problems. Thus, this problem requires efficient, well-tested heuristic optimization methods that do not require gradient information, to search the parameter space efficiently while avoiding local minima.

3.1. The Ferret Genetic Algorithm

Our company specializes in the design of algorithms for large-scale parameter search and optimization problems. Our most powerful optimizer is an evolutionary code called "Ferret", which is part of our Qubist Global Optimization Toolbox. Ferret began in 2002 as a multi-objective genetic algorithm, but has since "evolved" beyond the genetic algorithm paradigm (Holland, J. H. (1992) and Goldberg, D. E. (2002)). Ferret incorporates a unique linkage-learning algorithm that automatically searches for opportunities to divide large problems into several smaller, and easier, semi-independent ones. Thus, it has the ability to greatly simplify a parameter search by learning and exploiting the mathematical structure of the problem. Ferret also includes a self-adaptation algorithm that models and optimizes its own performance during a run, much in the spirit of Evolution Strategies codes. While Ferret is no longer a genetic algorithm in the strict sense of the term, we have retained the name "Ferret Genetic Algorithm", partially for historical reasons, and partly because the code remains closer to the genetic algorithm paradigm than other types of evolutionary optimizers.

We use Ferret in Sections 4.1 and 4.2 to demonstrate that our model can be inverted *exactly* when no noise is present in the data. We also show that all parameters are well determined (although not exactly) when 10% noise is added to the data, demonstrating our model's robustness to noise.

4. Results and Discussion

4.1. Non-Spatial Efficient Floor Model

We first present results obtained from our non-spatial efficient floor model. This model is idealized for the reasons discussed in Section 2.1. However, it forms the foundation of our more realistic non-efficient spatial models, and provides a simple illustration of the non-linearity introduced by the finite capacity of the gaming floor.

We generated an artificial data set for 25 machine types by constructing a random, but normalized, player preference distribution function p_n^{EF}, and a random distribution of baseline coin-in performance CI_i^0:

$$p_i^{EF} = N_p|1 + 0.5\xi| \quad \textbf{(28)}$$

$$\theta_q = \exp\left[-\left(\frac{Q_q - Q_0}{\sigma_q}\right)^2\right] * \max(0, Q_{max} - Q_q), \quad \textbf{(29)}$$

where N_p is a normalization factor, and we have assumed $Q_0 = 1, \sigma_q = 0.25, Q_{max} = 1.25$. We assumed 10 equally spaced bins for Q ranging from 0.1 to 1.25, and the distribution function $\theta_q(Q_q)$ is shown in Figure 1a. We used equations 12 and 15 to calculate the duty cycle and coin-in values for all machine types, at all values of Q, which represented our artificial data.

We used our evolutionary optimizer Ferret to solve for all parameters *exactly* (essentially to machine precision), thus demonstrating that the efficient floor model is uniquely defined by the data, and that the efficient floor inverse problem is feasible.

We also solved the forward optimization problem, in which we varied the number of machines of each type to maximize coin-in, with all model parameters fixed. Figures 1b and 1c respectively show the duty cycle and coin-in for each machines type, for the optimized model, and Figure 1d shows the total coin-in. We observe that the total coin-in rises approximately linearly with Q, until a threshold is reached, beyond which no further improvement is possible. Each machine type reaches its maximum duty cycle (equal to one, corresponding to full utilization) and coin-in at a different value of Q. Thus, the total coin-in reaches its saturation value when all machines are fully utilized.

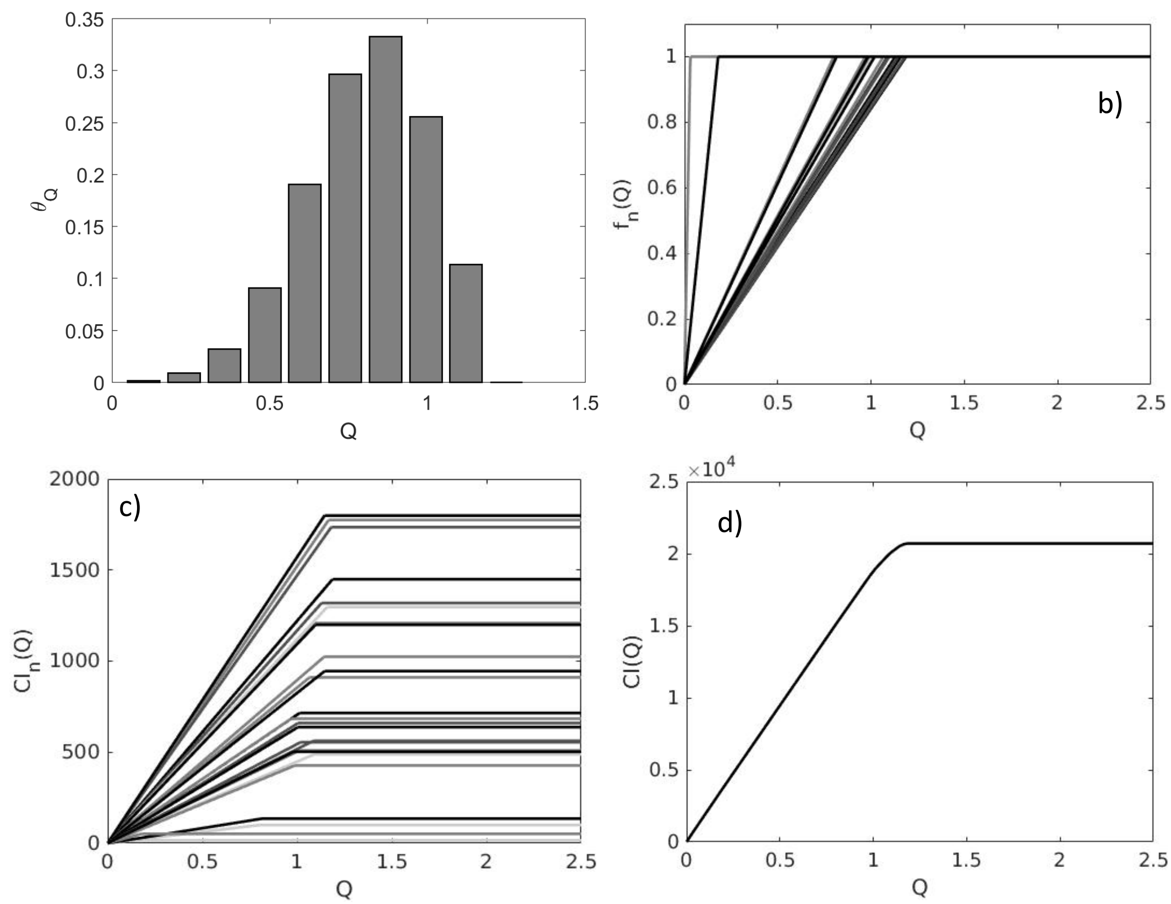

Figure 1: Forward optimization problem for the efficient floor model with 25 machine types. a) Duty cycle f_n for each machine type as a function of busyness Q. Each machine type reaches the $f_n = 1$ asymptote at a different value of Q. b) Coin-in as a function of Q for each machine type. c) Histogram showing the probability distribution function used for Q. d) Total coin-in for the floor as a function of Q.

4.2. Spatial Δ model – Finding Model Parameters from Data (Inverse Problem)

Figure 2 shows an example of the convergence of a run that solves an inverse problem to find the parameters of the spatial Δ model, in the absence of statistical noise. The Ferret evolutionary optimizer was configured to use 4 populations (nearly independent, internally interacting groups) of 250 individuals (trial solutions), which were allowed to evolve (refine) for 800 generations (iterations). The results show that the objective function F decreased approximately exponentially as a function of generation, reaching a value of approximately 10^{-12} by the end of the run. We conducted many numerical experiments like this one, without added noise, finding that the model parameters can *always* be found to almost arbitrary accuracy, for a wide range of optimizer configurations. Noiseless data are of course unrealistic.

However, this experiment verified that our model can be inverted, and does not suffer from mathematical degeneracy, in which many models might result in an identical minimum F.

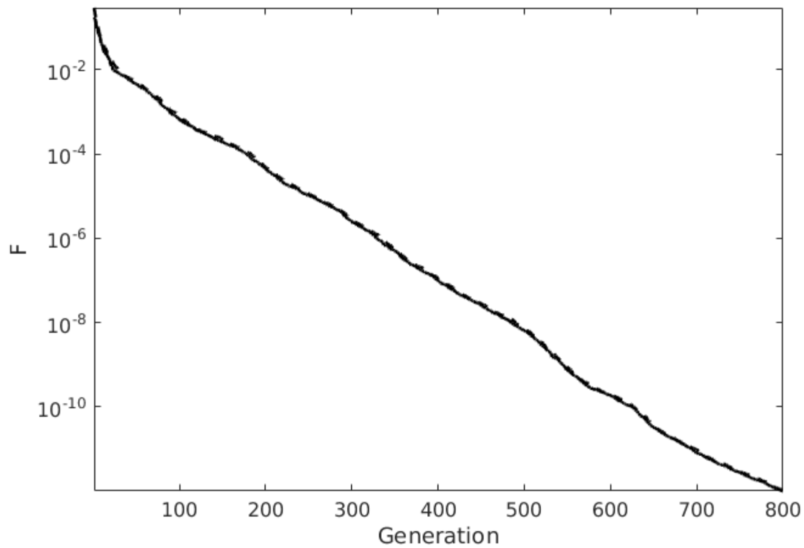

Figure 2: Approximately exponential convergence of an inverse model in the absence of statistical noise.

Figure 3 shows the convergence of an inverse model when 10% Gaussian noise was added to each of $f_{n,q}^{data}$ and $CI_{n,q}^{data}$. This run was computed using a single population of 250 individuals, and reaches its minimum value of ~ 0.1 by approximately generation 40.

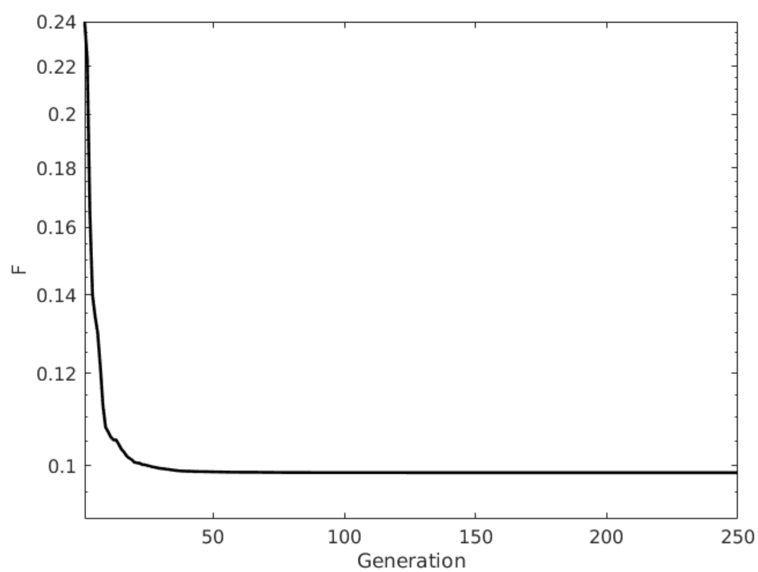

Figure 3. Convergence of an inverse model in the presence of 10% statistical noise.

We performed many inverse model runs to understand how the presence of error in $f_{n,q}^{data}$ and $CI_{n,q}^{data}$ affects the accuracy of our parameter determination. For each run, we used a single population of 250 individuals, noting that the minimum value was always reached long before generation 250. Figure 4 shows the results of this parameter recovery experiment, where 10% Gaussian noise was added to the artificial data for 25 runs. For each run, we plotted the recovered parameters against the artificial data parameters. All recovered parameters show good correlation with the data parameters, although the accuracy of the recovery varies. The power law γ was determined within a small range containing the data value of $\gamma = 2$. (We note that this particular model was run with the constraint that $\gamma^S = \gamma^C = \gamma$.) Player preferences are recovered most accurately, followed closely by the Δ^R (region delta) parameters. The clustering strength parameters associated with Δ^C were also recovered quite well. The parameters showing the poorest recovery were those associated with Δ^S (spatial delta) and Δ^B (bank delta). We do not presently understand why these parameters are poorly determined relative to others, but we are investigating this effect.

We performed a similar test using 10% Laplace noise instead of Gaussian noise, with no notable differences. The Laplace distribution was chosen as an alternative distribution function because of its simple analytical form, and its heavy tails that contrast with the Gaussian distribution.

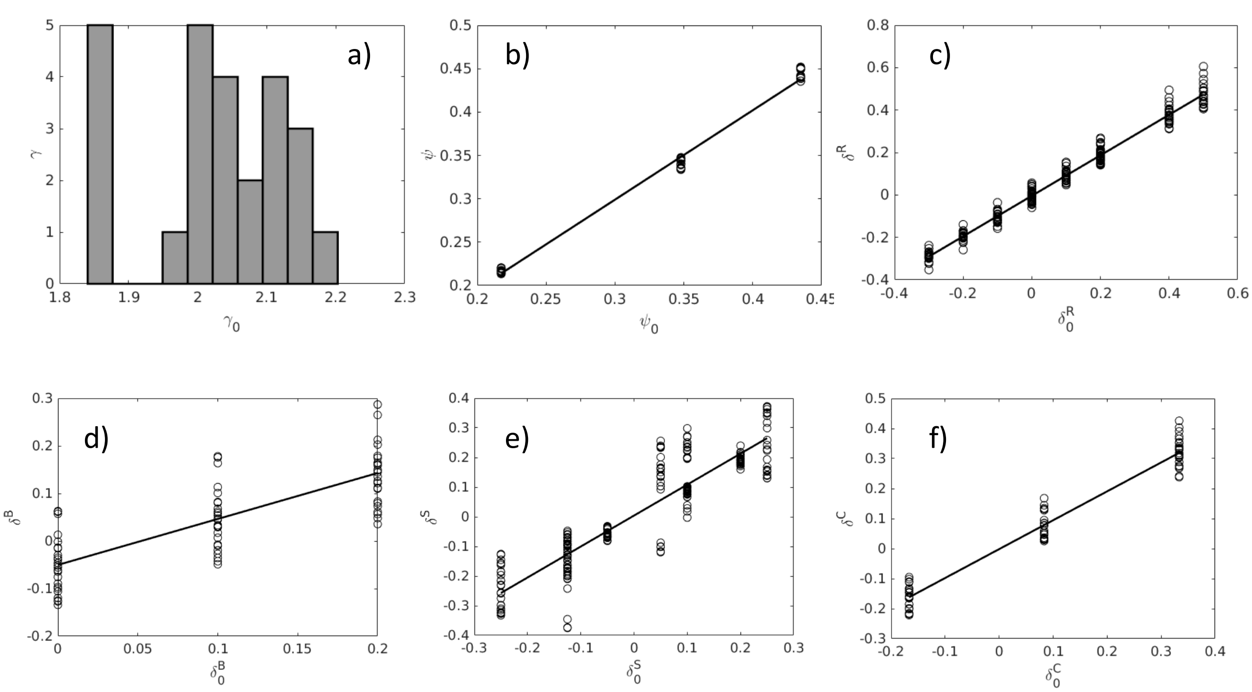

Figure 4: Results from 25 data modeling (inverse problem) runs with 10% Gaussian noise added to the artificial data. a) Distribution of γ. Note that we constrained $\gamma^S = \gamma^C = \gamma$ for these runs, but this is not required in general. b) Correlation of modeled player preference probability with data. c) Correlation of modeled regional Δ strength parameter with data. d) Correlation of modeled bank Δ strength parameter with data. e) Correlation of modeled spatial Δ strength parameter with data. f) Correlation of modeled clustering Δ strength parameter with data.

4.3. Spatial Δ model – Optimizing the Slot Floor (Forward Problem)

With all parameters determined by modelling the data, it is now possible to optimize the slot floor. We again use our Ferret optimizer to solve this forward problem, taking advantage of its multi-objective capabilities to simultaneously maximize the coin-in and win objectives. Figure 5 shows an example of a trade-off curve between these objectives. The trade-off curve (Pareto front) is defined by the property that coin-in cannot be increased without decreasing win, and vice-versa. Each point shown in Figure 5 represents a different configuration of the slot floor, and all are equally optimal in the Pareto sense. Figure 5 is taken from a graphical user interface, which allows a user to move the gray circle to highlight, visualize, and compare different optimal floor configurations, selected from different parts of the curve.

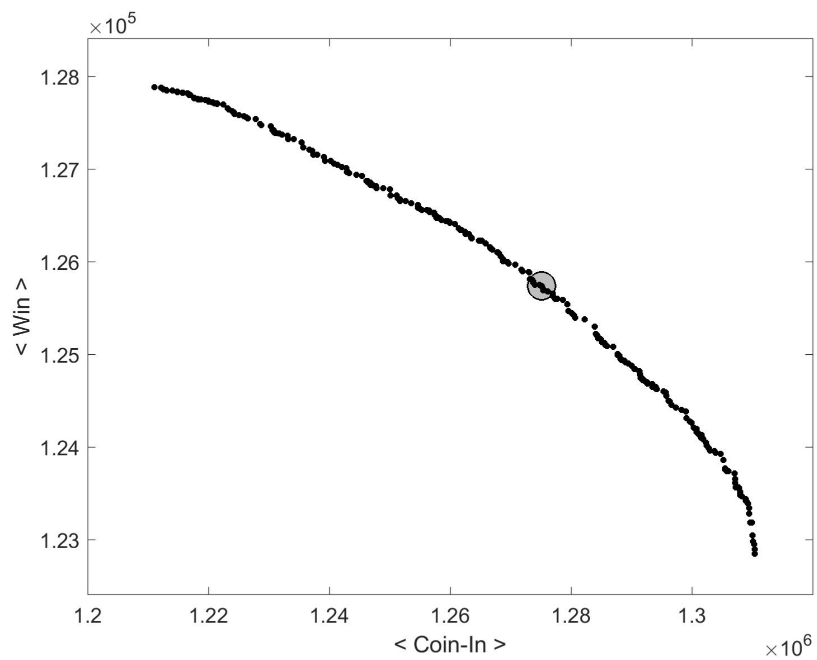

Figure 5. Example of a Win vs. Coin-In trade-off curve resulting from an optimization of our model. All model parameters were held fixed, while the machine type at each site was allowed to vary. Each dot represents a different configuration of the slot floor. The point at the center of the gray circle is the slot configuration chosen for visualization in Figures 6-8.

Figure 6 shows the slot floor configuration corresponding to the point selected by the gray disk shown in Figure 5. The background image is based on a map of Paris Las Vegas casino from Google Maps. The locations of slot machine banks, table games, walls, and walkways were sketched in by hand. Significant variations from the actual configuration of the Paris casino floor are certain, but this is unimportant for this artificial data study.

Note that we have only used 3 machine types, to help with visualization, but it is possible to perform the optimization for larger numbers of machine types. We note that the overall pattern shows considerable geometric structure that is discovered automatically during the optimization. For the model parameters used, machine type 1 (x symbols) generally prefers the interiors of large groups of machines, machine type 2 (circles) is often found on the perimeters of large groups, and machine type 3 often clusters in specific regions, but is sometimes interspersed with other machine types. The mix of machine types and the self-organized patterns observed are repeatable between optimization runs, and display a high level of complexity. The patterns become even more complex when the software is allowed to use a larger number of machine types.

Figure 6. The optimized slot floor corresponding to the point at the center of the gray circle in Figure 5. Each symbol type corresponds to a different machine type. Only 3 machine types were used to make visualization easier, but it is possible to run the optimization with larger numbers of machine types. The background image is based on a map of Paris Las Vegas casino from Google Maps.

Figure 7 shows heat maps and histograms of duty cycle, coin-in, and win for the same optimized model shown in Figure 6. All quantities are averaged over busyness Q. The duty cycle heat map can be thought of as a time-averaged map of where players spend time on the slot floor. Overall, the duty cycle map is more uniform than the coin-in and win maps, but a higher level of player clustering can be observed when the map is not averaged over Q. Not surprisingly, the coin-in and win heat maps are quite similar in appearance, which is consistent with the notion that both are good, synergistic metrics of performance. The win map shows more spatial variation though, and it is clear that there are certain spatial concentrations of very profitable machines, for example the cluster of type 3 (triangle) machines in the bottom right of the map. Note however that there would be no advantage to putting more type 3 machines on the floor. The mix and spatial distribution shown have been thoroughly optimized; any change would at best correspond to a different model on the trade-off curve, but more likely would correspond to a non-optimal model with decreased performance.

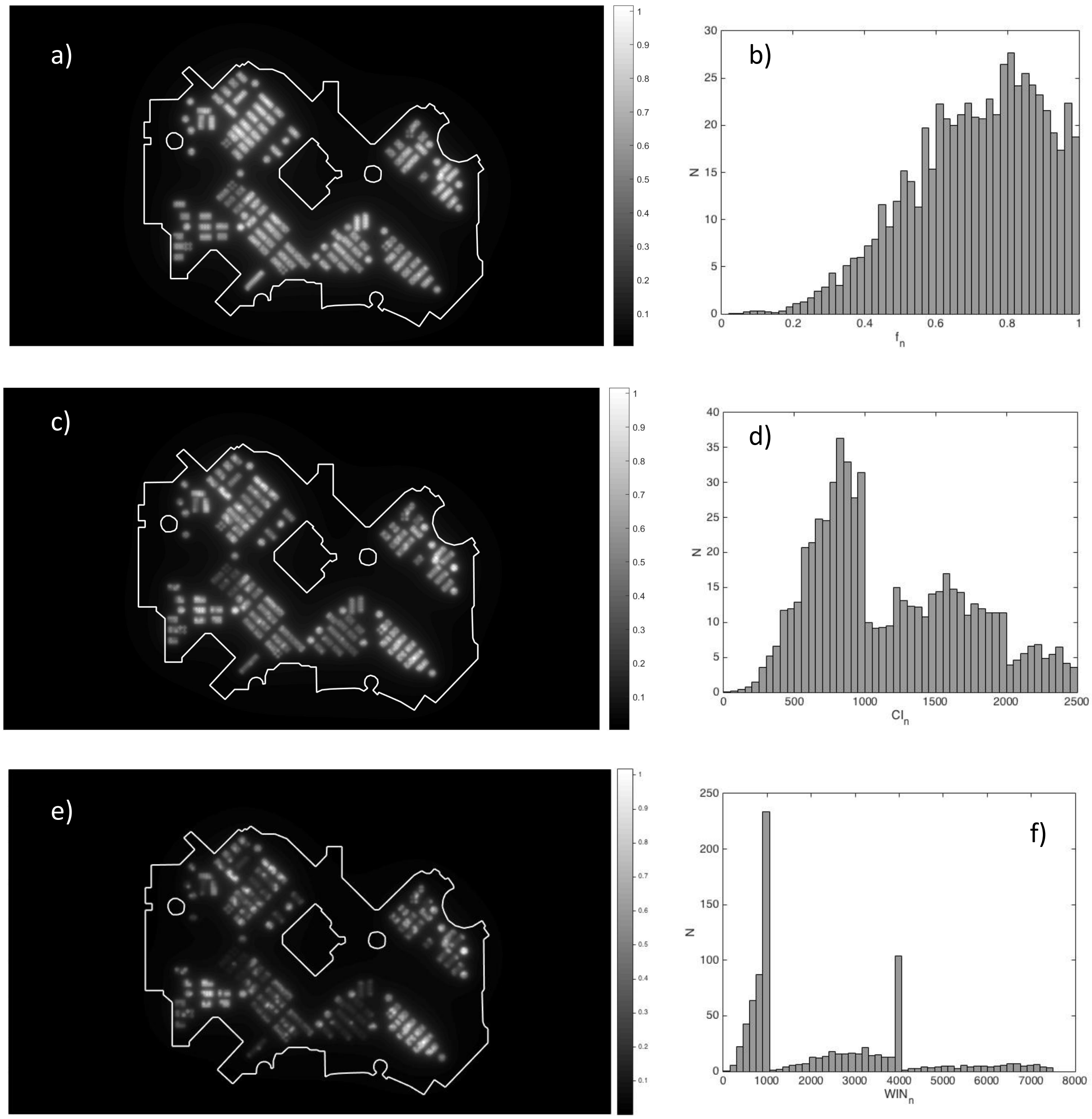

Figure 7. Heat maps and histograms corresponding to the point at the center of the gray circle in Figure 5. Lighter shades in the heat maps correspond to higher values, but all values are normalized to a maximum value of one. All quantities are averaged over busyness Q. a) Duty cycle heat map. b) Duty cycle histogram for all slot machines. c) Coin-in heat map. d) Coin-in histogram for all slot machines. e) Win heat map. f) Win histogram for all slot machines.

5. Conclusions and Future Work

This paper puts forward the mathematical framework for a non-linear, data-driven, spatial model of the slot floor. Our computational experiments using artificial data have demonstrated that model parameters can be uniquely determined from machine-level data, without encountering problems of mathematical degeneracy. Parameter estimation is reasonably robust in the presence of statistical noise.

Our results demonstrate a method for finding the optimal mix and locations of slot machines on the casino floor by varying the type of each machine on the slot floor to simultaneously optimize coin-in and win objectives. The model takes into account the influence of several "Δ" variables that affect performance. These include the region of the floor that a particular machine is located in; its position within a bank of machines (or the type of bank it's located in); and spatial effects such as distance to table games, walkways, entrances, etc. The model also includes a player "clustering" Δ that takes into account and predicts how busy or isolated the floor is in the proximity of each machine. Thus, we simultaneously model the slot floor performance and this aspect of player behaviour, which we believe is unique in this area of research.

The results presented here used a very small number of slot machine types. This was done partly for visualization purposes, and partly to make it feasible to conduct the large number of runs needed to study the statistical effects of noise on parameter determination. We are working to increase the number of machine types to a number that would be of practical use. We have recently discovered a novel computational method that simplifies the problem substantially and speeds up the computation, allowing us to scale this problem up to handle the larger number of parameters required for a realistic number of machine types. We will report on this method elsewhere.

Work done in parallel to that presented here considers time dependent model tracking in which the configuration of the floor is slowly varied, in response to changing business conditions, so that it always remains on the optimal trade-off curve, while also minimizing the number of machines that need to be swapped or moved. This capability may allow operators to maximize profitability in time-variable, trending markets, with minimal effort.

6. References

Dunn, W.T. (2014) *Slot Performance Analysis: An Essential Resource for Casino Operations Management* (book), Dunn Gaming Solutions.

Ghaharian, K. C. (2010). A mathematical approach for optimizing the casino slot floor: A linear programming application.

Goldberg, D. E. (2002) The Design of Innovation: Lessons From and for Competent Genetic Algorithms. Norwell, MA: Kluwer Academic Publishers.

Holland, J. H. (1992). Adaptation in natural and artificial systems. 1975. *Ann Arbor, MI: University of Michigan Press*.

Lucas, A. F. (2014). Exploring the Relationship between Race and Sports Book Wagering Activity and Daily Slot and Table Game Play. *UNLV Gaming Research & Review Journal*, *18*(2), 85.

Lucas, A. F., & Dunn, W. T. (2005). Estimating the effects of micro-location variables and game characteristics on slot machine volume: A performance-potential model. *Journal of Hospitality & Tourism Research*, *29*(2), 170-193.

Lucas, A. F., & Roehl, W. S. (2002). Influences on video poker machine performance: Measuring the effect of floor location. *Journal of Travel & Tourism Marketing*, *12*(4), 75-92.

Lucas, A., Singh, A. K., Gewali, L. P., & Singh, A. (2009). An application of spatial population partitioning using Voronoi Diagrams: Principal component regression analysis of slot machine performance data. *Journal of Travel & Tourism Marketing*, *26*(4), 415-429.

Lucas, A. F., & Santos, J. (2003). Measuring the effect of casino-operated restaurant volume on slot machine business volume: An exploratory study. *Journal of Hospitality & Tourism Research*, *27*(1), 101-117.