

a restaurant limit bookings for people only wanting to have one course at a peak time of say 7:30pm which may mean that the restaurant cannot use the table before that time and after that time when the preference should be to only offer a three course menu for bookings to be accepted at 7:30pm?

2. Secondly, yield management is essentially the application of a defined algorithm or mathematical formula to the control of inventory. As such, a yield management module in the context of a software solution must be able to specifically control and release inventory. Bossert fails to provide a yield management solution that is capable of controlling the release of inventory, as the disclosure, at Figure 3 describes a process where a capacity calculation module ("60") firstly determines capacity (using constraint programming and its inherent limitations) and only then does the proposed yield management calculation module ("70") accept or reject a reservation request. By taking this approach, Bossert is essentially "putting the cart before the horse".

[00128] The yield management disclosure of Bossert has no practical utility, nor is it an enabling disclosure, as it does not adhere to yield management principles, it does not properly apply yield management principles and it does not describe a workable solution. It is, at best, a mere "paper anticipation" for a set of ideas, not a workable embodiment that utilises yield management terminology.

Summary

[00128a] There is provided a computer enabled method for optimising and allocating booking requests within a venue having one or more spaces, comprising the steps of, at a computing system including at least one software module arranged to allocate bookings autonomously to one of a table or table combination, electronically receiving a booking request via a network, the booking request including constraint information including a plurality of constraint values associated with one of the venue and the requestor, and for each received booking request, the module categorising the booking request into one of at least two categories, the at least two categories being selected from a group of categories including a selection category including constraint information identifying a specific table or table combination, and a non-selection category including booking requests with constraint information not identifying a specific table or table combination, and for each of the at least two categories, the module selecting one of a plurality of iterative methods allocate the booking request to the table or table combination, the selection being dependent on the constraint information of the each of the at least two categories the one of the plurality of iterative methods including at least one of the following steps:

a. allocating the received booking request to a specific table or table combination requested by the booking requestor;

- b. allocating the received booking request to a specific table or table combination requested by the booking requestor by firstly identifying one or more individual requestors that comprise the booking request, and using the identity of at least one of the one or more requestors to retrieve requestor constraint information from a database, the requestor constraint information including a requestor ranking value that ranks the at least one requestor relative to other requestors in the database, whereby the booking request is allocated utilising the requestor ranking as one of the plurality of constraint values;
- c. where the module attempts to allocate a request to a specific table requested by a booking requestor and determines that a requested table or table combination is allocated to a previously received booking request, the module further determines the identity of at least one requestor associated with the booking request and uses the identity of the at least one requestor to retrieve requestor constraint information including a requestor ranking value from a database that ranks the at least one requestor relative to the requestors associated with the previously allocated booking request, and if the ranking of the requestor is higher than the ranking of the previously allocated booking requestor, reallocating the at least one previously allocated booking request to a different table or table combination and allocating the received booking request to the requested booked table;
- d. upon requiring a reallocation of at least one booking to accommodate a received booking request, reallocating the at least one previously allocated booking request by allocating the booking request of the largest size first and reallocating all other booking requests in descending order of size;
- e. upon requiring a reallocation of at least one booking to accommodate a received booking request, determining a booking size metric of the received booking and each of the allocated bookings, the size metric being determined by calculating a size metric which utilises the number of persons that comprise the booking request and the service time duration for the booking request as inputs, and utilising the size metric to reallocate all bookings in order from the largest size metric booking to the smallest size metric booking;
- f. utilising booking request constraint information and venue constraint information to determine a difficulty metric utilising the size metric and a peak period seating time value to determine a difficulty measure, the difficulty measure representing a measure of the relative difficulty of allocating the booking request relative to the constraints of the venue, whereby booking requests are ranked from most difficult to least difficult and allocated in descending order from most difficult to least difficult;
- g. utilising booking request constraint information and venue constraint information to determine sub-service periods within a service period, and for all booking requests that fall within the service period, firstly allocating all booking requests that fall across one or more sub-service periods in order of descending size, and subsequently allocating all booking

requests that do not fall across the one or more sub-service periods in order of descending size;

- h. utilising constraint information to determine a difficulty measure, the difficulty measure being representative of the relative difficulty of allocating a booking request, whereby bookings are allocated in descending order of difficulty;
- i. reallocating at least one previously allocated booking request to optimise the number of bookings within each of the one or more spaces;
- j. reallocating at least one previously allocated booking whereby bookings of identical or similar size are clustered, in both physical proximity and chronological proximity;
- k. reallocating at least one previously allocated booking whereby the total time that the each table or table combination remains unused between bookings during a single service period is minimised;
- l. reallocating at least one previously allocated booking to cluster bookings such that physically adjacent tables have similar start times;
- m. reallocating at least one previously allocated booking such that physically adjacent tables have similar finish times;
- n. reallocating at least one previously allocated booking so that smaller size bookings are physically clustered adjacent to larger size bookings;
- o. Reallocating at least one previously allocated booking such that a previously joined table for an earlier booking in a service period is reutilised for a later booking in the service period;
- p. reallocating at least one previously allocated booking such that the at least one booking is allocated in a manner where a minimal number of tables are joined to allocate the booking;
- q. reallocating at least one previously allocated booking such that the total number of bookings within a service period are arranged in a manner that requires the least possible number of table movements during the service period;
- r. allocating at least one potentially conflicting booking to the smallest fitting table irrespective of availability, and where a conflicting booking is generated, reallocating the previously allocated booking as a result of the newly created conflicting allocation;
- s. reallocating at least one previously allocated booking whereby an empty table is retained between one or more booked table;
- t. utilising constraint information to reallocate all bookings from the highest ranked available table in a descending order of rank;
- u. reallocating at least one previously allocated booking whereby the ranking of the booking requestor determines a table to which the booking requestor is reallocated;

v. reallocating at least one previously allocated booking utilising one or more qualitative constraints derived from information associated with the booking requestor including but not limited to a stated occasion associated with the booking, a menu or courses selected by the requestor, the courses selected by the requestor, ancillary products selected by the requestor and the date of the booking; and

w. reallocating all bookings to one or more different table solution sets to determine whether at least one of the one or more different table solution sets results in a more optimal outcome, and if so, selecting the at least one of the one or more different table solution sets that results in the more optimal outcome.

[00129] In a first aspect, the present disclosure provides a computing system for optimising and allocating bookings within a space for a defined period of time in a venue, comprising an optimisation and allocation module in communication with a processor and arranged to receive the at least one request and communicate with a database to determine whether other requests for spaces have been made by other users, and if so, utilise the processor to retrieve information regarding the other requests for spaces and combine the at least one request with other requests to form a pool of requests, retrieve constraint information regarding the venue including information regarding the size of the venue and the size and quantity of available furniture that may be allocated to the space in the venue, and in a structured way iteratively allocate all requests from the pool of requests utilising the restraint information to produce an optimised and/or prioritised allocation instruction set, wherein the optimised and/or prioritised allocation instruction set saved in the database and provided by a space allocation user interface to one or more users associated with the venue.

[00130] In one embodiment, to provide a computing system and a user interface to at least one remote user to input the restaurant's floor plans to scale to be stored in the data base.

[00131] In one embodiment, to provide a computing system and a user interface to at least one remote user to input and divide the restaurant's scaled plans into