US 20010047287A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2001/0047287 A1

Jacobs et al. (43) Pub. Date: Nov. 29, 2001

(54) FINDING TECHNIQUE FOR A SCHEDULING SYSTEM

(76) Inventors: **Simon Jacobs**, Vancouver (CA); **Guy Druce**, Richmond (CA)

Correspondence Address:
D. C. Peter Chu, Esq.
DORSEY & WHITNEY LLP
1420 Fifth Avenue, Suite 3400
Seattle, WA 98101 (US)

(21) Appl. No.: **09/824,850**

(22) Filed: **Apr. 2, 2001**

Related U.S. Application Data

(63) Non-provisional of provisional application No. 60/193,705, filed on Mar. 31, 2000. Non-provisional of provisional application No. 60/193,834, filed on Mar. 31, 2000. Non-provisional of provisional application No. 60/193,917, filed on Mar. 31, 2000. Non-provisional of provisional application No. 60/193,832, filed on Mar. 31, 2000. Non-provisional of provisional application No. 60/193,833, filed on Mar. 31, 2000.

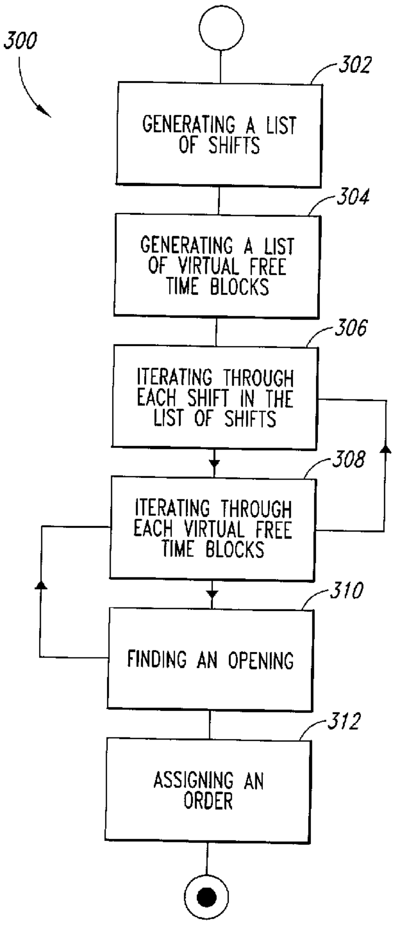
Publication Classification

(51) **Int. Cl.**^{7} **G06F 17/60**

(52) **U.S. Cl.** **705/9**

(57) **ABSTRACT**

An aspect of the present invention includes a method for finding an opening in which to fit an order in a schedule. The method includes computing an amount of time required in a shift to fit the order, and creating a schedulable time block from a virtual free time block. The schedulable time block includes a primary block. The method further includes examining the primary block. The primary block is a candidate for fitting the order if the duration of the primary block, excluding at least one break, is greater than or equal to the amount of free time required in the shift to fit the order. The method further includes creating at least one opening in the shift from the schedulable time block so as to present to a customer at least one option of fitting the order in the schedule to perform a desired service.


300
302
GENERATING A LIST OF SHIFTS
304
GENERATING A LIST OF VIRTUAL FREE TIME BLOCKS
306
ITERATING THROUGH EACH SHIFT IN THE LIST OF SHIFTS
308
ITERATING THROUGH EACH VIRTUAL FREE TIME BLOCKS
310
FINDING AN OPENING
312
ASSIGNING AN ORDER

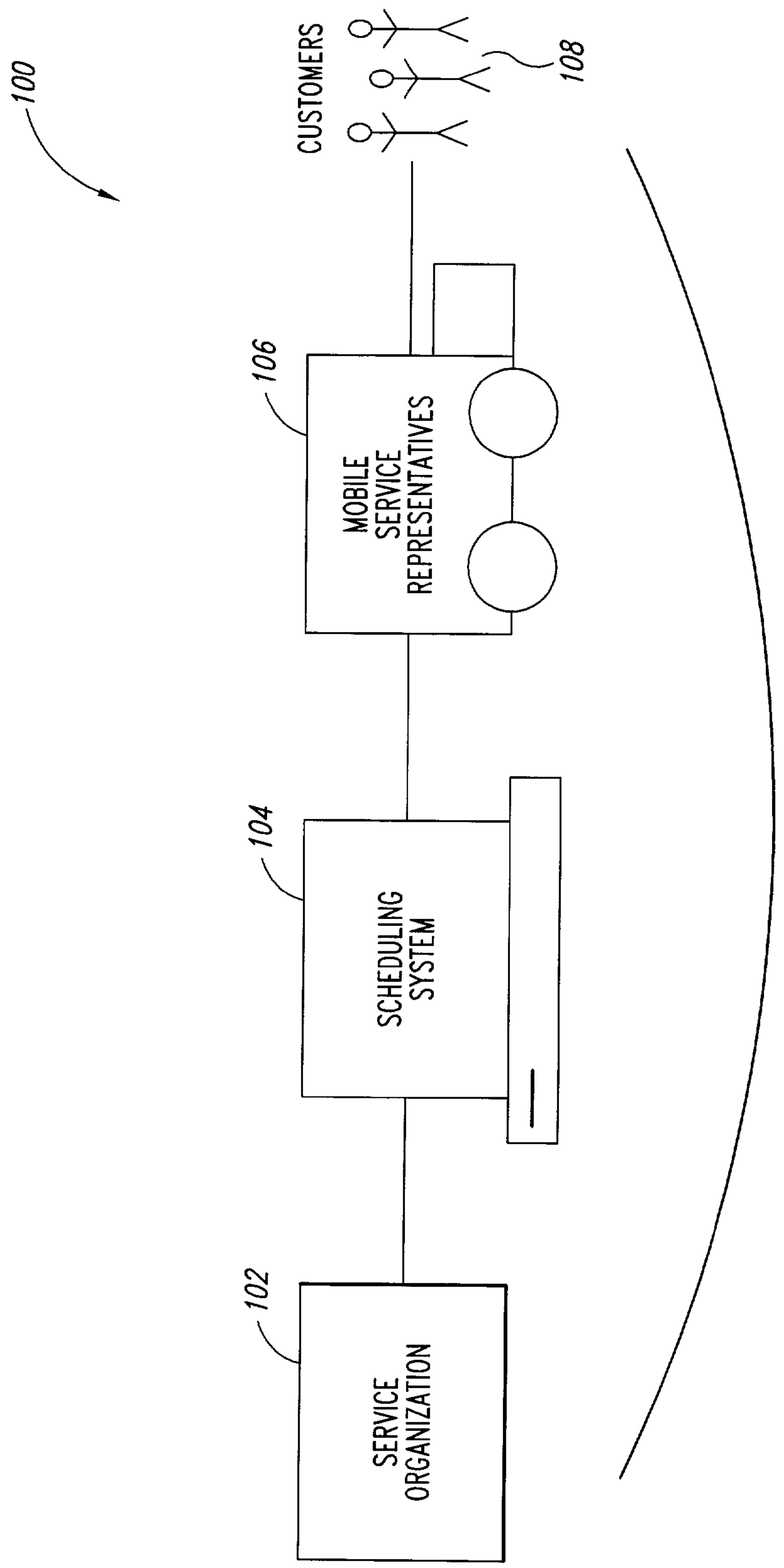

100
CUSTOMERS
108
106
MOBILE SERVICE REPRESENTATIVES
104
SCHEDULING SYSTEM
102
SERVICE ORGANIZATION

Fig. 1

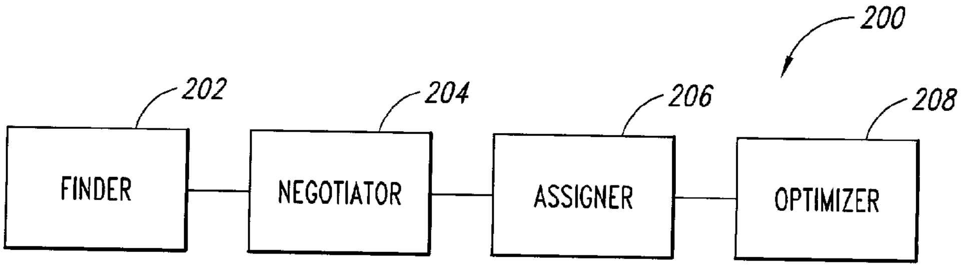

200
202
FINDER
204
NEGOTIATOR
206
ASSIGNER
208
OPTIMIZER

Fig. 2

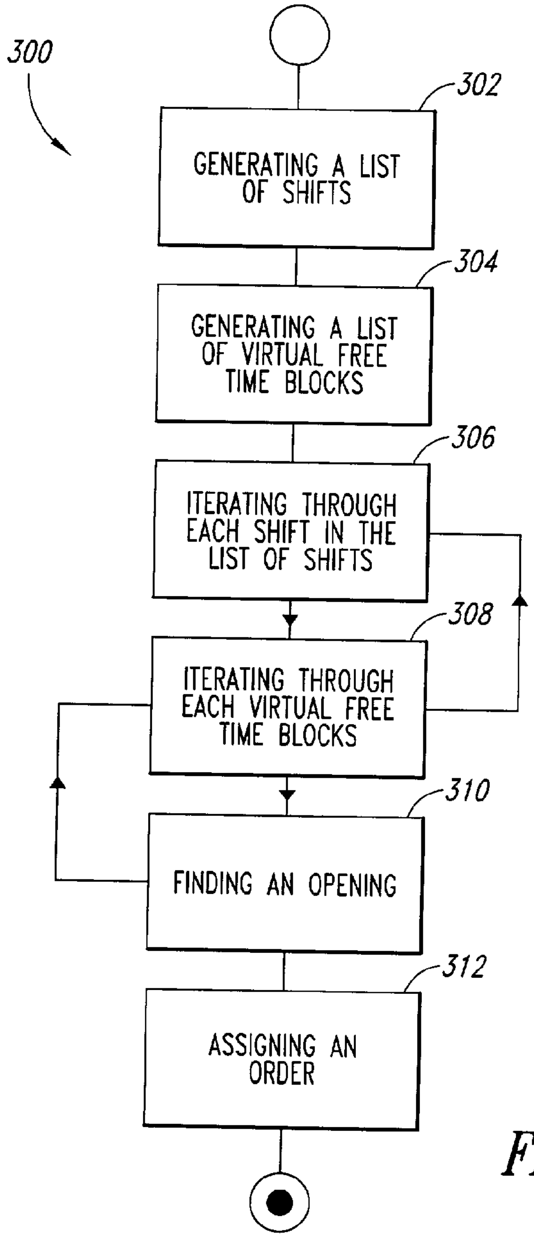

300
302
GENERATING A LIST OF SHIFTS
304
GENERATING A LIST OF VIRTUAL FREE TIME BLOCKS
306
ITERATING THROUGH EACH SHIFT IN THE LIST OF SHIFTS
308
ITERATING THROUGH EACH VIRTUAL FREE TIME BLOCKS
310
FINDING AN OPENING
312
ASSIGNING AN ORDER

Fig. 3

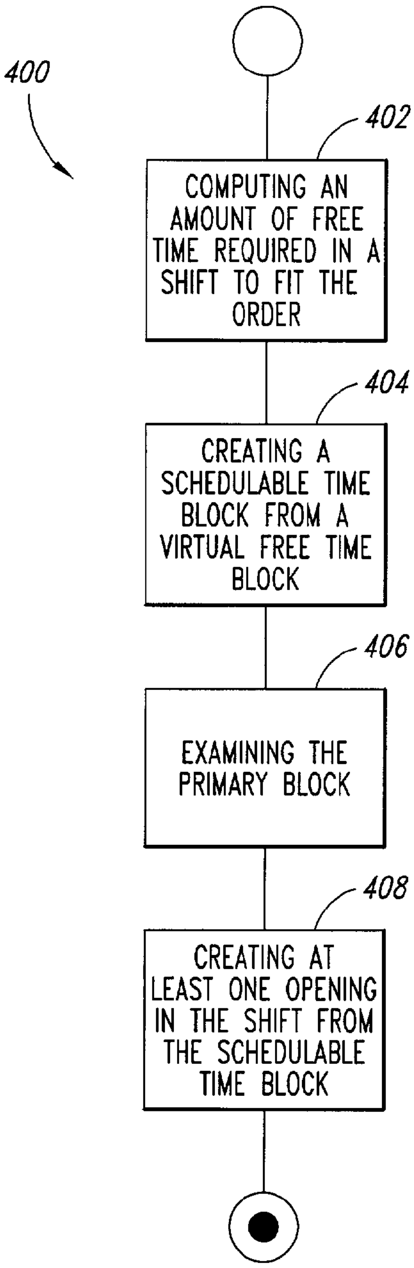

400
402
COMPUTING AN AMOUNT OF FREE TIME REQUIRED IN A SHIFT TO FIT THE ORDER
404
CREATING A SCHEDULABLE TIME BLOCK FROM A VIRTUAL FREE TIME BLOCK
406
EXAMINING THE PRIMARY BLOCK
408
CREATING AT LEAST ONE OPENING IN THE SHIFT FROM THE SCHEDULABLE TIME BLOCK

Fig. 4

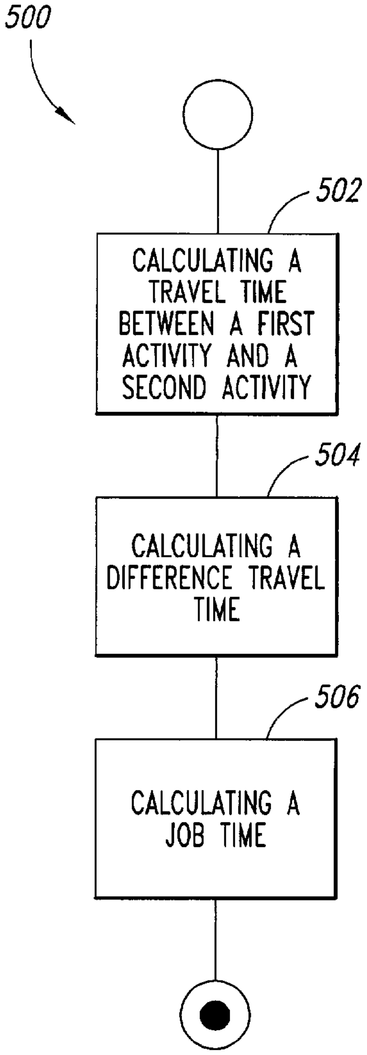

500
502
CALCULATING A TRAVEL TIME BETWEEN A FIRST ACTIVITY AND A SECOND ACTIVITY
504
CALCULATING A DIFFERENCE TRAVEL TIME
506
CALCULATING A JOB TIME

Fig. 5

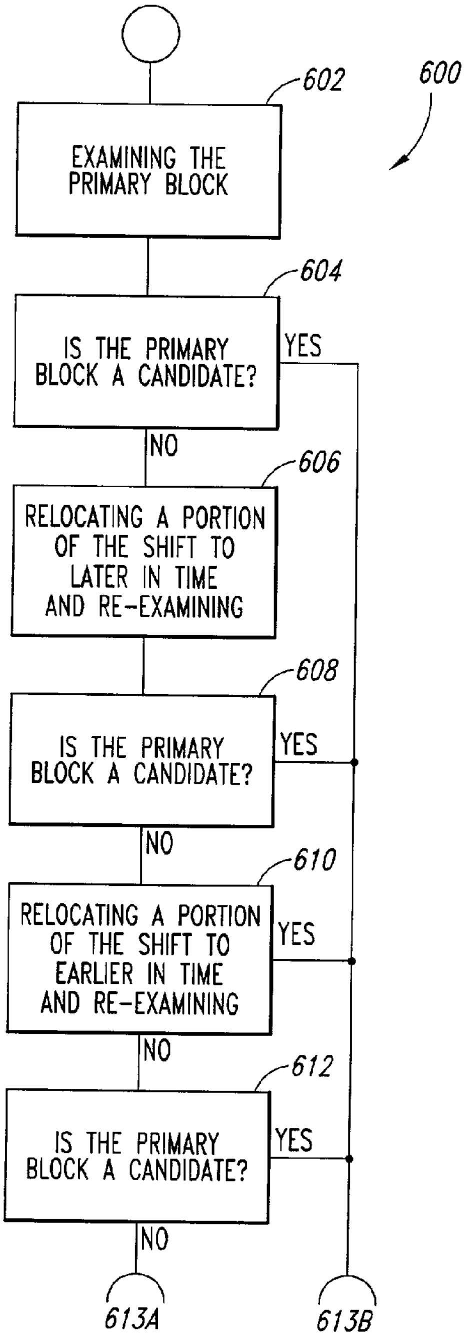

600
602
EXAMINING THE PRIMARY BLOCK
604
IS THE PRIMARY BLOCK A CANDIDATE?
YES
NO
606
RELOCATING A PORTION OF THE SHIFT TO LATER IN TIME AND RE-EXAMINING
608
IS THE PRIMARY BLOCK A CANDIDATE?
YES
NO
610
RELOCATING A PORTION OF THE SHIFT TO EARLIER IN TIME AND RE-EXAMINING
YES
NO
612
IS THE PRIMARY BLOCK A CANDIDATE?
YES
NO
613A
613B

Fig. 6A

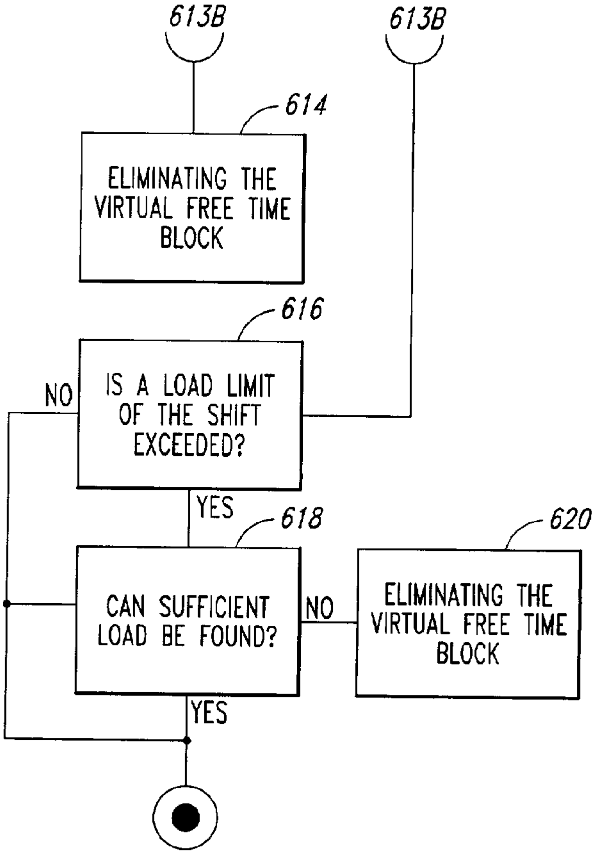

613B
613B
614
ELIMINATING THE VIRTUAL FREE TIME BLOCK
616
IS A LOAD LIMIT OF THE SHIFT EXCEEDED?
NO
YES
618
CAN SUFFICIENT LOAD BE FOUND?
NO
YES
620
ELIMINATING THE VIRTUAL FREE TIME BLOCK

Fig. 6B

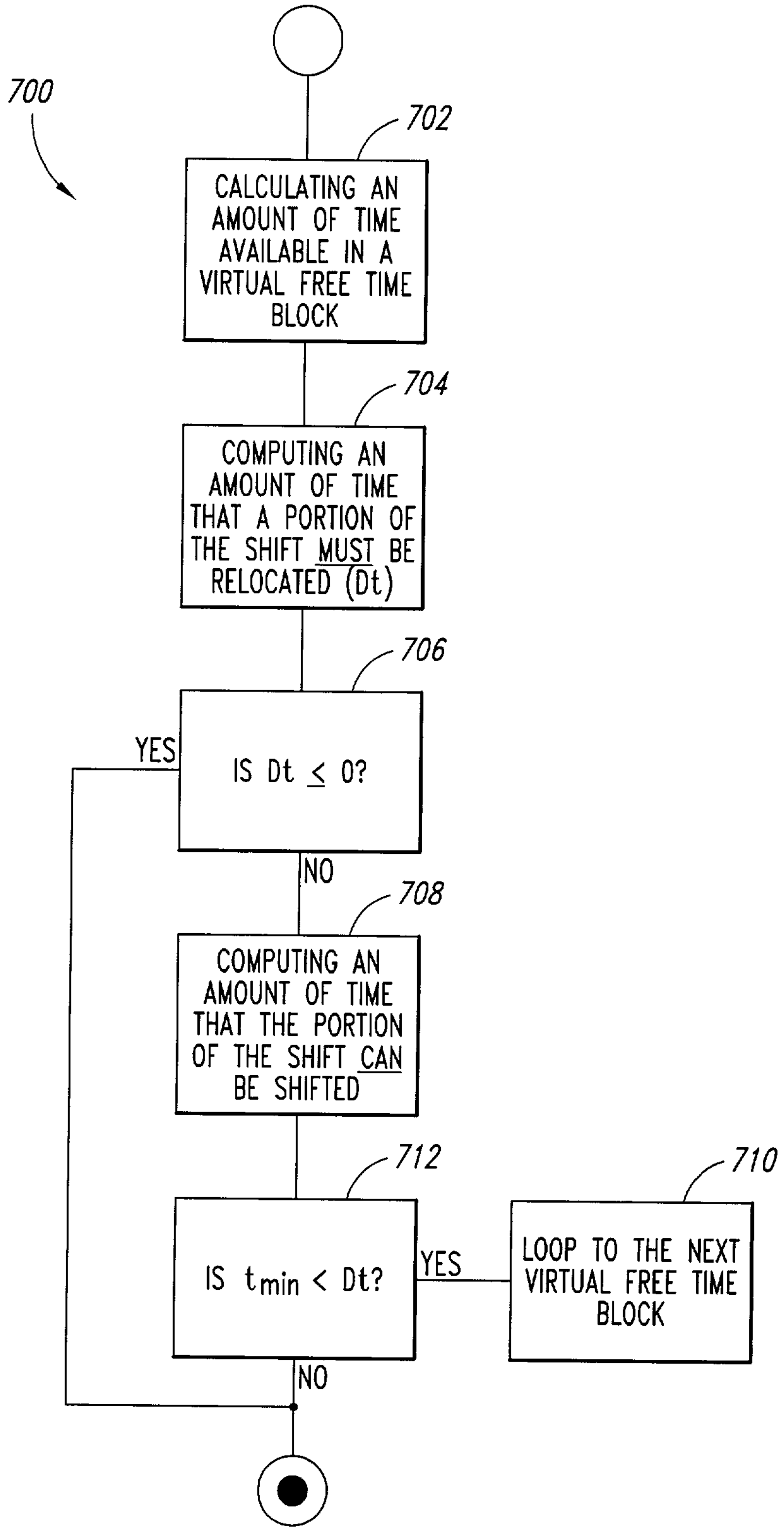

700
702
CALCULATING AN AMOUNT OF TIME AVAILABLE IN A VIRTUAL FREE TIME BLOCK
704
COMPUTING AN AMOUNT OF TIME THAT A PORTION OF THE SHIFT MUST BE RELOCATED (Dt)
706
IS Dt ≤ 0?
YES
NO
708
COMPUTING AN AMOUNT OF TIME THAT THE PORTION OF THE SHIFT CAN BE SHIFTED
712
IS t_{min} < Dt?
YES
NO
710
LOOP TO THE NEXT VIRTUAL FREE TIME BLOCK

Fig. 7

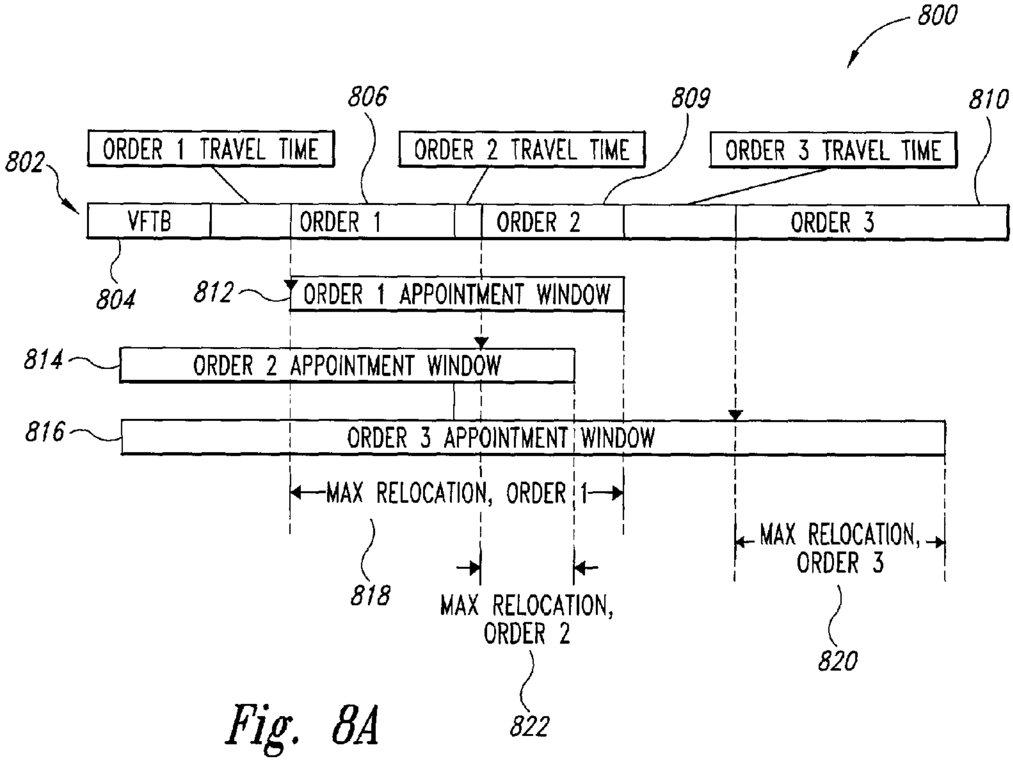

800
806
809
810
802
ORDER 1 TRAVEL TIME
ORDER 2 TRAVEL TIME
ORDER 3 TRAVEL TIME
VFTB
ORDER 1
ORDER 2
ORDER 3
804
812
ORDER 1 APPOINTMENT WINDOW
814
ORDER 2 APPOINTMENT WINDOW
816
ORDER 3 APPOINTMENT WINDOW
MAX RELOCATION, ORDER 1
818
MAX RELOCATION, ORDER 2
822
MAX RELOCATION, ORDER 3
820

Fig. 8A

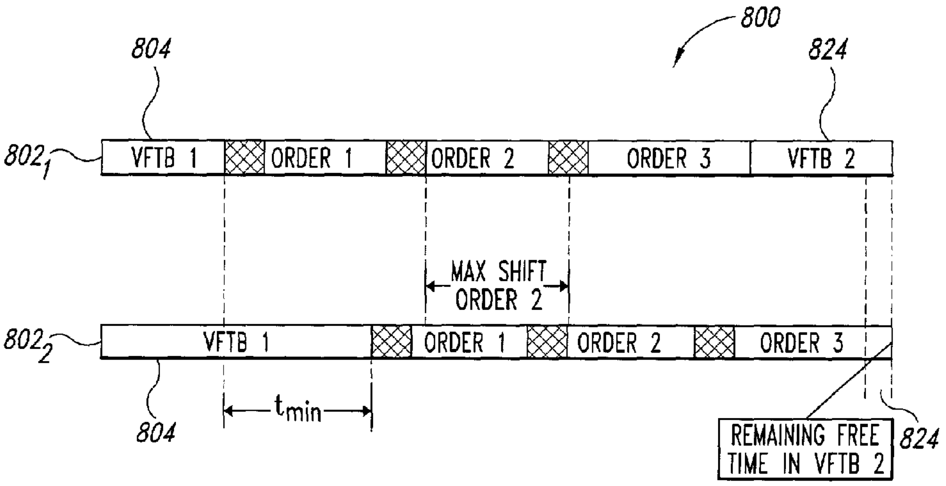

800
804
824
802_1
VFTB 1
ORDER 1
ORDER 2
ORDER 3
VFTB 2
MAX SHIFT ORDER 2
802_2
VFTB 1
ORDER 1
ORDER 2
ORDER 3
804
t_{min}
REMAINING FREE TIME IN VFTB 2
824

Fig. 8B

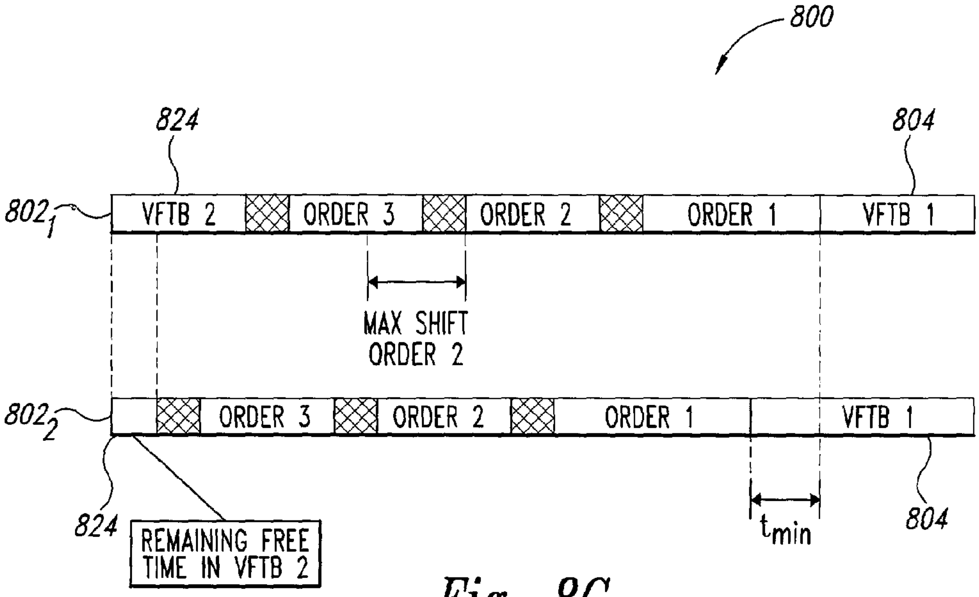

800
824
804
802_1
VFTB 2
ORDER 3
ORDER 2
ORDER 1
VFTB 1
MAX SHIFT ORDER 2
802_2
ORDER 3
ORDER 2
ORDER 1
VFTB 1
824
REMAINING FREE TIME IN VFTB 2
t_{min}
804

Fig. 8C

FINDING TECHNIQUE FOR A SCHEDULING SYSTEM

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application incorporates by reference and claims the benefit of U.S. Provisional Application No. 60/193,834, filed Mar. 31, 2000; U.S. Provisional Application No. 60/193,917, filed Mar. 31, 2000; U.S. Provisional Application No. 60/193,832, filed Mar. 31, 2000; U.S. Provisional Application No. 60/193,705, filed Mar. 31, 2000; and U.S. Provisional Application No. 60/193,833, filed Mar. 31, 2000.

TECHNICAL FIELD

[0002] The technical field relates generally to scheduling. More particularly, it pertains to finding an opening in a schedule within a predetermined period of time.

BACKGROUND OF THE INVENTION

[0003] To schedule a desired service to be performed, a customer calls a service organization to place an order for the desired service. The customer may be kept on the phone for an undesirably long period of time while the service organization searches the schedule in an attempt to fit the order into the schedule. To avoid annoying the customer, some service organizations do not search the schedule while the customer is on the phone, but simply accept the order and dump it into a time bucket without scheduling the order to a particular shift of a worker. The night before the order is to be performed, the service organization executes a batch process that assigns orders to various workers. One problem with this approach is that the service organization lacks the ability to accurately assess the capacity of the workforce to do work.

[0004] The approach as described above limits the ability of a customer to specify his preferences in regard to when and how the order is to be performed. The customer is unable to select a desired appointment window or a desired worker with a set of desired skills. As the expectations of customers have increased over time, scheduling systems that fail to meet increased customer expectations may result in customer dissatisfaction and lead to the eventual lack of acceptance in the marketplace. Thus, what is needed are systems and methods for enhancing the scheduling process that allow customers' increased expectations to be met.

SUMMARY OF THE INVENTION

[0005] An illustrative aspect of the invention includes a method for scheduling. The method includes forming a list of shifts in a schedule and determining a shift in which an order can be scheduled. The method also includes assigning the order to the shift that is determined by the act of determining. The acts of forming, determining, and assigning are contemporaneously executed while a customer is on a phone attempting to schedule the order.

[0006] Another illustrative aspect of the invention includes a method for finding an opening to fit an order in a schedule. The method includes computing an amount of free time required in a shift to fit the order and creating a schedulable time block from a virtual free time block. The schedulable time block includes a primary block, at least one expansion block, and at least one load block. The method also includes examining the primary block. Under the act of examining, the primary block is a candidate to fit the order if the duration of the primary block, excluding at least one break, is greater than or equal to the amount of free time required in the shift to fit the order. The method further includes creating at least one opening in the shift from the schedulable time block so as to present a customer with at least one option of fitting the order in the schedule to perform a desired service.

[0007] Another illustrative aspect includes a method for assigning an order to an opening in a schedule after a customer has selected an appointment window in the schedule. The opening and the appointment window are specified. The method includes generating a list of schedulable time blocks for a shift identified in the opening and then intersecting the opening and the appointment window to obtain a time range. The method also includes choosing the opening to assign the order if a schedulable time block from the list of schedulable time blocks includes the opening. The opening is within the time range obtained by the act of intersecting.

[0008] Another illustrative aspect includes a method for assigning an order to a schedule after a customer has specified an appointment window in the schedule. The method includes checking a list of openings for overlap with the appointment window and generating a list of schedulable time blocks in a shift if there is no overlap. The method includes assigning the order to the schedule if there is an opening in the list of openings that overlaps with the appointment window or an opening in the list of schedulable time blocks that overlaps with the appointment window.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] **FIG. 1** is a block diagram of a system showing the relationship between a service organization, a scheduling system, mobile service representatives, and customers according to one aspect of the present invention.

[0010] **FIG. 2** is a block diagram of a scheduling system according to one aspect of the present invention.

[0011] **FIG. 3** is a process diagram of a method for finding an opening in a schedule to fit an order to a shift of a worker and assigning the order to the opening according to one aspect of the present invention.

[0012] **FIG. 4** is a process diagram of a method for finding an opening in a schedule to fit an order to a shift of a worker according to one aspect of the present invention.

[0013] **FIG. 5** is a process diagram of a method for computing an amount of free time required to accommodate an order in a shift.

[0014] FIGS. **6**A-**6**B are a process diagram of a method for examining a primary block according to one aspect of the present invention.

[0015] **FIG. 7** is a process diagram of a method for computing extra time that can be obtained by relocating assigned orders earlier in the shift or later in the shift according to one aspect of the present invention.

[0016] FIGS. **8**A-C are graphical diagrams showing graphs to illustrate techniques for relocating a portion of a shift to earlier in time or later in time to find extra time to fit an order.

DETAILED DESCRIPTION OF THE INVENTION

[0017] In the following detailed description of exemplary embodiments of the invention, reference is made to the accompanying drawings, which form a part hereof, and in which are shown, by way of illustration, specific exemplary embodiments in which the invention may be practiced. In the drawings, like numerals describe substantially similar components throughout the several views. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. Other embodiments may be utilized and structural, logical, electrical, and other changes may be made without departing from the spirit or scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims. The present application incorporates by reference the following reference: Guy Druce, *Level*-1 *Algorithm,* v. 1.6 (Dec. 4, 2000).

[0018] **FIG. 1** is a block diagram of a system **100** according to one aspect of the present invention. The system **100** includes a service organization **102**. The service organization **102** performs services for a roster of customers **108**. When a customer **108** is interested in having a service performed by the service organization **102**, the customer **108** calls the service organization **102** to make a reservation for the service performed.

[0019] Using the scheduling system **104**, the service organization **102** negotiates with the customer **108** to place a reservation in a schedule. The scheduling system **104** provides to the service organization **102** several appointment windows from which the customer **108** may choose. The customer **108** selects an appointment window in which the service requested will be performed.

[0020] During the negotiation, the scheduling system **104** determines which mobile service representatives **106** can do the work. The customer **108** picks an appointment window and the scheduling system **104** assigns to the order to a mobile service representative. A mobile service representative **106** is one who is capable of performing the service requested by the customer **108**. Each time a reservation is placed, the scheduling system **104** accounts for many factors in assigning the reservation to a mobile service representative **106**, such as time availability, skill sets, geographic area, duration of each job, travel times between jobs, and equipment requirements. As customers **108** place more reservations for services, the scheduling system **104** periodically optimizes the schedule by relocating reservations assigned to a single mobile service representative or swapping reservations between different mobile service representatives.

[0021] The scheduling system **104** can be implemented over one machine or several machines with different processors. This allows the scheduling system **104** to be scalable depending on the needs of the service organization **102**. The scheduling system **104** is also configurable. This allows the scheduling system **104** to meet different constraints and objectives. In one embodiment, the scheduling system **104** is implemented over Common Object Request Broker Architecture (CORBA). In another embodiment, the scheduling system **104** is implemented using a database, such as an Oracle database.

[0022] **FIG. 2** is a block diagram of a scheduling system **200** according to one aspect of the present invention. The scheduling system **200** is similar to the scheduling system **104** of **FIG. 1**. The scheduling system **200** includes a finder **202**. When a customer calls the service organization to place an order for service, the finder **202** finds one or more openings in the schedule in which the order may fit. Recall that prior systems may take a prohibitively long period of time to find openings in the schedule. In contrast, the embodiments of the present invention through the implementation of the finder **202** return one or more openings within an amount of time so that the customer need not wait on the phone for an undesirably long period of time.

[0023] After the finder **202** finds one or more openings, the negotiator **204** may present these openings as appointment windows to the service organization. The service organization can discuss these appointment windows with the customer and negotiate with the customer to select an appointment window to perform the service in accordance with the order. The customer may make specific demands, such as a request for a particular worker to perform the service or a request for a particular set of skills in the performance of the service. The finder **202** can find openings in the schedule associated with a particular worker.

[0024] When the customer has selected an appointment window, the assigner **206** proceeds to assign the order to the opening associated with the appointment window. When the order is assigned, the optimizer **208** optimizes the schedule so as to accommodate additional orders. In one embodiment, the optimizer **208** defragments the available spaces in a shift to fit additional orders in the shift.

[0025] **FIG. 3** is a process diagram of a method **300** for finding an opening in a schedule to fit an order to a shift of a worker according to one aspect of the present invention. The method **300** begins with an act **302** for generating a list of shifts. The act **302** generates the list of shifts using a window over the schedule. The window helps to control the amount of information to find each shift in the list of shifts. Each shift in the list of shifts may contain enough time to accommodate the order. The act **302** also includes a set of candidacy rules that allow the act to be more selective about each shift in the list of shifts.

[0026] The method **300** includes an act **304** for generating a list of virtual free time blocks. The list of virtual free time blocks is generated for a shift of a worker. A virtual free time block is an amount of time that could be used to fit an order by either bumping one or more contiguous orders within a shift or by a free time block. The term "free time block" means a block of free time between the end of the work on one order and the start of travel to another order. Thus, by bumping one or more contiguous orders within a shift, a virtual free time block may accumulate enough free time to fit an order into the shift. The term "contiguous" means next to or near in time or sequence.

[0027] The method **300** includes an act **306** for iterating through each shift in the list of shifts generated in the act **302**. The act **306** allows a set of acts to execute for each shift in the list of shifts. This set of acts help to find one or more openings in the list of shifts so as to fit an order into the schedule for a customer. Recall that the list of shifts is generated from a window over the schedule to control the amount of information that must be processed to find one or more openings. This set of acts is described in greater detail hereinbelow.

[0028] The method **300** includes an act **308** for iterating through each virtual free time block from a list of virtual free time blocks. The list of virtual free time blocks is generated from the act **304** for each shift in the list of shifts generated in the act **302**. In one embodiment, each virtual free time block is a free time block. Another abstraction of the free time block called a schedulable time block keeps track of the virtual free time blocks that can be gathered to fit an order into a shift. The schedulable time block is discussed in greater detail hereinbelow.

[0029] The act **308** allows a set of acts to execute for each virtual free time block in the list of virtual free time blocks. This set of acts help to identify whether the virtual free time block is a candidate to become an opening in the shift so as to fit an order. The combination of act **306** and act **308** searches through the list of shifts and the virtual free time blocks in the list of shifts for one or more openings to fit an order.

[0030] The method **300** includes an act **310** for finding an opening. The act **310** comprises other acts that are executed to find one or more openings to fit an order. The act **310** is iterated by the act **308** for each virtual free time block in the list of virtual free time blocks and by the act **306** for each shift in the list of shifts.

[0031] The method **300** also includes an act **312** for assigning an order. When one or more openings are found, these openings are presented to a customer. The customer selects a desired opening. The act **312** attempts to assign the order to the opening. If the order is successfully assigned to the opening, the order is assigned to a shift of a worker. On the appointed date and time, the worker proceeds to perform the service as specified in the order.

[0032] **FIG. 4** is a process diagram of a method **400** for finding an opening in a schedule to fit an order to a shift of a worker according to one aspect of the present invention. The method **400** discusses the act **310** of **FIG. 3** in greater detail. The method **400** includes an act **402** for computing an amount of free time required to accommodate the order. The amount of free time required to accommodate the order is a sum of three terms. One of the three terms is a travel time, another is a job time, and the third term is a difference travel time. These terms are explained by proceeding to **FIG. 5**.

[0033] **FIG. 5** is a process diagram of a method **500** for computing an amount of free time required to accommodate an order in a shift. The method **500** includes an act **502** for calculating a travel time T between a first activity and a second activity. The travel time T is defined as the time to travel from a location a to another location b. The travel time T can be mathematically described as T=t(a,b). The term "t" is a function that returns a time period for traveling between the location a and the location b.

[0034] The method **500** also includes an act **504** for calculating a difference travel time. The difference travel time can be explained by an example. Suppose the shift comprises sabFcd. The term "s" signifies the start of the shift. The terms "a,""b,""c," and "d" are work activities that are to be performed by a worker. The term "F" signifies a virtual free time block. Thus, the time travel to a equals t(s,a). The time travel to b equals t(a,b). The time travel to F equals 0. The time travel to c equals t(b,c). The time travel to d equals (c,d).

[0035] Suppose that a new order n is to be inserted into the virtual free time block F. The time travel to n equals t(b,n). And the difference travel time is mathematically defined as: $\delta=t(n,c)-t(b,c)$. Thus, the difference travel time is a result of the travel time from the new order to the activity c subtracting the travel time from the activity b to the activity c. This difference travel time is the time calculated by the act **504**.

[0036] The method **500** includes an act **506** for calculating a job time. The job time is the amount of time required to perform a service as specified by an order. The method **500** sums the travel time T between a first activity and a second activity, the difference travel time, and the job time to form a summation. This summation is the amount of free time required in a shift to fit the order.

[0037] Returning to **FIG. 4**, act **402** produces the amount of free time required in a shift from a sum of the travel time T between a first activity and a second activity, the difference travel time, and the job time. The method **400** also includes an act **404** for creating a schedulable time block from a virtual free time block. The schedulable time block includes a primary block. The schedulable time block may include at least one expansion block, at least one load block, or both. The primary block is a block of time into which the order may be inserted. The primary block may not have adequate time to fit an order however. The expansion block and the load block help to provide some extra time that may increase the time of the primary block to fit the order. The expansion block finds extra time in the shift by relocating other assigned orders in the shift. The load block finds extra time in the shift by removing assigned orders in the shift so that the total load of the shift is kept below a worker's load limit.

[0038] The method **400** includes an act **406** for examining the primary block to see if the order can be fitted into the primary block. The act **406** comprises a set of acts that determine whether primary block is a candidate to fit the order. This set of acts may try to find extra time in the shift by shuffling assigned orders earlier or later in time in the shift or by removing assigned orders. To focus on this set of acts in greater detail, the discussion proceeds to FIGS. **6**A-**6**B.

[0039] FIGS. **6**A-**6**B are a process diagram of a method **600** for examining a primary block according to one aspect of the present invention. The method **600** includes an act **602** for examining the primary block. The act **602** deems that the primary block is a candidate to fit the order if the duration of the primary block, excluding any breaks, is greater than or equal to the amount of free time required in the shift to fit the order. An act **604** follows the act **602** to check whether the primary block is a candidate. If the primary block is a candidate, the method **600** progresses to node **613**B to other acts.

[0040] If the primary block is not a candidate, this indicates that the primary block has insufficient time to fit the order. To find extra time, the method **600** progresses to an act **606** for finding the extra time by relocating a portion of the shift that is filled with assigned orders to the right of the primary block. In other words, the relocation of the portion of the shift is a shuffle of assigned orders to later in the time of the shift so as to create extra time that can be added to the primary block to fit an unassigned order. If extra time is found, the extra time is added as an expansion block to the schedulable time block created in the act **404** of **FIG. 4**. In order to focus in greater detail on how the extra time in the shift is found, the discussion further proceeds to **FIG. 7**.

[**0041**] **FIG. 7** is a process diagram of a method **700** for computing extra time that can be obtained by relocating assigned orders earlier in the shift or later in the shift according to one aspect of the present invention. The method **700** begins at the act **702** for calculating an amount of time available in a virtual free time block for fitting an order. This amount of time is represented as t.

[**0042**] The method **700** includes an act **704** for computing an amount of time in which a portion of the shift must be relocated. This amount of time is represented as Dt. Dt equals EOT −t. EOT is the amount of free time required in a shift to fit the order, which is calculated by act **404** of **FIG. 4**. t is the amount of time available in the virtual free time block for fitting an order as discussed above.

[**0043**] When Dt is computed, the method **700** progresses to an act **706** for comparing Dt to zero. If Dt is a negative quantity, this indicates that there is enough time in the virtual free time block to fit the order, and the method **700** ends its execution. If Dt is a positive quantity, this indicates that a portion of the shift must be relocated, and the method **700** progresses to the act **708**.

[**0044**] The act **708** computes an amount of time in which the portion of the shift can be relocated. This amount is represented as t_{min}. The portion of the shift can be relocated earlier in time or later in time relative to the virtual free time block in which an order is inserted. By relocating the portion of the shift, more free time may become available to add to the virtual free time block to accommodate the order.

[**0045**] Visually, relocating the portion of the shift to the left means relocating the portion of the shift to an earlier point in time. Correspondingly, relocating the portion of the shift to the right means relocating to a later point in time. To illustrate in greater detail the computation of the amount of time in which the portion of the shift can be relocated, the discussion proceeds to FIGS. **8**A-C.

[**0046**] FIGS. **8**A-C are graphical diagrams showing graphs to illustrate techniques for relocating a portion of the shift to earlier in time or later in time to find extra time to fit an order. **FIG. 8A** is a graphical diagram showing a graph **800** to illustrate a technique for relocating a portion of the shift. The graph **800** includes a shift **802** that defines the work period of a worker.

[**0047**] The shift **802** includes a virtual free time block **804**. For the purpose of this illustration, the virtual free time block **804** is the time block into which an order is to be fitted. The shift **802** includes other assigned orders, such as order **806**, order **808**, and order **810**. Order **806** is designated as Order **1**. Order **808** is designated as Order **2**. And order **810** is designated as Order **3**. The shift **802** includes travel time to get from one assigned order to the next. The shift **802** includes appointment windows **812**, **814**, and **816**. The appointment window **812** defines a time period during the shift **802** in which a worker could start to work on Order **1**. The appointment window **814** similarly defines a time period in which the worker could start to work on Order **2**. And the appointment window **816** defines a time period for a worker to start to work on Order **3**.

[**0048**] The graph **800** also illustrates a maximum relocation for each of the Orders **1**, **2**, and **3**. There are three maximum relocation figures, **818**, **820**, and **822**. The maximum relocation figure denotes the maximum relocation that can be done for a particular assigned order. To relocate to a point later in time, the maximum relocation is formed from taking the absolute value of the subtraction of the starting time of an assigned order from the end of the appointment window for the assigned order. To relocate to a point earlier in time, the maximum relocation is formed from taking the absolute value of the subtraction of the starting time of an assigned order from the start of the appointment window for the assigned order. To calculate t_{min}, which is the amount of time the portion of the shift can be relocated, the minimum of the three maximum relocation figures **820**, and **822** is taken. Thus, in the graph **800**, it can be seen that the maximum relocation figure **822** is the minimum.

[**0049**] To summarize what has been discussed hereinbefore, suppose a new order is to be inserted in place of the virtual free time block **804**. Because the virtual free time block **804** is insufficient to accommodate the new order, an extra amount of time is required. This amount of time is Dt. The extra amount of time can be obtained by relocating Orders **1**, **2**, and **3**. For example, Orders **1**, **2**, and **3** can be relocated to the right of the virtual free time block **804**. Order **2** has the least amount of time available between the end of its appointment window and the start time of the Order **2**. Hence, the maximum amount of time in which the Orders **1**, **2**, and **3** may be relocated is given by t_{min}. t_{min} is given by the maximum relocation figure **822** as shown in the graph **800**.

[**0050**] **FIG. 8B** is a graphical diagram illustrating a technique for relocating a portion of the shift to a later point in time. The graph **800** includes a shift $\mathbf{802}_1$. The shift $\mathbf{802}_1$ is similar to the shift **802** discussed in **FIG. 8A** but is different in that the shift $\mathbf{802}_1$ includes another virtual free time block **824** in addition to the virtual free time block **802**. Recall that the virtual free time block **804** is the block in which the order is to be inserted. Given the presence of the virtual free time block **824**, Orders **1**, **2**, and **3** and the travel time associated with these orders can be relocated to the right. The maximum amount of relocation is given by t_{min}. Thus, in the shift $\mathbf{802}_2$, the virtual free time block **804** has been enlarged by t_{min}, and correspondingly, the virtual free time block **824** has been compressed by t_{min}.

[**0051**] **FIG. 8C** is a graphical diagram illustrating a technique for relocating a portion of the shift to an earlier point in time. The graph **800** of **FIG. 8C** is similar to the graph **800** of **FIG. 8B** except for the location of the virtual free time blocks **804** and **824**. Recall that the virtual free time block **804** is the block in which the order is to be inserted. Given the presence of the virtual free time block **824**, Orders **1**, **2**, and **3** and the travel time associated with these orders can be relocated to the left. The maximum amount of relocation is given by t_{min}. Thus, in the shift $\mathbf{802}_2$, the virtual free time block **804** has been enlarged by t_{min}, and correspondingly, the virtual free time block **824** has been compressed by t_{min}.

[**0052**] Returning to act **708** of **FIG. 7**, $\mathbf{t}_{min}$ is now computed. Thus, at this point in the method **700**, two quantities are known: the amount of time by which the portion of the shift must be relocated Dt and the amount of time by which the portion of the shift can be relocated t_{min}. The method **700** progresses to an act **712** for comparing t_{min} to Dt.

[0053] If t_{min} is less than Dt, this means that no extra time can be found in the shift even with the relocation of the portion of the shift, and the method **700** progresses to an act **710**. The act **710** loops to the next virtual free time block in the list of virtual free time blocks as discussed in the act **308** of **FIG. 3**. Otherwise, if t_{min} is greater than Dt, this means that there is extra time to fit the order, and the method **700** completes its execution.

[0054] Returning to the act **606** of **FIG. 6**A, recall that the act **606** finds the extra time by relocating the portion of the shift to the right of the primary block. If the relocation succeeds in finding enough time so that the extra time added to the duration of the primary block is greater than or equal to the amount of free time required in the shift to fit the order, then the primary block can be used to fit the order. In this case, the extra time is added as expansion block to the schedulable time block created in the act **404** of **FIG. 4**.

[0055] An act **608** follows the act **606** to check whether the primary block is a candidate. If the primary block is a candidate, the method **600** progresses to the node **613**B and to other acts. Otherwise, the method **600** progresses to an act **610**.

[0056] The act **610** finds the extra time by relocating the portion of the shift to the left of the primary block. In other words, the method **600** attempts to relocate the portion of the shift to an earlier point in time so as to find extra time in which to fit the order. If the relocation succeeds in finding enough time so that the extra time added to the duration of the primary block is greater than or equal to the amount of free time required in the shift to fit the order, then the primary block can be used to fit the order. The extra time is added as expansion block to the schedulable time block created in the act **404** of **FIG. 4**.

[0057] An act **612** follows the act **610** to check whether the primary block is a candidate. If the primary block is a candidate, the method **600** progresses to the node **613**B and to an act **616** of **FIG. 6B**. Otherwise, the method **600** progresses to a node **613**A and to an act **614**. The act **614** eliminates the virtual free time block from further consideration. The virtual free time block is derived from the schedulable free time block that was created in the act **404** of **FIG. 4**. The next virtual free time block is then considered as discussed in act **308** of **FIG. 3**.

[0058] If the method **600** progresses to the act **616**, the virtual free time block from which the schedulable time block is derived can be expanded to suit the order. However, because each shift has a load limit, the addition of the amount of free time required to fit the order to the shift may exceed this load limit. If the load limit is not exceeded, the method **600** completes its execution.

[0059] Otherwise, the method **600** progresses to an act **618** to reduce the load of the shift so as to fit the order. In order to reduce the load of the shift, the act **618** finds at least one virtual free time block in the shift whose orders can be removed to reduce the load. If the act **618** finds an order in a virtual free time block to be removed, that virtual free time block will be added to the load block, which was created in the act **404** of **FIG. 4**. The method **600** then terminates and returns to the end of the act **406** of **FIG. 4**.

[0060] If the act **618** cannot find any virtual free time block in the shift that can be removed, the method **600** progresses to an act **620**. The act **620** eliminates the virtual free time block, from which the schedulable time block was derived, from further consideration, and the method **600** terminates its execution. The next virtual free time block is then considered as discussed in the act **308** of **FIG. 3**.

[0061] Returning to the end of the act **406** of **FIG. 4**, recall that the act **406** examines the primary block to see whether the order can be fit into the primary block. The act **406** comprises a set of acts that determine whether the primary block is a candidate to fit the order. This set of acts, which are discussed in greater detail in FIGS. **6**A-**6**B, may try to find extra time in the shift by relocating assigned orders earlier or later in time in the shift or by removing assigned orders. The method **400** progresses to an act **408**.

[0062] The act **408** creates one or more openings from the schedulable time block created in the act **404**. These openings are added to a list of openings that are returned to the service organization. The service organization may use these openings to define a range of times in which work on the order may start. This can be used to negotiate with a customer to set a time for an appointment to perform the order.

CONCLUSION

[0063] A technique has been discussed for finding one or more openings in a schedule so as to allow a service organization to address the preferences of customers while allowing the service organization to satisfy constraints placed on the scheduling system and to meet business objectives. A scheduling system comprises three main components. The negotiator interacts with customers to negotiate an appointment window in which the service is performed as specified in the reservation. The assigner assigns the reservation to a shift of a desired mobile service representative. And the optimizer continuously optimizes the schedule in the background. The embodiments of the present invention focus on providing one or more openings within a predetermined period of time for the negotiator to interact with customer in a timely manner. Because customers dislike waiting on the phone for an extended period of time, the embodiments of the present invention enable the negotiator to work fast enough to obtain an agreeable time window and worker to perform the service requested.

[0064] Although the specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement, which is calculated to achieve the same purpose, may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the present invention. It is to be understood that the above description is intended to be illustrative, and not restrictive. Combinations of the above embodiments and other embodiments will be apparent to those of skill in the art upon reviewing the above description. The scope of the invention includes any other applications in which the above structures and fabrication methods are used. Accordingly, the scope of the invention should only be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.

I claim:

1. A method for scheduling, comprising:

forming a list of shifts in a schedule;

determining a shift in which an order can be scheduled; and

assigning the order to the shift determined by the act of determining, wherein the acts of forming, determining, and assigning are contemporaneously executed while a customer is on a phone attempting to schedule the order.

2. The method of claim 1, wherein forming includes forming a window over a schedule of a worker, wherein the schedule of the worker includes a number of shifts, wherein the window encompasses a portion of the number of shifts in the schedule of the worker, and wherein forming a list of shifts includes selecting at least one shift from the portion of the number of lists encompassed by the window.

3. The method of claim 1, wherein assigning includes assigning to an earlier time period of the shift before assigning to a later time period of the shift.

4. A method for finding an opening in which to fit an order in a schedule, comprising:

computing an amount of free time required in a shift to fit the order;

creating a schedulable time block from a virtual free time block, wherein the schedulable time free block includes a primary block, zero or more expansion blocks, and zero or more load blocks;

examining the primary block, wherein the primary block is a candidate to fit the order if a duration of the primary block, excluding at least one break is greater than or equal to the amount of free time required in the shift to fit the order; and

creating at least one opening in the shift from the schedulable time block so as to present to a customer at least one option of fitting the order in the schedule to perform a desired service.

5. The method of claim 4, further comprising generating a list of shifts from a window defined over a set of shifts of a worker.

6. The method of claim 5, further comprising generating a list of virtual free time blocks from a shift of a worker.

7. The method of claim 6, further comprising calculating a travel time between a first activity and a second activity.

8. The method of claim 7, wherein calculating a travel time includes calculating a difference travel time when the order is inserted into a virtual free time block of the shift of the worker, wherein the different travel time is defined as a result of a subtraction of the travel time between the first activity and the second activity and the travel time of the order and the second activity, and wherein the virtual free time block that the order is inserted into is between the first activity and the second activity.

9. The method of claim 8, further comprising calculating a job time, wherein the job time is defined as the time that the order will take to be performed in the shift.

10. The method of claim 9, wherein computing an amount of free time required in the shift to fit the order includes summing the travel time, the difference travel time, and the job time.

11. The method of claim 10, further comprising computing an extra time by relocating a portion of the shift to fit the order.

12. The method of claim 11, wherein computing an extra time by relocating a portion of the shift includes computing an amount of time that the portion of the shift must be shifted to fit the order, wherein the amount of time that the portion of the shift must be shifted is defined as a result of a subtraction of the amount of free time required in the shift to accommodate the order and a time available in the virtual free time block.

13. The method of claim 11, wherein computing an extra time by relocating a portion of the shift includes computing an amount of time that the portion of the shift can be shifted, wherein computing an amount of time that the portion of the shift can be relocated includes aggregating a number of virtual free time blocks in the portion of the shift.

14. The method of claim 11, further comprising executing the act of computing the extra time by relocating a portion of the shift to later in time in the shift if the act of examining the primary block determines that the primary block is not a candidate, wherein the primary block is a candidate to fit the order if the extra time plus the duration of the primary block is greater than or equal to the amount of free time required in the shift to fit the order, and updating the at least one expansion block if the primary block is a candidate.

15. The method of claim 14, further comprising executing the act of computing the extra time by relocating a portion of the shift to earlier in time in the shift if the act of executing the act of computing the extra time by relocating a portion of the shift to later in time and the act of examining the primary block determine that the primary block is not a candidate, executing the act of examining the primary block, wherein the primary block is a candidate to fit the order if the extra time plus the duration of the primary block is greater than or equal to the amount of free time required in the shift to fit the order, and updating the at least one expansion block if the primary block is a candidate.

16. The method of claim 15, further comprising eliminating the virtual free time block from further consideration if the act of executing the act of computing the extra time by relocating a portion of the shift to earlier in time in the shift and the act of examining the primary block determine that the primary block is not a candidate.

17. The method of claim 16, further comprising checking a load limit of the shift if the primary block is being expanded by the extra time to fit the order, wherein checking includes adding the amount of free time required in the shift to fit the order to a current load of the shift to define a new load, and wherein checking includes comparing the new load against the load limit.

18. The method of claim 17, further comprising reducing a total load of the shift by finding at least one virtual free time blocks to be removed, wherein if the act of finding finds at least one virtual free time block to be removed, the act of reducing executes an act of adding the at least one virtual free time block to be removed, and updating the at least one load block if the act of finding finds at least one virtual free time block to be removed.

19. The method of claim 18, further comprising eliminating the virtual free time block if the act of reducing fails to reduce the total load of the shift to fit the order.

20. The method of claim 19, further comprising iterating a set of the above acts for each virtual free time block in the list of virtual free time blocks, wherein the set excludes some of the above acts.

21. The method of claim 20, further comprising iterating the act of claim 20 for each shift in the list of shifts.

22. A computer-readable medium having instructions stored thereon for causing a computer to perform a method for finding an opening to fit an order in a schedule, the method comprising:

computing an amount of free time required in a shift to fit the order;

creating a schedulable time block from a virtual free time block, wherein the schedulable time block includes a primary block, at least one expansion block, and at least one load block;

examining the primary block, wherein the primary block is a candidate to fit the order if a duration of the primary block, excluding at least one break is greater than or equal to the amount of free time required in the shift to fit the order; and

creating at least one opening in the shift from the schedulable time block so as to present a customer with at least one option of fitting the order in the schedule to perform a desired service.

23. The method of claim 22, further comprising generating a list of shifts from a window defined over a set of shifts of a worker.

24. The method of claim 23, further comprising generating a list of virtual free time blocks from a shift of a worker.

25. The method of claim 24, further comprising calculating a travel time between a first activity and a second activity.

26. The method of claim 25, wherein calculating a travel time includes calculating a difference travel time when the order is inserted into a virtual free time block of the shift of the worker, wherein the different travel time is defined as a result of a subtraction of the travel time between the first activity and the second activity and the travel time of the order and the second activity, and wherein the virtual free time block that the order is inserted into is between the first activity and the second activity.

27. The method of claim 26, further comprising calculating a job time, wherein the job time is defined as the time that the order will take to be performed in the shift.

28. The method of claim 27, wherein computing an amount of free time required in the shift to fit the order includes summing the travel time, the difference travel time, and the job time.

29. The method of claim 28, further comprising computing an extra time by relocating a portion of the shift to fit the order.

30. The method of claim 29, wherein computing an extra time by relocating a portion of the shift includes computing an amount of time that the portion of the shift must be shifted to fit the order, wherein the amount of time that the portion of the shift must be relocated is defined as a result of a subtraction of the amount of free time required in the shift to accommodate the order and a time available in the virtual free time block.

31. The method of claim 29, wherein computing an extra time by relocating a portion of the shift includes computing an amount of time that the portion of the shift can be shifted, wherein computing an amount of time that the portion of the shift can be shifted includes aggregating a number of virtual free time blocks in the portion of the shift.

32. The method of claim 29, further comprising executing the act of computing the extra time by relocating a portion of the shift to later in time in the shift if the act of examining the primary block determines that the primary block is not a candidate, wherein the primary block is a candidate to fit the order if the extra time plus the duration of the primary block is greater than or equal to the amount of free time required in the shift to fit the order, and updating the at least one expansion block if the primary block is a candidate.

33. The method of claim 32, further comprising executing the act of computing the extra time by relocating a portion of the shift to earlier in time in the shift if the act of executing the act of computing the extra time by relocating a portion of the shift to later in time and the act of examining the primary block determine that the primary block is not a candidate, executing the act of examining the primary block, wherein the primary block is a candidate to fit the order if the extra time plus the duration of the primary block is greater than or equal to the amount of free time required in the shift to fit the order, and updating the at least one expansion block if the primary block is a candidate.

34. The method of claim 33, further comprising eliminating the virtual free time block from further consideration if the act of executing the act of computing the extra time by relocating a portion of the shift to earlier in time in the shift and the act of examining the primary block determine that the primary block is not a candidate.

35. The method of claim 34, further comprising checking a load limit of the shift if the primary block is being expanded by the extra time to fit the order, wherein checking includes adding the amount of free time required in the shift to fit the order to a current load of the shift to define a new load, and wherein checking includes comparing the new load against the load limit.

36. The method of claim 35, further comprising reducing a total load of the shift by finding at least one virtual free time blocks to be removed, wherein if the act of finding finds at least one virtual free time block to be removed, the act of reducing executes an act of adding the at least one virtual free time block to be removed, and updating the at least one load block if the act of finding finds at least one virtual free time block to be removed.

37. The method of claim 36, further comprising eliminating the virtual free time block if the act of reducing fails to reduce the total load of the shift to fit the order.

38. The method of claim 37, further comprising iterating a set of the above acts for each virtual free time block in the list of virtual free time blocks, wherein the set excludes some of the above acts.

39. The method of claim 38, further comprising iterating the act of claim 17 for each shift in the list of shifts.

* * * * *