

[0021] Embodiment 10 is the computer-implemented method of any one of embodiments 1 through 9, further comprising: determining, by the computing system, a third value that indicates utilization of the multiple moving machines based on the multiple moving machines performing the multiple queues of item-movement tasks while the multiple queues of item-movement tasks have a third state; modifying, by the computing system, a third item-movement task from the first queue of item-movement tasks while the multiple queues of item-movement tasks have the third state (i) from selecting an original item from an original source location (ii) to selecting an alternative item from an alternative location to form a fourth state of the multiple queues of item-movement tasks, the alternative item being a same type of item as the original item; determining, by the computing system, a fourth value that indicates utilization of the multiple moving machines based on the multiple moving machines performing the multiple queues of item-movement tasks while the multiple queues of item-movement tasks have the fourth state; determining, by the computing system based on comparison of the third value to the fourth value, that utilization of the multiple moving machines is greater when the multiple queues of item-movement tasks have the third state; and modifying, by the computing system based on having determined that the utilization of the multiple moving machines is greater when the multiple queues of item-movement tasks have the third state, the third item-movement task to select the original item from the original source location, to revert the multiple queues of item-movement tasks to the third state.

[0022] Embodiment 11 is a computing system comprising one or more processors and one or more computer-readable devices including instructions that, when executed by the one or more processors, cause the computing system to perform the computer-implemented method of any one of embodiments 1 to 10.

[0023] The devices, system, and techniques described herein may provide one or more of the following advantages. For example, the disclosed technology can be used to reduce an amount of travel time or distance between tasks for warehouse workers. Typically, a warehouse worker may be assigned only inbound tasks or only outbound tasks but not both. For example, the warehouse worker may be assigned a queue of inbound tasks and another warehouse worker may be assigned a queue of outbound tasks. Using the disclosed technology, the warehouse worker can be assigned both inbound and outbound tasks in order to reduce an amount of time that the

[0055] FIG. 2 is a conceptual diagram of a state of the warehouse environment 100 after a first swap of task assignments. Although one swap is depicted in FIG. 2, multiple swaps can be performed by the computing system 114 at a given time. The computing system 114 can perform multiple swaps at a time (e.g., every second) in order to keep up with how fast a state of the warehouse environment 100 can change. Thus, a most optimal task swap and selection can be made in a short period of time.

[0056] At time = 1, the computer system 114 can swap tasks (step A, 204). In other words, the computer system 114 can assign a task to be completed by the forklift 104B to the forklift 104A and vice versa. In some implementations, the computer system 114 can swap destinations for completing one or more tasks that are assigned to one or more of the forklifts 104A-B. The computer system 114 can also swap source items/pallets for completing one or more tasks that are assigned to one or more of the forklifts 104A-B. In some implementations, the computer system 114 can swap first tasks that are assigned to each of the forklifts 104A-B. In some implementations, the computer system 114 can randomly select which tasks to swap between the forklifts 104A-B. Thus, inbound tasks can be swapped with inbound tasks, outbound tasks can be swapped with outbound tasks, and inbound tasks can be swapped with outbound tasks.

[0057] When choosing which tasks to swap, the computer system 114 can first determine whether it is feasible for either of the forklifts 104A-B to perform the swapped tasks. For example, the computer system 114 can determine whether the forklift 104B has necessary equipment to complete a task that would have been completed by the forklift 104A (e.g., forklifts can complete tasks on different storage levels in the storage area 106 while a warehouse jack may only be able to complete tasks on low storage levels in the storage area 106). If the forklift 104B does have the necessary equipment to complete the task, then the computer system 114 can continue to assess whether implementing the swap can result in an improvement of global efficiency and forklift efficiency.

[0058] Once the computer system 114 swaps the tasks, the computer system 114 can determine a change in time efficiency for each of the forklifts 104A-B (step B, 206). Based on the change in time efficiency, the computer system 114 can determine whether swapping the tasks resulted in an improvement on global warehouse and/or forklift efficiency.

with 1 minute of empty travel time along the empty travel route 110A. Thus, the computer system 114 can determine that it may be efficient to keep the swap since the forklift 104B can complete the task 112A' with 30 seconds of empty travel time along the empty travel route 110A'. Regardless, when the 30 seconds of empty travel time is combined with the other empty travel times to determine a total empty travel time for the forklift 104B based on the swap, the total empty travel time for the forklift 104B may not be improved.

[0063] The computer system 114 also projected that based on the swap, the forklift 104B can complete the originally assigned task 112E next in 3 minutes. However, the computer system 114 projected that the forklift 104B can experience 5 minutes of empty travel time along empty travel route 110E' in order to complete the task 112E. In the initial task assignment depicted in FIG. 1, the computer system had projected that the forklift 104B could complete the task 112E in 3 minutes and experience 2 minutes of empty travel time along the empty travel route 112E. Thus, the computer system 114 can determine that although the forklift 104B can complete the task 112E in 3 minutes whether or not the tasks are swapped, the forklift 104B can experience 3 more minutes of empty travel time along the empty travel route 110E'. Therefore, swapping the tasks may not be efficient.

[0064] Based on the swap, the computer system 114 can also generate a task swap time efficiency table 202. The table 202 can indicate projected total empty travel time per forklift based on the swap. The table 202 also indicates projected global empty travel time for the warehouse environment 100. Based on assigning the forklift 104A the task 112D' and assigning the forklift 104B the task 112A', the forklift 104A is projected to experience a total of 10 minutes and 30 seconds of empty travel time. Under the initial task assignments depicted in FIG. 1, the forklift 104A was only expected to experience 5 minutes and 30 seconds of empty travel time. Thus, the swap would cause the forklift 104A to experience 5 more minutes of empty travel time, which is an indication of increased inefficiency resulting from the swap.

[0065] The forklift 104B is projected to experience a total of 6 minutes and 30 seconds of empty travel time based on the swap. Under the initial task assignments depicted in FIG. 1, the forklift 104B was expected to experience 4 minutes and 30 seconds of empty travel time. Thus, the swap would cause the forklift 104B to experience 2 more minutes of empty travel time, which is an indication of increased inefficiency resulting from the swap.

[0066] Finally, the projected global empty travel time for the warehouse environment 100 amounts to 17 minutes based on the swap. Under the initial task assignments depicted in FIG. 1, the projected global empty travel time was 10 minutes. Thus, the swap would cause increased inefficiency in the overall warehouse environment 100.

[0067] Based on such analysis and as described above, the computer system 114 can revert the swap and keep the initial task assignments (step C, 208).

[0068] FIG. 3 is a conceptual diagram of a state of the warehouse environment 100 after a second swap of task assignments. At time = 2, the computer system 114 can swap location(s) of task(s) (step A, 304). Not only did the computer system 114 assign task 112D' to forklift 104A, the computer system 114 also changed a location of the original task 112B, thereby modifying the original task 112B to task 112B'. The computer system 114 also assigned the original task 112A to the forklift 104B and changed a location of the original task 112A, thereby modifying the original task 112A to task 112A''. The swaps made by the computer system 114 are depicted in location swap table 300.

[0069] Once the computer system 114 swaps the tasks, the computer system 114 can determine a change in time efficiency for each of the forklifts 104A-B (step B, 306). Based on the change in time efficiency, the computer system 114 can determine whether swapping the tasks resulted in an improvement on global warehouse and/or forklift efficiency.

[0070] As shown in FIG. 3, the computer system 114 can keep the swap (step C, 308). The computer system 114 can make this determination when the change in time efficiency for each of the forklifts 104A-B is improved from the time efficiency metrics that were projected for the initial task assignments and/or any previous swaps. Moreover, the computer system 114 can assign the swapped tasks to each of the forklifts 104A-B (step D, 310).

[0071] Based on the example swap in FIG. 3, the computer system 114 projected that the forklift 104A can complete the task 112D' in 1 minute 30 seconds, but will experience 2 minutes of empty travel time along empty travel route 110D'. In the first attempted swap depicted in FIG. 2, the computer system 114 had projected that the forklift 104B could complete the task 112D' in 1 minute and 30 seconds with 2 minutes of empty travel time along the empty travel route 110D'. Thus, the computer system 114 can determine that there is no improvement in efficiency with the swap of FIG. 3.

[0072] The computer system 114 also projected that based on changing the task 112B' location from 122B to 122B', the forklift 104A can complete the task 112B' in 4 minutes and 30 seconds and experience 3 minutes of empty travel time along the empty travel route 110B''. In the first attempted swap depicted in FIG. 2, the computer system 114 projected that the forklift 104A could complete the task 112B in 6 minutes and experience 5 minutes and 30 seconds of empty travel time along the empty travel route 110B. The computer system 114 can therefore determine that changing the task 112B' location from 122B to 122B' resulted in saving the forklift 104A 2 minutes and 30 seconds in empty travel time. Thus, efficiency of the forklift 104A improved from the swap in FIG. 3.

[0073] Additionally, based on assigning the task 112A'' to the forklift 104B and changing the task 112A'' location from 122A to 122A', the forklift 104B can complete the task 112A'' in 1 minute and experience 30 seconds of empty travel time along empty travel route 110A'. In the first attempted swap depicted in FIG. 2, the computer system 114 projected that the forklift 104B could complete the task 112A' in 5 minutes and experience 30 seconds of empty travel time along the empty travel route 110A'. Although the empty travel time is the same in both swaps, the computer system 114 can determine that the swap in FIG. 3 is better than the swap in FIG. 2 because the forklift 104B can complete the task 112A'' in 3 minutes and 30 seconds less time than completing the task 112A'. Thus, changing or swapping the destination location of the task 112A'' from location 122A to location 122A' improved efficiency of the forklift 104B.

[0074] Based on changing the destination location 122A' for the task 112A'', the computer system 114 also projected that the forklift 104B can complete the task 112E in 3 minutes and experience 1 minute of empty travel time along empty travel route 110E''. In the first attempted swap depicted in FIG. 2, the computer system 114 projected that the forklift 104B could complete the task 112E in 3 minutes and experience 5 minutes of empty travel time along the empty travel route 110E'. The long amount of empty travel time is because the forklift 104B would have to travel from location 122A in aisle 124B to location 122B in aisle 124A. By changing the destination location of the task 112A'' to the location 122A', the forklift 104B can complete the task 112E with 4 minutes less of empty travel time along the empty travel route 110E''. This is because the location 122A' is closer to the location 122E of the task 112E. Thus,

pallet. Thus, the computer system can revert to the task assignment involving the original source pallet rather than the randomly selected alternative source pallet.

[0115] As yet another example, the randomly selected alternative source pallet can be located behind pallets in a storage rack. The original source pallet can be located in front of other pallets in a storage rack. However, the randomly selected alternative source pallet can be in a location that is closer to the current location of the operator and the original source pallet can be located farther away from the current location of the operator. Thus, by completing the task with the randomly selected alternative source pallet, the operator can spend less time completing the task, even if the operator must pull the other pallets out before accessing the alternative source pallet. Since the overall completion time of the task with the randomly selected alternative source pallet can be less than the overall completion time of the task with the original source pallet, the computer system can keep the modification and assign the task to the operator with the randomly selected alternative pallet.

[0116] The computer system can continuously repeat blocks 414-458 in the process 400. For example, in some implementations, whenever the computer system determines that a modification or swap should not be kept (e.g., refer to blocks 426, 442, and 458), the computer system can try swapping or modifying different tasks for the same or different operators. Since the computer system can continuously repeat blocks 414-458, swaps and/or modifications can be made relative to dynamic changes to a state of the warehouse. The computer system can therefore make accurate and effective swap and/or modification determinations as well as global efficiency metric assessments based on the current state of the warehouse, which can be advantageous to improve overall warehouse efficiency.

[0117] Still referring to the process 400 in FIGS. 4A-E, the computer system can determine whether the operator requested a task in 460 (e.g., refer to FIG. 5). The operator can have a user device, such as a laptop, computer, mobile device, smartphone, or other computing device. The operator can provide input to the user device indicating that the operator has completed a task and is ready to execute another, new task.

[0118] If the operator requested a task, then the computer system can send the operator a highest task in the operator's queue in 462. The highest task in the operator's queue can be an initially assigned task, a swapped task, and/or a task having a modified destination location

conveyor belt, inbound to a dock lane, reworking at a platform, reworking from a platform for a fix pallet, receiving on a dock lane, moving a pallet to a slat area, moving a pallet into a freezer unit, moving a pallet out of a freezer unit, de-slatting a pallet, re-ingesting a pallet, outbound staging, and outbound loading. The computer system can retrieve information about queues of automated dock tasks. In addition to the information previously described, the information for each of the automated dock tasks can further include a current location name of a pallet for each task (e.g., door, conveyor, dock lane, freezer unit, truck, etc.), a set of mechanical equipment that can be used to execute the task, a set of IT equipment that can be used to execute the task, and a set of target locations to which a pallet needs to be automatically moved to. The computer system can also retrieve information about all locations in the dock area, which can include all dock doors (e.g., trucks), dock lanes, conveyor positions on an inbound section of the conveyor belts (e.g., before T-car, conveyor loop, RGV loop, etc.), conveyor positions on an inbound section of the conveyor belts (e.g., after T-car, conveyor loop, RGV loop, etc.), floor locations identified for slatting, and locations in freezer units. Each of these locations can include additional data, such as location type, capacity (e.g., maximum number of pallets), set of pallets currently in the location, and data associated with each of the set of pallets. The data associated with each pallet can include a location, associated truck or order number, classification of an outbound pallet, and depth in the location (e.g., for multi-deep locations). Moreover, the computer system can retrieve configuration information associated with the warehouse, which can include all locations in the warehouse. This can include all conveyor locations and edges between neighboring automation locations. Length and capacity data associated with each edge can also be included in the configuration information. The location and capacity data can be used, for example, by the computer system to determine empty travel time between tasks and projected task completion times.

[0146] Similarly, the disclosed techniques can be used for selecting tasks in automated warehouses with conveyor lanes. Automated conveyor tasks can pertain to staging pallets on outbound conveyor belts or lanes. The automated conveyor tasks can include inbound to infeed conveyor belts, outbound loading of pallets from conveyor lanes into trucks, reworking pallets at a platform, reworking from platforms to fix pallets, inbound to dock lanes, inbound to freezer units, receiving pallets on a dock lane, receiving pallets from a freezer unit, moving pallets to a

implementations, the computer program product can be received in a propagated signal, for example, over the transceiver 668 or the external interface 662.

[0159] The mobile computing device 650 can communicate wirelessly through the communication interface 666, which can include digital signal processing circuitry where necessary. The communication interface 666 can provide for communications under various modes or protocols, such as GSM voice calls (Global System for Mobile communications), SMS (Short Message Service), EMS (Enhanced Messaging Service), or MMS messaging (Multimedia Messaging Service), CDMA (code division multiple access), TDMA (time division multiple access), PDC (Personal Digital Cellular), WCDMA (Wideband Code Division Multiple Access), CDMA2000, or GPRS (General Packet Radio Service), among others. Such communication can occur, for example, through the transceiver 668 using a radio-frequency. In addition, short-range communication can occur, such as using a Bluetooth, WiFi, or other such transceiver (not shown). In addition, a GPS (Global Positioning System) receiver module 670 can provide additional navigation- and location-related wireless data to the mobile computing device 650, which can be used as appropriate by applications running on the mobile computing device 650.

[0160] The mobile computing device 650 can also communicate audibly using an audio codec 660, which can receive spoken information from a user and convert it to usable digital information. The audio codec 660 can likewise generate audible sound for a user, such as through a speaker, e.g., in a handset of the mobile computing device 650. Such sound can include sound from voice telephone calls, can include recorded sound (e.g., voice messages, music files, etc.) and can also include sound generated by applications operating on the mobile computing device 650.

[0161] The mobile computing device 650 can be implemented in a number of different forms, as shown in the figure. For example, it can be implemented as a cellular telephone 680. It can also be implemented as part of a smart-phone 682, personal digital assistant, or other similar mobile device.

[0162] Various implementations of the systems and techniques described here can be realized in digital electronic circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These various implementations can include implementation in one or more computer