METHOD FOR PLANNING TRANSPORT ROUTES OF OVERHEAD HOIST VEHICLES BASED ON ELIMINATION OF CONFLICTS

[0001] The present application claims the benefit of priority to Chinese Patent Application No. 202210744605.6 filed on June 29, 2022, each of which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

[0002] The present disclosure relates to the field of wafer manufacturing and material transport technology, and in particular to a method for planning transport routes of overhead hoist vehicles based on elimination of conflicts.

BACKGROUND

[0003] In foundries for chips, wafers are processed at machines according to specific process flows, and the flow of wafers between machines is handled by an overhead hoist transport system (OHTS). Due to the fact that a track network for overhead hoist vehicles forms a sparse directed cyclic graph, and that only unidirectional travelling is allowed on tracks, there may be two conflicts: 1) a track is blocked by an overhead hoist vehicle due to failure of the overhead hoist vehicle or to loading or unloading of wafers, causing a conflict with a subsequent overhead hoist vehicle travelling on the same track; 2) two overhead hoist vehicles arrive at an intersection of tracks concurrently, in other words, a conflict occurs due to the two overhead hoist vehicles travelling on different tracks arrive at a same intersection concurrently. Therefore, if conflicts cannot be eliminated in time, waiting and congestion of overhead hoist vehicles will occur in some areas, thereby affecting the handling efficiency of the OHTS seriously.

[0004] In view of this, the present disclosure provides technical solutions for addressing the above technical problems.

SUMMARY

[0005] The present disclosure provides a method for planning transport routes of overhead hoist vehicles based on elimination of conflicts, including: S101: calculating a shortest distance between each two intersections of intersections in a track map, and establishing a routing table; S102: matching each respective overhead hoist vehicle with a respective issued handling task of issued

handling tasks, until each overhead hoist vehicle has a respective issued handling task or each issued handling task has a respective matched overhead hoist vehicle; S103: querying, based on data of the issued handling tasks, the routing table, and finding a shortest route to be travelled through of the respective overhead hoist vehicle for the respective issued handling task; S104: calculating a leaving duration of the respective overhead hoist vehicle starting from the respective overhead hoist vehicle travelling through a current intersection on the shortest route and ending at the respective overhead hoist vehicle leaving a next intersection, and sorting leaving durations in an ascending way; S105: detecting, according to sort of the leaving durations, whether a conflict occurs on the respective overhead hoist vehicle during travelling from the current intersection to the next intersection; in response to the conflict not occurring on the respective overhead hoist vehicle and a moment at which the respective overhead hoist vehicle leaving the next intersection being t1, setting a scheduling moment t for the respective overhead hoist vehicle to t1, and in response to the conflict occurring on the respective overhead hoist vehicle, performing operation S106; S106: determining whether the conflict occurs at the next intersection; in response to the conflict occurring at the next intersection: commanding an overhead hoist vehicle having a highest task priority to travel through the next intersection, and commanding one or more remaining overhead hoist vehicles having lower task priorities to wait at the next intersection; or commanding each overhead hoist vehicle of the one or more remaining overhead hoist vehicles having lower task priorities to switch to another route at a corresponding intersection previous to the next intersection; and in response to the conflict not occurring at the next intersection: determining that the conflict occurs in a loading process or an unloading process of another overhead hoist vehicle; commanding the respective overhead hoist vehicle to wait until the loading process or the unloading process of the another overhead hoist vehicle completes; or commanding the respective overhead hoist vehicle to switch to another route at an intersection previous to the next intersection; and S107: repeating operations S102 to S106, until a scheduling period ends or all issued handling tasks are completed. Commanding each overhead hoist vehicle of the one or more remaining overhead hoist vehicles having lower task priorities to switch to another route at a corresponding intersection previous to the next intersection, or commanding the respective overhead hoist vehicle to switch to another route at an intersection previous to the next intersection, includes: querying the routing table, acquiring another shortest route between the corresponding intersection previous to the next intersection or the intersection previous to the next intersection and a destination, and determining whether there is a respective conflict on the another shortest route; in response to there not being the respective conflict, commanding each overhead hoist vehicle of the one or more remaining overhead hoist vehicles having lower task priorities to switch to the another shortest route, or commanding the respective overhead hoist vehicle to switch to the

another shortest route; and in response to there being the respective conflict, eliminating the respective conflict using an ejecting-chain algorithm. The ejecting-chain algorithm includes ejecting a route of one respective overhead hoist vehicle of overhead hoist vehicles associated with the respective conflict, performing neighborhood action on routes of remaining overhead hoist vehicles of the overhead hoist vehicles associated with the respective conflict, and planning a new route for the one respective overhead hoist vehicle, in order to eliminate the respective conflict.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The preferred embodiments will be illustrated below in a clear and understandable manner, in conjunction with the accompanying drawings, in order to further illustrate the above features, technical features, advantages, and implementation methods.

[0007] FIG. 1 is a flow chart of some embodiments of the present disclosure.

[0008] FIG. 2 shows an example of a construction of a space-time state diagram and space-time routes used in the ejecting-chain algorithm according to some embodiments of the present disclosure.

[0009] FIG. 3 shows an example of initial routes used in ejecting operation of the ejecting-chain algorithm according to some embodiments of the present disclosure.

[0010] FIG. 4 shows an example of a conflict occurred in the initial routes used in the ejecting operation of the ejecting-chain algorithm according to some embodiments of the present disclosure.

[0011] FIG. 5 shows an example of ejecting a route in the ejecting operation of the ejecting-chain algorithm according to some embodiments of the present disclosure.

[0012] FIG. 6 shows an example of a trial operation in the ejecting operation of the ejecting-chain algorithm according to some embodiments of the present disclosure.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0013] In the following, the embodiments of the present disclosure will be illustrated in detail with reference to the accompanying drawings, in order to illustrate the embodiments of the present disclosure or the technical solutions in the prior art more clearly. Obviously, the illustrated drawings only show some the embodiments of the present disclosure. Those skilled in the art can obtain other drawings and other embodiments based on these drawings, without any inventive effort.

[0014] Based on the present disclosure, those skilled in the art should understand that one respective aspect illustrated herein can be implemented independently of any other aspect, and that two or more of the aspects can be combined in various ways. For example, the device and/or the method can be implemented using any number and aspect illustrated in the present disclosure. Moreover, the device and/or the method may be implemented using other structures and/or functionalities other than one or more aspects of the aspects illustrated in the present disclosure.

[0015] It should be further noted that the diagrams provided in the following embodiments only illustrate the basic concept of the present disclosure in a schematic manner. The diagrams only show the components related to the present disclosure and are not drawn according to the actual numbers, shapes, and sizes of the components in practice. The form, number, and scale of each component in practice may be arbitrarily changed, and the layout of the components may also be more complex.

[0016] Moreover, in the following description, specific details are provided to facilitate a thorough understanding of the examples. However, those skilled in the art shall understand that the aspects can be implemented without these specific details. The terms “first”, “second”, or the like are only for the purpose of illustration and shall not be understood as indicating or implying relative importance or implying the quantity of technical features indicated. Therefore, features limited by “first”, “second”, or the like may explicitly or implicitly include one or more features. In the illustration of the present disclosure, unless otherwise specified, “a plurality of” means two or more.

[0017] In foundries for chips, wafers are processed at machines according to specific process flows, and the flow of wafers between machines is handled by OHTS. Upon one respective machine completes the processing task, a material control system issues a handling instruction, and an automatic material handling system matches idle overhead hoist vehicles with the issued handling task(s) and schedules transport routes. Upon receiving the handling instruction, an overhead hoist vehicle goes to the one respective machine to load the wafer and transport the wafer to a next designated machine.

[0018] Upon receiving the handling instruction, an overhead hoist vehicle goes to a starting machine to load the wafer and travels to a destination machine to unload the wafer, thereby completing a handling task. Due to the fact that a track network for overhead hoist vehicles forms a sparse directed cyclic graph, and that only unidirectional travelling is allowed on tracks, there may be two conflicts: 1) a track is blocked by an overhead hoist vehicle due to failure of the overhead hoist vehicle or to loading or unloading of wafers, causing a conflict with a subsequent overhead hoist vehicle travelling on the same track; 2) two overhead hoist vehicles arrive at an intersection of

tracks concurrently, in other words, a conflict occurs due to the two overhead hoist vehicles travelling on different tracks arrive at a same intersection concurrently. Therefore, if conflicts cannot be eliminated in time, waiting and congestion of overhead hoist vehicles will occur in some areas, thereby affecting the handling efficiency of the OHTS seriously.

[0019] Embodiments of the present disclosure provide a method for planning transport routes of overhead hoist vehicles based on elimination of conflicts. The method mainly includes: a certain strategy is selected to match all overhead hoist vehicles with handling tasks issued in real time, then a shortest distance between each two intersections is calculated using an algorithm, i.e., route planning is performed for each pair of overhead hoist vehicle and task. At last, time is used as a baseline and deduction is performed forward, an earliest moment at which one respective overhead hoist vehicle of all overhead hoist vehicles arrives at a respective next node is regarded as a scheduling moment. It is detected whether a conflict occurs, and in response to the conflict occurring, the conflict is eliminated using different strategies, for example, an overhead hoist vehicle is commanded to wait, or an overhead hoist vehicle is commanded to switch to another route at a previous intersection, until a scheduling period ends or all issued handling tasks are completed. The earliest conflict is eliminated first, the conflicts include loading or unloading of wafers, conflicts occurred at intersections, failure of one or more overhead hoist vehicles, and the like.

[0020] The algorithms according to embodiments of the present disclosure implement two stages of scheduling methods including assignment of real-time tasks and planning of routes of overhead hoist vehicles. In this way, conflicts including loading or unloading of wafers, conflicts occurred at intersections, failure of one or more overhead hoist vehicles, and the like can be detected in real time and can be eliminated in time, thereby reducing time taken for handling, improving handling efficiency of the AMHS, and achieving effective and stable transport of materials.

[0021] In the following, the technical solutions provided by embodiments of the present disclosure are illustrated in conjunction with the accompanying drawings.

[0022] As shown in FIG. 1, some embodiments of the present disclosure include the following operations.

[0023] At S101, a shortest distance between each two intersections of intersections in a track map is calculated, and a routing table is established.

[0024] In some embodiments of the present disclosure, the shortest distance between each two intersections of intersections in the track map is calculated using one of A-star (A*) algorithm, Dijkstra algorithm, and shortest path faster algorithm (SPFA).

[0025] A* algorithm is the most effective direct search method for finding the shortest routes in a static route network, and it is also an effective algorithm for solving many search problems. The closer the estimated distance value in the algorithm is to the actual value, the greater the search speed.

[0026] Dijkstra algorithm is an algorithm for finding the shortest route(s) from one vertex to the remaining vertices, and is used to solve the shortest route problem in a weighted graph. The Dijkstra algorithm mainly includes traversing, starting from a starting vertex and using a greedy algorithm strategy, the nodes adjacent to the vertex that is closest to a previous vertex and has not been visited, in an iteration way, until a destination vertex is traversed.

[0027] SPFA is a type of algorithm for finding a single-source shortest path. It is a queue optimization of Bellman-ford algorithm and is a highly efficient algorithm for finding the shortest path.

[0028] In some embodiments, the shortest distance between each two intersections of the intersections in the track map may be calculated using for example the A* algorithm. The A* algorithm is used to quickly plan the transport routes of the overhead hoist vehicles. The heuristic function $h(n)$ of the evaluation function $f(n) = g(n) + h(n)$ represents an estimated value of a minimum cost of routes from a current node n to a target node. By calculating average duration taken for an overhead hoist vehicle to travel through a certain route, the cost of sides is regularly updated, so that the heuristic function can reflect the traffic situation in the past period of time, and guide the algorithm to avoid congested sections and find the route having the lowest cost.

[0029] The routing table stores one or more nodes between one respective intersection and other intersections. For example, each row of the routing table is represented by a triple <src, dest, next>, where src represents a current intersection, dest represents a destination intersection, and next represents a next node to be passed through from src to dest.

[0030] At S102, each respective overhead hoist vehicle is matched with a respective issued handling task of issued handling tasks, until each overhead hoist vehicle has a respective issued handling task or each issued handling task has a respective matched overhead hoist vehicle.

[0031] The operation S102 may include the following operations.

[0032] At S201, the data of the issued handling tasks is pre-processed, to obtain a task sequence and a set of handling tasks.

[0033] In some embodiments, pre-processing includes: setting each respective issuing moment for a respective issued handling task of the issued handling tasks to a number of seconds elapsed from the disposing moment 0 at which the respective issued handling task is issued to the

respective issuing moment, and sorting the issued handling tasks in an ascending way according to issuing moments, to obtain the task sequence.

[0034] At S202, a disposing moment t for the respective overhead hoist vehicle is set to 0, states of all overhead hoist vehicles are set to *idle*, and each scheduling moment for the respective overhead hoist vehicle at which the respective overhead hoist vehicle arrives at a corresponding starting node is set to 0, to obtain a set of overhead hoist vehicles.

[0035] At S203, the set of overhead hoist vehicles of which the respective overhead hoist vehicle is in *idle* state at a moment t is matched with the set of handling tasks using a bipartite graph, until each overhead hoist vehicle has a respective issued handling task or each issued handling task has a respective matched overhead hoist vehicle.

[0036] In the A* algorithm, the matching strategy for the overhead hoist vehicles and tasks includes matching, at a certain or random moment t, the set of overhead hoist vehicles of which the respective overhead hoist vehicle is in *idle* state with the set of issued handling tasks using a bipartite graph. The all overhead hoist vehicles being in the *idle* state at the moment t are sorted in a descending way based on each respective duration for which the respective overhead hoist vehicle has been in the *idle* state. Then, a distance between the respective overhead hoist vehicle and each respective handling task of the issued handling tasks is calculated, the respective overhead hoist vehicle of the all overhead hoist vehicles is matched with a handling task nearest to the respective overhead hoist vehicle, and one or more overhead hoist vehicles have been matched are removed from the set of overhead hoist vehicles and one or more matched handling tasks are removed from the set of handling tasks, until each overhead hoist vehicle has a respective issued handling task or each issued handling task has a respective matched overhead hoist vehicle.

[0037] In some embodiments, matching the set of overhead hoist vehicles of which the respective overhead hoist vehicle is in the *idle* state at the moment t with the set of handling tasks using the bipartite graph, includes the following two situations.

[0038] When a number of overhead hoist vehicles of which states are *idle* at the moment t is greater than a number of handling tasks of the set of handling tasks, some idle overhead hoist vehicles are left, but idle overhead hoist vehicles shall not wait on tracks, in order to prevent congestions/conflicts. In view of this, a respective virtual task is generated for each overhead hoist vehicle of overhead hoist vehicles that are not matched with any handling task, and the overhead hoist vehicles that are not matched with any handling task are commanded to operate at on-call stations (cycling areas) on one or more Intrabay loops (secondary travelling tracks), in order to

prevent increased congestion possibility due to these overhead hoist vehicles operating on one or more Interbay loops (main travelling tracks). A starting node and a destination of one respective virtual task are selected as two positions on the one or more Interbay loops, respectively, and a state of the respective overhead hoist vehicle is set to *orderrec* (operation without loads).

[0039] When a number of overhead hoist vehicles of which states are *idle* at the moment t is less than a number of handling tasks of the set of handling tasks, each overhead hoist vehicle is matched with a respective handling task of the set of handling tasks, and handling tasks that are not matched with any overhead hoist vehicle are configured to wait at corresponding machines until a state of one or more overhead hoist vehicles turns to *idle*.

[0040] At S103, the routing table is queried based on the data of the issued handling tasks, and a shortest route to be travelled through of the respective overhead hoist vehicle for the respective issued handling task is acquired.

[0041] At S104, a leaving duration of the respective overhead hoist vehicle starting from the respective overhead hoist vehicle travelling through a current intersection on the shortest route and ending at the respective overhead hoist vehicle leaving a next intersection is calculated, leaving durations are sorted in an ascending way, and a state of the respective overhead hoist vehicle is updated, where the state of the respective overhead hoist vehicle includes at least one of *delivery*, *pickup*, or *unload*.

[0042] For example, when the respective overhead hoist vehicle is loaded at one respective loading/unloading node between the current intersection and the next intersection, the leaving duration is a sum of a duration for arriving at the next intersection and a loading duration. When the respective overhead hoist vehicle unloads at the respective loading/unloading node, the leaving duration is a sum of the duration for arriving at the next intersection and an unloading duration. When the respective overhead hoist vehicle only passes through the respective loading/unloading node, the leaving duration is equal to the duration for arriving at the next intersection.

[0043] At S105, it is detected, according to sort of the leaving durations, whether a conflict occurs during the respective overhead hoist vehicle travelling from the current intersection to the next intersection.

[0044] In response to the conflict not occurring, it is determined that a moment at which the respective overhead hoist vehicle leaves the next intersection is t1, and a scheduling moment t for the respective overhead hoist vehicle is set to t1. When the respective overhead hoist vehicle arrives at

an unloading node, the state of the respective overhead hoist vehicle is set to *idle* after unloading, and operation S203 is performed again to assign a new task to the respective overhead hoist vehicle.

[0045] In response to the conflict occurring, operation S106 is performed.

[0046] In the embodiments of the present disclosure, due to the fact that in plants for manufacturing semiconductor wafers, overhead hoist vehicles travel in a unidirectional way and in a designated direction on tracks, when a distance between two overhead hoist vehicles is less than a preset threshold, it is determined that a conflict and a blockage occurs.

[0047] At S106, it is determined whether the conflict occurs at the next intersection.

[0048] In response to the conflict occurring at the next intersection, an overhead hoist vehicle having a highest task priority is commanded to travel through the next intersection, one or more remaining overhead hoist vehicles having lower task priorities are commanded to wait at the next intersection, and a state of the one or more remaining overhead hoist vehicles that have lower task priorities and wait at the next intersection is updated into *wait*; or

[0049] each overhead hoist vehicle of the one or more remaining overhead hoist vehicles having lower task priorities is commanded to switch to another route at a corresponding intersection previous to the next intersection.

[0050] In response to the conflict not occurring at the next intersection, it is determined that the conflict occurs in a loading process or an unloading process of another overhead hoist vehicle, and the respective overhead hoist vehicle is commanded to wait. The leaving duration of the respective overhead hoist vehicle is a sum of a duration for arriving at the next intersection and a waiting duration, and a state of the respective overhead hoist vehicle that is blocked and waits in situ is updated into *wait*, until the loading process or the unloading process of the another overhead hoist vehicle completes; or

[0051] the respective overhead hoist vehicle is commanded to switch to another route at an intersection previous to the next intersection.

[0052] In some embodiments of the present disclosure, due to the fact that there are intersections between Interbay loops and Intrabay loops, a conflict occurs when overhead hoist vehicles arrive at a same intersection concurrently, the respective overhead hoist vehicle is in a loading/unloading process, or the respective overhead hoist vehicle malfunctions. When a conflict occurs, it is eliminated using the following strategies: (1) commanding one or more overhead hoist vehicles to wait in situ; (2) commanding one or more overhead hoist vehicles to switch to another route.

[0053] Thus, it is determined first whether the conflict occurs at the next intersection. If yes, due to the fact that a duration for an overhead hoist vehicle to travel through the intersection is usually short, the overhead hoist vehicle having the highest task priority may be commanded to travel through the next intersection, and the one or more overhead hoist vehicles that have lower task priorities may be commanded to wait in situ or to switch to another route at an intersection previous to the next intersection. If no, it is determined that the conflict occurs at one respective loading/unloading node between the current intersection and the next intersection for loading/unloading of another overhead hoist vehicle. When there is another overhead hoist vehicle in the loading/unloading process at the respective loading/unloading node or has failure, priority may be given to switching to another route depending on the waiting duration. That is to say, an overhead hoist vehicle which is going to travel through a previous intersection to the section having a conflict may be commanded to switch to another route at the previous intersection, and the one or more overhead hoist vehicle that have been blocked may be commanded to wait in situ.

[0054] In some embodiments, the task priorities may be set according to actual situations. For example, the task priorities may be set according to any one of a descending sort of delay durations for current routes of overhead hoist vehicles, a descending sort of priorities of wafer tasks, and a descending sort of waiting durations of overhead hoist vehicles due to conflict(s).

[0055] In some embodiments, when one respective overhead hoist vehicle is commanded to switch to another route at a corresponding intersection previous to the next intersection, another optimal route starting at the intersection previous to the next intersection may be selected.

[0056] In some embodiments, selecting another optimal route may include: querying the routing table, acquiring another shortest route between the corresponding intersection previous to the next intersection or the intersection previous to the next intersection and the destination, and determining whether there is a respective conflict on the another shortest route.

[0057] If no, the one respective overhead hoist vehicle is commanded to switch to the another shortest route.

[0058] It yes, the respective conflict on the another shortest route is eliminated using an ejecting-chain algorithm.

[0059] In some embodiments, the ejecting-chain algorithm includes ejecting one or more routes on which the respective conflict does not occur, performing neighborhood action on a route on which the respective conflict occurs, and planning one or more new routes in place of the one or more ejected routes, in order to eliminate the respective conflict occurred at a corresponding node.

[0060] In some embodiments, in order to further control congestion and disperse traffic of overhead hoist vehicles on the tracks, the spatial nodes on the tracks are distributed along the time axis to obtain a space-time state diagram. Then, a meta-heuristic algorithm of the ejecting-chain algorithm is used on the space-time state diagram for dispersing traffic.

[0061] The space-time state diagram is based on the spatial nodes in the classic vehicle routing problem, and the spatial nodes are distributed into a series of space-time nodes along the time axis. The conflict detection of the overhead hoist vehicles is modeled in a form of a directed graph, a construction of which is shown in FIG. 2. All tracks are discretized into a series of nodes to ensure that each time step an overhead hoist vehicle can move from a current node to an adjacent node. Each node v_i is expanded into a series of space-time nodes $\{v_{i0}, v_{i1}, \ldots, v_{iT}\}$ according to the number of time steps included in the entire planning cycle, and each side $v_i \rightarrow v_j$ in original topological graph is accordingly expanded into $\{v_{i0} \rightarrow v_{j1}, v_{i1} \rightarrow v_{j2}, \ldots, v_{i(T-1)} \rightarrow v_{jT}\}$. A respective side from v_{it} to $v_{i(t+1)}$ is added to represent waiting in situ, and the spatial routes are also accordingly converted into space-time routes. In this way, the conflict constraint of the overhead hoist vehicles is converted into a degree constraint of nodes, that is, the in-degrees and out-degrees of each space-time node on a respective space-time route do not exceed 1, respectively.

[0062] In some embodiments, the ejecting-chain algorithm includes ejecting one or more routes having at least one alternative route, performing neighborhood action on a route on which the respective conflict occurs, and planning one or more new routes in place of the one or more ejected routes, in order to eliminate the respective conflict. The ejecting-chain algorithm is an efficient meta-heuristic algorithm and is highly suitable for solving resource allocation problems with strong capacity constraints. As shown in FIGS. 3 to 6, the basic idea is to select an object to release its occupied resources first, form a partial solution with more relaxed resource constraints, and then perform neighborhood searches on the partial solution to obtain a reference structure. Based on an optimal reference structure, neighborhood searches action may be continued to obtain a new reference structure, or trial operations may be performed to obtain a complete solution again and end the current round of ejection-chain search.

[0063] The specific operations of the ejecting-chain algorithm are illustrated below referring to the algorithm example as shown in FIGS. 3 to 6.

[0064] FIG. 3 shows information of the algorithm example, including 4 nodes, 4 overhead hoist vehicles, and 4 transport routes for the overhead hoist vehicles.

[0065] FIG. 4 shows an initial solution having a conflict, i.e. at moment T1, a conflict between A→B and B→C occurs at node C.

[0066] FIG. 5 is a schematic diagram of an ejecting operation. The ejecting operation is performed to remove the route of D→C. A neighborhood search action is performed to reroute B→C, in order to eliminate the conflict and obtain a reference structure.

[0067] FIG. 6 is a schematic diagram of a trial operation. The trial operation is performed to recalculate a shortest route having fewest conflicts for the removed route of D→C, in order to obtain a trial solution. Then, one or more further ejecting operations may be performed, or neighborhood searches may be continued from (b). For example, a trial solution obtained by ejecting and then putting back the route of B→C can eliminate all conflicts.

[0068] In the example of implementing process of the ejecting-chain algorithm as shown in FIGS. 3 to 6, based on the initial relaxation solution by which conflict of overhead hoist vehicles is not completely eliminated, by ejecting the route from node D to node C, the remaining routes can be optimized. The neighborhood action is performed by recalculating each route of overhead hoist vehicles to find one or more shortest routes having the fewest conflicts, thereby reducing the number of routes having conflict(s) (i.e. two overhead hoist vehicles arrive at a same node at the same time), and obtaining the reference structure. Next, neighborhood searches may be continued, but since the optimal reference structure for the present round has no conflict any more, there is no need to further perform neighborhood search actions. Therefore, the trial action is performed, i.e. putting the ejected route back. Similarly, the next round of ejecting-chain operation is continued, and all conflicts can be eliminated and feasible solution(s) can be obtained.

[0069] At S107, operations S102 to S106 are repeated, until a scheduling period ends or all issued handling tasks are completed.

[0070] In the embodiments of the present disclosure, before the scheduling period ends or all the issued handling tasks are completed, states of overhead hoist vehicles may include: (1) *idle*, having no task; (2) *orderrec*, going to, without loads, a destination for loading as commanded; (3) *pickup*, loading materials; (4) *delivery*, operating with loads; (5) *unload*, unloading materials; (6) *wait*, waiting; and (7) *failure*, malfunctioning.

[0071] In some embodiments, in operation S106, in response to determining that the conflict occurs in the loading process or the unloading process of the another overhead hoist vehicle, a loading duration or an unloading duration of the another overhead hoist vehicle is determined.

[0072] In response to the loading duration or the unloading duration being less than or equal to a preset duration, the blocked respective overhead hoist vehicle is commanded to wait until the loading process or the unloading process of the another overhead hoist vehicle completes.

[0073] In response to the loading duration or the unloading duration being greater than the preset duration, the blocked respective overhead hoist vehicle is commanded to switch to another route at an intersection previous to the next intersection.

[0074] In response to determining that the conflict occurs in the loading process or the unloading process of the another overhead hoist vehicle, a loading duration or an unloading duration of the another overhead hoist vehicle is determined, in order to determine whether the another overhead hoist vehicle is in a loading/unloading process, or malfunctions. In some embodiments, determination is made by comparing the loading/unloading duration with a threshold. In response to the loading/unloading duration being greater than the threshold, it is determined that the another overhead hoist vehicle malfunctions, and the blocked respective overhead hoist vehicle is commanded to switch to another route at an intersection previous to the next intersection. In response to the loading/unloading duration being less than or equal to the threshold, it is determined that the another overhead hoist vehicle is in a loading/unloading process, and the blocked respective overhead hoist vehicle is commanded to wait until the loading process or the unloading process of the another overhead hoist vehicle completes.

[0075] In some embodiments, before operation S101, the track map is pre-processed.

[0076] An original track map includes intersections, barcodes, positions of machines, positions of stockers (automatic repositories or solid storages or storage facilities for storing wafers), and the like. The pre-processing refers to removing elements other than the intersections in the original track map to construct a new and simple graph. Then the shortest distance between each two intersections is calculated using a shortest-path algorithm. Thus, the pre-processing is intended to reduce the scale of the track map, thereby reducing calculating time of the algorithm.

[0077] In the algorithms for AMHS according to the embodiments of the present disclosure, in order to simplify the models, all overhead hoist vehicles travel at a same and constant speed. When a blockage occurs, waiting in situ of one respective overhead hoist vehicle at a certain node does not take braking, acceleration and deceleration during steering, or the like into account. Moreover, in order to simulate actual operating scenarios, the input data of the algorithms for AMHS is all the actual handling tasks executed in a certain wafer manufacturing plant over a period of time (such as a day). The map adopts a map of a real plant. At the beginning of the algorithm, the map is pre-

processed, the A* algorithm is used to find a respective shortest distance between each two intersections, and one or more nodes or intersections to be passed through from each intersection to any other intersection are recorded. After processing, during execution of the algorithm, a shortest distance between any two nodes can be found by searching for the shortest route between two intersections that are closest to the two nodes, respectively. In other words, the shortest distance can be found by querying the table.

[0078] In the embodiments of the present disclosure, the input of the algorithms includes:

[0079] a track map of a wafer manufacturing plant, including nodes, sides, properties of the nodes, maintenance positions, and the like;

[0080] original data of tasks over a period of time, including series numbers of tasks, starting nodes, destinations, priorities of tasks; and

[0081] data of overhead hoist vehicles, including initial positions of overhead hoist vehicles, travelling speeds of overhead hoist vehicles, historical mileage, priorities of overhead hoist vehicles, cycling rules of overhead hoist vehicles, on-call stations.

[0082] The output of the algorithms includes: one or more routes for each task, assigned overhead hoist vehicles, and all nodes to be travelled through and moments at which one respective overhead hoist vehicle travels through these nodes.

[0083] With the method for planning transport routes of overhead hoist vehicles based on elimination of conflicts according to the embodiments of the present disclosure, wafers can be quickly and accurately handled to their destinations, and cycle time of wafers between machines can be reduced, thereby improving system throughput, and achieving efficient and stable transport of materials.

[0084] The above are only the specific embodiments of the present disclosure, but the scope of protection of the present disclosure is not limited to this. Any changes or replacements that are easily conceivable to those skilled in the art within the scope of disclosure of the present disclosure shall fall within the scope of protection of the present disclosure. Therefore, the protection scope of the present disclosure should be determined by the protection scope of the claims.

CLAIMS

1. A method for planning transport routes of overhead hoist vehicles based on elimination of conflicts, comprising:

S101: calculating a shortest distance between each two intersections of intersections in a track map, and establishing a routing table;

S102: matching each respective overhead hoist vehicle with a respective issued handling task of issued handling tasks, until each overhead hoist vehicle has a respective issued handling task or each issued handling task has a respective matched overhead hoist vehicle;

S103: querying, based on data of the issued handling tasks, the routing table, and finding a shortest route to be travelled through of the respective overhead hoist vehicle for the respective issued handling task;

S104: calculating a leaving duration of the respective overhead hoist vehicle starting from the respective overhead hoist vehicle travelling through a current intersection on the shortest route and ending at the respective overhead hoist vehicle leaving a next intersection, and sorting leaving durations in an ascending way;

S105: detecting, according to sort of the leaving durations, whether a conflict occurs during the respective overhead hoist vehicle travelling from the current intersection to the next intersection;

in response to the conflict not occurring, determining that a moment at which the respective overhead hoist vehicle leaves the next intersection is t1, and setting a scheduling moment t for the respective overhead hoist vehicle to t1, and in response to the conflict occurring, performing operation S106;

S106: determining whether the conflict occurs at the next intersection;

in response to the conflict occurring at the next intersection:

commanding an overhead hoist vehicle having a highest task priority to travel through the next intersection, and commanding one or more remaining overhead hoist vehicles having lower task priorities to wait at the next intersection; or

commanding each overhead hoist vehicle of the one or more remaining overhead hoist vehicles having lower task priorities to switch to another route at a corresponding intersection previous to the next intersection; and

in response to the conflict not occurring at the next intersection:

determining that the conflict occurs in a loading process or an unloading process of another overhead hoist vehicle;

commanding the respective overhead hoist vehicle to wait until the loading process or the unloading process of the another overhead hoist vehicle completes; or

commanding the respective overhead hoist vehicle to switch to another route at an intersection previous to the next intersection; and

S107: repeating operations S102 to S106, until a scheduling period ends or all issued handling tasks are completed;

wherein commanding each overhead hoist vehicle of the one or more remaining overhead hoist vehicles having lower task priorities to switch to another route at a corresponding intersection previous to the next intersection, or commanding the respective overhead hoist vehicle to switch to another route at an intersection previous to the next intersection, includes:

querying the routing table, acquiring another shortest route between the corresponding intersection previous to the next intersection or the intersection previous to the next intersection and a destination, and determining whether there is a respective conflict on the another shortest route;

in response to there not being the respective conflict, commanding each overhead hoist vehicle of the one or more remaining overhead hoist vehicles having lower task priorities to switch to the another shortest route, or commanding the respective overhead hoist vehicle to switch to the another shortest route; and

in response to there being the respective conflict, eliminating the respective conflict using an ejecting-chain algorithm;

wherein the ejecting-chain algorithm includes ejecting a route of one respective overhead hoist vehicle of overhead hoist vehicles associated with the respective conflict, performing neighborhood action on routes of remaining overhead hoist vehicles of the overhead hoist vehicles associated with the respective conflict, and planning a new route for the one respective overhead hoist vehicle, in order to eliminate the respective conflict.

2. The method according to claim 1, wherein operation S102 includes:

S201: pre-processing the data of the issued handling tasks, to obtain a task sequence and a set of handling tasks;

S202: setting states of all overhead hoist vehicles to *idle* and each disposing moment for the respective overhead hoist vehicle at which the respective overhead hoist vehicle arrives at a corresponding starting node to 0, to obtain a set of overhead hoist vehicles;

S203: matching the set of overhead hoist vehicles of which the respective overhead hoist vehicle is in *idle* state at a moment t with the set of handling tasks using a bipartite graph, until each overhead

hoist vehicle has a respective issued handling task or each issued handling task has a respective matched overhead hoist vehicle.

3. The method according to claim 2, wherein in operation S201, pre-processing the data of the issued handling tasks, includes:

setting each respective issuing moment for a respective issued handling task of the issued handling tasks to a number of seconds elapsed from the disposing moment 0 at which the respective issued handling task is issued to the respective issuing moment, and sorting the issued handling tasks in an ascending way according to issuing moments, to obtain the task sequence.

4. The method according to claim 2, wherein in operation S203, matching the set of overhead hoist vehicles of which the respective overhead hoist vehicle is in the *idle* state at the moment t with the set of handling tasks using the bipartite graph, includes:

sorting the all overhead hoist vehicles being in the *idle* state at the moment t in a descending way based on each respective duration for which the respective overhead hoist vehicle has been in the *idle* state;

matching the respective overhead hoist vehicle of the all overhead hoist vehicles with a handling task nearest to the respective overhead hoist vehicle; and

removing one or more overhead hoist vehicles have been matched from the set of overhead hoist vehicles and removing one or more handling tasks have been matched from the set of handling tasks, until each overhead hoist vehicle has a respective issued handling task or each issued handling task has a respective matched overhead hoist vehicle.

5. The method according to claim 2 or 4, wherein in operation S203, matching the set of overhead hoist vehicles of which the respective overhead hoist vehicle is in the *idle* state at the moment t with the set of handling tasks using the bipartite graph, includes:

in response to a number of overhead hoist vehicles of which states are *idle* at the moment t being greater than a number of handling tasks of the set of handling tasks, generating a respective virtual task for each overhead hoist vehicle of overhead hoist vehicles that are not matched with any handling task, and commanding the overhead hoist vehicles that are not matched with any handling task to operate at on-call stations on one or more Intrabay loops.

6. The method according to claim 2 or 4, wherein in operation S203, matching the set of overhead hoist vehicles of which the respective overhead hoist vehicle is in the *idle* state at the moment t with the set of handling tasks using the bipartite graph, includes:

in response to a number of overhead hoist vehicles of which states are *idle* at the moment t being less than a number of handling tasks of the set of handling tasks, each overhead hoist vehicle is matched with a respective handling task of the set of handling tasks, and handling tasks that are not matched with any overhead hoist vehicle are configured to wait at corresponding machines until a state of one or more overhead hoist vehicles turns to *idle*.

7. The method according to claim 1, wherein in operation S106, in response to determining that the conflict occurs in the loading process or the unloading process of the another overhead hoist vehicle, determining a loading duration or an unloading duration of the another overhead hoist vehicle;

in response to the loading duration or the unloading duration being less than or equal to a preset duration, commanding the respective overhead hoist vehicle to wait until the loading process or the unloading process of the another overhead hoist vehicle completes; and

in response to the loading duration or the unloading duration being greater than the preset duration, commanding the respective overhead hoist vehicle to switch to another route at an intersection previous to the next intersection.

8. The method according to claim 1, wherein operation S104 further includes: updating a state of the respective overhead hoist vehicle, wherein the state of the respective overhead hoist vehicle includes at least one of *delivery*, *pickup*, or *unload*.

9. The method according to claim 1, wherein in operation S106:

in response to the conflict occurring at the next intersection, commanding the overhead hoist vehicle having the highest task priority to travel through the next intersection, and updating a state of the one or more remaining overhead hoist vehicles that have lower task priorities and wait at the next intersection into *wait*; and

in response to determining that the conflict occurs in the loading process or the unloading process of the another overhead hoist vehicle, updating a state of the respective overhead hoist vehicle into *wait* until the loading process or the unloading process of the another overhead hoist vehicle completes.

10. The method according to claim 1, wherein in operation S101, the shortest distance between each two intersections of intersections in the track map is calculated using one of A-star (A*) algorithm, Dijkstra algorithm, and shortest path faster algorithm (SPFA).

11. The method according to claim 10, wherein in operation S101, the shortest distance between each two intersections of intersections in the track map is calculated using the A* algorithm.

12. The method according to claim 1, wherein before operation S101, the method further includes: pre-processing the track map.