

基于冲突消解的天车运输路径规划方法

本申请基于申请号为“202210744605.6”、申请日为 2022 年 06 月 29 日的中国专利申请提出，并要求该中国专利申请的优先权，该中国专利申请的全部内容在此以引入方式并入本申请。

技术领域

本申请属于晶圆制造及物料运输技术领域，更确切的说涉及一种基于冲突消解的天车运输路径规划方法。

背景技术

在芯片代工厂中，晶圆按照特定的工艺流程在各个机台上进行加工作业，晶圆在不同机台之间的流转通过天车系统负责搬运。由于天车的轨道网络是一个稀疏的有向有环图，且轨道上只允许单向行驶，因此会存在两种冲突情形：1) 前车因故障或装卸晶圆而阻塞轨道，与同一轨道上的后车造成冲突。2) 两车在轨道的汇合口相遇，即两车由不同的轨道驶入同一汇合口造成冲突。因此，若不能对冲突情形进行及时消解，就会造成部分区域车辆出现等待、拥塞的现象，严重影响 AMHS 系统的搬运效率。

基于此，本申请提供了解决以上技术问题的技术方案。

发明概述

本申请提供一种基于冲突消解的天车运输路径规划方法，包括如下步骤：S101、计算轨道地图中任意两个交叉点之间的最短距离，建立路由表；S102、将小车和已下达的搬运任务进行最优匹配，至所有小车或搬运任务全部被匹配；S103、根据已下达的搬运任务数据查询所述路由表，获得每个小车送达搬运任务需要经过的最短路径；S104、计算每辆小车在所述最短路径上从当前交叉点到离开下一交叉点的离开时间，并将所述离开时间按从小到大的顺序进行排序；S105、按所述离开时间的排序，依次检测小车从当前交叉点到下一交叉点是否遇到冲突，若否，则确定该小车离开下一交叉点的时间为 t_1 ，并设调度时刻 $t=t_1$ ，

若是，则进行 S106；S106、判断冲突是否发生在交叉点，若是，则使任务优先级高的小车优先通过，任务优先级低的小车在该交叉点等待，或，使任务优先级低的小车在上一交叉点选择另一条分支，更换路径；若否，则认定该冲突发生在小车装/卸货过程中，则使被阻塞的小车原地等待，至另一辆小车完成装卸货过程，或，使被阻塞的小车在上一交叉点选择另一条分支，更换路径；S107、反复执行所述 S102-S106，直到调度周期结束或所有搬运任务均执行完成；其中，当小车在上一交叉点更换路径时，首先查询所述路由表，获取该小车从上一交叉点至目标点的最短路径，再判断该最短路径中是否发生冲突；若否，执行该最短路径；若是，则将该最短路径中的冲突采用弹射链算法进行消解；所述弹射链算法是将产生冲突的小车路径执行弹出动作，剩余小车执行邻域动作，再对弹出的小车重新规划路径，以消解冲突。

附图说明

下面将以明确易懂的方式，结合附图说明优选实施方式，对上述特性、技术特征、优点及其实现方式予以进一步说明。

图 1 是本申请实施例的流程示意图；

图 2 是本申请实施例的弹射链算法中时空状态图的构造方式与时空路径示例；

图 3 是本申请实施例的弹射链算法中弹射链操作的初始路径示例；

图 4 是本申请实施例的弹射链算法中弹射链操作的初始存在冲突示例；

图 5 是本申请实施例的弹射链算法中弹射链操作的路径弹出示例；

图 6 是本申请实施例的弹射链算法中弹射链操作的试探动作示例。

本发明的实施方式

为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对照附图说明本申请的具体实施方式。显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图，并获得其他的实施方式。

基于本申请，所属领域的技术人员应了解，本文中所描述的一个方面可与

任何其它方面独立地实施，且可以各种方式组合这些方面中的两者或两者以上。举例来说，可使用本文中所阐述的任何数目和方面来实施设备及/或实践方法。另外，可使用除了本文中所阐述的方面中的一或多者之外的其它结构及/或功能性实施此设备及/或实践此方法。

还需要说明的是，以下实施例中所提供的图示仅以示意方式说明本申请的基本构想，图式中仅显示与本申请中有关的组件而非按照实际实施时的组件数目、形状及尺寸绘制，其实际实施时各组件的型态、数量及比例可为一种随意的改变，且其组件布局型态也可能更为复杂。

另外，在以下描述中，提供具体细节是为了便于透彻理解实例。然而，所属领域的技术人员将理解，可在没有这些特定细节的情况下实践所述方面。术语“第一”、“第二”等仅用于描述目的，而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此，限定有“第一”、“第二”等描述的特征可以明示或者隐含地包括一个或者更多个该特征。在本申请的描述中，除非另有说明，“多个”的含义是两个或两个以上。

在芯片代工厂中，晶圆按照特定的工艺流程在各个机台上进行加工作业，晶圆在不同机台之间的流转通过天车系统负责搬运。当机台完成加工任务后，系统下达搬运指令，AMHS（Automatic Material Handling System 自动物料搬运系统）对空闲天车和已下达的搬运任务进行匹配并规划运输路径，天车接到指令后，前往该机台装载晶圆，并将晶圆运输至下一个指定的机台。

当天车接到搬运指令后，前往晶圆起始机台装载晶圆，并送至目的机台卸载晶圆，完成一次运输任务。由于天车的轨道网络是一个稀疏的有向有环图，且轨道上只允许单向行驶，因此会存在两种冲突情形：1）前车因故障或装卸晶圆而阻塞轨道，与同一轨道上的后车造成冲突。2）两车在轨道的汇合口相遇，即两车由不同的轨道驶入同一汇合口造成冲突。因此，若不能对冲突情形进行及时消解，就会造成部分区域车辆出现等待、拥塞的现象，严重影响 AMHS 系统的搬运效率。

本申请实施例提供了一种基于冲突消解的天车运输路径规划方法，该方法主要手段是针对所有小车，根据实时下达的搬运任务，选择一定的策略进行匹

配指派。然后采用算法计算小车在任意两个交叉点之间的最短距离，即为每一对小车和任务进行路径规划。最后，以时间为准线不断向前推进，所有小车行进到下一个节点的最早时刻就是调度时刻，检测是否遇到冲突，遇到冲突就按照不同的策略解决冲突，例如其中一辆车等待，或者其中一辆车在上一个交叉口更换路径，直到调度周期结束或所有任务均完成。每次处理最先发生的冲突，包括装卸货、交叉路口冲突、小车故障等。

本申请实施例中的 AMHS 算法实现了实时任务指派和小车路径规划两个阶段的调度方法。能够实时检测冲突情况，并对冲突情形进行及时消解，包括装卸货、交叉路口冲突、小车故障等，减少搬运时间，提高 AMHS 系统的搬运效率，实现高效稳定的物料传输。

以下结合各附图，说明本申请各实施例提供的技术方案。

如图 1 所示，本申请第一实施例包括以下步骤：

S101、计算轨道地图中任意两个交叉点之间的最短距离，建立路由表。

本申请实施例中，采用包括 A*算法、Dijkstra 算法、SPFA 算法中的任一算法计算轨道地图中任意两个交叉点之间的最短距离。

A*算法，A* (A-Star)算法是一种静态路网中求解最短路径最有效的直接搜索方法，也是解决许多搜索问题的有效算法。算法中的距离估算值与实际值越接近，最终搜索速度越快。

Dijkstra 算法是从一个顶点到其余各顶点的最短路径算法，解决的是有权图中最短路径问题。Dijkstra 算法主要特点是从起始点开始，采用贪心算法的策略，每次遍历到始点距离最近且未访问过的顶点的邻接节点，直到扩展到终点为止。

SPFA(Shortest Path Faster Algorithm)算法是求单源最短路径的一种算法，它是 Bellman-ford 的队列优化，它是一种十分高效的最短路算法。

在一些实施中，可采用例如 A*算法计算获取轨道地图中任意两个交叉点之间的最短距离。利用 A*算法快速规划天车运输路径。其中估价函数 $f(n) = g(n) + h(n)$ 中的启发函数 $h(n)$ 表示当前节点 n 到目标节点的最小代价路径估计值，统计天车经过某一段路径的平均时间，定期更新边的代价，使启发

式函数能够反映过去一段时间内的交通状况，引导算法避开拥塞路段寻找代价最小的路径。

所述路由表中保存每一个交叉点去其他交叉点下一步要经过的点。例如，路由表的每一行由一个三元组 $\langle \text{src}, \text{dest}, \text{next} \rangle$ 表示，其中 src 是当前交叉点， dest 是目的地交叉点， next 是路过 src 之后，前往 dest 要经过的下一个点。

S102、将小车和已下达的搬送任务进行最优匹配，至所有小车或搬送任务全部被匹配；

所述 S102 的过程可以包括：

S201、对已下达的搬送任务数据进行预处理，形成任务序列，得到搬送任务集合；

在一些实施中，所述预处理是指：将搬送任务的下达时间改为从 0 时刻开始经过的秒数，并将搬送任务按下达时间从小到大排序，形成所述任务序列；

S202、设调度时刻 $t=0$ ，将所有小车状态设为 idle ，小车到达初始点的时刻设为 0，得到小车集合；

S203、将 t 时刻状态为 idle 的所述小车集合和所述搬送任务集合进行二分图匹配，至所有小车或搬送任务全部被匹配；

A*算法中，小车和任务的指派策略为在某一调度时刻 t ，对处于空闲状态的小车集合和已经下达的搬送任务集合进行二分图匹配：首先，将 t 时刻状态为 idle 的小车按照已空闲时间由高到低进行排序，其次，计算小车跟已下达的搬送任务之间的距离，对每一辆小车，依次选择距离最近的搬送任务进行匹配，并将已匹配的小车和搬送任务从各自集合中删除，反复执行，至所有小车或搬送任务全部被匹配。

在一些实施中，在将 t 时刻状态为 idle 的小车集合和搬送任务集合进行二分图匹配的过程中分两种情况：

若 t 时刻状态为 idle 的小车的数量大于搬送任务的数量，则有多余小车空闲，由于空闲小车不能在轨道上等待，以免造成阻塞/冲突，因此，对未匹配到搬送任务的小车生成一个虚拟任务并执行，使该小车在其 on call station （轮询区域）的 Intrabay （副航道）中空载运行，避免在 Interbay （主航道）上运行而

增加堵塞的几率。该虚拟任务的起点和终点分别所选择的 Interbay 的两个端点，此时小车的状态改为 orderrec（空载运行）。

若 t 时刻状态为 idle 的小车的数量小于搬运任务的数量，则小车全部被分配到搬运任务，剩下未被分配的搬运任务在机台上继续等待，直到有小车状态更新为 idle 时再分配。

S103、根据已下达的搬运任务数据查询所述路由表，获得每个小车送达搬运任务需要经过的最短路径；

S104、计算每辆小车在所述最短路径上从当前交叉点到离开下一交叉点的离开时间，并将所述离开时间按从小到大的顺序进行排序，同时将所有小车状态更新为 delivery，或 pickup，或 unload；

如果小车在该点装货，则离开时间=到达时间+装货时间；如果小车在该点卸货，则离开时间=到达时间+卸货时间。否则，小车仅经过该点，则离开时间=到达时间。

S105、按所述离开时间由小到大排序，依次检测小车达从当前交叉点到下一交叉点是否遇到冲突，

若否，则确定该小车离开下一交叉点的时间为 t_1 ，并设调度时刻 $t=t_1$ ，如果该小车达到该点为卸货点，卸货完成后，重置小车状态为 idle，并返回 S203，对小车分配新的任务。

若是，则进行 S106；

本申请实施例中，由于在半导体晶圆制造厂中，小车在轨道都是按照固定的方向单向运行的，因此当两辆小车距离小于设定阈值时，即认定冲突产生，发生阻塞。

S106、判断冲突是否发生在交叉点，

若是，则使任务优先级高的小车优先通过，任务优先级低的小车在该交叉点等待，同时将等待在该交叉点的任务优先级低的小车的状态更新为 wait，

或，使任务优先级低的小车在上一交叉点选择另一条分支，即更换路径；

若否，则认定该冲突发生在小车装/卸货过程中，则使被阻塞的小车原地等待，小车离开时间=到达时间+等待时间，同时将被阻塞原地等待的小车的状态

更新为 wait，至另一辆小车完成装卸货过程，

或，使被阻塞的小车在上一交叉点选择另一条分支，更换路径。

本申请实施例的该步骤中，由于在 Interbay 和 Intrabay 之间有交叉口，阻塞主要出现在两车通过交叉口、小车装货/卸货时、以及小车出现故障时。当阻塞发生时，采取如下策略进行解除：(1) 原地等待；(2) 其中一辆车更换路径。

因此，首先判断冲突是否发生在交叉口，对于两车在交叉口发生阻塞，由于通过路口时间较短，可以选择装载高优先级的任务的小车优先通过，装载低优先级的任务的小车原地等待，或者在上一个交叉口选择另一条分支，即更换路径。如果判定冲突不是发生在交叉口，则认定该冲突发生在小车装/卸货过程中，对于前面有小车装货/卸货或出现故障，根据等待的时长，可以优先选择更换路径，即：对即将通过该冲突路段的上一个交叉口到达该冲突路段的小车，在该冲突路段的上一个交叉口选择另一条分支，使已经被阻塞的小车原地等待。

本实施例中，上述任务优先级的设定规则可根据具体情况设置。例如按照天车现有路径的延迟时间由高到低排序、按照晶圆任务的优先级由高到低排序、按照天车因冲突而等待的时间由高到低排序中的任一种。

在一些实施中，当小车在上一交叉点选择另一条分支，即更换路径时，可在上一交叉点上重新选择最优路径。

在一种实施例中，小车更换路径时，在上一交叉点上重新选择最优路径。操作可以是：首先查询所述路由表，获取该小车从上一交叉点至目标点的最短路径，再判断该最短路径中是否发生冲突；

若否，执行该最短路径；

若是，则将该最短路径中的冲突采用弹射链算法进行消解；

所述弹射链算法是将产生冲突的小车路径执行弹出动作，剩余小车执行邻域动作，再对弹出的小车重新规划路径，以消解冲突。

在一些实施中：为进一步对轨道上的天车进行拥塞控制和流量疏导，将轨道上的空间节点沿时间轴展开得到时空状态图，然后在时空状态图上使用弹射链的元启发式算法进行流量疏导。

时空状态图以经典车辆路由问题中的空间节点为基础，按时间轴展开为一

系列时空节点，将天车的碰撞检测用有向图的形式建模，其构造方式如图 2 所示。首先将所有轨道离散化成一系列节点，确保每个时间节拍天车可以从当前节点移动到其相邻节点。对于每个节点 v_i ，根据整个规划周期包含的时间节拍数展开为一系列时空节点 $\{v_{i0}, v_{i1}, \dots, v_{iT}\}$ ，原拓扑图中的边 $v_i \rightarrow v_j$ 也随之展开为 $\{v_{i0} \rightarrow v_{j1}, v_{i1} \rightarrow v_{j2}, \dots, v_{i(T-1)} \rightarrow v_{jT}\}$ ，同时增加从 v_{ii} 到 $v_{ii(t+1)}$ 的边表示原地等待，同时空间中的路径也相应地变成了时空路径。通过以上表述，运输车的碰撞约束转换为节点的度约束，即时空路径经过的每个时空节点的入度和出度分别不超过 1。

弹射链是一种高效的元启发式算法，十分适合用于求解容量约束较强的资源分配问题。如图 3-图 6 所示，其基本思想为先挑选一个对象释放其占用的资源，形成一个资源限制更宽松的部分解，然后对其执行邻域动作得到参考结构。从最优参考结构出发，可以继续执行邻域动作得到新的参考结构；也可以执行试探动作重新得到一个完整解，结束本轮弹射链搜索。

为了描述弹射链算法的具体操作，以图 3-图 6 所示算例为例进行说明：

图 3 中：算例信息。表示 4 个节点，4 个天车，分别执行 4 条运输路径的算例。

图 4：初始解，存在冲突。即 $A \rightarrow B$ 与 $B \rightarrow C$ 于 $T1$ 时刻在 C 点发生冲突。

图 5：弹射操作示意图。执行弹出动作移除 $D \rightarrow C$ 的路径，执行邻域动作对 $B \rightarrow C$ 重路由，消除一个冲突，得到参考结构。

图 6：试探动作示意图。执行试探动作，为被移除的路径 $D \rightarrow C$ 重新计算冲突数最少的最短路径，得到试探解。之后可以执行新的弹出动作，也可以考虑从(b)出发继续执行邻域动作。例如，弹出 $B \rightarrow C$ 再放回得到试探解可以消除所有冲突。

在图 3-图 6 所示的弹射链执行过程示例中，从初始的未完全消除天车碰撞的松弛解出发，通过弹出从节点 D 到节点 C 的路径，可以对剩余路径进行优化，以对每一条路径重新计算冲突最少的最短路径为邻域动作，减少冲突的路径数（即两辆天车同时经过同一节点），得到参考结构。接下来，可以继续执行邻域动作，但由于本轮的最优参考结构已经没有冲突，无需进一步执行邻域动作，

故直接执行试探动作，将弹出的路径放回。类似地，继续进行下一轮弹射链可以消除全部冲突，得到可行解。

S107、反复执行所述 S102-S106，直到调度周期结束或所有搬运任务均执行完成。

本申请实施例中，直至调度周期结束或所有搬运任务均执行完成的过程中，小车可存在的运动状态包括：(1) idle，空闲，未分配任务；(2) orderrec，小车接到命令，前往目的地取货，空载运行；(3) pickup，小车取货；(4) delivery，小车载货运行；(5) unload，小车卸货；(6) wait，小车等待；failure，(7) 小车故障。

在本申请第二实施例中，所述 S106 中，若认定该冲突发生在小车装/卸货过程中，则判断当前装/卸货小车的装/卸货时间，

若所述装/卸货时间不大于预设时间，则使被阻塞的小车原地等待，至当前装/卸货小车完成装卸货过程，

若所述装/卸货时间大于预设时间，则使被阻塞的小车在上一交叉点选择另一条分支，即更换路径。

本实施例中，若判定冲突是发生在小车装/卸货过程中，则判断当前装/卸货小车的装/卸货时间，以确定堵塞小车是装货/卸货，还是该小车出现故障。在一些实施中，通过装/卸货时间与阈值进行比较判断，若装/卸货时间大于阈值，则判定该小车出现故障，则使被阻塞的小车在上一交叉点选择另一条分支，更换路径。若装/卸货时间小于或等于阈值，则判定该小车是装货/卸货，被阻塞的小车原地等待，至当前装/卸货小车完成装卸货过程。

在本申请第三实施例中，所述 S101 中，在计算获取轨道地图中任意两个交叉点之间的最短距离之前，先对所述轨道地图进行预处理。

由于原始轨道地图中包括交叉点，barcode（条形码），机台点位，stocker（晶圆自动仓储器或立体库或储存库，用于存储晶圆）点位等。所述的预处理是将原始轨道地图中的其他点去掉，仅保留交叉点，构造一个新的简单图。然后采用最短路算法，求解任意两个交叉点之间的最短距离。因此，所述的预处理的目的是缩小轨道地图的规模，以减小算法运行时间。

在本申请实施例的 AMHS 算法中，为了简化模型，所有小车均以相同的速度匀速运行，遇到阻塞时，在某一个节点上原地等待，不考虑实际刹车、拐弯时候的加速减速等情况。同时，为了模拟实际运行场景，AMHS 算法的输入数据为某晶圆制造厂一段时间内（如一天）的真实执行的所有搬送任务。地图采用真实工厂地图，算法运行开始时，对地图进行预处理，采用 A* 算法找到任意两个交叉点之间的最短距离，针对每个交叉点，记录到其他交叉点的下一步该经过的点。经过处理后，算法在执行过程中，查找任意两个点之间的最短距离就可以通过查找与它们最相近的交叉点的最短路径，即查表可得。

本申请实施例在具体实施时，其中的算法输入包括：

晶圆制造厂轨道地图：点，边，点的属性，维修位等；

一段时间内的原始任务数据：任务编号，起点，终点，任务优先级；

小车数据：小车初始位置，小车运行速度，历史行驶里程，小车优先级，
小车 cycling rule（轮询规则），on call station（轮询区域）；

算法输出包括：每个任务的执行路径，包括分配的小车，经过的所有点以及经过这些点的时间。

本申请实施例的基于冲突消解的天车运输路径规划方法，可以快速准确的将 wafer（晶圆）搬送到目的地，减少晶圆在机台设备之间的循环时间，提高系统的吞吐率，实现高效稳定的物料传输。

以上所述，仅为本申请的具体实施例，但本申请的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到的变化或替换，都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应以权利要求的保护范围为准。

权利要求书

1、一种基于冲突消解的天车运输路径规划方法，包括如下步骤：

S101、计算轨道地图中任意两个交叉点之间的最短距离，建立路由表；

S102、将小车和已下达的搬送任务进行最优匹配，至所有小车或搬送任务全部被匹配；

S103、根据已下达的搬送任务数据查询所述路由表，获得每个小车送达搬送任务需要经过的最短路径；

S104、计算每辆小车在所述最短路径上从当前交叉点到离开下一交叉点的离开时间，并将所述离开时间按从小到大的顺序进行排序；

S105、按所述离开时间的排序，依次检测小车从当前交叉点到下一交叉点是否遇到冲突，

若否，则确定该小车离开下一交叉点的时间为 t_1 ，并设调度时刻 $t=t_1$ ，若是，则进行 S106；

S106、判断冲突是否发生在交叉点，

若是，则使任务优先级高的小车优先通过，任务优先级低的小车在该交叉点等待，

或，使任务优先级低的小车在上一交叉点选择另一条分支，更换路径；

若否，则认定该冲突发生在小车装/卸货过程中，则使被阻塞的小车原地等待，至另一辆小车完成装卸货过程，

或，使被阻塞的小车在上一交叉点选择另一条分支，更换路径；

S107、反复执行所述 S102-S106，直到调度周期结束或所有搬送任务均执行完成；

其中，当小车在上一交叉点更换路径时，首先查询所述路由表，获取该小车从上一交叉点至目标点的最短路径，再判断该最短路径中是否发生冲突；

若否，执行该最短路径；

若是，则将该最短路径中的冲突采用弹射链算法进行消解；

所述弹射链算法是将产生冲突的小车路径执行弹出动作，剩余小车执行邻域动作，再对弹出的小车重新规划路径，以消解冲突。

2、根据权利要求1所述的基于冲突消解的天车运输路径规划方法，其中，

所述 S102 的过程具体包括：

S201、对已下达的搬运任务数据进行预处理，形成任务序列，得到搬运任务集合；

S202、将所有小车状态设为 idle，小车到达初始点的时刻设为 0，得到小车集合；

S203、将 t 时刻状态为 idle 的所述小车集合和所述搬运任务集合进行二分图匹配，至所有小车或搬运任务全部被匹配。

3、根据权利要求 2 所述的基于冲突消解的天车运输路径规划方法，其中，所述 S201 中，对已下达的搬运任务数据进行预处理的过程包括：将搬运任务的下达时间改为从 0 时刻开始经过的秒数，并将搬运任务按下达时间从小到大排序，形成所述任务序列。

4、根据权利要求 2 所述的基于冲突消解的天车运输路径规划方法，其中，所述 S203 中，将 t 时刻状态为 idle 的小车集合和搬运任务集合进行二分图匹配的过程包括：将 t 时刻状态为 idle 的小车按照已空闲时间由高到低进行排序，对每一辆小车，依次选择距离最近的搬运任务进行匹配，并将已匹配的小车和搬运任务从各自集合中删除，反复执行，至所有小车或搬运任务全部被匹配。

5、根据权利要求 2 或 4 所述的基于冲突消解的天车运输路径规划方法，其中，所述 S203 中，在将 t 时刻状态为 idle 的小车集合和搬运任务集合进行二分图匹配的过程中，若 t 时刻状态为 idle 的小车的数量大于搬运任务的数量，则对未匹配到搬运任务的小车生成一个虚拟任务并执行，使该小车在其 on call station 的 Intrabay 中空载运行。

6、根据权利要求 2 或 4 所述的基于冲突消解的天车运输路径规划方法，其中，所述 S203 中，在将 t 时刻状态为 idle 的小车集合和搬运任务集合进行二分图匹配的过程中，若 t 时刻状态为 idle 的小车的数量小于搬运任务的数量，则小车全部被分配到搬运任务，剩下未被分配的搬运任务在机台上继续等待，直到有小车状态更新为 idle 时再分配。

7、根据权利要求 1 所述的基于冲突消解的天车运输路径规划方法，其中，

所述 S106 中，若认定该冲突发生在小车装/卸货过程中，则判断当前装/卸货小车的装/卸货时间，

若所述装/卸货时间不大于预设时间，则使被阻塞的小车原地等待，至当前装/卸货小车完成装卸货过程，

若所述装/卸货时间大于预设时间，则使被阻塞的小车在上一交叉点选择另一条分支，更换路径。

8、根据权利要求 1 所述的基于冲突消解的天车运输路径规划方法，其中，所述 S104 中，同时将所有小车状态更新为 delivery，或 pickup，或 unload。

9、根据权利要求 1 所述的基于冲突消解的天车运输路径规划方法，其中，所述 S106 中，若冲突是发生在交叉点，使任务优先级高的小车优先通过，则将等待在该交叉点的任务优先级低的小车的状态更新为 wait；

若认定该冲突发生在小车装/卸货过程中，则将被阻塞原地等待的小车的状态更新为 wait，至另一辆小车完成装卸货过程。

10、根据权利要求 1 所述的基于冲突消解的天车运输路径规划方法，其中，所述 S101 中，采用包括 A*算法、Dijkstra 算法、SPFA 算法中的任一算法计算轨道地图中任意两个交叉点之间的最短距离。

11、根据权利要求 10 所述的基于冲突消解的天车运输路径规划方法，其中，所述 S101 中，采用 A*算法计算获取轨道地图中任意两个交叉点之间的最短距离。

12、根据权利要求 1 所述的基于冲突消解的天车运输路径规划方法，其中，还包括，所述 S101 中，在计算获取轨道地图中任意两个交叉点之间的最短距离之前，先对所述轨道地图进行预处理。