**说 明 书 摘 要**

本说明书公开了一种任务执行方法、装置、存储介质及电子设备，本说明书实施例根据原始有向图中涉及的待变化边的起始顶点的状态以及受待变化边影响的目标顶点的入边连接的邻居顶点的状态中的至少一种状态，确定出通过待变化边对原始有向图进行调整后目标顶点的变化后状态，并基于目标顶点的变化后状态以及原始有向图中未受待变化边影响的顶点的状态，执行目标任务。在此方法中，在有向图调整之后，无需重新计算调整后有向图中所有顶点的状态，而是基于原始有向图中至少部分顶点的状态，只对受到插入边或删除边影响的顶点的状态进行重新计算，不受影响的顶点的状态无需重新计算，从而减少了冗余计算。

**摘 要 附 图**



**权 利 要 求 书**

1、一种任务执行方法，其特征在于，包括：

获取原始有向图，确定所述原始有向图中每个顶点的状态，以及确定所述原始有向图中各顶点之间需要变化的边，作为待变化边，其中，针对所述原始有向图中包含的每个顶点，该顶点通过入边或出边与邻居顶点相连，该顶点的入边是以邻居顶点为起始顶点出发指向该顶点的边，该顶点的出边是以该顶点为起始顶点出发指向邻居顶点的边；

基于所述待变化边的指向，从所述原始有向图中查找出受所述待变化边影响的顶点，作为目标顶点；

根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态；

在接收到目标任务请求后，基于所述目标顶点的变化后状态以及所述原始有向图中未受到所述待变化边影响的顶点的状态，执行所述目标任务。

2、如权利要求1所述的方法，其特征在于，在获取原始有向图之后，所述方法还包括：

确定高速缓存区的大小以及存储一个顶点的状态所需占用的存储空间；

基于所述高速缓存区的大小以及所述存储空间，确定所述高速缓存区所能存储的顶点数量；

将所述原始有向图中各顶点按照度数从大到小排序，得到顶点序列；

按照度数从大到小的顺序，从所述顶点序列中确定出所述顶点数量的顶点，并将确定出的顶点的状态保存于所述高速缓存区中。

3、如权利要求1所述的方法，其特征在于，所述待变化边包括：待插入边和/或待删除边。

4、如权利要求1所述的方法，其特征在于，根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态，具体包括：

基于所述待变化边的指向，确定出所述待变化边的起始顶点；

基于所述起始顶点的状态，确定出在所述原始有向图中所述起始顶点通过所述起始顶点的每条出边所传递的状态变化量，作为第一状态变化量；

在所述原始有向图中查找出受所述起始顶点的每条出边影响的其他顶点；

从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到所述其他顶点的抵消后状态；

基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；

根据所述调整后有向图中所述起始顶点的状态，确定出在所述调整后有向图中所述起始顶点通过所述起始顶点的出边所需传递的状态变化量，作为第二状态变化量；

基于所述第二状态变化量，对所述调整后有向图中所述目标顶点的抵消后状态进行更新，得到所述目标顶点的变化后状态。

5、如权利要求4所述的方法，其特征在于，从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到所述其他顶点的抵消后状态，具体包括：

基于所述起始顶点通过所述起始顶点的每条出边所传递的状态变化量，确定出通过所述起始顶点的每条出边所需传递的负增量；

以所述负增量为传递参数，沿所述起始顶点的每条出边进行传递，以从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到抵消后状态。

6、如权利要求1所述的方法，其特征在于，根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态，具体包括：

基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；

若所述待变化边为待插入边，基于所述待变化边的起始顶点的状态以及所述待变化边的权重，确定出所述调整后有向图中所述待变化边的目的顶点的预计状态，所述目的顶点是指所述待变化边从所述起始顶点出发所指向的顶点，所述目的顶点为所述目标顶点中的一个顶点；

若所述预计状态小于所述原始有向图中所述待变化边的目的顶点的状态，则将所述预计状态确定为在所述调整后有向图中所述目的顶点的状态，并基于所述目的顶点的预计状态，对所述调整后有向图中所述目标顶点中除所述目的顶点之外的其他目标顶点的状态进行更新，得到变化后状态；

若所述预计状态不小于所述原始有向图中所述待变化边的目的顶点的状态，则保持所述目的顶点在所述原始有向图中的状态以及无需对所述调整后有向图中除所述目的顶点之外的其他目标顶点的状态进行更新。

7、如权利要求1所述的方法，其特征在于，根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态，具体包括：

基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；

若所述待变化边为待删除边，对所述调整后有向图中所述目标顶点的状态进行初始化，得到初始状态；

针对所述目标顶点通过入边连接的每个邻居顶点，基于该邻居顶点的状态以及该邻居顶点与所述目标顶点之间的入边的权重，确定出该邻居顶点为所述目标顶点所贡献的状态量，作为该邻居顶点对应的状态量；

从每个邻居顶点对应的状态量中选择出最小的状态量，作为在所述调整后有向图中所述目标顶点的变化后状态。

8、一种任务执行装置，其特征在于，包括：

获取模块，用于获取原始有向图，确定所述原始有向图中每个顶点的状态，以及确定所述原始有向图中各顶点之间需要变化的边，作为待变化边，其中，针对所述原始有向图中包含的每个顶点，该顶点通过入边或出边与邻居顶点相连，该顶点的入边是以邻居顶点为起始顶点出发指向该顶点的边，该顶点的出边是以该顶点为起始顶点出发指向邻居顶点的边；

查找模块，用于基于所述待变化边的指向，从所述原始有向图中查找出受所述待变化边影响的顶点，作为目标顶点；

确定模块，用于根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态；

执行任务模块，用于在接收到目标任务请求后，基于所述目标顶点的变化后状态以及所述原始有向图中未受到所述待变化边影响的顶点的状态，执行所述目标任务。

9、如权利要求8所述的装置，其特征在于，所述获取模块，还用于确定高速缓存区的大小以及存储一个顶点的状态所需占用的存储空间；基于所述高速缓存区的大小以及所述存储空间，确定所述高速缓存区所能存储的顶点数量；将所述原始有向图中各顶点按照度数从大到小排序，得到顶点序列；按照度数从大到小的顺序，从所述顶点序列中确定出所述顶点数量的顶点，并将确定出的顶点的状态保存于所述高速缓存区中。

10、如权利要求8所述的装置，其特征在于，所述确定模块，具体用于基于所述待变化边的指向，确定出所述待变化边的起始顶点；基于所述起始顶点的状态，确定出在所述原始有向图中所述起始顶点通过所述起始顶点的每条出边所传递的状态变化量，作为第一状态变化量；在所述原始有向图中查找出受所述起始顶点的每条出边影响的其他顶点；从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到所述其他顶点的抵消后状态；基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；根据所述调整后有向图中所述起始顶点的状态，确定出在所述调整后有向图中所述起始顶点通过所述起始顶点的出边所需传递的状态变化量，作为第二状态变化量；基于所述第二状态变化量，对所述调整后有向图中所述目标顶点的抵消后状态进行更新，得到所述目标顶点的变化后状态。

11、如权利要求10所述的装置，其特征在于，所述确定模块，具体用于基于所述起始顶点通过所述起始顶点的每条出边所传递的状态变化量，确定出通过所述起始顶点的每条出边所需传递的负增量；以所述负增量为传递参数，沿所述起始顶点的每条出边进行传递，以从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到抵消后状态。

12、如权利要求8所述的装置，其特征在于，所述确定模块，具体用于基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；若所述待变化边为待插入边，基于所述待变化边的起始顶点的状态以及所述待变化边的权重，确定出所述调整后有向图中所述待变化边的目的顶点的预计状态，所述目的顶点是指所述待变化边从所述起始顶点出发所指向的顶点，所述目的顶点为所述目标顶点中的一个顶点；若所述预计状态小于所述原始有向图中所述待变化边的目的顶点的状态，则将所述预计状态确定为在所述调整后有向图中所述目的顶点的状态，并基于所述目的顶点的预计状态，对所述调整后有向图中所述目标顶点中除所述目的顶点之外的其他目标顶点的状态进行更新，得到变化后状态；若所述预计状态不小于所述原始有向图中所述待变化边的目的顶点的状态，则保持所述目的顶点在所述原始有向图中的状态以及无需对所述调整后有向图中除所述目的顶点之外的其他目标顶点的状态进行更新。

13、如权利要求8所述的装置，其特征在于，所述确定模块，具体用于基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；若所述待变化边为待删除边，对所述调整后有向图中所述目标顶点的状态进行初始化，得到初始状态；针对所述目标顶点通过入边连接的每个邻居顶点，基于该邻居顶点的状态以及该邻居顶点与所述目标顶点之间的入边的权重，确定出该邻居顶点为所述目标顶点所贡献的状态量，作为该邻居顶点对应的状态量；从每个邻居顶点对应的状态量中选择出最小的状态量，作为在所述调整后有向图中所述目标顶点的变化后状态。

14、如权利要求8所述的装置，其特征在于，所述待变化边包括：待插入边和/或待删除边。

15、一种计算机可读存储介质，其特征在于，所述存储介质存储有计算机程序，所述计算机程序被处理器执行时实现上述权利要求1-7任一项所述的方法。

16、一种电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，其特征在于，所述处理器执行所述程序时实现上述权利要求1-7任一项所述的方法。

**说 明 书**

一种任务执行方法、装置、存储介质及电子设备

**技术领域**

本说明书涉及动态图处理技术领域，尤其涉及一种任务执行方法、装置、存储介质及电子设备。

**背景技术**

随着大数据时代的来临，图作为一种能够良好表达数据关联性的数据结构，在互联网应用、商业金融、科学计算等许多领域得到了广泛应用。例如：商品推荐、路径预测、金融风险检测、社交网络分析等应用场景。

在将有向图应用于不同应用场景的过程中，随着时间的推移，有向图的结构会随着时间变化而变化，比如增加或删除边和顶点。而有向图中顶点的状态值也会随着图的结构变化而变化，这意味着变化前计算得到的顶点的状态值在变化后的有向图中都是无效的。一般的解决方法是舍弃之前计算得到的所有顶点的状态值，在变化后的有向图中从头开始计算每个顶点的状态值。

然而，从变化前的有向图变化到变化后的有向图的过程中所涉及的顶点和边的改变通常是比较小的，舍弃之前计算到的所有顶点的状态值意味着舍弃了许多在变化后的有向图中依然有效的顶点状态，从而导致在变化后的有向图中重新计算每个顶点的状态值的过程中会产生冗余计算。

**发明内容**

本说明书实施例提供一种任务执行方法、装置、存储介质及电子设备，以部分解决上述现有技术存在的问题。

本说明书实施例采用下述技术方案：

本说明书提供的一种任务执行方法，包括：

获取原始有向图，确定所述原始有向图中每个顶点的状态，以及确定所述原始有向图中各顶点之间需要变化的边，作为待变化边，其中，针对所述原始有向图中包含的每个顶点，该顶点通过入边或出边与邻居顶点相连，该顶点的入边是以邻居顶点为起始顶点出发指向该顶点的边，该顶点的出边是以该顶点为起始顶点出发指向邻居顶点的边；

基于所述待变化边的指向，从所述原始有向图中查找出受所述待变化边影响的顶点，作为目标顶点；

根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态；

在接收到目标任务请求后，基于所述目标顶点的变化后状态以及所述原始有向图中未受到所述待变化边影响的顶点的状态，执行所述目标任务。

可选地，在获取原始有向图之后，所述方法还包括：

确定高速缓存区的大小以及存储一个顶点的状态所需占用的存储空间；

基于所述高速缓存区的大小以及所述存储空间，确定所述高速缓存区所能存储的顶点数量；

将所述原始有向图中各顶点按照度数从大到小排序，得到顶点序列；

按照度数从大到小的顺序，从所述顶点序列中确定出所述顶点数量的顶点，并将确定出的顶点的状态保存于所述高速缓存区中。

可选地，所述待变化边包括：待插入边和/或待删除边。

可选地，根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态，具体包括：

基于所述待变化边的指向，确定出所述待变化边的起始顶点；

基于所述起始顶点的状态，确定出在所述原始有向图中所述起始顶点通过所述起始顶点的每条出边所传递的状态变化量，作为第一状态变化量；

在所述原始有向图中查找出受所述起始顶点的每条出边影响的其他顶点；

从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到所述其他顶点的抵消后状态；

基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；

根据所述调整后有向图中所述起始顶点的状态，确定出在所述调整后有向图中所述起始顶点通过所述起始顶点的出边所需传递的状态变化量，作为第二状态变化量；

基于所述第二状态变化量，对所述调整后有向图中所述目标顶点的抵消后状态进行更新，得到所述目标顶点的变化后状态。

可选地，从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到所述其他顶点的抵消后状态，具体包括：

基于所述起始顶点通过所述起始顶点的每条出边所传递的状态变化量，确定出通过所述起始顶点的每条出边所需传递的负增量；

以所述负增量为传递参数，沿所述起始顶点的每条出边进行传递，以从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到抵消后状态。

可选地，根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态，具体包括：

基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；

若所述待变化边为待插入边，基于所述待变化边的起始顶点的状态以及所述待变化边的权重，确定出所述调整后有向图中所述待变化边的目的顶点的预计状态，所述目的顶点是指所述待变化边从所述起始顶点出发所指向的顶点，所述目的顶点为所述目标顶点中的一个顶点；

若所述预计状态小于所述原始有向图中所述待变化边的目的顶点的状态，则将所述预计状态确定为在所述调整后有向图中所述目的顶点的状态，并基于所述目的顶点的预计状态，对所述调整后有向图中所述目标顶点中除所述目的顶点之外的其他目标顶点的状态进行更新，得到变化后状态；

若所述预计状态不小于所述原始有向图中所述待变化边的目的顶点的状态，则保持所述目的顶点在所述原始有向图中的状态以及无需对所述调整后有向图中除所述目的顶点之外的其他目标顶点的状态进行更新。

可选地，根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态，具体包括：

基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；

若所述待变化边为待删除边，对所述调整后有向图中所述目标顶点的状态进行初始化，得到初始状态；

针对所述目标顶点通过入边连接的每个邻居顶点，基于该邻居顶点的状态以及该邻居顶点与所述目标顶点之间的入边的权重，确定出该邻居顶点为所述目标顶点所贡献的状态量，作为该邻居顶点对应的状态量；

从每个邻居顶点对应的状态量中选择出最小的状态量，作为在所述调整后有向图中所述目标顶点的变化后状态。

本说明书提供的一种任务执行装置，包括：

获取模块，用于获取原始有向图，确定所述原始有向图中每个顶点的状态，以及确定所述原始有向图中各顶点之间需要变化的边，作为待变化边，其中，针对所述原始有向图中包含的每个顶点，该顶点通过入边或出边与邻居顶点相连，该顶点的入边是以邻居顶点为起始顶点出发指向该顶点的边，该顶点的出边是以该顶点为起始顶点出发指向邻居顶点的边；

查找模块，用于基于所述待变化边的指向，从所述原始有向图中查找出受所述待变化边影响的顶点，作为目标顶点；

确定模块，用于根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态；

执行任务模块，用于在接收到目标任务请求后，基于所述目标顶点的变化后状态以及所述原始有向图中未受到所述待变化边影响的顶点的状态，执行所述目标任务。

可选地，所述获取模块，还用于确定高速缓存区的大小以及存储一个顶点的状态所需占用的存储空间；基于所述高速缓存区的大小以及所述存储空间，确定所述高速缓存区所能存储的顶点数量；将所述原始有向图中各顶点按照度数从大到小排序，得到顶点序列；按照度数从大到小的顺序，从所述顶点序列中确定出所述顶点数量的顶点，并将确定出的顶点的状态保存于所述高速缓存区中。

可选地，所述确定模块，具体用于基于所述待变化边的指向，确定出所述待变化边的起始顶点；基于所述起始顶点的状态，确定出在所述原始有向图中所述起始顶点通过所述起始顶点的每条出边所传递的状态变化量，作为第一状态变化量；在所述原始有向图中查找出受所述起始顶点的每条出边影响的其他顶点；从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到所述其他顶点的抵消后状态；基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；根据所述调整后有向图中所述起始顶点的状态，确定出在所述调整后有向图中所述起始顶点通过所述起始顶点的出边所需传递的状态变化量，作为第二状态变化量；基于所述第二状态变化量，对所述调整后有向图中所述目标顶点的抵消后状态进行更新，得到所述目标顶点的变化后状态。

可选地，所述确定模块，具体用于基于所述起始顶点通过所述起始顶点的每条出边所传递的状态变化量，确定出通过所述起始顶点的每条出边所需传递的负增量；以所述负增量为传递参数，沿所述起始顶点的每条出边进行传递，以从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到抵消后状态。

可选地，所述确定模块，具体用于基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；若所述待变化边为待插入边，基于所述待变化边的起始顶点的状态以及所述待变化边的权重，确定出所述调整后有向图中所述待变化边的目的顶点的预计状态，所述目的顶点是指所述待变化边从所述起始顶点出发所指向的顶点，所述目的顶点为所述目标顶点中的一个顶点；若所述预计状态小于所述原始有向图中所述待变化边的目的顶点的状态，则将所述预计状态确定为在所述调整后有向图中所述目的顶点的状态，并基于所述目的顶点的预计状态，对所述调整后有向图中所述目标顶点中除所述目的顶点之外的其他目标顶点的状态进行更新，得到变化后状态；若所述预计状态不小于所述原始有向图中所述待变化边的目的顶点的状态，则保持所述目的顶点在所述原始有向图中的状态以及无需对所述调整后有向图中除所述目的顶点之外的其他目标顶点的状态进行更新。

可选地，所述确定模块，具体用于基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；若所述待变化边为待删除边，对所述调整后有向图中所述目标顶点的状态进行初始化，得到初始状态；针对所述目标顶点通过入边连接的每个邻居顶点，基于该邻居顶点的状态以及该邻居顶点与所述目标顶点之间的入边的权重，确定出该邻居顶点为所述目标顶点所贡献的状态量，作为该邻居顶点对应的状态量；从每个邻居顶点对应的状态量中选择出最小的状态量，作为在所述调整后有向图中所述目标顶点的变化后状态。

可选地，所述待变化边包括：待插入边和/或待删除边。

本说明书提供的一种计算机可读存储介质，所述存储介质存储有计算机程序，所述计算机程序被处理器执行时实现上述的任务执行方法。

本说明书提供的一种电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，所述处理器执行所述程序时实现上述的任务执行方法。

本说明书实施例采用的上述至少一个技术方案能够达到以下有益效果：

本说明书实施例中根据原始有向图中涉及的待变化边的起始顶点的状态以及受待变化边影响的目标顶点的入边连接的邻居顶点的状态中的至少一种状态，确定出通过待变化边对原始有向图进行调整后目标顶点的变化后状态，并基于目标顶点的变化后状态以及原始有向图中未受待变化边影响的顶点的状态，执行目标任务。在此方法中，在有向图调整之后，无需重新计算调整后有向图中所有顶点的状态，而是基于原始有向图中至少部分顶点的状态，只对受到插入边或删除边影响的顶点的状态进行重新计算，不受影响的顶点的状态无需重新计算，从而减少了冗余计算。

**附图说明**

此处所说明的附图用来提供对本说明书的进一步理解，构成本说明书的一部分，本说明书的示意性实施例及其说明用于解释本说明书，并不构成对本说明书的不当限定。在附图中：

图1为本说明书实施例提供的任务执行方法的流程示意图；

图2为本说明书实施例提供的图处理单元的位置示意图；

图3为本说明书实施例提供的图处理单元的结构示意图；

图4为本说明书实施例提供的在原始有向图中查找出目标顶点的示意图；

图5为本说明书实施例提供的原始有向图与调整后有向图的对照示意图；

图6为本说明书实施例提供的插入边前的原始有向图与插入边后的调整后有向图的对照示意图；

图7为本说明书实施例提供的删除边前的原始有向图与删除边后的调整后有向图的对照示意图；

图8为本说明书实施例提供的任务执行装置结构示意图；

图9为本说明书实施例提供的电子设备的结构示意图。

**具体实施方式**

本说明书提供的任务执行方法旨在基于上一次有向图中每个顶点的状态，重新计算对上一次有向图进行局部调整后，受调整边影响的顶点的状态，对于不受调整边影响的顶点的状态无需重新计算，保持不变即可。

为使本说明书的目的、技术方案和优点更加清楚，下面将结合本说明书具体实施例及相应的附图对本说明书技术方案进行清楚、完整地描述。显然，所描述的实施例仅是本说明书一部分实施例，而不是全部的实施例。基于本说明书中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本说明书保护的范围。

以下结合附图，详细说明本说明书各实施例提供的技术方案。

图1为本说明书实施例提供的任务执行方法的流程示意图，该任务执行方法可以适用于服务器，包括：

S100：获取原始有向图，确定所述原始有向图中每个顶点的状态，以及确定所述原始有向图中各顶点之间需要变化的边，作为待变化边，其中，针对所述原始有向图中包含的每个顶点，该顶点通过入边或出边与邻居顶点相连，该顶点的入边是以邻居顶点为起始顶点出发指向该顶点的边，该顶点的出边是以该顶点为起始顶点出发指向邻居顶点的边。

在本说明书实施例中，图1所述的任务执行方法中针对有向图中各顶点的状态的迭代更新方法可以应用于服务器中的图处理单元，图处理单元与L1缓存区并行，如图2所示。在图2中，图处理单元位于L2缓存区的上层，且图处理单元与L2缓存区进行数据交互，交互的数据可以是指顶点的状态。

其中，图处理单元包含有：读取器、暂存器、状态变化量生成器、状态变化量处理器以及状态变化量队列。如图3所示。在图3中，读取器用于从内存中读取原始有向图、原始有向图中每个顶点的状态以及待变化边。暂存器用于存放初始化后的顶点的顶点标识，比如：顶点ID。状态变化量生成器用于基于原始有向图、原始有向图中每个顶点的状态以及待变化边，生成通过待变化边对原始有向图进行调整后的调整后有向图中每个顶点传递的状态变化量。状态变化量队列用于存储状态变化量。状态变化量处理器用于在原始有向图或调整后有向图中进行状态变化量的传递，并在传递过程中迭代更新每个顶点的状态以及每个顶点所需传递的状态变化量。

在本说明书实施例中，可以通过读取器，从内存中获取原始有向图、原始有向图中每个顶点的状态，以及确定出原始有向图中各顶点之间需要变化的边，作为待变化边。其中，原始有向图中各顶点之间需要变化的边可以是指在任意两个顶点之间增加边或将任意两个顶点之间的边删除。也就是，待变化边可以包括：待插入边和/或待删除边。

针对有向图，有向图是由有方向的边和顶点构成，而边可以包含出边和入边。在本说明书中，针对原始有向图中包含的每个顶点，该顶点通过入边或出边与邻居顶点相连，该顶点的入边是以邻居顶点为起始顶点出发指向以该顶点为目的顶点的边，该顶点的出边是以该顶点为起始顶点出发指向以邻居顶点为目的顶点的边。

其中，有向图的应用场景不同，有向图中每个顶点的表征、顶点与顶点之间的边的表征、顶点的状态也有不同。比如：在商品推荐场景中，有向图中的顶点可以是指用户、商品等，顶点与顶点之间的边可以用于表征用户与商品之间的交互关系（如浏览、点击等）。在这种场景下，顶点的状态可以是指用户的用户特征值或商品的商品信息特征值。在路径预测场景下，有向图中的顶点可以是指兴趣点（Point of Interest， POI），顶点与顶点之间的边可以表征各POI之间的通行路径，顶点的状态可以表征最短路径长度。在金融风险检测场景中，顶点可以是指用户信息，顶点与顶点之间的边可以表征用户与用户之间的联系，顶点的状态可以是指用户的欺诈概率。用户信息可以包括：手机号、邮箱等。若存在多个用户信息中的手机号、邮箱等信息相同，则这几个用户的欺诈概率越大。在社交网络分析场景中，顶点可以是指用户信息（如社交账号等），顶点与顶点之间的边可以表征用户与用户之间的联系，顶点的状态可以是指用户的重要程度。用户的重要程度越大，说明用户社交能力越强。

在获取到原始有向图之后，可以根据原始有向图中每个顶点的边的数量，确定出度数比较高的顶点，并把度数比较高的顶点的状态存储到高速缓存区中，以减少内存访问开销。

具体的，可以先确定出高速缓存区的大小以及存储一个顶点的状态所需占用的存储空间，然后，基于高速缓存区的大小以及存储一个顶点的状态所需占用的存储空间，确定出高速缓存区所能存储的顶点数量。同时，将原始有向图中各顶点按照度数从大到小排序，得到顶点序列。其中，度数可以是指与顶点相连接的边的总数量。之后，可以按照度数从大到小的顺序，从顶点序列中确定出顶点数量的顶点，并将确定出的顶点的状态保存于高速缓存区中。

当需要对高速缓存区中的顶点的状态进行改变时，可以从高速缓存区中读取顶点的状态，而无需从内存中读取顶点的状态，从而减少内存访问开销。

S102：基于所述待变化边的指向，从所述原始有向图中查找出受所述待变化边影响的顶点，作为目标顶点。

在本说明书实施例中，在基于待变化边对原始有向图进行调整时，可以只确定出受待变化边影响的顶点，然后，只对受待变化边影响的顶点的状态进行更新。对于不受待变化边影响的顶点的状态，可以保持原本的状态不变，无需对不受待变化边影响的顶点的状态进行更新，从而减少计算量。

在读取器获取到原始有向图、原始有向图中每个顶点的状态以及待变化边之后，可以将原始有向图、原始有向图中每个顶点的状态以及待变化边发送给状态变化量生成器。通过状态变化量生成器可以基于待变化边的指向、待变化边的起始顶点、待变化边的目的顶点以及原始有向图，从原始有向图中查找出受待变化边影响的顶点，作为目标顶点。如图4所示。其中，目标顶点可以是一个或多个。受待变化边影响的目标顶点可以是指沿原始有向图顶点之间的边的指向进行参数传递所能到达的顶点。任意两个顶点之间传递的参数可以是一个顶点需要传递给另一个顶点的状态变化量，由于顶点的状态更新算法不同，状态变化量也不同。比如：状态变化量可以是指一个顶点的状态值的改变量，也可以是指一个顶点的状态值与边的权重之和。

在图4中，原始有向图中包含有四个顶点，分别为：a、b、c、d。以待变化边为待删除边为例，若将bc的边作为待变化边（用虚线表示），则受待变化边影响的目标顶点为：c、d、b。

S104：根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态。

在本说明书实施例中，在确定出受待变化边影响的目标顶点之后，可以通过状态变化量生成器，生成目标顶点所需更新的状态变化量，并基于生成的状态变化量，对原始有向图中目标顶点的状态进行更新，以得到通过待变化边对原始有向图进行调整后得到的调整后有向图中目标顶点的变化后状态。其中，任意顶点所需更新的状态变化量可以是指任意顶点所需传递的状态变化量。

具体的，通过状态变化量生成器，根据待变化边的起始顶点的状态以及目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，生成调整后有向图中目标顶点所需更新的状态变化量，并基于目标顶点所需更新的状态变化量，确定出调整后有向图中目标顶点的变化后状态。

进一步，在生成调整后有向图中目标顶点所需更新的状态变化量之后，可以将生成的目标顶点所需更新的状态变化量增加到状态变化量队列中。然后，通过状态变化量处理器，从状态变化量队列中获取目标顶点所需更新的状态变化量，之后，基于目标顶点所需更新的状态变化量，确定出调整后有向图中目标顶点的变化后状态。

其中，在对原始有向图中的顶点的状态进行迭代更新的过程中，由于采用的状态更新算法不同，导致生成目标顶点所需更新的状态变化量的方法也不同。在本说明书中，可以将状态更新算法分为两大类，第一类为累积更新算法，第二类为选择更新算法。其中，累积更新算法可以包括：PageRank算法等，选择更新算法可以包括：单源最短路径算法等。

针对第一类的累积更新算法：

先介绍累积更新算法的算法原理：在有向图的初始化阶段，有向图中每个顶点的状态值为0，每个顶点对应一个初始化的总状态变化量，总状态变化量可以是指状态值的改变量。针对某个顶点，对该顶点的状态进行更新的方法为：将该顶点对应的总状态变化量累加到该顶点的状态值上，得到新的状态值。该顶点对应的总状态变化量的传递方法为：先基于该顶点对应的总状态变化量、该顶点对的出边数量以及预设参数，确定该顶点的每条出边传递给目的顶点的状态变化量。针对某个目的顶点，该目的顶点将接收到的状态变化量与该目的顶点对应的总状态变化量进行相加，将相加的结果重新作为该目的顶点对应的总状态变化量。其中，计算每条出边传递的状态变化量的公式为：，为顶点对应的总状态变化量，为顶点的出边数量，为预设参数。在这种算法原理下，针对每个顶点，该顶点的状态值是对其他顶点的状态变化量和自身状态变化量的累加。

在累积更新算法的情况下，计算基于待变化边对原始有向图进行调整后的目标顶点的变化后状态的原理是：先在原始有向图的基础上，假设删除待变化边的起始顶点的所有出边，并针对每条出边生成相应的负增量，以负增量来抵消起始顶点对出边连接的邻居顶点所贡献的状态变化量，得到邻居顶点的抵消后状态。然后，重新计算起始顶点通过调整后有向图中依然存在的出边或插入的出边所贡献的状态变化量，并根据重新计算出的状态变化量，对邻居顶点的抵消后状态进行更新，得到变化后状态。

具体的，可以基于待变化边的起始顶点的状态以及目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出在原始有向图中目标顶点所传递的状态变化量以及在调整后有向图中目标顶点所需更新的状态变化量，然后，根据在原始有向图中目标顶点所传递的状态变化量以及在调整后有向图中目标顶点所需更新的状态变化量，确定出调整后有向图中目标顶点的变化后状态。在累积更新算法中，不需求区分待变化边是待插入边或待删除边。

在确定出在原始有向图中目标顶点所传递的状态变化量时，可以基于待变化边的指向，确定出待变化边的起始顶点。然后，基于在原始有向图中待变化边的起始顶点的状态，确定在原始有向图中起始顶点通过起始顶点的每条出边所传递的状态变化量，作为第一状态变化量。之后，在原始有向图中查找出受起始顶点的每条出边影响的其他顶点。从其他顶点的状态中抵消起始顶点传递第一状态变化量过程中所增加的状态量，以得到其他顶点的抵消后状态。需要说明的是，受起始顶点的每条出边影响的其他顶点中包含有受待变化边影响的目标顶点以及至少部分目标顶点通过入边连接的邻居顶点。

其中，确定起始顶点的每条边所传递的状态变化量的方法为：基于起始顶点的状态、起始顶点的出边数量以及预设参数，确定出通过起始顶点的每条出边所传递的状态变化量。计算起始顶点的每条边所传递的状态变化量的公式为：，为起始顶点的状态值，为起始顶点的出边数量，为预设参数。

其中，从其他顶点的状态中抵消起始顶点传递第一状态变化量过程中所增加的状态量的方法可以包括：在确定出每条出边所传递的第一状态变化量之后，可以基于通过起始顶点的每条出边所传递的状态变化量，确定出通过起始顶点的每条出边所需传递的负增量。其中，负增量可以是指第一状态变化量的负值。以负增量为传递参数，沿起始顶点的每条出边进行传递，以从其他顶点中抵消起始顶点传递第一状态变化量时所增加的状态量，得到其他顶点的抵消后状态。

需要说明的是，从其他顶点中抵消起始顶点传递第一状态变化量时所增加的状态量所得到其他顶点的抵消后状态相当于在原始有向图中将起始顶点的所有出边删除后重新计算得到的其他顶点在起始顶点无贡献情况下的状态。

在得到其他顶点的抵消后状态之后，可以基于待变化边，对原始有向图进行调整，得到调整后有向图。即，调整待变化边，比如：删除边和/或插入边。

在对原始有向图进行调整之后，确定出在原始有向图中目标顶点所需更新的状态变化量的方法可以包括：可以根据调整后有向图中起始顶点的状态，确定出调整后有向图中起始顶点通过起始顶点的出边所需传递的状态变化量，作为第二状态变化量。最后，基于第二状态变化量，对调整后有向图中目标顶点的抵消后状态进行更新，得到变化后状态。

具体的，以第二状态变化量为传递参数，沿起始顶点的每条出边进行传递，以对调整后有向图中目标顶点的抵消后状态进行更新，得到变化后状态。也就是，在目标顶点的抵消后状态中累加第二状态变化量。

需要说明的是，调整后有向图中起始顶点可能还存在出边和/或额外增加出边的情况，而其他顶点的抵消后状态没有考虑起始顶点的任何出边，因此，在调整后有向图中需要将在原始有向图中多删的出边增加回来和/或增加额外的出边。

其中，计算出的第二状态变化量相当于起始顶点通过在原始有向图中多删或额外增加的出边所贡献的状态变化量。基于第二状态变化量对目标顶点的抵消后状态进行更新所得到的变化后状态相当于是在目标顶点的抵消后状态中累加起始顶点通过多删或额外增加的边所贡献的第二状态变化量。

基于上述计算目标顶点的变化后状态的描述，本说明书实施例中提供原始有向图与调整后有向图的对照示意图，如图5所示。

在图5中，原始有向图中包含三个顶点，分别为A、B、C。在原始有向图的初始化阶段，顶点A、顶点B、顶点C的状态值均为零，顶点A、顶点B、顶点C各自对应的总状态变化量均为1，预设参数假设为1。在对原始有向图中三个顶点的状态进行迭代更新过程中，顶点A先对自身的状态进行更新，得到状态值为1。将顶点A对应的总状态变化量通过两个出边平均传递给顶点B和顶点C，并将顶点A对应的总状态变化量删除。针对顶点C，顶点C接收顶点A发送的0.5状态变化量，并加上自身的总状态变化量，得到1.5。将1.5与自身的状态值进行相加，得到1.5。所以，顶点C的状态值更新为1.5。顶点C将1.5的总状态变化量传递给顶点B，顶点C将1.5的总状态变化量删除。针对顶点B，顶点B接收顶点A的0.5状态变化量和顶点C的1.5总状态变化量，对自身的总状态变化量进行更新，得到3，以3的总状态变化量更新自身的状态值，得到3。最终，顶点B的状态值为3。顶点B将3的总状态变化量删除。

在对原始有向图进行调整，以删除AC的边为例，AC的边为待变化边，起始顶点为顶点A，在原始有向图中顶点A的出边为AC和AB。假设删除顶点A的所有出边，计算原始有向图中顶点A分别为顶点B和顶点C所贡献的负增量，即，。将负增量传递给顶点B和顶点C。针对顶点C，顶点C对应的总状态变化量为，对自身的状态值进行更新，得到1。顶点C将传递给顶点B，顶点C删除的总状态变化量。针对顶点B，顶点B接收顶点A传递的状态变化量和顶点C传递的总状态变化量，顶点B自身的总状态变化量为-1。以-1对自身的状态值进行更新，得到2。在对原始有向图进行调整之后，调整后有向图中存在AB的边，需要重新计算顶点A通过AB边所贡献的状态变化量，即，。将1的状态变化量传递给顶点B，顶点B以1对自身的状态值进行更新，得到2+1=3。最终，在调整后有向图中，顶点A的状态值为1，顶点B的状态值为3，顶点C的状态值为1。

针对第二类的选择更新算法：

先介绍选择更新算法的算法原理：在有向图中确定出源顶点，针对某个顶点，该顶点的状态值表示源顶点离该顶点的最短距离。在对该顶点的状态值进行更新时，可以将小于该顶点的当前状态值的距离重新作为该顶点的状态值。

在确定调整后有向图中目标顶点的变化后状态时，可以基于待变化边的起始顶点的状态，确定出调整后有向图中目标顶点的变化后状态。或者，可以基于目标顶点通过入边连接的邻居顶点的状态，确定出调整后有向图中目标顶点的变化后状态。

基于待变化边对原始有向图进行调整，得到调整后有向图。然后，在待变化边为待插入边的情况下，可以基于待变化边的起始顶点的状态以及待变化边的权重，确定出调整后有向图中待变化边的目的顶点的预计状态。其中，目的顶点是指待变化边从起始顶点出发所指向的顶点，目的顶点为目标顶点中的一个顶点。目的顶点的预计状态可以是指待变化边的起始顶点为目的顶点所贡献的状态变化量。

其中，将待变化边的起始顶点的状态与待变化边的权重进行相加，得到待变化边的目的顶点的预计状态。

若预计状态小于原始有向图中待变化边的目的顶点的状态，则将预计状态确定为在调整后有向图中目的顶点的状态，并基于目的顶点的预计状态，对调整后有向图中目标顶点中除目的顶点之外的其他目标顶点的状态进行更新，得到各目标顶点的变化后状态。

若预计状态不小于原始有向图中待变化边的目的顶点的状态，则保持目的顶点在原始有向图中的状态。

在保持目的顶点在原始有向图中的状态的情况下，若对原始有向图进行调整时只涉及一个待变化边的插入，无需对调整后有向图中的各目标顶点的状态进行更新。

若对原始有向图进行调整时涉及多个待变化边的插入，可以基于一个待变化边的目的顶点在原始有向图中的状态，对调整后有向图中所有目标顶点中除目的顶点之外的其他目标顶点的状态进行更新，得到其他目标顶点的变化后状态。

基于上述计算插入边后的有向图中目标顶点的变化后状态的方法，本说明书实施例中提供插入边前的原始有向图与插入边后的调整后有向图的对照示意图，如图6所示。

在图6中，在原始有向图中，顶点1为源顶点，顶点1的状态值为0，顶点2的状态值为8，顶点3的状态值为9，顶点4的状态值为12，顶点5的状态值为14，顶点6的状态值为17，顶点7的状态值为19。在原始有向图中插入14边，受14边影响的目标顶点分别为：顶点4、顶点5、顶点6和顶点7。然后，计算14边的目的顶点4的预计状态，即，0+3=3。将预计状态3与顶点4的状态值12进行比较，可以将顶点4的状态值更新为3。接着，对顶点5、顶点6和顶点7的状态进行依次更新，得到变化后状态。最终，顶点5的变化后状态值为8，顶点6的变化后状态值为13，顶点7的变化后状态值为10。

在待变化边为删除边的情况下，在对得到调整后有向图之后，对目标顶点的状态进行初始化，得到初始状态。其中，初始状态可以是极大值。然后，针对目标顶点通过入边连接的每个邻居顶点，基于该邻居顶点的状态以及该邻居顶点与目标顶点之间的入边的权重，确定出该邻居顶点为目标顶点所贡献的状态量，作为该邻居顶点对应的状态量。最后，从每个邻居顶点对应的状态量中选择出最小的状态量，作为目标顶点的变化后状态。其中，每个邻居顶点对应的状态量可以是指邻居顶点为目标顶点所贡献的状态变化量。

在对目标顶点的状态进行初始化之后，可以将初始化后的目标顶点的顶点标识存储于暂存器中，在对初始化后的目标顶点的状态进行重新计算时，状态变化量生成器可以从暂存器中读取目标顶点，以减少遍历有向图的时间。

其中，在确定出该邻居顶点为目标顶点所贡献的状态量时，可以将该邻居顶点的状态和该邻居顶点与目标顶点之间的入边的权重进行求和，得到求和结果，作为该邻居顶点为目标顶点所贡献的状态量。

基于上述计算删除边后的有向图中目标顶点的变化后状态的方法，本说明书实施例中提供删除边前的原始有向图与删除边后的调整后有向图的对照示意图，如图7所示。

在图7中，在原始有向图中，顶点1为源顶点，顶点1的状态值为0，顶点2的状态值为8，顶点3的状态值为9，顶点4的状态值为12，顶点5的状态值为14，顶点6的状态值为17，顶点7的状态值为19。在原始有向图中删除13边，受13边影响的目标顶点分别为：顶点3、顶点5、顶点6和顶点7。然后，将每个目标顶点的状态进行初始化，得到初始状态。假设初始状态为。之后，针对每个目标顶点，重新计算每个目标顶点的变化后状态。针对顶点3，顶点3没有通过入边连接的邻居顶点，因此，顶点3的状态还是。针对顶点5，与顶点5通过入边连接的邻居顶点有顶点2、顶点4、顶点7和顶点3。计算每个邻居顶点为顶点5所贡献的状态量，顶点2为16，顶点3为，顶点4为17，顶点7为，这样，从多个邻居顶点所贡献的状态量中选择最小的状态量，作为顶点5的变化后状态，即，16。同理可得，顶点6的变化后状态为21，顶点7的变化后状态为19。

S106：在接收到目标任务请求后，基于所述目标顶点的变化后状态以及所述原始有向图中未受到所述待变化边影响的顶点的状态，执行所述目标任务。

在本说明书实施例中，在确定出调整后有向图中目标顶点的变化后状态之后，接收目标任务请求，在接收到目标任务请求之后，可以基于调整后有向图中目标顶点的变化后状态以及原始有向图中未受到待变化边影响的顶点的状态，执行目标任务。其中，目标任务可以包括：商品推荐、道路预测、金融风险检测、页面推荐等任务。

需要说明的是，在调整后有向图中未受到待变化边影响的顶点的状态与原始有向图中未受到待变化边影响的顶点的状态相同。

当目标任务为道路预测时，可以根据用户发送的目标任务请求中携带的起始地址和目的地址，从调整后有向图中查找出与所述起始地址匹配的顶点，作为第一匹配顶点，从调整后有向图中查找出与所述目的地址匹配的顶点，作为第二匹配顶点。根据第一匹配顶点的状态以及第二匹配顶点的状态，在调整后有向图中搜索出从第一匹配顶点到第二匹配顶点的最短顶点链路。根据搜索出的最短顶点链路，确定出与最短顶点链路相匹配的最短路径，将确定出的最短路径返回给用户，并在用户的终端设备上进行展示。

当目标任务为页面推荐时，调整后有向图中每个顶点表示页面，顶点与顶点之间的边表示页面与页面之间引用关系，顶点的状态表示页面的重要程度。接收用户发送的目标任务请求，根据目标任务请求中携带的检索关键词，在调整后有向图中查找与该检索关键词相匹配的各页面顶点。然后，根据各页面顶点对应的重要程度，对各页面进行排序，得到页面序列，并按照该页面序列，向用户展示与该检索关键词相匹配的各页面。

通过上述图1所示的方法可见，本说明书根据原始有向图中涉及的待变化边的起始顶点的状态以及受待变化边影响的目标顶点的入边连接的邻居顶点的状态中的至少一种状态，确定出通过待变化边对原始有向图进行调整后目标顶点的变化后状态，并基于目标顶点的变化后状态以及原始有向图中未受待变化边影响的顶点的状态，执行目标任务。在此方法中，在有向图调整之后，无需重新计算调整后有向图中所有顶点的状态，而是基于原始有向图中至少部分顶点的状态，只对受到插入边或删除边影响的顶点的状态进行重新计算，不受影响的顶点的状态无需重新计算，从而减少了冗余计算。

以上为本说明书实施例提供的任务执行方法，基于同样的思路，本说明书还提供了相应的装置、存储介质和电子设备。

图8为本说明书实施例提供的一种任务执行装置的结构示意图，所述装置包括：

获取模块801，用于获取原始有向图，确定所述原始有向图中每个顶点的状态，以及确定所述原始有向图中各顶点之间需要变化的边，作为待变化边，其中，针对所述原始有向图中包含的每个顶点，该顶点通过入边或出边与邻居顶点相连，该顶点的入边是以邻居顶点为起始顶点出发指向该顶点的边，该顶点的出边是以该顶点为起始顶点出发指向邻居顶点的边；

查找模块802，用于基于所述待变化边的指向，从所述原始有向图中查找出受所述待变化边影响的顶点，作为目标顶点；

确定模块803，用于根据所述待变化边的起始顶点的状态以及所述目标顶点通过入边连接的邻居顶点的状态中的至少一种状态，确定出通过所述待变化边对所述原始有向图进行调整后所述目标顶点的变化后状态；

执行任务模块804，用于在接收到目标任务请求后，基于所述目标顶点的变化后状态以及所述原始有向图中未受到所述待变化边影响的顶点的状态，执行所述目标任务。

可选地，所述获取模块801，还用于确定高速缓存区的大小以及存储一个顶点的状态所需占用的存储空间；基于所述高速缓存区的大小以及所述存储空间，确定所述高速缓存区所能存储的顶点数量；将所述原始有向图中各顶点按照度数从大到小排序，得到顶点序列；按照度数从大到小的顺序，从所述顶点序列中确定出所述顶点数量的顶点，并将确定出的顶点的状态保存于所述高速缓存区中。

可选地，所述确定模块803，具体用于基于所述待变化边的指向，确定出所述待变化边的起始顶点；基于所述起始顶点的状态，确定出在所述原始有向图中所述起始顶点通过所述起始顶点的每条出边所传递的状态变化量，作为第一状态变化量；在所述原始有向图中查找出受所述起始顶点的每条出边影响的其他顶点；从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到所述其他顶点的抵消后状态；基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；根据所述调整后有向图中所述起始顶点的状态，确定出在所述调整后有向图中所述起始顶点通过所述起始顶点的出边所需传递的状态变化量，作为第二状态变化量；基于所述第二状态变化量，对所述调整后有向图中所述目标顶点的抵消后状态进行更新，得到所述目标顶点的变化后状态。

可选地，所述确定模块803，具体用于基于所述起始顶点通过所述起始顶点的每条出边所传递的状态变化量，确定出通过所述起始顶点的每条出边所需传递的负增量；以所述负增量为传递参数，沿所述起始顶点的每条出边进行传递，以从所述其他顶点的状态中抵消所述起始顶点传递所述第一状态变化量时所增加的状态量，得到抵消后状态。

可选地，所述确定模块803，具体用于基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；若所述待变化边为待插入边，基于所述待变化边的起始顶点的状态以及所述待变化边的权重，确定出所述调整后有向图中所述待变化边的目的顶点的预计状态，所述目的顶点是指所述待变化边从所述起始顶点出发所指向的顶点，所述目的顶点为所述目标顶点中的一个顶点；若所述预计状态小于所述原始有向图中所述待变化边的目的顶点的状态，则将所述预计状态确定为在所述调整后有向图中所述目的顶点的状态，并基于所述目的顶点的预计状态，对所述调整后有向图中所述目标顶点中除所述目的顶点之外的其他目标顶点的状态进行更新，得到变化后状态；若所述预计状态不小于所述原始有向图中所述待变化边的目的顶点的状态，则保持所述目的顶点在所述原始有向图中的状态以及无需对所述调整后有向图中除所述目的顶点之外的其他目标顶点的状态进行更新。

可选地，所述确定模块803，具体用于基于所述待变化边对所述原始有向图进行调整，得到调整后有向图；若所述待变化边为待删除边，对所述调整后有向图中所述目标顶点的状态进行初始化，得到初始状态；针对所述目标顶点通过入边连接的每个邻居顶点，基于该邻居顶点的状态以及该邻居顶点与所述目标顶点之间的入边的权重，确定出该邻居顶点为所述目标顶点所贡献的状态量，作为该邻居顶点对应的状态量；从每个邻居顶点对应的状态量中选择出最小的状态量，作为在所述调整后有向图中所述目标顶点的变化后状态。

可选地，所述待变化边包括：待插入边和/或待删除边。

本说明书还提供了一种计算机可读存储介质，所述存储介质存储有计算机程序，所述计算机程序被处理器执行时可用于执行上述图1提供的任务执行方法。

基于图1所示的任务执行方法，本说明书实施例还提供了图9所示的电子设备的结构示意图。如图9，在硬件层面，该电子设备包括处理器、内部总线、网络接口、内存以及非易失性存储器，当然还可能包括其他业务所需要的硬件。处理器从非易失性存储器中读取对应的计算机程序到内存中然后运行，以实现上述图1所述的任务执行方法。

当然，除了软件实现方式之外，本说明书并不排除其他实现方式，比如逻辑器件抑或软硬件结合的方式等等，也就是说以下处理流程的执行主体并不限定于各个逻辑单元，也可以是硬件或逻辑器件。

在20世纪90年代，对于一个技术的改进可以很明显地区分是硬件上的改进（例如，对二极管、晶体管、开关等电路结构的改进）还是软件上的改进（对于方法流程的改进）。然而，随着技术的发展，当今的很多方法流程的改进已经可以视为硬件电路结构的直接改进。设计人员几乎都通过将改进的方法流程编程到硬件电路中来得到相应的硬件电路结构。因此，不能说一个方法流程的改进就不能用硬件实体模块来实现。例如，可编程逻辑器件（Programmable Logic Device, PLD）（例如现场可编程门阵列（Field Programmable Gate Array，FPGA））就是这样一种集成电路，其逻辑功能由用户对器件编程来确定。由设计人员自行编程来把一个数字系统“集成”在一片PLD上，而不需要请芯片制造厂商来设计和制作专用的集成电路芯片。而且，如今，取代手工地制作集成电路芯片，这种编程也多半改用“逻辑编译器（logic compiler）”软件来实现，它与程序开发撰写时所用的软件编译器相类似，而要编译之前的原始代码也得用特定的编程语言来撰写，此称之为硬件描述语言（Hardware Description Language，HDL），而HDL也并非仅有一种，而是有许多种，如ABEL（Advanced Boolean Expression Language）、AHDL（Altera Hardware Description Language）、Confluence、CUPL（Cornell University Programming Language）、HDCal、JHDL（Java Hardware Description Language）、Lava、Lola、MyHDL、PALASM、RHDL（Ruby Hardware Description Language）等，目前最普遍使用的是VHDL（Very-High-Speed Integrated Circuit Hardware Description Language）与Verilog。本领域技术人员也应该清楚，只需要将方法流程用上述几种硬件描述语言稍作逻辑编程并编程到集成电路中，就可以很容易得到实现该逻辑方法流程的硬件电路。

控制器可以按任何适当的方式实现，例如，控制器可以采取例如微处理器或处理器以及存储可由该（微）处理器执行的计算机可读程序代码（例如软件或固件）的计算机可读介质、逻辑门、开关、专用集成电路（Application Specific Integrated Circuit，ASIC）、可编程逻辑控制器和嵌入微控制器的形式，控制器的例子包括但不限于以下微控制器：ARC 625D、Atmel AT91SAM、Microchip PIC18F26K20 以及Silicone Labs C8051F320，存储器控制器还可以被实现为存储器的控制逻辑的一部分。本领域技术人员也知道，除了以纯计算机可读程序代码方式实现控制器以外，完全可以通过将方法步骤进行逻辑编程来使得控制器以逻辑门、开关、专用集成电路、可编程逻辑控制器和嵌入微控制器等的形式来实现相同功能。因此这种控制器可以被认为是一种硬件部件，而对其内包括的用于实现各种功能的装置也可以视为硬件部件内的结构。或者甚至，可以将用于实现各种功能的装置视为既可以是实现方法的软件模块又可以是硬件部件内的结构。

上述实施例阐明的系统、装置、模块或单元，具体可以由计算机芯片或实体实现，或者由具有某种功能的产品来实现。一种典型的实现设备为计算机。具体的，计算机例如可以为个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任何设备的组合。

为了描述的方便，描述以上装置时以功能分为各种单元分别描述。当然，在实施本说明书时可以把各单元的功能在同一个或多个软件和/或硬件中实现。

本领域内的技术人员应明白，本说明书的实施例可提供为方法、系统、或计算机程序产品。因此，本说明书可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本说明书可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

本说明书是参照根据本说明书实施例的方法、设备（系统）、和计算机程序产品的流程图和／或方框图来描述的。应理解可由计算机程序指令实现流程图和／或方框图中的每一流程和／或方框、以及流程图和／或方框图中的流程和／或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和／或方框图一个方框或多个方框中指定的功能的装置。

这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和／或方框图一个方框或多个方框中指定的功能。

这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和／或方框图一个方框或多个方框中指定的功能的步骤。

在一个典型的配置中，计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器(RAM)和/或非易失性内存等形式，如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括暂存电脑可读媒体(transitory media)，如调制的数据信号和载波。

还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

本领域技术人员应明白，本说明书的实施例可提供为方法、系统或计算机程序产品。因此，本说明书可采用完全硬件实施例、完全软件实施例或结合软件和硬件方面的实施例的形式。而且，本说明书可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

本说明书可以在由计算机执行的计算机可执行指令的一般上下文中描述，例如程序模块。一般地，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本说明书，在这些分布式计算环境中，由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中，程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于系统实施例而言，由于其基本相似于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

以上所述仅为本说明书的实施例而已，并不用于限制本说明书。对于本领域技术人员来说，本说明书可以有各种更改和变化。凡在本说明书的精神和原理之内所作的任何修改、等同替换、改进等，均应包含在本说明书的权利要求范围之内。

**说 明 书 附 图**



图1



图2



图3



图4



图5



图6



图7



图8



图9