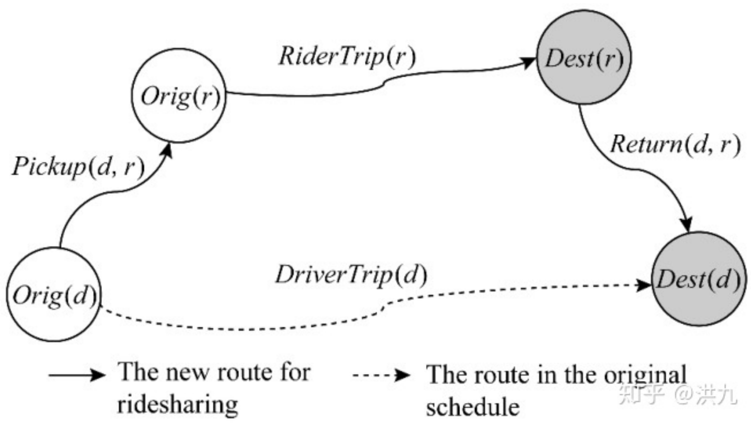
二部图应用

Assign	
Status	
create	
source	https://www.zhihu.com/collection/618549038
source2	

这一部分介绍两个场景，一个是二部图最优匹配在多对多拼车算法中的应用，另一个是在推荐算法中的应用，最后简单介绍二部图在计算广告中的应用

二部图最优匹配在多对多拼车算法中的应用



图片中相关符号定义：

$Orig(d)$：司机d的出发位置

$Dest(d)$：司机d的目的地

$Orig(r)$：乘客r的出发位置

$Dest(r)$：乘客r的目的位置

$Pickup(d,r)$：接驾距离，距离也可以表示为时间

$RiderTrip(r)$：送驾距离，也可表示为时间

$DriverTrip(d)$：司机原来的行驶路径

$Return(d,r)$：司机返回原始目的地的距离

$Detour(d,r)$：由于接送新乘客r所产生的绕路距离，计算公式如下

$Detour(d,r) = Pickup(d,r) + RiderTrip(r) + Return(d,r) - DriverTrip(d)$

$Price(d,r)$：乘客r需要支付给司机d的车费，计算公式如下：

$Price(d,r) = RiderTrip(r) + Detour(d,r)$

接下来描述一些时间相关定义：

$DepartureTime(d)$：司机d出发的时间

$ArrivalTime(d)$：司机d的最晚到达时间

$departureTimeMin(r)$：乘客r最早出发时间

$departureTimeMax(r)$：乘客r最晚出发时间

$feedbackTime(D,R)$：拼车结果反馈给司机和乘客的时间

基于实际的情况，这个问题有如下约束：

- 司机要在乘客的出发时间范围内到达乘客的出发位置

  $departureTimeMin(r) \le DepatureTime(d) + Pickup(d,r) \le departureTimeMax(r)$

- 司机在完成乘客的拼车订单后，还能再自己的最晚到达时间之前到达司机自身的目的地，保证司机形成不被耽误

  $DepatureTime(d) + Price(d,r) + RiderTrip(r) + Reture(d,r) \le ArrivalTime(d)$

- 乘客支付的车费必须小于他的提出的最大拼车费用

$$Price(d,r) \le maxPrice(r)$$

- 平台视角希望所有司机产生的绕路距离最小

$$Min \sum_{i=1}^{n} Detour(d_i, d_j)$$

有了如上定义和约束后，提出了如下**匹配策略**。

匹配分为两个阶段，第一个阶段是单乘客对多司机的匹配，目的是对于特定乘客r，选择n个合适的司机；第二个阶段是多乘客与多司机的最优组合匹配

步骤一：单乘客对多司机的匹配

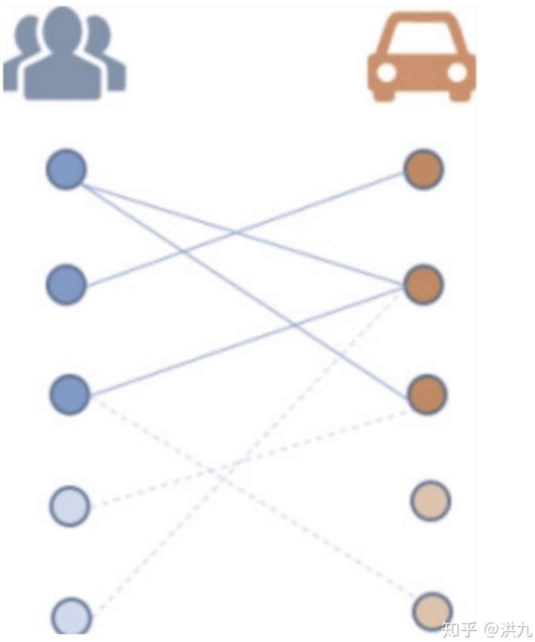
有如下要求：

- 司机的出发时间$DepartureTime(d)$必定早于乘客的最晚出发时间$depatureTimeMax(r)$
- 需要$Pickup(d,r)+Return(d,r)$的距离尽量小

使用一些基于方法和规则，可以得到乘客r_i的候选司机集合$D_i=\{d_1,d_3\}$

步骤二：多乘客对多司机的最优匹配

目的是让所有司机拼车绕路的距离达到最小，所以基于上一步得到的候选集合，以$Detour(d,r)$为节点之间的边权重，可以得到一个二部图



利用KM算法即可求解得到最优的匹配。

*为解决在稀疏矩阵中KM算法效率不高的问题，将整个矩阵拆分成为多个子矩阵并分别做KM匹配的方法

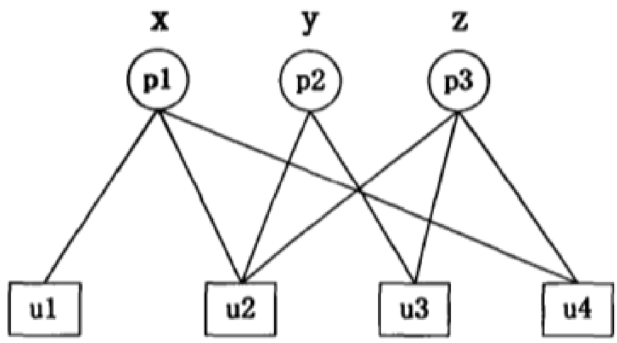
二部图在推荐算法中的应用

假定系统中有m个用户和n个物品，用户合集用$U=\{u_1,u_2,...,u_m\}$表示，物品集合用$P=\{p_1,p_2,...,p_n\}$表示，依据此构建一个$m*n$的用户-物品关系矩阵$\{a_{i,j}\}$，$i\in\{1,...m\}$，$j\in\{1,...,n\}$。矩阵元素的取值公式如下

$$a_{i,j}=\{^{1,u_i choose\ p_j}_{0,u_i\ notchoose\ p_j}$$

这样可以得到一个用户-物品矩阵可以被抽象成为一个二部图，同一个物品从不同用户处获取的初始权重不一样的。

初始资源的流向是物品到用户



对任意商品p_i，其被分配的输出资源值为$f(p_i) \geq 0$，用户合集和物品合计表示如上

首先，物品的初始资源从物品节点流向用户节点，流动完成后，对于任意的用户u_i，其得到的资源值计算公式如下：

$$f(u_i) = \sum_{j=1}^{n} \frac{f(p_j)*a_{i,j}}{k(p_j)}$$

其中$k(p_j)$表示二部图中物品类节点p_j对应的度

在资源流向用户后，需要回流到物品，对应的公式如下：

$$f^{'}(p_j) = \sum_{i=1}^{m} \frac{f(u_i)*a_{i,j}}{k(u_i)}$$

上述公式中的$f(u_i)$可以被第一个公式代替，综合两个公式可以得到物品资源更新的表达式：

$$f^{'}(p_J) = \sum_{j=1}^{n} \omega_{J,j} f(p_j)$$

令矩阵$W = \{\omega_{i,j}\}_{m*n}$，则基于二部图的推荐算法资源分配流程可以有如下矩阵表示：

$$f^{'} = Wf$$

其中，f表示推荐系统中n个物品的n维初始资源的列向量，而$f^{'}$表示n个物品的n维最终资源列向量。进行推荐时，将目标用户未选择过的物品集合按照对应的$f^{'}$中的最终资源进行排序，按照一定的规则生成推荐结果提供给用户。

二部图在计算广告中的应用

在接入AdX时，一次广告请求可能含有多个广告位，但是我们不能拿相同广告主的广告进行重复竞价。

一个简单的例子，假设有A，B两个广告位，且有a，b，c三个广告主的广告参与竞价。如果我们拿a广告去参与A广告位的竞价，a就不能被拿去参与啊B和C的竞价了。

基于这样的限制，整个问题可以被抽象成为一个带权值的二部图，可以用KM算法求解