

西安电子科技大学 2020 年数学建模校内赛

承 诺 与 产 权 转 让 书

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们同意将参赛论文以及支撑材料中的所建模型、算法以及程序产权归属西安电子科技大学所有。2020 年数学建模校内赛竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号是（从 A/B/C 中选择一项填写）： B

参赛报名队号为 20B560

队长所属学院（请填写完整的全名）： 人工智能学院

参赛队员姓名与学号（此处附电子签名）：

1. 闫浩霖 19200300129 电子签名： 

2. 凌少鹏 19200300130 电子签名： 

3. 廖依诺 19200100077 电子签名： 

日期： 2020 年 5 月 3 日

西安电子科技大学 2020 年大学生数学建模校内赛

评 阅 专 用 页

	评阅人 1	评阅人 2	评阅人 3	总评
成绩				

基于时段流量模型和图网络模型的洗手池规划配置

摘要

本文通过在对学生进行问卷调查,收集相关数据信息的基础上,分区域建立了校园主要地点的时段人流量变化模型,并使用计算机模拟病毒传播情况,研究病毒传播机理,建立了洗手率决策模型,模拟计算出人均洗手率,确定了新增洗手池的数目,根据区域地理特征建立图网络模型确定其分布,进行相关对比分析,并就资源使用,防控疫情的角度评估模型,最后根据模拟结果向后勤部门提出相关建议。

针对问题一,根据校区地理特征将校区分区域研究,主要研究在早上 6:30 至 8:30 期间学生从宿舍到食堂、教学楼的人员流动特征,在假设学生离开宿舍的时间分布满足正态分布的基础上,根据实际距离、人均步行速率等因素推算出食堂与教学楼的人流量变化情况,并就人流量变化情况分析各个区域道路拥挤的时段,在该时段通过计算机模拟病毒的传播情况,结果显示在 6:50 至 7:30 (以丁香区域为例),区域人流量变化快,人群相互接触频繁,推测是病毒传播的主要时间。

针对问题二,在人流量变化的基础上,根据一系列洗手决策的规则,使用 Monte Carlo 模拟出在现有洗手池资源情况下的人均洗手率,考虑到洗手对病毒传播的阻隔作用,对问题一的病毒传播模型进行调整,以丁香区域为例模拟出在早晨人均洗手率为 15%的情况下病毒传播情况,结果显示有 16 人感染。而在增设洗手池的情况下模拟计算出的洗手率得到提高,并有效阻止了病毒的传播,依此证明了增设洗手池的必要性。

针对问题三,通过对校园区块的主要地点抽象出图网络模型,我们可以得到各个地点区域的人流量在一定时段内随着时间变化的曲线关系。以各个地点在某一时刻的人流量和其与其他地点的最短图模型距离为参考指标,计算出在各个地点增设洗手池的必要性权重分配变化曲线,为流动洗手池的流动和固定洗手池的配置提供参考。在考虑错时作息和线上线下混合教学的情况下,调整人流量网络模型的参数,继而分析得出增设洗手池的配置和规模,从而减缓校方的管控压力,有利于控制校园的疫情。

针对问题四,在能源消耗角度看,根据模型计算出洗手池规模,使用洗手率决策规则模拟计算出人均洗手率,假设人均每次洗手用水 550ml,则根据最终统计,每天该时段的总水资源使用量达 3.31 立方米,根据西安市居民平均水价估计水费约为 10 元;从防疫新冠的有效性角度看,为研究增设洗手池对病毒传播是否有阻隔作用,使用第一问的病毒传播模拟试验做对照分析,最终结果显示增设洗手池能够有效阻隔病毒传播,由这两个方面确定我们的模型是合理有效的。

问题五中,我们根据模型结果分析结论,向校后勤部写信提出有关错时作息下各地点洗手点的数量规模与位置等建议。

【关键词】：时段人流量模型、计算机模拟、图网络模型、对照试验

一、问题重述

1.1 背景资料与条件

新型冠状病毒疫情给复工复产带来了不小的挑战，人们采取多种预防手段来阻断病毒的传播，而其中洗手是最直接、最简单、最有效的手段之一。但是，目前仅有教学楼内有洗手资源，且其中的洗手资源也十分有限，较少的洗手资源也会导致人流聚集，不利于疫情的防控。另外，在校园活动时也少有洗手的地方。

1.2 需要解决的问题

问题一：分析我校教学楼、宿舍楼、食堂等分布特征，建立人员流动网络模型，在此基础上给出潜在的病毒感染机理分析。

问题二：评估我校目前的洗手资源、复学后人流密度，分析增加室外固定洗手池以及设立流动洗手池的必要性。

问题三：设计洗手池规模、龙头点数、流量等参数，根据学生校园内的日常活动建立模型，在满足必要的防疫要求的前提下，求出最少的洗手池的配置地点、流动性配置路线计划。考虑在错时作息以及线上线下混合教学等情形下，模型的调整方案

问题四：在设定参数下从能源消耗和预防新冠有效性两方面对模型进行评估。

问题五：根据模型分析结论，给我校后勤部门写一封建议信

二、模型假设

1. 假设校园的不同区块内的一定区域人员流动情况近似相同，可用同一个区块进行表示。
2. 假设校园的不同区块之间的路径可取区块间多条道路距离的平均值为权重。
3. 假设人员流动网络模型研究的对象是在校学生群体。
4. 假设人员流动的区域只有校园里的这些区块。
5. 假设学生离开宿舍的概率随时间呈正态分布。
6. 假设存在少部分不在食堂区块就餐的群体。
7. 假设每个洗手池有一个龙头，同时刻内仅可供一人使用。

三、主要符号说明

符号	说明
P_c	学生选择不去就餐的概率
N_i	某一区域人流量
S	洗手间某一时刻的人数
P_I	接触感染者后被感染的概率
P_0	学生选择洗手的最高概率
M	增设洗手池的数目
D_{ij}	结点 i, j 之间的最短图模型距离
$Score_i$	在某结点增设洗手池的必要性指数
\hat{P}_w	时段内最终人均洗手率

四、模型的建立与分析

4.1 问题一的模型的建立与分析

4.1.1 区域分块

假设学生的活动范围是以教学楼为中心的区块分布，以学生日常活动地点：宿舍、食堂、教学楼这三个区域作为一个区块，根据宿舍楼建筑群的分布特征将校区分为如图 1 中的红绿白三个区块。分别为丁香区（大一学生），海棠区（大二学生），竹园区（大三大四学生），将教学楼 A 至 G 分为 AB 教学区，CD 教学区，EFG 教学区，三个区域的学生均有可能在这三个教学区上课。

宿舍 \longleftrightarrow 食堂 \longleftrightarrow 教学楼



图 1 校区地图分块

4.1.2 人员流动统计

早上 6:30 至 8:30 这一时间段是学生大范围流动且具有明显流动方向规律(宿舍->食堂->教学楼)的时间段, 因此取这一时间段来进行人员流动分析。

对于各个宿舍区域, 假设在 6:30 至 8:30 时间段内, 离开宿舍的学生人数变化符合正态分布

$$N_1(t) = P \times \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{(t-KT)^2}{2\sigma^2}\right)}$$

其中 P 为宿舍总人数, KT 为人流高峰的时刻, $\sigma = 20$ 。设学生选择不在于食堂就餐直接去教学楼的概率为 P_c , 从宿舍到食堂的距离为 $D_{apt-res}$, 平均步行速度为 $\bar{v} = 5 \text{ km/h}$, 则时段内进入食堂的人数变化 $N_2(t)$ 为:

$$N_2(t) = N_1\left(t - \frac{D_{apt-res}}{\bar{v}}\right) \times (1 - P_c)$$

设在食堂吃饭的人均用时为 T_{eat} , 从食堂到达某教学区域的距离为 $D_{res-class}$, 宿舍到该教学区的距离为 $D_{apt-class}$ 则时段内该教学区域的人数变化 $N_3(t)$ 为:

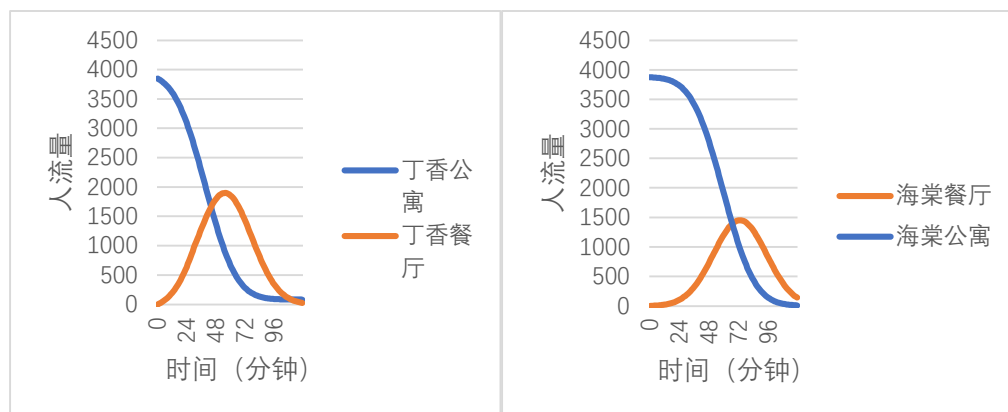
$$N_3(t) = \sum \left(N_1\left(t - \frac{D_{apt-class}}{\bar{v}}\right) \times P_c + N_2\left(t - T_{eat} - \frac{D_{res-class}}{\bar{v}}\right) \right)$$

根据调查问卷估计三个区域的 P_c, T_{eat} 如表 1

表 1 三个区域的调查结果

区域	P_c	T_{eat}	KT(分钟)
丁香区	0.1	30	60
海棠区	0.2	25	40
竹园区	0.25	27	48

三个区域的食堂、宿舍至三个教学区域的距离信息见附录[1], 计算出各个地点的人流量变化如图 2



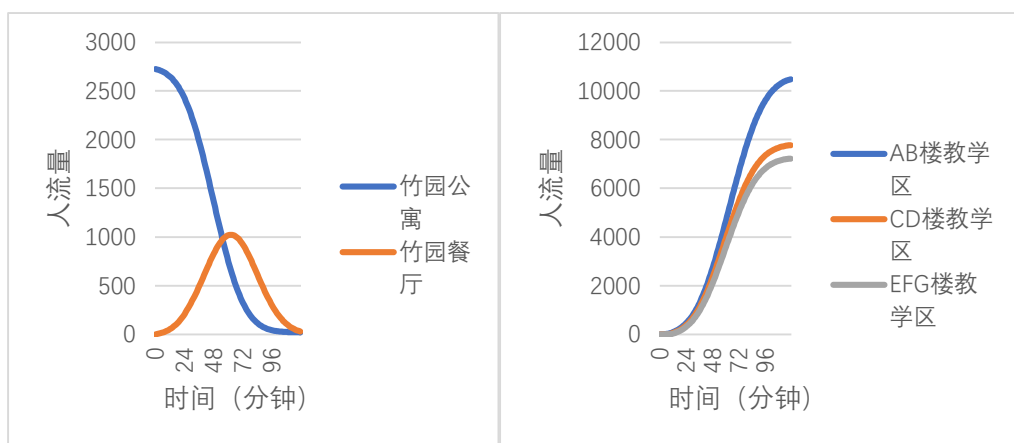


图 2 各个地点人流量变化统计

由图 1 可知, 三个食堂的平均人流量高峰期在 7:18 至 8:06, 平均食堂最高人流量可达约 1500 人 (丁香 1964, 海棠 1493, 竹园 1090)。对于三个教学楼区域, AB 楼区域人流量最大, 在 8:18 时达到最高约 12000 人。

三个区域的每时刻离开宿舍的人数统计如图 3

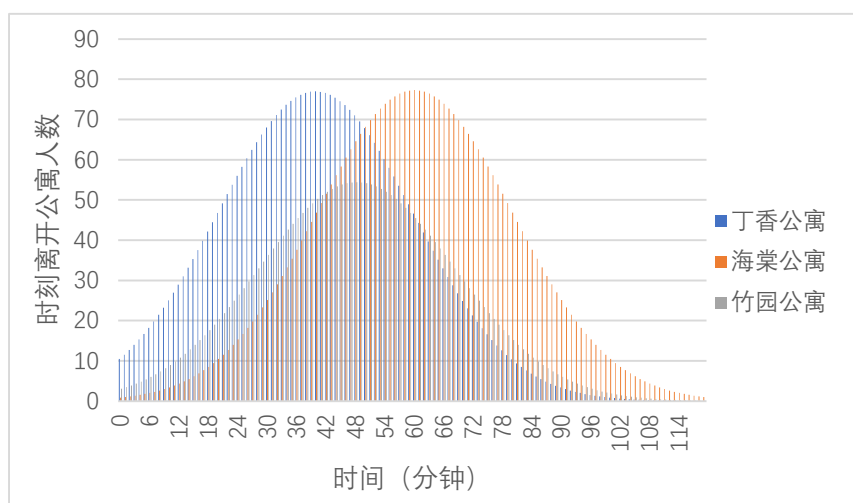


图 3 每时刻离开宿舍人数统计

分析可知丁香区域内, 在 6:50 至 7:30 时离开宿舍的人数最多, 可以推知此时区域内道路人流密度最大。同理推出其他区域如表 2

表 2 三个区域内道路人流密度最大的时间段统计

区域	丁香	海棠	竹园
道路拥挤时段	6:50~7:30	7:24~7:48	7:06~7:36

每时刻到达三个教学区域的人数统计如图 4

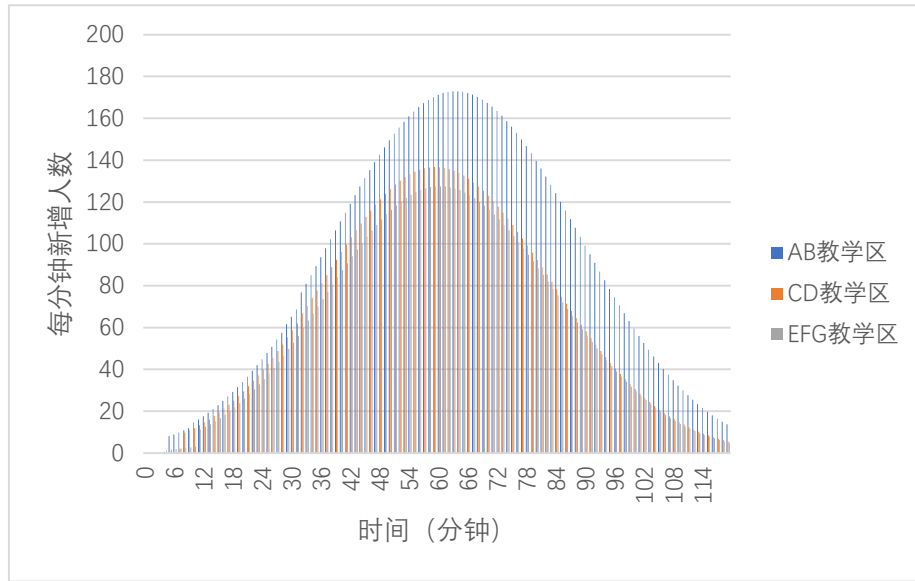


图 4 三个教学区每时刻新增人数

分析知在三个区域的道路拥挤时间约在 7:30 至 7:54 之间。AB 区域的道路拥挤程度要显然大于其他两个区域。

4.1.3 潜在病毒感染机理分析

根据表 2，已知各区域道路拥挤的时段，以丁香区域为研究对象，丁香区域内，在 6:50 至 7:30 时区域内人员流动最多，食堂人流量最大，人群间相互接触的次数最频繁，以此作为模拟病毒传播的人员流动数据。假设接触一个无症状病毒携带者被感染的概率为 P_i ，在该时段内某学生接触人数为 A ，区域总人流量为 N ，其中无症状感染者人数为 m ，则该学生在该时段内被感染的概率为：

$$P_I = P_i^{(A \times \frac{m}{N})}$$

假设 $m = 1$ ，每次在一定范围内随机模拟接触人数 A ，通过 Monte Carlo 模拟，计算出时段内感染者的数量为 24 人，由此推测，在刚复学的每天早上，由于在区域道路上和在食堂内的人流量较大，人员流动速率快，人群间相互接触的次数频繁，导致了病毒的传播。

4.2 问题二模型的建立与分析

4.2.1 模拟模型的建立

设某区域内共有洗手池 M 个，每个洗手池有一个龙头，同一时刻只可供一人使用。在该地点人均洗手的最高概率为 P_0 ，在某时刻已有洗手人数为 S ，该区域人流量变化数据由第一个问题的模型提供，考虑到个体差异性，某位学生在这一点的洗手概率为：

$$P_w = P_0 - u$$

其中 $0 \leq u \leq \frac{P_0}{2}$ ， u 为随机干扰项。则模拟每个个体的决策过程如下：

1. 根据概率 P_w 决定是否洗手，若不洗手则离开，不进行以下流程。
2. 对当前洗手间的拥挤程度进行评价，分为不拥挤、有点拥挤、拥挤。
3. 根据拥挤程度调整概率 P_w

评价标准与概率调整如下：

根据洗手间的当前时刻人数 S 评价拥挤程度

若 $S \leq M$ 即有空余洗手池则评价为不拥挤， P_w 随机增加但不超过 P_0 。

若 $M \leq S \leq M + n$ ，评价为有点拥挤， n 为可接受的等待人数， P_w 不变。

若 $S \geq M + n$ ，评价为拥挤， $P_w = P_w - \left(\frac{S}{n + M} - 1 \right)$ 。

关注研究丁香区域的学生在三个教学区域，丁香食堂的人均洗手率，模拟计算在这四个地方的洗手率 $\bar{P}_{wi}, i = 0, 1, 2, 3$ ，则在时段内最终人均洗手率为：

$$\hat{P}_w = 1 - \prod_{i=0}^3 (1 - \bar{P}_{wi})$$

4.2.2 模拟并分析结果

通过问题一的模型，可以计算出三个区域的学生流动到三个教学区的人数占三个区域的总人数比例，根据比例将三个教学区域的洗手池数分配给丁香区，调查得丁香食堂的洗手池为 12 个，统计如表 3

表 3 丁香区的洗手池数统计

区域	AB	CD	EFG	丁香	总计
数目	10	6	3	12	31

在现有的洗手池数量的基础上，设 W 为在丁香区域的食堂和宿舍附近新增洗手池数，将 W 分通过对四个地点的 Monte Carlo 模拟，最终计算出人均洗手率的变化情况如图 5

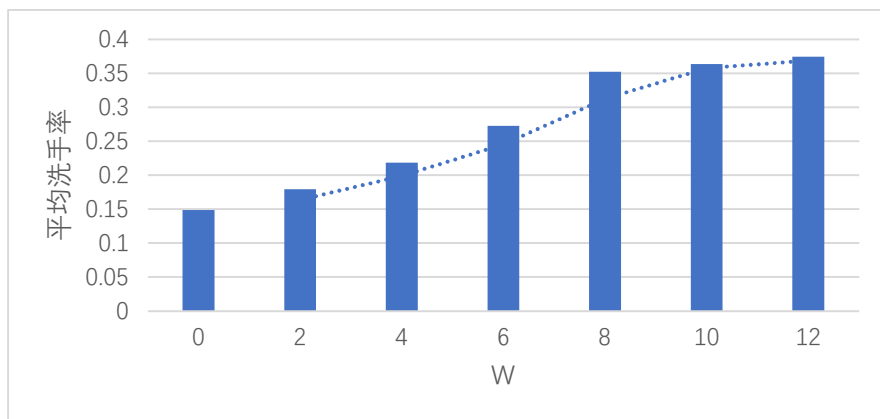


图 5 人均洗手率变化趋势

分析图 1 可知，在现有的洗手池数量下，人均洗手率只有约 15%，那么在该时段的大规模人员流动的情况下，粗略估计有 $R = 85\%$ 的学生都没有洗手，调整第一问的病毒传播模拟模型中的感染概率为：

$$P_I = P_i^{(A \times \frac{m}{N} \times R)}$$

假设该区域有一人感染，多次模拟后结果如图 6

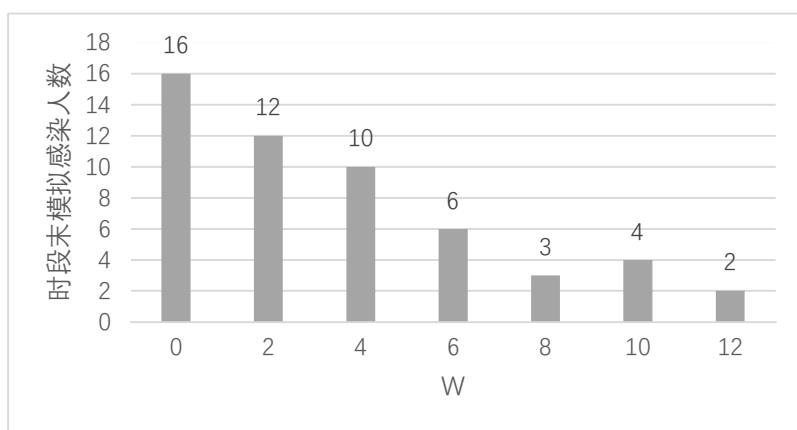


图 6 模拟感染人数变化

结果显示，若不增加该时段丁香区域最终会有 16 人感染，考虑到中午，下午的第二三次人流大规模活动，这将造成严重的后果。

因此必须增设新的洗手池。推荐在丁香区域增加 12 个洗手池。

4.3 问题三的模型建立

根据各个地点的分布特征，使用图论方法计算得出在每一个地点增设洗手池的必要性大小，在计划增设洗手池数目固定的情况下，以此必要性大小为权重进行分配。

4.3.1 校园区块图模型的建立

我们将校园区块的结点和主要道路进行抽象，以一个区块为结点，鉴于校园内道路曲折复杂，这里简化图模型的建立，取两点之间多条道路的平均值为图模型中两点之间边的权重，最终建立图模型示意图 7 如下：

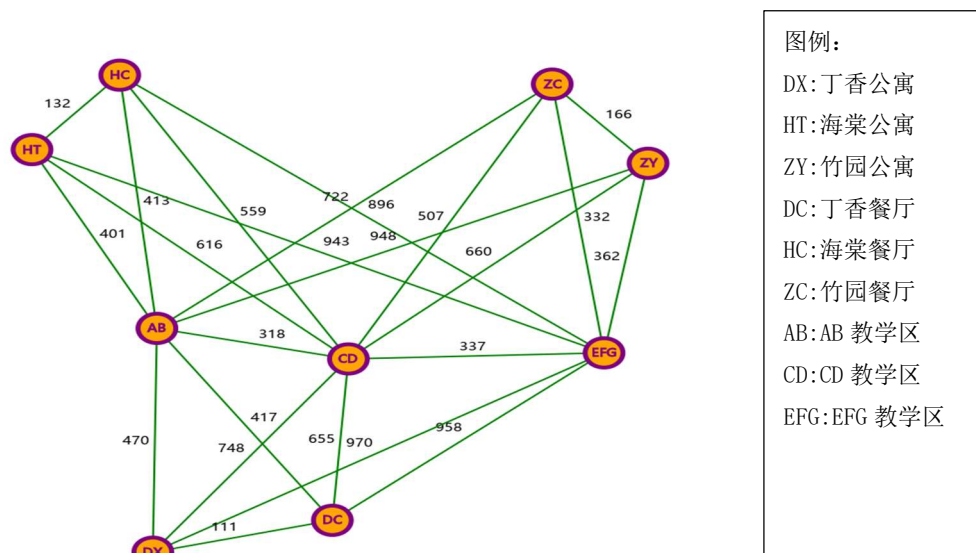


图 7 图模型示意图

4.3.2 求解各点评价价值

图中每个结点的人流数目在一定时间段内是关于时间 t 的函数，这里我们同样以早晨 6:30-8:30 这个时间段为研究对象，为确定在哪些地点增设洗手池效果最佳，我们对每一个结点计算评价价值。

$$Score_i(t) = \sum_j^n \frac{N_j(t)}{1 + D_{ij}}$$

其中， $N_j(t)$ 为结点 j 的人流量， D_{ij} 为 结点 i 到结点 j 之间的图模型距离 (以 100 米为单位)。如下图 8 是各个结点评价价值的变化曲线。

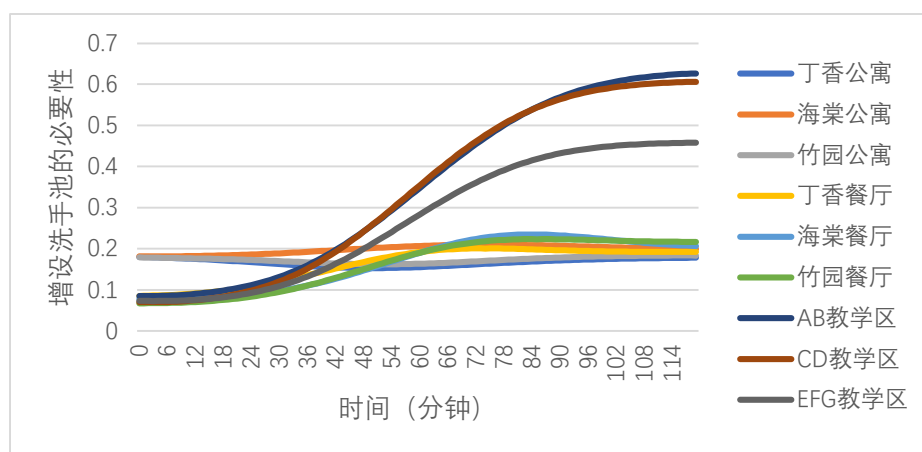


图 8 各地点增设洗手池的必要性变化曲线

通过这幅图，我们可以直观地分析出，整个时段内各个宿舍出口区域有较小的升高，食堂区域在时段中部有轻微的波动，整体无大的起伏，时段后部，教学区域设立洗手池的必要性达到最高。另外我们可以发现教学区评价值的峰值要远远地高于宿舍和食堂区域的评价值峰值，这是因为教学区的人流量来源于各个宿舍区，是整个校园区块的中心，是人流量最终汇聚的地方。而评价值计算函数则与每一点的人流量有较大依赖的关系。

4.3.3 洗手池数目的分配

针对流动洗手池，通过定量分析，基于上图，根据各个结点之间的评价值比例关系，我们可以对各个地点进行合理的流动洗手池数目分配，例如设置流动水车，在校园内不同的时段流动供水，在 6:30 至 7:15, 学生主要聚集在宿舍，食堂附近，可在道路周边设置流动洗手池，在 7:35 左右时开始，可以将流动水资源向三个教学区转移，由图知 8:00 前后，是教学区域尤其是 AB 楼附近人流量变化最多的时段，因此在教学区应增加更多的洗手资源。

针对固定洗手池，设各个地点之间的评价值比例为 W_i 则：

$$W_i = \frac{\int_0^T Score_i(t) dt}{\sum_{j=1}^N \int_0^T Score_j(t) dt}$$

计算得到各个地点之间的评价值对比关系，如下图

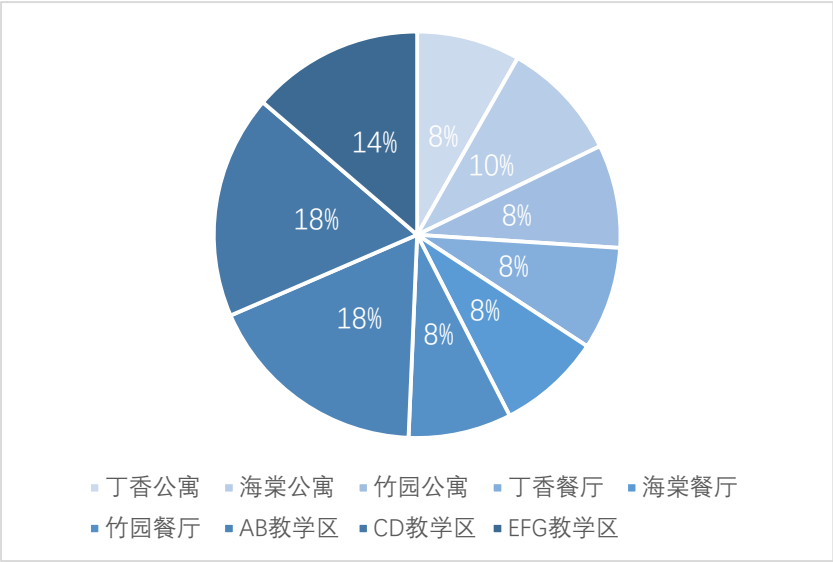


图 9 各个地点之间的评价值对比图

通过第二问模型的模拟计算，得到满足要求的丁香区域的 M 值，根据图 9 比例推算继而得到各地设立固定洗手池的数目，结果如下表 4：

地点	新增洗手池数
丁香公寓	6
海棠公寓	7
竹园公寓	6
丁香餐厅	6
海棠餐厅	6
竹园餐厅	6
AB 教学区	13
CD 教学区	13
EFG 教学区	10

为减缓我校目前现有的洗手池资源承载压力，同时降低对增设洗手池资源的消耗，下面考虑错时作息和线上线下错峰上课。

假设初始离开宿舍的学生均分为 $q=3$ 个批次，调整人流量模型 $N_i(t)$ ，分 q 次模拟每一批次的学生人流情况，得到人流量变化曲线

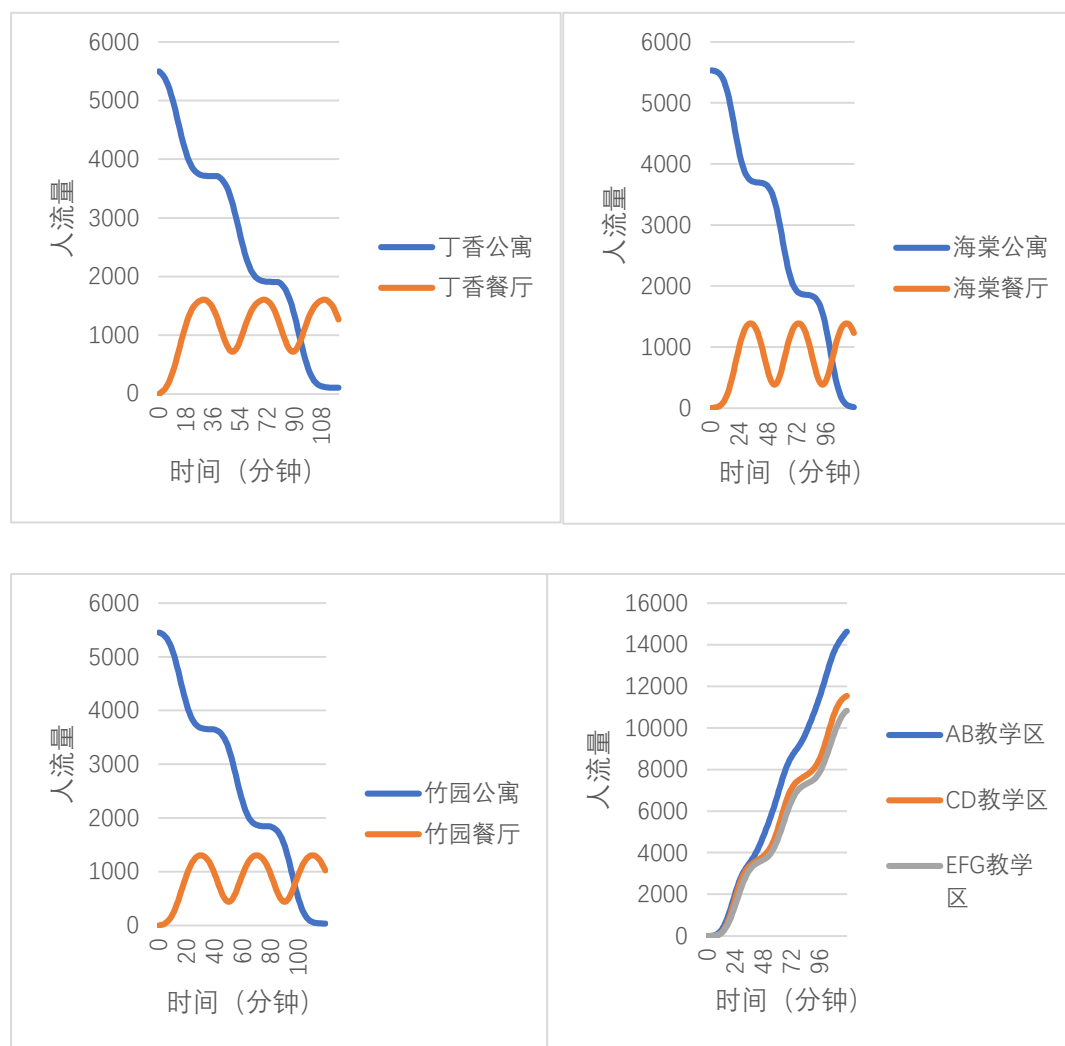


图 10 分批次各地点人流量变化图

由图 10 对比问题一的人流量变化图，可知通过分批次作息，可有效降低区域人流量的峰值，采用错时出行，极大缓解了道路交通压力，降低了人流密度。

在分批次人流量变化趋势下，重复前两问的模型模拟计算，得到各地增设洗手池数目如下表 5：

地点	新增洗手池数
丁香公寓	5
海棠公寓	5
竹园公寓	5
丁香餐厅	4
海棠餐厅	4
竹园餐厅	4
AB 教学区	9
CD 教学区	8
EFG 教学区	5

由表中我们可以看到，分批次错时作息使得需要增设的洗手池数目得到降低。因此减少了我校的财力支出，且更加方便学校的管理与调控。

增设上表数量的洗手池和不增设洗手池的人均洗手率对比如下表 6

区域	增设下人均洗手率	不增设下人均洗手率
丁香区域	35%	17%
海棠区域	36.70%	16.10%
竹园区域	37.10%	17.40%

表中所示增设下人均洗手率显著高于不增设下洗手率，证明在错时作息的情况下，也需要增设洗手池，保障人均洗手率。

4. 4 模型评估

4. 4. 1 能源消耗

根据统计每个人平均洗手用水量在 $\bar{L} = 550ml$ ，在考虑错时作息的情况下，由问题三模型估计出来的人均洗手率，粗略估计洗手人数，统计该时段需要的水资源统计如表 7：

	总人数	人均洗手率	洗手人数	资源消耗(m^3)
丁香区域	5513	35%	1946.089	1.070349
海棠区域	5535	36.70%	2031.345	1.11724
竹园区域	5457	37.10%	2041.984	1.123091
合计	16552		6019.418	3.31068

根据中国水价网查询知西安居民平均水价为 $2.9\text{元}/m^3$ ，由此可得出平均每天该时段水资源消费约为 10 元。由此判断能源资源的消耗是合理的。因此根据模型计算出来的增设水池规模是合理的，资源成本是可接受的。

4.4.2 防疫新冠有效性

利用问题一中的病毒传染模型，模拟在错时出行的人员流动背景下，假设每个区域初始有一例无症状感染者存在，使用 Monte Carlo 模拟出增设洗手池的情况下（实验组）时段结束后最终感染人数与不增设洗手池的情况下（对照组）的感染人数对比模拟结果如下表 8：

区域	总人数	实验组感染人数	对照组感染人数
丁香区域	5513	2	12
海棠区域	5535	2	13
竹园区域	5457	1	9

对照试验显示，增设洗手池的情况下，实验组感染人数明显小于对照组，证明模型计算出的新增洗手池的规模是合理的，能有效阻隔病毒的传播。

4.5 致校后勤部门工作人员的一封信

尊敬的后勤部门工作人员：

你们好！我们是西安电子科技大学的热心学生，想为贵部门提供一些在当今新型冠状病毒疫情形势下的复学提供一些建议。新型冠状病毒疫情给我们复学带来了不小的挑战，现在我们采取了多种预防手段来阻断病毒的传播，而其中洗手是最直接、最简单、最有效的手段之一。但是，目前仅有教学楼内有洗手资源，且其中的洗手资源也十分有限，较少的洗手资源也会导致人流聚集，不利于疫情的防控。另外，在校园活动时也少有洗手的地方。因此，我们利用数学建模的相关知识，得出了较为科学合理的洗手池设置方案，现叙述如下，供贵部门参考。

首先，我校生活区共分丁香、海棠、竹园三部分，每一部分地理位置相距较远，区域内部宿舍楼和食堂距离较近，且用餐、上课时间集中，因此在 6:30 至 8:30、12:00 至 13:30、17:00 至 18:30 均有较为聚集的人员流动，食堂人流量大。取复学后早晨情况为例，经推测，区域道路上和在食堂内的人流量较大，人员流动速率快，人群间相互接触的次数频繁，导致了病毒的传播。在仅有一个无症状感染者的情况下计算，该时段内的高风险感染者就高达 24 人。因此，对病毒传播的抑制手段亟需施行。

在预防新型冠状病毒的措施，洗手是最可行，最简单的方法之一。但是，经调查知，我校大学生洗手率意愿不高，再加上我校除教学楼和食堂外洗手资源匮乏，现有资源也不充裕，导致洗手率进一步降低，人流聚集度也相对提高，十分不利于当前新冠疫情的防控。

在经过数据处理、模型建立和合理分析后，我们对洗手池的建立提供以下建议：

在保持原有作息的情况下——在丁香公寓区域新增 6 个洗手池，在丁香餐厅区域新增 6 个洗手池，在海棠公寓区域新增 7 个洗手池，在海棠餐厅区域新增 6 个洗手池，在竹园公寓区域新增 6 个洗手池，在竹园餐厅区域新增 6 个洗

手池，在 AB 教学区新增 13 个洗手池，在 CD 教学区新增 13 个洗手池，在 EFG 教学区新增 10 个洗手池；

在进行错时作息的情况下——在丁香公寓区域新增 5 个洗手池，在丁香餐厅区域新增 4 个洗手池，在海棠公寓区域新增 5 个洗手池，在海棠餐厅区域新增 4 个洗手池，在竹园公寓区域新增 5 个洗手池，在竹园餐厅区域新增 4 个洗手池，在 AB 教学区新增 9 个洗手池，在 CD 教学区新增 8 个洗手池，在 EFG 教学区新增 5 个洗手池。

考虑到学生在道路上洗手意愿较低，且铺设成本较高，区域内的洗手池设置倾向于置于建筑附近人流量较大的地方附近，诸如建筑出口、交道口附近即可。

我们诚恳地向您们提出以上建议，希望我们的方案能为当前疫情的防控与复学的顺利进行做出一些贡献。我们相信我们能够迎接住这次疫情的挑战，在我们熟悉的西电校园，正常有序工作学习，勤奋努力，共克时艰！

闫浩霖、凌少鹏、廖怡诺

2020 年 5 月 4 日

五、附录

附录 1	
	丁香食堂
丁香宿舍群（取最大半径）	155
A 楼、B 楼	380
C 楼、D 楼	486
E 楼、F 楼	642
	丁香宿舍群（取最大半径）
丁香食堂	155
A 楼、B 楼	510
C 楼、D 楼	641
E 楼、F 楼	797
	海棠食堂
海棠宿舍群（取最大半径）	165
A 楼、B 楼	420
C 楼、D 楼	471
E 楼、F 楼	656
	海棠宿舍群（取最大半径）
海棠食堂	165
A 楼、B 楼	585
C 楼、D 楼	636

E 楼、F 楼	821
	竹园食堂
竹园宿舍群（取最大半径）	168
A 楼、B 楼	595
C 楼、D 楼	370
E 楼、F 楼	305
	竹园宿舍群（取最大半径）
竹园食堂	168
A 楼、B 楼	763
C 楼、D 楼	538
E 楼、F 楼	473

附录 2

```

1. import matplotlib.pyplot as plt
2. import numpy as np
3. from utils import *
4. import graph
5. import density
6. import datasheet
7. import csv
8.
9. class Graph(object):
10.
11.     # 初始化
12.     # size: 结点个数
13.     def __init__(self, nodes, edges):
14.
15.         # 存储变量
16.         self.Inf = 10 ** 9
17.         self.nodes = nodes
18.         self.edges = edges
19.         self.size = len(nodes)
20.
21.         # 初始化 ID 字典（对结点标号）
22.         self.idx = dict()
23.         for i in range(self.size):
24.             self.idx[nodes[i]] = i
25.
26.         # 初始化邻接矩阵
27.         self.graph = [[self.Inf for i in range(self.size)] for j in range(
            self.size)]
28.         for node in range(self.size):

```

```

29.         self.graph[node][node] = 0
30.     for edge in edges:
31.         self.add_edge(*edge)
32.
33.     # 初始化更新标记
34.     self.update = False
35.
36.
37.     # 添加边
38.     def add_edge(self, u, v, w=1):
39.         self.graph[self.idx[u]][self.idx[v]] = w
40.         self.graph[self.idx[v]][self.idx[u]] = w
41.         self.update = False
42.
43.
44.     # 获取边
45.     def get_edge(self, u, v):
46.         return self.graph[self.idx[u]][self.idx[v]]
47.
48.
49.     # 获取两点间最短距离
50.     def get_min_distance(self, u, v):
51.         if not self.update:
52.             self.flyod()
53.         return self.graph[self.idx[u]][self.idx[v]]
54.
55.
56.     # flyod 算法求最短路
57.     def flyod(self):
58.         for k in range(self.size):
59.             for u in range(self.size):
60.                 for v in range(self.size):
61.                     self.graph[u][v] = min(self.graph[u][v], self.graph[u]
[k] + self.graph[k][v])
62.             self.update = True
63.
64.
65.     # 求结点的权重
66.     # P: 一维数组, 表示各个结点的人流量
67.     # 返回 结点权重列表
68.     def get_weight(self, P):
69.         weight = dict()
70.         for u in range(self.size):
71.             weight[self.nodes[u]] = 0

```

```

72.         for v in range(self.size):
73.             weight[self.nodes[u]] += (self.graph[u][v] / 100 + 1) ** -
1 * P[v]
74.         return weight
75.
76.
77. if __name__ == '__main__':
78.
79.     graph_data = load_graph()
80.     graph = graph.Graph(*graph_data)
81.
82.     nodes, edges = graph_data[0], graph_data[1]
83.
84.     d, r = density.get_density(datasheet.number_of_origin_people, datasheet.percent_of_student_with_class, T=120, flag=False, batch=3)
85.     traffic = np.array(list(r.values()))
86.     weight_curve = {node: np.zeros(120) for node in nodes}
87.     weight_sum = dict()
88.
89.     for i in t:
90.         weight = graph.get_weight(traffic[:, i])
91.         for node in weight:
92.             weight_curve[node][i] = weight[node]
93.
94.     for node in weight_curve:
95.         weight_sum[node] = np.sum(weight_curve[node]) / len(weight_curve[node])
96.         plt.plot(t, weight_curve[node])
97.     plt.show()

```

附录 3

时段人流量模型

```

1. import numpy as np
2. import matplotlib.pyplot as plt
3. from utils import *
4. import graph
5. import csv
6. import datasheet
7.
8. def get_density(number_of_origin_people, percent_of_student_with_class, T=
120, sigma=20, flag=False, batch=3):
9.     G = graph.Graph(*load_graph())

```

```

10.     Dorm = datasheet.Dorm
11.     Canteen = datasheet.Canteen
12.     TeachingBuilding = datasheet.TeachingBuilding
13.     Corresponeding = datasheet.Corresponeding
14.     dur = datasheet.dur
15.     P = datasheet.P
16.     v = datasheet.v
17.     K = datasheet.K
18.
19.     number_of_delta_people = dict()
20.     delay = dict()
21.     V = 83.333333
22.     for dorm in Dorm:
23.         if flag:
24.             dT = T / batch
25.             dsigma = sigma / batch
26.             number_of_delta_people[dorm] = np.hstack([Normal(np.arange(dT)
, dT * v[dorm], dsigma) * number_of_origin_people[dorm] / batch] * batch)
27.         else:
28.             number_of_delta_people[dorm] = Normal(np.arange(T), T * v[dorm
], sigma) * number_of_origin_people[dorm]
29.
30.             delay[dorm] = dict()
31.             delay[dorm][Corresponeding[dorm]] = int(G.get_min_distance(dorm, C
orresponeding[dorm]) / V)
32.             for teachingbuilding in TeachingBuilding:
33.                 delay[dorm][teachingbuilding] = int(G.get_min_distance(dorm, t
eachingbuilding) / V)
34.         for canteen in Canteen:
35.             number_of_delta_people[canteen] = np.zeros(T)
36.
37.             delay[canteen] = dict()
38.             for teachingbuilding in TeachingBuilding:
39.                 delay[canteen][teachingbuilding] = int(G.get_min_distance(cant
een, teachingbuilding) / V)
40.         for teachingbuilding in TeachingBuilding:
41.             number_of_delta_people[teachingbuilding] = np.zeros(T)
42.
43.         for t in range(T):
44.             for i in range(len(Dorm)):
45.                 for j in range(len(TeachingBuilding)):
46.                     if t >= delay[Dorm[i]][TeachingBuilding[j]]:

```

```

47.         number_of_delta_people[TeachingBuilding[j]][t] += num
er_of_delta_people[Dorm[i]][t - delay[Dorm[i]][TeachingBuilding[j]]] * per
cent_of_student_with_class[Dorm[i]]
48.         if t >= delay[Dorm[i]][Canteen[i]] + dur[Canteen[i]]:
49.             number_of_delta_people[TeachingBuilding[j]][t] += numb
er_of_delta_people[Canteen[i]][t - delay[Dorm[i]][Canteen[i]] - dur[Cantee
n[i]]] * percent_of_student_with_class[Dorm[i]] * K[TeachingBuilding[j]]
50.             if t >= delay[Dorm[i]][Canteen[i]]:
51.                 number_of_delta_people[Canteen[i]][t] += number_of_delta_p
eople[Dorm[i]][t - delay[Dorm[i]][Canteen[i]]] * (1 - P[Dorm[i]])
52.
53.     number_of_remained_people = dict()
54.     for dorm in Dorm:
55.         number_of_remained_people[dorm] = get_erosion(number_of_delta_peop
le[dorm], number_of_origin_people[dorm])
56.     for canteen in Canteen:
57.         number_of_remained_people[canteen] = get_traffic(number_of_delta_p
eople[canteen], dur[canteen])
58.     for teachingbuilding in TeachingBuilding:
59.         number_of_remained_people[teachingbuilding] = get_accumulation(num
ber_of_delta_people[teachingbuilding])
60.     return number_of_delta_people, number_of_remained_people

```