1. **\*\*\***
2. **\*\*\***
3. **网络模型和问题描述**
   1. 任务描述（注意，此时仅有大体执行，一些因子的引入需要写着逐渐完善，造新东西嘛）

此模块梳理思路，先将用户群体分为可信任和不可信任两部分群体---此处引入信任度---，接着有三个部分：分配任务，处理数据和提供报酬分配任务:

分配任务：

平台方提供任务，由用户自己选择（初始位置和任务数量，以及对不同任务的报价,以及承诺数据质量）

此时平台会对所有任务提交进行评估进行任务分配（承诺质量不足75%的不选，同时可初步计算贡献效率来作为选择标准---计算看老论文---），被分配到的人称为优胜者----此模块可以修改为机器学习----

分配原则（对不同群体有不一样的，对于信任群体应遵循覆盖原则，对于非信任群体则应尽可能遵循重合原则，即与信任群体的任务尽可能重合，同时也应该满足覆盖原则）

----此处的实现可能会引入第一篇论文中的现状偏好因子来实现

-----此处目的：尽可能实现任务的覆盖，结合报酬实现低成本和宽覆盖，高质量的综合最优解（其中不是任何一项的最优，即不能一味追求数据质量这种东西）

处理数据：

首先对于信任人员，根据其提供数据计算得到不同等级的标准数据并储存在一个数组----或者其他存储函数----（其中若一个人基于其所报数据为标准数据，多人则根据信任度加权获得，取平均也行），接着更新所有人员的数据：

1. 信任度更新：根据实际等级与承诺等级进行对比，此处我们引入第一个修正函数，以标准数据表当前数据作为基准点，将实际数据离散化为等级的映射情况，代入数值即可修正信任度---此处函数可能需要重新设计，注意此处数据和等级是不一样的---
2. 数据质量评估：与标准数据表对比确认实际数据质量等级

---注意，实际上标准数据表并不一定会覆盖所有咱们需要的75%以上所有数据，我们可以根据大部分数据设计函数计算得到其他的，或者用户对比的干脆找最近的标准数据

---针对上一个注意，我们也可采用昨天的思路，没有确定的就按照不可信内部进行加权当作标准值进入标准数据表，不过此时更新信任度幅度小（就昨天说的那个）

分配报酬：

咱们分配报酬的原则是基础+额外，基础即报价，额外的计算：

由于涉及到不同的等级，所以等级也要作为一个参数套入计算额外的，且质量越高理论上报酬也应该越高，由此数据等级与完美数据的距离也要作为一个参数，由此有一个初步思路：

我们可以设计三个函数，一个针对承诺的计算额外报酬，距离承诺越近报酬越高；另一个我们针对实际效益（即数据越准越好）来给予额外报酬，最后综合前两个函数所得到的报酬进行综合确定最终额外报酬（>=-基础,有下界的函数应该好找）

* 1. 物理模型（要写定义内容）
  2. \*\*\*

1. **同类模型比较**
2. **模拟数据实验论证**