凸优化中期报告

伍维晨 李泽坤 窦鹏飞

课题二：**Algorithms for Large-scale Optimal Transport**

一、工作概述

本组目前的工作主要分两个部分，第一部分为研读两篇论文，对其进行分析，掌握其中所提到的算法，为进一步实现其中的算法打好基础；第二部分是运用 Gurobi 计算模块初步实现对问题（2.1）的解答，并对单纯形法和内点法在该问题上的表现进行对比。

二、论文概述

2.1 *DOTmark- A Benchmark for Discrete Optimal Transport*

2016 年 12 月，Jorn Schrieber 等在 IEEE 上发表论文 *DOTmark- A Benchmark for Discrete Optimal Transport*，提出了对离散最优运输问题（Discrete Optimal Transport,DOT）的各类算法的评价标准。这篇论文中提到了四种求解离散最优运输问题的算法，是本组完成期末项目的重要参考。这部分将首先叙述论文的主体脉络，并提出本组将以此为参考进行何种工作。

论文的第一部分介绍了问题的背景和研究的动机，第二部分明确了最优运输问题的定义。在一般意义下，最优运输问题定义为：

$$\min_{\pi\in\Pi(\mu,\nu)}\int_{X\times Y}\|x-y\|^p d\pi(\mathrm{x},\mathrm{y})$$
$$\mathrm{s.t.}\int_X\|x\|^p d\mu(\mathrm{x})<\infty$$
$$\int_Y\|y\|^p d\nu(\mathrm{y})<\infty$$

其中μ和ν分别是集合X和Y上的测度。在实际应用中，往往将问题离散化，转化为以下的线性规划问题:

$$\min\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}\pi_{ij}$$
$$s.t.\sum_j\pi_{ij}=\mu_i,\forall i=1,2,...,\mathrm{m}$$
$$\sum_i\pi_{ij}=\nu_j,\forall j=1,2,...,n$$
$$\pi_{ij}\geq 0$$

本文的第三部分提出了检验离散最优运输问题各类算法的标准，称为 DOTmark，这是本文最主要的创建。作者运用了十种不同的图片——其中有的是以各种方式随机生成的，有的是经典图片，有的是几何图形，还有的是实际应用

中的图像（显微镜下的线粒体图像）——两两配对，求解最优运输问题，并在不同的问题规模下对不同算法的运行时间进行考察。第四部分则介绍了作者考察的四种算法：

（1）Transportation Simplex（运输单纯形法）：该算法将 DOT 视为一个完全二部图匹配问题，并将任何一个可行匹配视为该二部图上的一棵生成树。算法从一个基础可行匹配开始，逐次加入变量，每次加入变量后都在此前的生成树上寻找由此生成的环；在该环上进行调整，尽可能减少匹配的代价，并将调整后变为 0 的变量从基础可行匹配中删去。如此循环，直至无法继续减少匹配的代价为止。

（2）Shortlist Method(短列表法)：该方法是运输单纯形法的变种，所不同的是加入了一个短列表和三个变量 s,q,l:短列表用于限制搜索空间，s 表示该短列表的长度，q 和 l 用于控制寻找新基本可行解时搜寻的变量个数。当短列表中的所有变量都无法使运输代价进一步减小时，调用运输单纯形法将此时得到的局部最优解转化为全局最优解。

（3）Shielding Neighborhood Method, or Shortcut Method(捷径算法)：该算法的核心思想是解决一系列的稀疏最优运输问题。

（4）AHA Method(AHA 算法)：该算法原本用于解决由连续测度向离散测度的最优运输问题，即所谓的半离散最优运输问题；为使该算法应用于完全离散的最优运输问题，本文作者对匹配中的第一幅图像做了连续化处理。该算法不一定是完全精确的。

接下来，本文作者应用 DOTmark 标准，对以上四种算法的性能做了检验，由于不涉及本组工作，此处对检验结果不再赘述。本组计划详细表述并实现 Transportation Simplex 算法，并将该算法的性能与其他算法进行比较。

2.2 *Multiscale Strategies for Computing Optimal Transport*

这篇论文提出了解决点集之间的近似优化运输问题的一种多尺度方案，对那些表面上高维但其实质是低维的点集尤其有效。其核心思想是，对于待解问题的点集进行多尺度的分解，产生一系列的优化运输问题，每次分解都将原本较为精细的问题规模转变为更加粗糙的形式，从而便于解决。在分解完成后，通过自顶向下的模式依次解决各子问题。文中数值实验证实，该算法的时间和空间复杂度是 O(n)的，其中 n 为点集中点的个数，精确度也大幅提高。不仅如此，由于分解算法从更大的粒度上来“观察”这个优化运输问题，我们有可能从中发现一些直接求解所看不到的新奇的特征。

文章的第一部分首先提出了优化运输问题的定义，这是一类特殊的线性规划问题。当然，线性规划问题有成熟的求解方法，但是文章指出直接套用线性规划的求解方法忽视了问题本身有用的几何特性。文章也介绍了一些相关的工作，这些算法也是本文中算法的思想来源，包括第一篇论文里提到的 **Shielding Neighborhoods Method**.

文章的核心在于第三部分，也就是算法部分。解决两个点集X与Y之间的优化运输问题，直观上讲，我们当然要考虑$|X||Y|$条路径，因此这个算法的复杂度应

当是平方级别的。于是，降低问题规模成为算法要考虑的目标，这在原文中叫做**Coarsening** 过程。具体来讲，原有问题中的一片区域许多点可以被“捏”成下一尺度优化运输问题中的一个点，这种方法可以递归地进行下去，直到问题规模足够小以至于我们可以用现有的成熟的优化运输问题的算法直接有效求解。这样我们就成功地减少了问题的规模。这种逐步规约最终可以得到一个层次结构，比较典型地体现了分治思想，这也是算法得名“multiscale”的原因。但同时我们的经验又提醒我们，更粗粒度的问题规约必然带来更不精确的解。为了解决这个问题，本文算法提出了具有原创性的 **Propagate** 过程，即较前一尺度下优化运输问题的解可以被传递到并应用于下一尺度问题的求解。这种应用可以是两方面的，一是前一层最优解可以排除掉一些路径，让我们求解下一层时不必考虑；二是前一层的解采用极为简单的变换可以变成第二层的一个较好初始解，这将大大减少迭代的次数。这样自顶向下的求解子问题就是一个对原问题最优解精确度不断提升的逼近。最后我们还要有求解每一个子问题的有效算法，回想一下，我们已经有了由上一层传来的较优初始解，所以这个过程称为 **Refinement**，其实质为 batch column generation methods.

除此之外，文章提到已经有一个 R 包 *mop* 实现了多尺度的框架，可以用在多尺度分解过程。这个包允许运用多种不同的方法来求解单个的优化运输问题，比如 MOSEK 或者 CPLEX.

针对这篇论文，我们将要做的工作是仔细研究 *mop* 的实现，主要分为以下几个部分：第一是 **Coarsening** 过程中构建多尺度点集表示的算法，比如迭代 K 均值算法；第二是不同的 **Propagate** 策略，比如 Simple propagation 和 Iterated capacity propagation;第三是不同的 **Refinement** 策略，比如 neighborhood refinement 和 potential refinement。之后我们打算用 Python 自己实现上述算法，同时还要比较求解单一优化运输问题的各种方法应用于上述算法框架的效果，比如网络单纯性法，Ford-Fulkerson 算法等。实验的数据集将包括文中主要提到的两个：Caffarelli data sets 和 ellipse data set。

三、利用 Gurobi 比较单纯形法和内点法求解（2.1）的表现

我们使用 Gurobi 计算模块对问题（2.1）在 Python 环境下进行了试验（代码见附件），为缩短试验时间，使用了规模为 1024×1024 的随机参数矩阵，并通过更改运行时的参数对单纯形法和内点法的表现进行了比较。

总共进行了 100 次试验，对于同一个问题二者能得到相同的解，单纯形法的平均运行时间为 18.73 秒，内点法平均运行时间为 19.04 秒，二者的总体表现不相上下，但在单个问题上表现各有千秋。

我们计划接下来对论文 DOTmark 中提供的数据集进行求解，并比较两种优化方法的表现。