

# Pipeline Route Optimization Design

Haodong Chen<sup>1</sup>, Jiakang Du<sup>1</sup>, Bohan Zhang<sup>2</sup>, Qingjun Ren<sup>1</sup>, Hongchun Sun<sup>1</sup>

<sup>1</sup>School of Mathematics and Statistics, Linyi University, Linyi Shandong

<sup>2</sup>School of Information Science and Engineering, Jinan University, Jinan Shandong

Email: 2101058556@qq.com, 614558183@qq.com

Received: Jun. 24<sup>th</sup>, 2018; accepted: Jul. 13<sup>th</sup>, 2018; published: Jul. 20<sup>th</sup>, 2018

---

## Abstract

With the rapid development of industry and urbanization, the demand for tap water is increasing in urban construction, and pipeline is an important way to transport tap water. The thought of this article, based on the Dijkstra algorithm, in view of the network system of pipeline route, the shortest route optimization design, algorithm steps are given; and by using the MATLAB calculation program of optimal design for the pipeline, the shortest path is obtained. Finally, through the numerical test, the rationality and practicability of the model are further verified. The establishment of the model has great social and economic benefits and saves social resources.

## Keywords

The Shortest Route, Optimal Design, Dijkstra's Algorithm

---

# 管道路线的优化设计

陈昊东<sup>1</sup>, 杜家康<sup>1</sup>, 张博瀚<sup>2</sup>, 任庆军<sup>1</sup>, 孙洪春<sup>1</sup>

<sup>1</sup>临沂大学数学与统计学院, 山东 临沂

<sup>2</sup>济南大学信息科学与工程学院, 山东 济南

Email: 2101058556@qq.com, 614558183@qq.com

收稿日期: 2018年6月24日; 录用日期: 2018年7月13日; 发布日期: 2018年7月20日

---

## 摘要

随着工业进程和城市化的加快, 在城市建设中对自来水的需求量越来越大, 而管道是运输自来水的重要途径。本文, 基于Dijkstra算法的思想, 针对管网系统中管道路线, 进行了最短路线的优化设计, 给出了算法步骤, 并利用MATLAB编制了优化设计的计算程序, 得出了管道铺设的最短路径。最后, 通过进行数值试验, 进一步的验证了模型的合理性、实用性, 模型的建立具有很大的社会效益, 节约了社会资源。

## 关键词

最短路线, 优化设计, Dijkstra算法

Copyright © 2018 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

水是生命之源, 是城市发展的基础。近些年来, 随着我国经济的迅猛发展, 城市化进程和人口迅速扩张, 城市用水量也日益剧增, 城市供水面临的问题越来越突出, 严重限制了城市的进一步发展。而管道是运输水源的重要途径, 所以说对管道的铺设进行优化设计显得刻不容缓。管道铺设问题的优化其根本目的在于能够最大限度的缩小管道的铺设长度, 其实也就是我们常说的最短路径问题。目前, 国内外的专家学者对最短路线问题已经有了很多的研究。比如, 王俊珺、夏华丽([1])选择 Dijkstra 算法作为核心算法; 并且针对原始 Dijkstra 算法的不足, 分别对原始 Dijkstra 算法的数据存储方式、执行效率和数据结构提出了优化方法; 施益昌、郑贤斌等([2])应用 Floyd 算法, 借助于 MATLAB 程序设计语言对动态规划中最短路径进行了有效的设计; 赵娟、王建新([3])运用动态规划方法的基本思想求解最短路径问题; 张水舰、刘学军等([4])定义了动态随机最短路径, 给出了动态随机最短路径优化数学模型, 提出了一种动态随机最短路径遗传算法; 王树西、吴政学([5])通过对 Dijkstra 算法进行改进, 解决了 Dijkstra 算法退出机制对不联通的有向图是无效的, 会陷入死循环等问题; 王荣、江东、韩惠([6])将 Floyd 与 Dijkstra 算法结合, 优化后的算法在很大程度上减少了运算次数和时间, 提高了算法的时间及空间复杂度, 算法效率较高。本文主要基于 Dijkstra 算法的思想并作出了改进, Dijkstra 算法是保证源点到各个顶点之间的路程最短, 而改进后的算法是在保证经过所有顶点的前提下, 使得总的路线最短, 这样就提出了管道铺设最短路线的优化设计方案, 给出了算法步骤, 并编制出了 MATLAB 程序。

## 2. 优化设计数学模型

假设有一个自来水厂为周边的城市进行供水, 自来水厂净化的水足够多, 完全可以供应周边城市, 那我们如何进行管道的铺设可以将自来水厂生产的自来水供应给周边的城市又保证所铺设的管道路线最短呢? 下面我们给出两种思路的管道铺设优化方案。

### 2.1. 单条线路的管道铺设方案(附录 1)

假设自来水厂的位置是固定的, 它要供给给周边  $n$  个城市的自来水, 并且我们已知任何一个城市之间的距离以及任何一个城市和自来水厂的距离。因为我们建造的是单条线路的管道, 即管道由自来水厂出发经过所有的城市, 最终到达最后一个城市, 所以我们先固定自来水厂找到和自来水厂距离最近的城市, 然后再以该城市为起点找到和它距离最近的城市, 进行迭代, 每次都选取最近的距离, 最终所形成的路线也是最短的。

例如, 在图 1 所示中, 1 表示自来水厂, 2~6 表示自来水厂要供给自来水的五个城市, 采用单条线路进行管道铺设, 每段路线长度如图 1 所示, 问如何才能使得铺设管道的总路线最短?

通过计算得知, 从自来水厂开始铺设管道先到达 3 号城市, 由 3 号城市再铺到 5 号城市, 再到 2 号城市, 再由 2 号城市到 4 号城市, 最后铺到 6 号城市结束。即: 1→3→5→2→4→6。图中红线部分即为

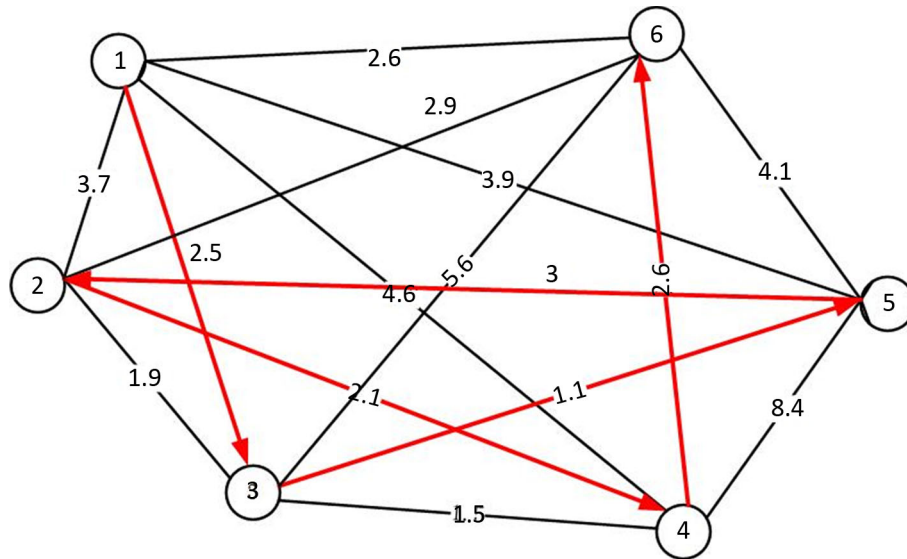


Figure 1. Single line pipe laying diagram  
图 1. 单条线路管道铺设图

管道铺设的分布图(长度单位为 km), 总长度为 11.3 km。

下面我们通过矩阵的形式进行解释:

第一步, 构造自来水厂与各个城市之间的距离以及各个城市之间的距离, 设  $a_{1j}$  为自来水厂和第  $j$  个城市之间的距离,  $b_{ij} (i=1,2,\dots,n; j=1,2,\dots,n)$  表示为第  $i$  个城市与第  $j$  个城市之间的距离, 这样就可以建立一个  $n \times n$  的矩阵  $A$ 。

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ b_{11} & b_{12} & \cdots & b_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix}$$

其中, 在矩阵  $A$  中当  $i = j$  时,  $b_{ij}$  表示的为自身之间的距离, 为避免对数据准确性造成影响, 故设为无穷大数。

第二步, 比较矩阵  $A$  的第一列, 得出第一列的最小值, 设为  $a_{1k}$ , 其含义为自来水厂与第  $k$  个城市之间的距离最短。

第三步, 以第  $k$  个城市为起点, 找与第  $k$  个城市距离最短的城市。我们需要先找到矩阵  $A$  的第  $k+1$  行, 因为第  $k+1$  行表示的是第  $k$  个城市与其他城市之间的距离。所以我们比较第  $k+1$  行中各个元素的大小, 可得第  $k+1$  行中最小的元素为  $b_{(k+1)m}$ , 其含义为第  $k$  个城市与第  $m$  个城市之间距离最短。输出城市  $m$  后, 将元素  $b_{(k+1)m}$  所在的列上所有的元素都变为无穷大值, 得到矩阵  $B$ 。

$$B = \begin{pmatrix} a_{11} & \cdots & a_{1(m-1)} & \infty & a_{1(m+1)} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{(k+1)1} & \cdots & b_{(k+1)(m-1)} & \infty & b_{(k+1)(m+1)} & \cdots & b_{(k+1)n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{n(m-1)} & \infty & b_{n(m+1)} & \cdots & b_{nn} \end{pmatrix}$$

第四步, 以城市  $m$  为起点, 找到与城市  $m$  距离最近的城市。先找到矩阵  $B$  的第  $m+1$  行, 比较  $m+1$  行中所有元素的大小, 得出  $b_{(m+1)p}$  为最小值。按照第三步的做法将  $p$  输出, 并将  $b_{(m+1)p}$  所在列的所有元素变为无穷大值, 得到矩阵  $C$ 。

$$C = \begin{pmatrix} a_{11} & \cdots & a_{1(p-1)} & \infty & a_{1(p+1)} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{(m+1)1} & \cdots & b_{(m+1)(p-1)} & \infty & b_{(m+1)(p+1)} & \cdots & b_{(m+1)n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{n(p-1)} & \infty & b_{n(p+1)} & \cdots & b_{nn} \end{pmatrix}$$

第五步，每次都按照第三步的方法依次进行迭代，每次输出一个值，直到最后输出完所有值，进行完  $n$  次迭代，根据依次输出的值就可以知道从自来水厂经过各个城市的最短单条线路的顺序。

本题是基于迭代思想并且运用矩阵给出的算最短路径的方法，该方法简单易懂，过程相对简单，不过对于计算机要求较高，运算起来较为麻烦，对前期数据要求较为具体，但是也从迭代的方面提出了最短路径的算法。

## 2.2. 树状的管道铺设方案(附录 2)

我们下面运用 Dijkstra 算法的思想对上述问题进行进一步的优化求解，在我们的日常生活中自来水厂给周围城市进行供水一般来说不会采用单线路的管道运水方案，因为这样来说不能够做到铺设管道总路线最短，现在一般都采用树状形式的管道输水路线，这样既可以做到铺设管道总路线最短，又可以保证使用的管材最省。通过 Dijkstra 算法求解管道铺设总路线最短的主要思想是每次找到离源点最近的一个点，然后将该点看做源点再寻找离它最近点，通过层层迭代，最终就可以找到铺设管道的最短路线。

例如，在图 2 所示中，1 表示自来水厂，2~6 表示自来水厂要供给自来水的五个城市，现在采用树状的管道铺设方案对管道进行铺设，每段路线长度如图 1 所示，问如何才能使得铺设管道的总路线最短？

同样是上面的问题，如果我们采用树状的管道铺设方案，其管道铺设方案如下图所示。

下面我们具体的进行解释：

第一步，我们先将源点到各个城市之间的距离和各个城市之间的距离用矩阵的形式表示出来。用 0 表示自来水厂，1~ $n$  表示  $n$  个城市， $a_{0j}$  ( $j=1,2,\dots,n$ ) 表示自来水厂到  $n$  个城市之间的距离， $a_{ij}$  ( $i=1,2,\dots,n; j=1,2,\dots,n$ ) 表示第  $i$  个城市到第  $j$  个城市之间的距离， $a_{i0}$  ( $i=1,2,\dots,n$ ) 表示  $n$  个城市到自来水的距离，则表示上述距离的矩阵  $A$  可表示为：

$$A = \begin{pmatrix} a_{00} & a_{01} & \cdots & a_{0n} \\ a_{10} & a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n0} & a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

其中矩阵  $A$  中  $a_{ij}$  ( $i=j$ ) 时表示的是自身之间的距离，为避免对数据准确性造成影响，故设为无穷大数。

第二步，先对矩阵第一行自来水厂与各个城市之间的距离大小进行比较，假设自来水厂和城市  $k$  之间的距离最小，即  $a_{0k}$  是第一行所有元素中最小的。先输出  $k$  值，然后将矩阵中的  $a_{0k}$  和第  $k+1$  行的元素  $a_{k0}$  都用无穷大值替换，得到矩阵  $B$ 。

$$B = \begin{pmatrix} a_{00} & \cdots & a_{0(k-1)} & \infty & a_{0(k+1)} & \cdots & a_{0n} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{(k-1)0} & \cdots & a_{(k-1)(k-1)} & a_{(k-1)k} & a_{(k-1)(k+1)} & \cdots & a_{(k-1)n} \\ \infty & \cdots & a_{k(k-1)} & a_{kk} & a_{k(k+1)} & \cdots & a_{kn} \\ a_{(k+1)0} & \cdots & a_{(k+1)(k-1)} & a_{(k+1)k} & a_{(k+1)(k+1)} & \cdots & a_{(k+1)n} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n0} & \cdots & a_{n(k-1)} & a_{nk} & a_{n(k+1)} & \cdots & a_{nn} \end{pmatrix}$$

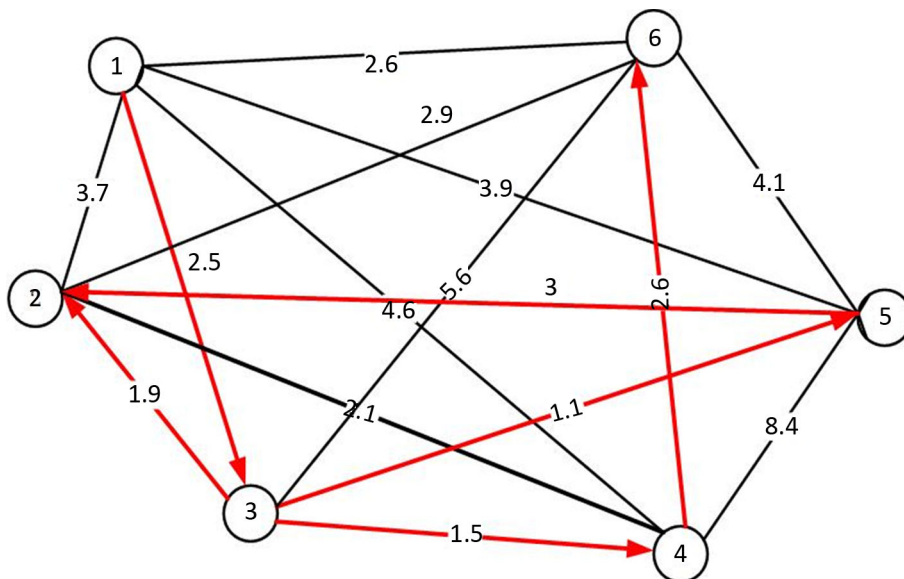


Figure 2. Tree piping layout  
图 2. 树状管道铺设图

第三步，比较第 1 行和第  $k+1$  行中所有的元素求出最小值，下面假设几种不同的情况：

① 若  $a_{0m}$  是其中的最小值，那么表示从自来水厂直接到城市  $m$  距离最小；

② 若  $a_{kp}$  是其中的最小值，那么表示从第  $k$  个城市到第  $p$  个城市之间距离最小，我们就可以从城市  $k$  直接铺设管道到达城市  $p$ ，这样就会减少管道铺设的总线路，达到最节约管材。

第四步，先假设第三步中的②成立，然后将  $a_{kp}$  以及矩阵  $B$  中第  $p+1$  行元素  $a_{pk}$  变为无穷大值，得到新的矩阵  $C$ ，按照上述的方法再比较矩阵  $C$  中的第 1 行、第  $k+1$  行和第  $p+1$  行中的所有元素得出最小的元素，下面对最小的元素进行分情况讨论：

① 若  $a_{0q}$  是最小的元素，即从自来水厂直接铺设管道到城市  $q$  会使得总路线最短；

② 若  $a_{kw}$  是最小的元素，即从城市  $k$  铺设管道到城市  $w$  会使得总路线最短；

③ 若  $a_{pr}$  是最小的元素，即从城市  $p$  铺设管道到城市  $r$  会使得总路线最短。

第五步，根据上述的做法进行层层迭代，就可以求出铺设管道到所有城市总路线最短的方案。

该方法较为贴合实际，对现实的生活中管道铺设方案有较为重要的指导意义，本算法是基于 Dijkstra 算法，在此基础上进行的进一步的改进，Dijkstra 算法研究的是源点到各个顶点之间的路程最短，而我们这个算法是在保证经过所有顶点的前提下，使得总的路线最短，这样就可以达到铺设管道所用管材最省，节约社会资源。

### 3. 数值算例

有一个自来水厂要给周围的 5 个城市供水，自来水厂和 5 个城市依次用 1~6 个数字进行表示，自来水厂和 5 个城市的分布图如图 1 所示，自来水厂到各个城市之间的距离以及 5 个城市之间的距离如表 1，求在经过所有城市的前提下，从自来水厂如何铺设管道？

求解思路：首先，用矩阵形式表示自来水厂和 5 个城市以及 5 个城市之间的距离，然后比较第一行中元素的大小，根据第一行元素的位置，找到下一个元素的位置，将两个元素都变为无穷大值，再比较两行的元素的大小得出最小值，再逐次往下计算。具体求解过程：

**Step1:** 用矩阵形式表示自来水厂和各个城市之间的距离以及 5 个城市之间的距离：

**Table 1.** Distance between points  
**表 1.** 各个点之间的距离

线段	距离( $\times 10^5$ m)
1~2	3.7
1~3	2.5
1~4	4.6
1~5	3.9
1~6	2.6
2~3	1.9
2~4	2.1
2~5	3.0
2~6	2.9
3~4	1.5
3~5	1.1
3~6	5.6
4~5	8.4
4~6	2.6
5~6	4.1

$$A = \begin{pmatrix} \infty & 3.7 & 2.5 & 4.6 & 3.9 & 2.6 \\ 3.7 & \infty & 1.9 & 2.1 & 3 & 2.9 \\ 2.5 & 1.9 & \infty & 1.5 & 1.1 & 5.6 \\ 4.6 & 2.1 & 1.5 & \infty & 8.4 & 2.6 \\ 3.9 & 3 & 1.1 & 8.4 & \infty & 4.1 \\ 2.6 & 2.9 & 5.6 & 2.6 & 4.1 & \infty \end{pmatrix}$$

**Step2:** 先比较第一行元素的大小值, 经过比较得出第一行中最小值为  $a_{13} = 2.5$ , 再将矩阵  $A$  中的  $a_{13}$  和  $a_{31}$  都用无穷大值替换, 得到矩阵  $B$ :

$$B = \begin{pmatrix} \infty & 3.7 & \infty & 4.6 & 3.9 & 2.6 \\ 3.7 & \infty & 1.9 & 2.1 & 3 & 2.9 \\ \infty & 1.9 & \infty & 1.5 & 1.1 & 5.6 \\ 4.6 & 2.1 & 1.5 & \infty & 8.4 & 2.6 \\ 3.9 & 3 & 1.1 & 8.4 & \infty & 4.1 \\ 2.6 & 2.9 & 5.6 & 2.6 & 4.1 & \infty \end{pmatrix}$$

即先从自来水厂出发铺设管道到第 3 个城市。

**Step3:** 比较矩阵中的第一行和第三行的元素, 得最小值  $a_{35} = 1.1$ , 再将矩阵  $B$  的  $a_{35}$  和  $a_{53}$  用无穷大值替换, 得到矩阵  $C$ :

$$C = \begin{pmatrix} \infty & 3.7 & \infty & 4.6 & 3.9 & 2.6 \\ 3.7 & \infty & 1.9 & 2.1 & 3 & 2.9 \\ \infty & 1.9 & \infty & 1.5 & \infty & 5.6 \\ 4.6 & 2.1 & 1.5 & \infty & 8.4 & 2.6 \\ 3.9 & 3 & \infty & 8.4 & \infty & 4.1 \\ 2.6 & 2.9 & 5.6 & 2.6 & 4.1 & \infty \end{pmatrix}$$

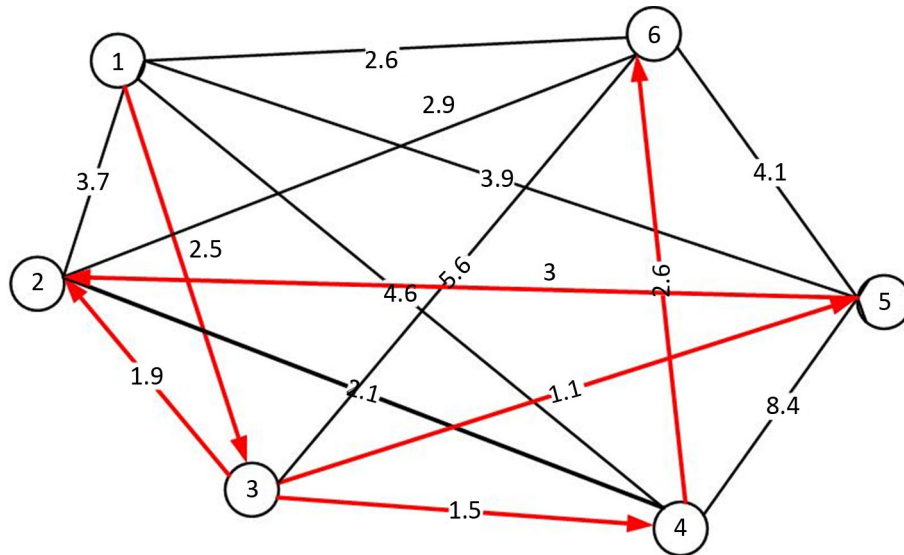


Figure 3. Pipe laying diagram  
图 3. 管道铺设图

即从自来水厂铺设管道到第 3 个城市，再从第 3 个城市铺设管道到达第 5 个城市

**Step4:** 比较矩阵中的第一行、第三行和第五行元素，得最小值是  $a_{34} = 1.5$ ，再将矩阵  $C$  中的  $a_{34}$  和  $a_{43}$  用无穷大值替换，得到矩阵  $D$ :

$$D = \begin{pmatrix} \infty & 3.7 & \infty & 4.6 & 3.9 & 2.6 \\ 3.7 & \infty & 1.9 & 2.1 & 3 & 2.9 \\ \infty & 1.9 & \infty & \infty & \infty & 5.6 \\ 4.6 & 2.1 & \infty & \infty & 8.4 & 2.6 \\ 3.9 & 3 & \infty & 8.4 & \infty & 4.1 \\ 2.6 & 2.9 & 5.6 & 2.6 & 4.1 & \infty \end{pmatrix}$$

即从自来水厂铺设管道到第 3 个城市，再从第 3 个城市铺设管道到达第 5 个城市，再由第 3 个城市铺设管道到第 4 个城市

**Step5:** 依次进行下去，就可以得到管道铺设方案为从自来水厂铺设管道到第 3 个城市，再从第 3 个城市铺设管道到达第 5 个城市，再由第 3 个城市铺设管道到第 4 个城市，由第 3 个城市铺设管道到第 2 个城市，再由第 4 个城市铺设管道到第 6 个城市，总长为  $9.6 \times 10^5$  m，分布图如图 3 所示。

如果采用单条线路铺设方案，需要铺设的管道长度为  $11.3 \times 10^5$  m，而采用树状管道铺设方案只需要铺设  $9.6 \times 10^5$  m，与单条线路管道铺设方案相比在管道铺设长度上减少了 15%，所以说树状管道铺设方案对于缩短管道铺设长度具有重要的意义。

### 参考文献

- [1] 王俊珺, 夏华丽, 等. 物流配送路线规划中的最短路径研究[J]. 农业网络信息, 2007(5): 60-62.
- [2] 施益昌, 郑贤斌, 等. 基于 MATLAB 动态规划中最短路线的实现程序[J]. 电脑学习, 2003(6): 37-38.
- [3] 赵娟, 王建新. 用动态规划方法求解最短运输路线问题[J]. 现代电子技术, 2012, 35(17): 120-122.
- [4] 张水舰, 刘学军, 杨洋. 动态随机最短路径算法研究[J]. 物理学报, 2012, 61(16): 7-16.
- [5] 王树西, 吴政学. 改进的 Dijkstra 最短路径算法及其应用研究[J]. 计算机科学, 2012, 5(39): 223-228.
- [6] 王荣, 江东, 韩惠. 基于 Floyd 方法的最短路径算法优化算法[J]. 甘肃科学报, 2012, 4(24): 114-118.

## 附录

### 附录 1 单条线路的管道铺设方案 MATLAB 程序

```
function [U,V,d]=lcxz(A,b)
[m,n]=size(A);
B=A;
U=zeros(1,n);
V=ones(1,n);
R=b*ones(1,n);
L=b*ones(m,1);
a=A(1,:);
[u,v]=min(a);
B(1,:)=R;
B(:,1)=L;
t=1;
U(1,2)=u;
V(1,2)=v;
for i=2:n-1
    s=v;
    f=B(v,:);
    t=i+t;
    [u,v]=min(f);
    U(1,i+1)=u;
    V(1,i+1)=v;
    B(s,:)=R;
    B(:,s)=L;
i=i+1;
end
U;
V;
d=sum(U);
```

### 附录 2 树状的管道铺设方案

```
function [U,V,d]=ndjtsl(A,b)
[m,n]=size(A);
C=inf*ones(n);
B=A;
U=zeros(1,n);
V=ones(1,n);
rep=b;
a=B(1,:);
```



```
[u,v]=min(a);
B(1,v)=rep;
B(v,1)=rep;
U(1,2)=u;
V(1,2)=v;
f=B(1,:);
for i=3:n
    r=v;
    F=[f;B(v,:)];
    [u,v]=min(F,[],2);
    rv=v;
    [u,v]=min(u);
    U(1,i)=u;
    x=v;
    s=V(x);
    y=rv(v);
    v=y;
    V(1,i)=y;
    B(r,y)=rep;
    B(y,r)=rep;
    B(s,y)=rep;
    B(y,s)=rep;
    f=[f;B(r,:)];
    f(x,y)=rep;
    i=i+1;
end
d=sum(U);
```

**知网检索的两种方式:**

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>  
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2324-7991, 即可查询
2. 打开知网首页 <http://cnki.net/>  
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: [aam@hanspub.org](mailto:aam@hanspub.org)