

网上任务分派问题数学实验的可视化设计^①

石彤菊

(华北电力大学 数理电学院, 保定 071003)

摘 要: 研究了各种任务分派问题算法, 对一般任务分派和有约束任务分配的求解方法进行算法设计, 以 Microsoft Visual J++ 6.0 作为实验开发平台, 采用 Java 语言及 Applet 嵌入网页技术, 实现了网上任务分派问题数学实验的可视化。用户可由可视化界面, 输入任意任务分配问题, 得到最佳的分配结果及满意率。为学生学习掌握相关知识, 提供了具有可视化和交互性功能的学习使用平台

关键词: 互联网; 任务分配; Visual J++ 6.0; Applet 小程序; 匈牙利算法

Visualization Design of Mathematical Experiment for Task Assignments Problem Based on Internet

SHI Tong-Ju

(Department of Mathematics and Physics, North China Electric Power University, Baoding 071003, China)

Abstract: Visualizing design of mathematical experiment for task assignments problem is studied in this paper. For task assignments problem with as well as without constraints, the study completed the design of algorithm by making use of Microsoft visual J++6.0 as experimental development platform. Java language and applet embedded in Web page are employed to realize the development of visualization. The software can display the optimal task assignments and the satisfaction rate according to the user's input. The developed system can provide a study platform with functions of visualization and interaction for user to study the task assignments problem.

Key words: Internet; task assignment; Visual J++; Applet; Hungarian algorithm

1 前言

随着计算机和互联网的迅速普及, 数学实验已成为是现代数学教育的一个必不可少的实践环节。我校开发了“基于 Internet 的数学实验系统”, 使学生能在互联网上做数学实验。这种动脑又动手的主动参与过程, 激发了学生的求知欲, 使学生能够更直观、更深刻的理解掌握数学的概念、理论和方法, 更好地培养学生科学研究的习惯和创新意识。学生可在家中访问校园网上的数学实验系统, 不出家门就能做数学实验, 这样也达到了远程教育的目的。

本文中主要介绍“基于 Internet 的数学实验系统”中的一部分: 各种任务分派问题的算法、程序界面设计及各功能的实例演示。

在生产实践中, 经常会遇到这样的一般分派问题:

将若干个任务分派给若干个人, 使完成任务的效率最高或费用最小。高校每年的毕业生论文课题分派就属于这类问题, 而且在一般情况下, 课题数量大于学生人数, 目标函数是求最大满意程度。目前一般解决此问题的方法是匈牙利算法及改进方法。本文首先讨论了任务分派问题的相关算法: 匈牙利算法及有关的改进和拓展方法, 并针对各算法进行可视化的程序设计。通过实例测试, 用户该软件的可视化界面, 输入要分派的任务、人员及权数矩阵, 即可求得目标函数的最优解, 并显示最佳分派方法及打印功能等。

2 算法原理及分析

2.1 匈牙利算法基本原理

匈牙利算法是求目标函数极小值的方法。它的权

^① 收稿时间:2011-03-17;收到修改稿时间:2011-04-28

数矩阵的元素都是非负的。并可转化为任一行和列中的最小成本元素为0。转化步骤为:

第1步: 选出分派矩阵的每行中的最小元素, 并从该行的所有元素中减去它, 这将在矩阵的每一行中至少产生一个零元素。

第2步: 选出每列的最小元素, 并从该列的所有元素中减去它。这一步将在矩阵的每一列中至少产生一个零元素。

2.2 匈牙利算法步骤如下:

第1步: 由有零元素最少的行开始, 圈出一个零元素, 将它圈起, 然后划去同行同列的其它零元素。这样依次做完各行, 已划去的就不能再圈了。如果这样能得到 n 个被圈起的零, 就得最优解。如果圈出的零不够 n 个, 继续执行第二步。

第2步: 作能覆盖所有零元素的最少的直线集合。

(1) 没有零元素的行打对号; (2) 对已打对号的行中所有含零元素的列打对号; (3) 对打有对号的列中有零元素的行打对号; (4) 重复(2), (3) 直到得不出新的打对号的行、列为止; (5) 对没有打对号的行划一横线, 有打对号的列划一纵线, 就得到覆盖所有零元素的最少直线数。

第3步: 若最少覆盖直线数小于矩阵维数, 则作第4步; 否则, 改变方案, 重新探测。

第四步: 有线通过的元素称为“被覆盖”元素, 没有线通过的元素称为“未覆盖”元素。选出矩阵中未覆盖的最小元素, 并从所有未覆盖元素中减去它。然后将这个元素加到两条覆盖线交叉处的各个覆盖元素上。

第5步: 重复第1到第3步, 直到找到一个最优分派^[1]。

2.3 匈牙利算法改进:

匈牙利算法是本身只适用于权数矩阵是 $n \times n$ 的方阵的情况。而一般情况下, 任务和人员的数目不一定恰好相等。此时就要对矩阵进行改造, 如添加空的任务, 或添加不存在的人员, 使权数矩阵变成 $n \times n$ 的方阵。另外, 当目标函数最大值时, 采用权数矩阵中的最大值减去各个元素, 经该变换后, 再利用匈牙利算法, 所得的最优解即原问题的最优解。

2.4 有约束任务分派问题算法原理及分析

在生产实际中, 有时会出现多人被分派去完成一个任务, 或者多个任务被分派给一个人的情况。我们

目标是如何给人员分派任务, 使得全部任务所花的总时间最少。并注意任何人员分派的一组任务所需时间, 不超过工人可用时间的限制。这类问题称为有约束条件的分派问题。

算法步骤:

第1步: 将问题写成矩阵格式。

第2步: 置 $k=1$ 。

第3步: 可用时间 a_k 与列的最小所需时间 (c_i, k) 比较, 如果 $\min(c_i, k) > a_k$, 划去列 k 。

第4步: 对 $k=2, 3, \dots, n$, 重复第三步。

第5步: 按照该行最小与次最小时间的差, 确定剩余矩阵的罚数, 并将此罚数填在每行的右边。

第6步: 对具有最大罚数的行中的最小成本元素作一分派, 并且这个分派不超过工人可用时间 a_j 。

第7步: 划去第6步所作分派的行, 减少有关可用时间。

第8步: 重复第2步到第7步, 直到要么所有的工作都被分派完, 要么所有列(人员)都被标记完成。

如果有多于一个的行拥有的零元素个数相同时, 采用如下探测方案: 从拥有两个零元素的行开始, 对于其它拥有大于两个零元素的行仍一律选择第一个零元素, 每次重新探测时只是改变拥有两个零元素的行中选择各零元素的顺序, 若仍不能得到最优解, 则固定拥有两个零元素的行中零元素的选择顺序, 而探测拥有三个零元素的行, 依次类推。这样组合顺序进行探测, 一定能得到最优解^[2]。

由于匈牙利算法需要对权数矩阵做许多标记, 并且当探测失败后, 要退回去, 改变方案重新探测, 所以要记录许多标记和中间状态。需要定义大量标志变量和数组, 变量定义时采用“见名知意”原则, 以方便编写和调试程序, 避免不必要的麻烦。

3 可视化过程的实现

3.1 编程语言的选择

选择 Visual J++6.0 作为开发平台, 并选择纯 Java 语言作为软件的编程语言, 这能够实现软件的简介、高效。

3.2 程序的界面设计

① Applet 界面设计

该界面用 Java Applet 编程实现, 使用 `this.setLayout(null)` 进行控件布局非常灵活, 可以自由设定各个控

件的位置,还可以通过 reshape 自由设定各个控件的大小。程序主界面设计详见图 1 所示^[3]。



图 1 任务分派问题可视化界面

② 网页界面的设计

首先将 Applet 小程序嵌入到网页中,再在网页中写明实验题目、实验原理、方法说明、使用说明、问题及结论等。页面的设计用 Microsoft Office FrontPage 2003 作为开发环境

3.3 程序设计步骤

① 确定用户通过界面要输入的变量和要输出的结果。输入:要分配的任务名称、人员名称及权数矩阵。输出:在“任务分派框”内显示分派结果和满意率。设定打印按钮,以打印输出结果。

② 按钮选择包括:一般分派问题和有约束的任务分配问题;下拉列表框中可选择求目标函数最大值或最小值。若是求解有约束分派问题,则可用时间文本框变成可用状态,输入数据即可。若选择了求最大值,则调用相应函数,将权数矩阵作处理,使之变成求最小值问题。调用相应求解方法求解。

③ 将程序嵌入到编制好的网页,以实现真正的

Applet。

3.4 程序的输入和输出

3.4.1 程序的输入

在图 1 左边三个文本域分别输入要分配的任务名称、人员名称及权数矩阵。以空格作为识别符号,提取各名称;矩阵的行间各元素以空格分隔提取,不同行以回车符分隔提取。右上角两个单选按钮,用于选择一般分派问题或有约束任务分配问题,下方的下拉按钮可用于选择求目标函数最大值或最小值。

3.4.2 程序的输出

① Applet 的输出:在图 1 的右下方设定文本框输出最佳的任务分派结果及满意率。

② 网页显示效果:在网页中主要包括两部分内容:文字说明和 Applet 程序。在文字说明部分主要由实验题目、实验原理方法步骤及使用说明等。Applet 程序部分如上所述^[4]。

4 结论

本次设计实现了任务分配问题数学实验的可视化,通过用户输入任务、人员及权数矩阵,完成了输入数据的提取功能、输入的差错控制功能、求得目标函数的最优解并显示最佳分派方法,及打印功能等。通过实例测试,该程序使用方便、运算准确,且具有很好的通用性、灵活性及良好的交互性能。达到了预期的设计目标。为学习和使用者提供了具有交互性和可视化功能的学习平台。

参考文献

- 1 武汉大学,山东大学,计算数学教研室.计算方法.北京:高等教育出版社,1979.
- 2 《运筹学》教材编写组.运筹学.第 2 版.北京:清华大学出版社,1990.
- 3 清源计算机工作室.Visual J++6.0 开发宝典.北京:工业出版社,2003.
- 4 魏泱,等.Visual J++程序设计.北京:北京航空航天大学社,2002.