

基于知识结构的运筹学模型和方法库设计

郭晓军 宋朝霞 (四川省西南石油学院)

摘要:本文从运筹学的知识结构入手,分析和设计了适应其知识结构的模型和方法库,并探讨了以模型和方法库为核心建立应用系统的途径。

一、概述

运筹学和计算机有着密切的关系,没有计算机作为运筹学的有力工具,大多数运筹学技术的实现是难以想象的。

目前,解决运筹学问题的计算机软件很多,概括起来主要有两类:其一,独立运行解决某一类运筹学问题的单个程序(如解决线性规划问题的软件,其不单可以求出最优解,且可以进行灵敏度分析等);其二是能求解常用运筹学模型的软件包。但这两类软件都有一个共同特点,即每一个能运行的程序模块都是针对一个模型的一种具体求解方法,模型和方法之间的关系是“1-1”关系。这就造成该类软件的模型和方法扩充性较差,从而限制了在其上开发应用软件。

由于运筹学知识结构中有“1-N”和“N-1”,即一种模型可用多种方法求解,和多种模型可以使用同一方法求解这样一种构造特点,所以,我们在认真分析运筹学知识结构特点的基础上,设计了基于知识结构的运筹学模型、方法库系统,并探讨了在此基础上建立各种应用软件的途径。这种基于知识结构的运筹学模型、方法库系统具有下述特点:

- (1)本质上反映了运筹学模型、方法之间内在关系;
- (2)模型方法均具有较大的独立性;
- (3)可以直接裁剪出模型、方法库的全部或部分作为决策支持系统的基础,或作为其它应用软件的核心部件;
- (4)模型、方法库系统可以方便地作为其它应用软件的开发平台。

二、系统总体构思

考虑到运筹学模型、方法的三类用户:科研人员、教师

和学生,我们设计了如下运筹学模型、方法库系统,如图1所示。

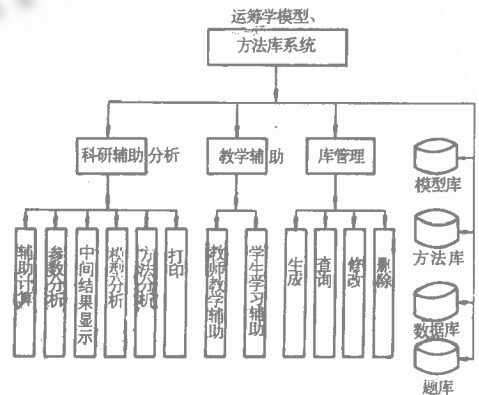


图1 系统总体结构图

说明:(1)科研辅助分析帮助运筹学研究人员进行运筹学题目的计算,模型、方法的分析,模型重要参数的分析。其中的模型分析是用户运用本系统的模型库提供的各种模型对原问题分别建模,并算出各种模型的结果,从而寻找一种最优的模型设计,见图2;方法分析对同一模型选择多种方法计算时,各种方法在计算精度、运行速度、占用存储容量等方面的比较,是选择最优方法的分析手段,见图3。

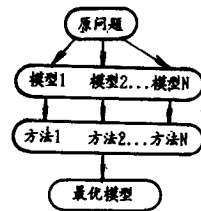


图2 模型优化分析示意图

(2)辅助教学提供运筹学的教师教学辅助和学生学习

辅助两大功能。为教师提供诸如教材选择、参考书选择、出练习题、出考卷、评阅试卷等方法的服务功能。辅助教学功能中凡与模型、方法有关的内容均在模型、方法库的直接支持下,通过调用模型、方法来完成。

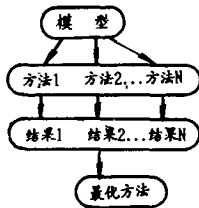


图3 方法优化分析示意图

(3)库管理主要提供模型库、方法库、题库、数据库的生成、查询、删除等功能。

(4)模型库是由相对独立的各种运筹学模型构成。

方法库是由各自相对独立的各种运筹学方法构成。

题库存放关于运筹学教学用的各种例题、习题、考题和标准答案。

数据库存放与前三库有关的数据,包括数据字典等;另外存放有关的科研用数据以备调用。

整个系统的运行借助于科研辅助分析、教学辅助、库管理三大子系统菜单界面,用户通过“点菜”并回答少许的屏幕问题,即可进行运筹学题目的分析和计算。模型和方法是本系统的核心和难点,下面将重点进行讨论。

三、模型库、方法库的分析设计

借助于“现实世界-信息世界-数据世界”这样一种思维方式,我们可以得到如下示意图,见图4。

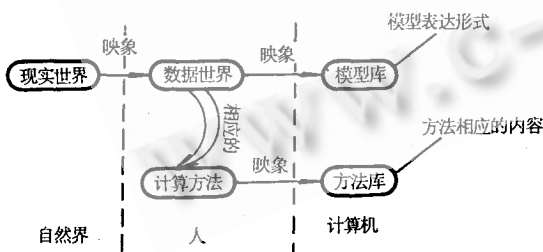


图4 模型、方法库映象示意图

从图中可以看出,计算机内组织的模型、方法库实际上只是我们人的大脑世界中的运筹学模型和计算方法在计算机内的一种映象。这种映象可以通过程序模型来构

造,通过程序的运行来实现。

1.模型库分析设计

通过对图4的分析,我们认为模型库可通过存放各种模型的输入、输出格式、数据进入方法库计算或运行时需要的变量管理程序等来实现。另外,模型库还要存放一些关于模型的知识,如对模型的讲解,模型求解步骤说明等。模型库中的各模型相对独立,结构图见图5。

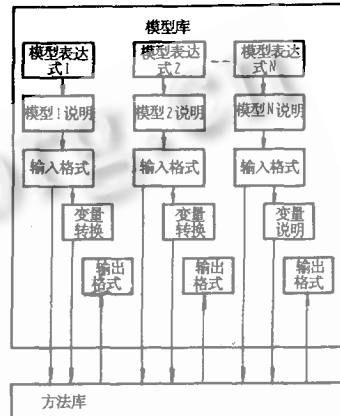


图5 模型库结构图

(1)模型表达式用于提供模型的直观映象,是每一个具体的模型呈现在用户面前的形式。模型表达式可以是数学模型,也可以是图表形式等等。

例如,线性规划模型呈现给用户的形式就是如下数学模型:

$$\text{Max(或 Min)}Z = C_1X_1 + C_2X_2 + \dots + C_NX_N$$

约束条件

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1N}x_N (*) b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2N}x_N (*) b_2$$

$$a_{M1}x_1 + a_{M2}x_2 + \dots + a_{MN}x_N (*) b_M$$

(2)模型说明部分包括与模型有关的知识,如对模型实际意义的解释、参数说明、适用范围、条件、模型符号说明、模型的求解方法和步骤概述等。

如线性规划模型的参数说明如下:

C_j 是目标系数; X_j 为未知变量;

a_{ij} 为模型系数; b_j 为约束常量;

(*)表示大于、小于、等于号之一。

(3)模型输入格式是指模型库给用户提供的数据输入格式。可以是一问一答的交互式格式,也可以是图表输入格式。它是模型库参数的主要输入口。

(4)模型变量转换是为模型调用多种方法而准备的。功能是将模型使用的变量个数和变量名与所调用方法使用的变量个数统一起来。图6是变量转换示意图。

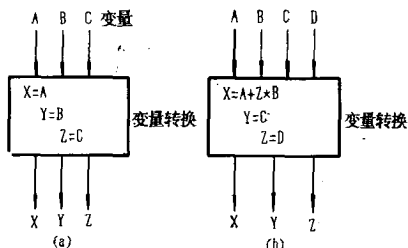


图6 变量转换示意图

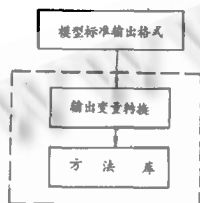


图7 模型输出格式示意图

(5)模型输出格式为保证模型调用任一方法返回结果的输出格式一致而设计的模块,它由两部分组成,一是输出变量转换;二是模型标准输出格式。见图7。

2.方法库分析设计

方法库是本系统运行的核心部件,担任着本系统的全部计算工作。它应提供如下功能:每一种方法模块均能和相应的模型配合正确运行;同时能为模型提供相应的分析算法(如模型参数分析);每一方法在一定条件下可以脱离模型独立运行。

图8是方法库结构图:

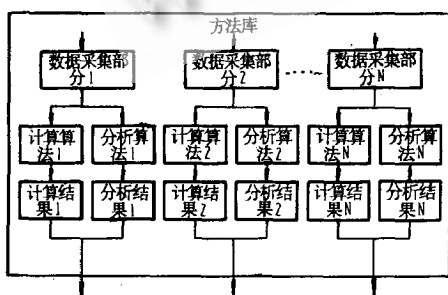


图8 方法库结构示意图

方法库中每一种方法的构成模块说明如下:

(1)数据采集部分。这一模块是考虑到方法输入数据来源多样,保证方法脱离模型独立运行而设计的。可以将来自模型库、数据库、屏幕(当独立运行输入数据时)的数据按运行需要的数据格式接受下来。

对来自模型库的数据,数据采集部分原样接受。对来自数据库的数据,由于是按调用方法要求的格式存储,故数据采集部分只需从库中读出数据即可;对来自于屏幕的数据,要求用户要熟知方法库中所用方法需输入数据的全部知识,能通过屏幕直接将数据赋给所调用方法,也能从屏幕读出计算结果。这里数据采集部分给用户提提供调用方法简单的输入输出格式。

(2)计算算法。在这里是纯粹的计算程序,只负责计算,其数据的输入和输出主要由模型来完成。

(3)参数分析部分。模型库内的有些模型除了要求对指定参数进行分析,还要求对重要参数进行分析,例如,线性规划模型要求对结果进行灵敏度分析等。参数部分就是提供这一功能。

(4)方法库输出结果。从图8可知,方法库输出的结果主要有两种,一是计算结果,二是分析结果;除此之外,在方法计算过程中还有一些中间结果,这可根据用户要求随时提供。

四、结束语

由于模型库和方法库设计中采用了高度模块化方法,所以保证了库内的每一个模块都是一个标准的积木部件,任何需用到运筹学模型或方法的用户均可从库中方便地取出自己需用的模块,构造自己的应用系统。用户也可以根据自己需要重新组合模型方法库,或对其赋与新内容,在其基础上开发出自己的应用软件系统。

另一方面,我们认为用计算机语言来实现这一设计思想并不困难,象目前流行的C语言、FoxBASE、FoxPRO等均可实现模型、方法库。为了更有效地实现本设计构思,充分发挥各种计算机语言的长处,实现过程中同时采用几种语言将更为有效。

参考文献:

- [1]李德、钱颂迪主编《<运筹学>>,清华大学出版社,1982年
- [2]B.E吉勒特《<运筹学导论>>,机械工业出版社,1982年