

优化建模软件 LocalSolver 简介

万保成,牛太阳,王田娥

(吉林农业大学信息技术学院,长春 130118)

摘要:

优化建模工具 LocalSolver 是近年出现的一个基于局部搜索的商业软件,依赖其特有的局部搜索技术,对于组合优化问题表现出强大的求解能力。介绍该软件的主要特点、基本的工作流程和同 C 语言语法的不同之处,并用经典的背包问题和指派问题两个实例来说明 LocalSolver 的用法和求解能力。

关键词:

优化建模; LocalSolver; 背包问题

基金项目:

吉林农业大学校级重点教研课题(No.2013xjzd057)、吉林省高等教育教学研究课题项目、吉林省青年科研基金(No.201201095)

0 引言

作为培养创新性人才的重要环节,数学建模在各高等院校倍受青睐,越来越多的大学生参加数学建模竞赛活动。纵观历届数学建模竞赛题目,许多都可建成优化模型,大部分都是用数学规划表示。时至今日,所建模型的决策变量和约束条件动辄成千上万,使用在线性规划等课程上所学的图解法或使用单纯型法自行编程进行求解要么不可能,要么稳定性差,均不实用。使用现成的商用或开源数学软件是一个很好的选择。这样做的好处是学生可以把更多精力放在建立优化模型上。当然也不是说求解的事情全都扔给软件去做,实际上也要结合软件特点来建立优化模型,否则有时达不到预期的效果。根据问题的不同,不同的软件对于所建模型也有不同的适用性。这是学生在学习软件时要注意的事情。

1 LocalSolver 的特点和工作流

LocalSolver^[1]建模工具包含了一个 LocalSolver 编程语言(LocalSolver Programming language,LSP)和一个内置的高效求解器。LSP 语言是一门描述性、命令式、结

构化、动态的解释型建模语言,它擅长求解高度非线性的 0-1 规划模型,特别适用于大规模组合优化和次序优化问题。

LocalSolver 提供了命令式、结构化编程语言具有的字面量、变量、LSP 表达式、数据类型(整型、字符型、浮点型、布尔值、映射)、程序结构(顺序、分支、循环)等概念,只要事先学过一门通用编程语言,例如 C、Java、VB 等,就可以对 LSP 快速上手。当然毕竟这门语言是专用于优化建模的,它所具有的描述性和动态的特点需要学生学习时注意。实际上,LSP 的语言可以归纳为顺次执行的 5 个预定义函数组成的建模框架:

- input():声明数据或从文件中读取数据
- model():声明优化模型
- param():为局部搜索求解器指定参数
- display():在求解期间输出信息到控制台或某些文件
- output():在求解完成时输出信息到控制台或某些文件

所有内建变量和函数除了 model()外都有默认值。

2 LocalSolver 的语法

LSP 语言的目标是建立 LocalSolver 模型的语法树

并对解析过程参数化。模型用 LSP 提供的运算符和关键字 (minimize, maximize, constraint) 来定义。为此, LSP 语言提供了一系列函数来进行命令式和结构化编程 (循环、条件、变量等)。总体上看, LSP 语言与 C 语言比较相似。例如, 变量名和函数名等标识符的构成, 标识符大小写敏感, 语句块用一对花括号“{}”, 语句结束用“;”, 注释用“//”和“/* */”, 转义字符, 数值的表示, 数据类型如整数、浮点数、布尔值、字符串, 全局变量, 局部变量, 变量的作用域, 顺序结构, 分支结构, 循环结构等同 C 语言大致一样。C 语言中没出现的语法现象或类型详述如下。

● Nil

Nil 是一个特别的类型。该类型的单个值通过关键字 Nil 访问。它在标记一个缺省值时特别有用, 并在函数没有明确的返回值时作为默认返回值, 还作为局部和全局变量的默认值。

● 映射

映射是一种关联式容器, 一个键对应一个值。键可以是整数 (不一定是连续的或正的) 也可以是字符串, 值可以是除了 Nil 外的任何类型的值。获取或重新设置一个键对应的值可以使用“方括号”运算符。例如, a[k] 返回映射 a 中键 k 对应的值。如果没有值与指定的键关联就返回 Nil。设置一个键对应的值将会覆盖该键以前对应的值。映射既可以是关联表也可以是数组。空映射使用 map() 或 {} 来创建。一个映射可以通过在花括号中罗列一系列值来创建, 此时其对应的键顺序为 0, 1, …。

● LS 表达式

LS 表达式是 LSP 程序中所建数学模型的变量或表达式, 仅能使用连接运算符 <- 引入和操作。只有通过这个运算符引入和操作的表达式能够传给求解器进行求解。

● 局部变量

局部变量仅仅在定义它的程序块或被这个块包含的块中可见。用户要能使用 local 关键字来引入。

● 内存管理

解释器使用引用计数来管理内存。每个值都关联一个引用计数器。这个计数器根据这个值上的操作进行增或减。当引用计数器为 0 时, 关联的值将被销毁。

3 示例

下面以著名的背包问题^[2]为例简单说明 LocalSolver 的编程过程。背包问题的定义如下: 任给一组物品 I_1, I_2, \dots, I_n , 假定物品 I_i 具有体积 s_i 和价值 c_i , 另给定一个容积为 S 的背包。目标是挑选 n 个物品的一个子集使得该子集中的物品都可以放进背包中, 且这些物品的价值之和最大。背包问题是一个组合优化问题, 属于 NP 完全类。如果用 0-1 变量 x_i 表示是否将物品 I_i 放进背包中:

$$x_i = \begin{cases} 1, & I_i \text{ 放进背包} \\ 0, & \text{否则} \end{cases}$$

由此背包问题可以表述为如下 0-1 规划问题:

$$\max c(x) = \sum_{i=1}^n c_i x_i$$

$$s.t. \sum_{i=1}^n s_i x_i \leq S,$$

$$x_i \in \{0, 1\}, i=1, \dots, n$$

下面是背包问题的小实例: $n=8$, 物品体积依次为 10, 60, 30, 40, 30, 20, 20, 2, 价值依次为 1, 10, 15, 40, 60, 90, 100, 15, 背包的体积为 102。LSP 代码如下:

```
function model() {
  x_0 <- bool(); x_1 <- bool(); x_2 <- bool(); x_3 <- bool();
  x_4 <- bool(); x_5 <- bool(); x_6 <- bool(); x_7 <- bool();
  knapsackWeight <- 10*x_0 + 60*x_1 + 30*x_2 + 40*x_3 +
  30*x_4 + 20*x_5 + 20*x_6 + 2*x_7;
  constraint knapsackWeight <= 102;
  knapsackValue <- 1*x_0 + 10*x_1 + 15*x_2 + 40*x_3 +
  60*x_4 + 90*x_5 + 100*x_6 + 15*x_7;
  maximize knapsackValue;
}
```

模型的所有变量, 也称为表达式, 用左箭头 <- 进行声明。决策变量, 只能是布尔变量 (0 或 1), 通过数学运算符 bool 引入。中间变量是通过其他运算符建立在这些决策变量之上; 乘积 (*), 和 (+), 小于等于 (<=)。大量其他数学运算符是可用的, 允许我们建立并且求解高度非线性的组合优化问题。关键字 constraint 或者 maximize 分别用来标记表达式是约束或者最大化的目标。

为了求解这个模型, 先把上述代码存储到 toy.lsp

文件中,然后在控制台命令提示符后输入 LocalSolver toy1.sp,回车即可。在控制台上会显示模型的主要特征:表达式数目、操作数、决策变量、约束条件、目标函数。接着罗列求解器的参数:时间限制、线程数、模拟退火水平等。每1秒,LocalSolver 输出已使用的时间、迭代数目和当前最优目标函数值。

下面再看一例。在 MatLab³中调用 LocalSolver 求解器,用于测试 LocalSolver 求解指派问题⁴的效率。MatLab 代码如下:

```
time=10; % 指定 LocalSolver 的运行时间,单位秒
m=200; % 任务数
n=m; % 人数
cost=ceil(100*rand(m,n)+5); % 随机产生费用矩阵
dlmwrite('time.txt',time); % 写到文件中,在 LocalSolver 中读取
dlmwrite('m.txt',m); % 同上
dlmwrite('n.txt',n); % 同上
dlmwrite('cost.txt',cost,' '); % 同上
! localsolver assign.bp % 调用 LocalSolver 求解
[assign,mc]=assignmentproblem(cost); %调用 MatLab 中本
地的指派问题求解程序
```

下面是 assign.lsp 文件的内容:

```
function input()
{
    f=openRead("time.txt");
    time=readInt(f);
    close(f);
    f=openRead("m.txt");
    m=readInt(f);
    close(f);
    f=openRead("n.txt");
    n=readInt(f);
    close(f);
}
```

参考文献:

[1]LocalSolver 帮助文档. <http://www.LocalSolver.com/>
 [2]堵丁柱,葛可一,胡晓东. 近似算法的设计与分析. 北京: 高等教育出版社, 2011
 [3]MatLab 帮助文档. <http://www.mathworks.cn/>
 [4]Hamdy A. Taha. 运筹学导论初级篇(第8版). 北京: 人民邮电出版社, 2008

作者简介:

万保成(1977-),男,河南唐河人,博士研究生,副教授,研究方向为计算机代数、优化建模
 牛太阳(1991-),男,河南焦作人,在读本科,专业为信息与计算科学
 王田娥(1977-),女,山西平定人,硕士,讲师,研究方向为优化建模
 收稿日期:2013-12-16 修稿日期:2013-12-30

```
f=openRead("cost.txt");
for[i in 1..m][j in 1..n]
    cost[i][j]=readInt(f);
close(f);
}
function model(){
    x[1..m][1..n]<-bool();
    for[i in 1..m]
    {
        C1[i] <- sum[j in 1..n](x[i][j]);
        constraint C1[i] ==1;
    }
    for[j in 1..n]
    {
        C2[j] <- sum[i in 1..m](x[i][j]);
        constraint C2[j]==1;
    }
    obj <- sum[i in 1..m][j in 1..n](cost[i][j]*x[i][j]);
    minimize obj;
}
function param()
{
    lsTimeLimit=time;
    lsNbThreads=3;
    lsAnnealingLevel=9;
    lsVerbosity=1;
}
```

4 结语

总之,LocalSolver 是一个功能强大、容易掌握、不断发展的优化建模软件。在数学建模竞赛中,它既可用于解模型,也可用于模型的验证。希望通过学习,能够在数学建模中得到适当且灵活的运用。

Introduction of LocalSolver Optimization Modeling Software

WAN Bao-cheng, Niu Tai-yang, WANG Tian-e

(College of Information Technology, Jilin Agricultural University, Changchun 130118)

Abstract:

LocalSolver, the optimization modeling tools, is a commercial software that emerged in recent years. Relying on high-performance local-search techniques, it shows the strong ability of solving combinatorial optimization problems. The basic characteristics, introduces the basic workflow and the grammar of this software. Takes knapsack problem and assignment problem as examples to illustrate the usage and power of LocalSolver.

Keywords:

Optimization Model; LocalSolver; Knapsack Problem



(上接第 49 页)

How to Develop Computational Thinking Ability in Computer Basis Teaching

LI Lin¹, WANG Cheng², ZHAO Feng²

(1. School of Computer, Wuhan University of Science and Technology, Wuhan 430062;

2. Department of Common Required Courses, Hubei Institute of Fine Arts, Wuhan 430205)

Abstract:

On the basis of analyzing the computational thinking at home and abroad, proposes some effective methods to culture the computational thinking ability of students, for example, deeply expounds the basic principles, uses some vivid examples to guide the students, changes examination-driven memory to applied-type comprehension, and makes good uses of the convenience of mainstream network communication methods to provide differential teaching for students. Applies computational thinking to the practice of Computer Basis course teaching, finds it can effectively improve the comprehensive ability and quality of students, get excellent teaching effect.

Keywords:

Computational Thinking; Computer Basis Course; Teaching Reformation