第30卷 第13期 计 算 机 工 程 2004年7月
Vol.30 № 13 Computer Engineering July 2004

·软件技术与数据库· 文章编号：1000—3428(2004)13—0069—03 文献标识码：A 中图分类号：TP319

线性规划软件包GLPK的分析与应用

陈 慧，谷寒雨
（上海交通大学自动化研究所，上海 200030）

摘 要：GLPK是一个求解大规模的线性规划问题(LP)、混合整数规划问题(MIP)以及相关问题的自由软件包。该文分析了GLPK的算法结构与数值计算等多方面的实现技术，并应用于解决NP-hard的调度问题。数值结果表明GLPK是研究LP和MIP问题强有力的工具。

关键词：GLPK；线性规划；混合整数规划；调度；LP松弛模型

Analysis and Application of Linear Programming Software Kit GLPK

CHEN Hui，GU Hanyu
(Dept. of Automation, Shanghai Jiaotong Univ., Shanghai 200030)

【Abstract】GLPK is a package intended for solving large-scale linear programming (LP), mixed integer linear programming (MIP) problems. The article analyzes the algorithmic and numerical aspects in developing GLPK, and applies it to solve the NP-hard scheduling problem. The numerical tests lead to the conclusion that GLPK is a valuable tool for studying LP and MIP problems.

【Key words】GNU Linear programming kit(GLPK)；LP；MIP；Scheduling；LP relaxation model

调度是研究如何在一段时间内为相互竞争的活动分配稀缺资源。在经济全球化背景下，调度在制造业、服务业、计算机通信等行业的广泛应用对提高企业竞争力至关重要。然而除少数问题外，实际调度问题大都是NP-hard的，这也是调度问题在应用数学、计算机科学、控制等领域受到广泛关注的重要原因。近年来多面集理论（polyhedral theory）已迅速发展成为研究调度问题的有力工具，它把调度问题转化为一线性规划问题，从而可利用现有的线性规划软件包求解。

近几十年来，随着计算机和算法理论的发展，准确快速地解决大规模线性规划问题已成为可能，然而高品质的线性规划软件产品往往价格十分昂贵。另一方面，有些问题需要针对具体要求设计专门的优化算法，显然那些不开放源码的软件难以满足需求。

GLPK (GNU Linear Programming Kit) 是包含在GNU系统中的一个线性规划软件包，可以用来求解大型的线性规划问题(LP)和部分混合整数规划问题(MIP)。作为一个自由软件，GLPK不但免费而且开放完整的程序源码。本文通过分析其算法原理和实现技术，并应用于调度问题的研究，来对其实际性能做一评估。

1 GLPK软件包分析

GLPK[1]是由一组用标准C写成的函数组成的库，完全遵从ANSI标准，确保了软件的可移植性。GLPK的开发者遵照自由软件发布惯例提供了易于下载解压的发布包、标准详尽的文档、以及围绕GLPK建立的网上社区。

1.1 GLPK的算法结构

线性规划问题有各种不同的形式，为了便于计算求解，GLPK 规定了一种标准形式，可表示为

$$
\begin{aligned}
\min(\max) \quad & Z = \hat{c}^T x_s + c_0 \\
& x_R = \hat{A} x_S \\
& l_R \le x_R \le u_R \\
& l_S \le x_S \le u_S
\end{aligned}
\tag{1}
$$

其中 $x_R \in R^m$：辅助变量向量；$x_S \in R^n$：结构变量向量；$\hat{c}$：目标函数系数向量；c_0：目标函数常数项；$\hat{A}$：约束矩阵；l：变量下界向量；u：变量上界向量。

很容易把任意线性规划数学模型都转化为GLPK的标准形式：如果约束为等式，那么约束左边的辅助变量就取值常数，它的上界等于下界；如果约束为不等式，那么把原不等式变为等式，根据约束确定辅助变量的边界，这样得到的标准型与原来的模型是完全等价的。

GLPK提供了单纯形法（simplex method）和内点法（interior point method）求解LP问题。自从1947年Dantzig首次发明单纯形法以来，单纯形法仍然是解决大部分实际线性规划问题最有效的算法之一。通过分析源代码，可以发现GLPK中单纯形算法的基本步骤是：

(1) 计算初始解，判断初始解是否原问题可行（primal feasible）或者对偶可行（dual feasible）。如果初始解即原问题可行，又对偶可行，则当前解就是最优解。如果初始解只是对偶可行，则执行步骤(2)，如果初始解非对偶可行，则执行步骤(3)。

(2) 运用对偶单纯形法(dual simplex)，从对偶可行的初始解出发，在保持对偶可行的条件下，迭代求得原始可行的解，即最优解。

(3) 运用两阶段法(two-phase method)，第1阶段判断原问题是否存在基可行解，如原问题可行，进入第2阶段计算，否则停止。第2阶段从第1阶段得到的基可行解出发，用单纯形法求得最优解。

GLPK采用了改进单纯形法(revised simplex method)，只对单纯形表的一部分元素进行迭代计算，略去了其他不必要的计算。这不仅提高了运算效率，而且节省了存储空间，对于求解大规模线性规划问题是很重要的。

内点法具有多项式时间复杂度，在实际应用中能有效地解决大规模线性规划问题。GLPK采用了基于Mehrotra 预估校正技术(Mehrotras predictor-corrector technique)的原始一对

基金项目：国家自然科学基金资助项目(60274031)
作者简介：陈 慧（1981—），女，本科生，研究方向：自动化；谷寒雨，副教授、博士
收稿日期：2003-06-04 **E-mail**：guhy@sjtu.edu.cn

偶内点法（primal-dual interior point method）。

MIP问题是NP-hard的，对于此类问题目前的计算理论和算法还不成熟。目前GLPK用结合了某些简单启发式算法的分支定界法(branch and bound method)来解MIP，只能解决部分小规模MIP问题。添加分支切割法（branch-and-cut method）和其它一些启发式算法可以提高GLPK求解MIP的性能。

1.2 GLPK的实现技术

(1) 平衡化方法(scaling method)

对标准形式(1)中的原始数据，GLPK会做如下形式转化：

$$\begin{aligned} \min(\max) \quad & Z = (\hat{c}')^T (x'_S) + c_0 \\ & x'_R = \hat{A}' x'_S \\ & l'_R \le x'_R \le u'_R \\ & l'_S \le x'_S \le u'_S \end{aligned} \tag{2}$$

其中，$\hat{A}' = R\hat{A}S, \hat{c}' = S^T\hat{c}$，

$x'_R = Rx_R$，$l'_R = Rl_R$，$u'_R = Ru_R$，

$x'_S = S^{-1}x_S$，$l'_S = S^{-1}l_S$，$u'_S = S^{-1}u_S$ 。

这样经过平衡化后的数据，相比原始数据具有更稳定的数值特性。

(2) 最陡边技术（Steepest edge technique）

在单纯形法中，当多个非基变量都满足进基条件时，如何选择进基变量呢？GLPK采用D.Goldfarb和J.K.Reid提出的最陡边技术：新的迭代方向对应于目标函数负梯度有最小夹角。这一启发式算法能大大提高单纯形法的效率。

(3) 哈里斯技术（Harris technique）

当多个基变量同时达到边界时，如何选择出基变量？这个问题在基解是退化的时候尤为突出。哈里斯指出，由于计算误差的存在，在退化基解的情况下，基变量的当前值可略微超过它们的边界。GLPK采用哈里斯提出的二关比例测试(two-pass ratio test)选择出基变量，不但得到更快的收敛速度，更优的数值稳定性，而且在遇到退化问题时，还可防止死循环。

(4) 稀疏矩阵支持（sparse-matrix support）

在GLPK中，约束矩阵是以稀疏矩阵的形式存储的，其中除了约束系数的数值，还有一些辅助信息。GLPK动态开辟和释放内存，对于包含稀疏矩阵所有非零元素的稀疏向量区域，就分配以大的存储块。稀疏矩阵支持技术使得GLPK可以解决非零约束系数上百万的大规模LP问题。

(5) 应用程序接口函数（API）

用GLPK求解具体的线性规划问题，首先要将原问题转换成GLPK的标准形式，然后就可以使用GLPK提供的接口函数按照“输入数据 -> 建立问题 -> 求解问题 -> 输出结果”的流程编写应用程序。新建一个LP问题对象之后，GLPK将问题涉及的数据以通用结构的形式加以组织、存储与调用，这对用户修改、扩充算法非常有利。GLPK提供了非常容易使用的用户接口函数，当用户为求解不同的线性规划问题而编写程序时，这些库函数可以被直接调用，大大提高了编程效率。各接口函数的参数与功能可参考GLPK的使用手册[1]。

(6) 建模语言GLPK/L

为了便于建模，GLPK还提供了专门的建模语言GLPK/L,详见GLPK/L的说明[2]。

2 GLPK在调度问题中的应用

对最小化加权平均完工时间的单机动态调度问题，文献[3]中利用LP松弛法进行了深入研究。此方法不仅能得到非常好的下界（lower bound），而且还可以利用LP的求解结果得到有性能保证的近似算法（approximation method），因此有很高的理论价值。但在应用此方法时，必须求解一个转化的线性规划问题，而且计算难度随问题规模迅速增加，文献中通常采用商用软件包CPLEX。通过求解此调度问题，并和CPLEX比较，可以较好地评估GLPK的实用价值。

2.1 调度问题描述

考虑只有一台机器的单机调度问题。设有n个需要加工的工件j,(j=1,…, n), 每个工件有一个非负的加工时间 p_j、非负的权重 w_j 以及到达时间 r_j。任何时刻机器只能同时加工一个工件，工件加工不能被打断。要求所有工件加权完成时间之和最小，此问题在文献中记作 $1|r_j|w_jC_j$。以下是此问题基于时间离散化的整数规划模型：

$$\min \sum_{j=1}^{n} \sum_{t=r_j+p_j}^{T} w_j t x_{jt} \tag{3}$$

$$\text{s.t.} \sum_{t=r_j+p_j}^{T} x_{jt} = 1 \quad (j = 1,\cdots,n) \tag{4}$$

$$\sum_{j=1}^{n} \sum_{s=t}^{\min(T,\, t+p_j-1)} x_{js} \le 1 \quad (t = 1,\ldots T) \tag{5}$$

$$x_{jt} = 0, \ (t = 1,\cdots,r_j + p_j - 1, j = 1,\cdots,n) \tag{6}$$

$$x_{jt} \in \{0,1\}, \ (t = r_j + p_j,\cdots,T, j = 1,\cdots,n) \tag{7}$$

其中 $T = r_{\max} + \sum_{i=1}^{n} p_i$ 。$x_{jt} = 1$ 表示第j个工件在t时刻加工完成。式(4)确保每个工件到达后必须完成，并且只完成一次;式(5)限制机器在任意时间段（t-1,t]内,最多只能加工一个工件;式(6)限制每个工件到达之前不被加工。

如果松弛整数约束式(7)，则得到 $1|r_j|w_jC_j$ 的 $x_{jt}-LP$ 模型，其最优解是一个很好的下界，文献[3]中对此有广泛深入的研究。

2.2 GLPK求解方法

如前所述，首先要将调度问题式(3)-式(7)转化为GLPK的标准形式。转化好的标准形式如下：

$$x_S^T = [x_{11}\, x_{12} \ldots x_{1T} \ldots\ldots x_{n1}\ x_{n2} \ldots x_{nT}]$$

$$l_S^T = [l_{11}\, l_{12} \ldots l_{1T} \ldots\ldots l_{n1}\ l_{n2} \ldots l_{nT}]$$

$$u_S^T = [u_{11}\, u_{12} \ldots u_{1T} \ldots\ldots u_{n1}\ u_{n2} \ldots u_{nT}]$$

$$x_R^T = [x_{R1}\ x_{R2} \ldots x_{Rn}\, x_{R(n+1)} \ldots x_{R(n+T)}]$$

$$l_R^T = [l_{R1}\, l_{R2} \ldots l_{Rn}\ l_{R(n+1)} \ldots l_{R(n+T)}]$$

$$u_R^T = [u_{R1}\ u_{R2} \ldots u_{Rn}\ u_{R(n+1)} \ldots u_{R(n+T)}]$$

$$c_0 = 0\ ;$$

$$\hat{c}^T = [w_1\ 2w_1\ 3w_1 \ldots Tw_1 \ldots\ldots w_n\ 2w_n \ldots Tw_n]$$

$$\hat{A} =$$

$$
\begin{bmatrix}
\underbrace{00\ldots0}_{r_1+p_1-1}\ \underbrace{11\ldots1}_{T-r_1-p_1+1}\ 00\ldots\ldots\ldots\ldots\ldots\ldots0 \\
00\ldots\ldots0\ \underbrace{00\ldots0}_{r_2+p_2-1}\ \underbrace{11\ldots1}_{T-r_2-p_2+1}\ 00\ldots\ldots\ldots0 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
00\ldots\ldots\ldots\ldots\ldots\ldots0\ \underbrace{00\ldots0}_{r_n+p_n-1}\ \underbrace{11\ldots1}_{T-r_n-p_n+1} \\
\underbrace{11\ldots1}_{p_1}\ \underbrace{00\ldots0}_{T-p_1}\ \underbrace{11\ldots1}_{p_2}\ \underbrace{00\ldots0}_{T-p_2}\ldots\ldots\underbrace{11\ldots1}_{p_n}\ \underbrace{00\ldots0}_{T-p_n} \\
0\underbrace{11\ldots1}_{p_1}\underbrace{00\ldots0}_{T-p_1-1}\ 0\underbrace{11\ldots1}_{p_2}\underbrace{00\ldots0}_{T-p_2-1}\ldots\ldots0\underbrace{11\ldots1}_{p_n}\underbrace{00\ldots0}_{T-p_n-1} \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\underbrace{0\ldots\ldots0}_{T-1}1\ \underbrace{0\ldots\ldots0}_{T-1}1\ldots\ldots\underbrace{0\ldots\ldots0}_{T-1}1
\end{bmatrix}
$$

$x_{jt}=0,\ l_{jt}=u_{jt}=0,\quad (t=1,\cdots,r_j+p_j-1,\ j=1,\cdots,n)$

$0\le x_{jt}\le 1,\ l_{jt}=0,\ u_{jt}=1,\quad (t=r_j+p_j,\cdots,T,\ j=1,\cdots,n)$

$x_{Ri}=1,\ l_{Ri}=u_{Ri}=1,\ (i=1,\ldots,n)$

$x_{Ri}\ \le 1,\ l_{Ri}=-\infty,\ u_{Ri}=1,$

$$(i=n+1,\ldots,n+T)$$

此松弛模型有n×T个结构变量，n+T个辅助变量，即n+T个约束，$\hat{\mathbf{A}}$ 是一个(n+T)×(nT)维的稀疏矩阵，其中非零元的个数为 $n(T+1)-\sum_{i=1}^{n}r_i-\frac{1}{2}\sum_{i=1}^{n}p_i^{\,2}+(T-\frac{1}{2})\sum_{i=1}^{n}p_i$ ，占全部元素的百分比很小(5%以下)。由于 $T\ge n$,当工件数目增加时，变量数目至少以 $O(n^2)$ 迅速增加。利用GLPK的应用程序接口，很容易对此LP问题建模并求解。

3 GLPK性能测试

为测试GLPK求解LP和MIP问题的性能，采用和文献[3]相同的测试方法。用于测试的问题分为3类：(1) OPT：此类问题规模较小，n=30； w_j 为1到10的随机整数； p_j 为 1~5或者1~10的随机整数； r_j 为均匀分布(参考文献[4])。(2) SYN：n= 50或100； r_j 服从泊松分布：平均每10个单位时间有5 或 20个工件到达； p_j 和 w_j 服从两种分布：1) 服从期望为5，标准差为2.5的正态分布；2) 服从二峰分布——以0.5的概率服从期望为2.5，标准差为0.5的正态分布；以0.5的概率服从期望为7.5，标准差为0.5的正态分布。(3) HARD：n= 50或100； w_j 为1到10的随机整数； p_j 为1 或20， p_j 为1的概率为0.9, p_j 为20的概率为0.1。测试结果见表1。

表1 测试结果

	OPT	SYN	HARD
平均变量数	5 541	37 909	39 630
平均约束数	215	518	552
约束矩阵非零元素个数	32 149	225 984	150 663
平均百分误差(%)	MIP: 0	0.001 7	0.001 2
	LP: 0		
平均计算时间(s)	MIP: 91	86.7	26.5
	LP: 2		

测试环境：版本GLPK 3.2.1/ 计算机：1200MHz，Pentium4, 256MB物理内存;操作平台:Windows XP。编译环境:Visual C++6.0。

百分误差根据 $\frac{BEST-LB}{LB}\times 100(\%)$ 计算，在OPT/MIP测试组中，BEST和LB的值分别为Van den Akker. Van Hoesel 和 Savelsbergh的分支定界程序以及GLPK的分支定界法求得的整数规划模型的解；在OPT/LP、SYN和HARD中，BEST和LB分别为CPLEX 和GLPK给出的LP松弛模型的解。

从表1可以看出，对小规模问题OPT、GLPK的整数规划算法能正确求得最优解。对规模较大的SYN和HARD问题，GLPK的计算精度与CPLEX相差无几。根据测试看，GLPK执行相当稳定，计算速度也比较快。

4 总结

本文介绍了开放源代码的线性规划软件包GLPK，分析了其算法原理和实现技术，并基于GLPK在单机调度问题中的应用，评估GLPK的实际性能。可以看出，GLPK已集成了线性规划大部分的先进计算方法，求解精度高，运算速度快，执行稳定可靠，具有很高的应用和研究价值。

参考文献

1 Makhorin A. GNU Linear Programming Kit Reference Manual(Draft Edition). http://www.gnu.org/software/glpk, 2002-07

2 Makhorin A. GNU Linear Programming Kit Modeling Language GLPK /L (Draft Edition). http://www.gnu.org/software/glpk, 2002-06

3 Uma R N, Wein J. On the Relationship between Combinatorial and LP -based Approaches to NP-hard Scheduling Problems. http://ebbets.poly.edu/PDC-lab/wein.html, 1997-12-02

4 Belouadah H, Posner M E, Potts C N. Scheduling with Release Dates on a Single Machine to Minimize Total Weighted Completion Time. Discrete Applied Mathematics,1992, 36:213-231

☆☆

(上接第43页)

6 结论

通过以上的分析可以看出，基于数据库的引信虚拟设计是引信设计发展的必然趋势，利用数据库强大的查询功能和数据处理能力，可以为设计人员提供高效的技术数据支持和专家智能的技术表达能力，并且在计算机上实现虚拟设计的全过程，可以提高设计效率、减少设计误差、在设计的起始阶段通过虚拟测试就可以找出存在的问题，在设计的中间阶段通过虚拟环境实现用户和设计的有效交互，在设计的完成阶段通过虚拟仿真技术验证设计的可靠性等原则并进行动力学、控制学、电磁干扰、可靠性、安全性等方面的分析。

参考文献

1 刘永红, 任工昌, 张优云. 基于网络的CAD标准件数据库实体建模. 计算机工程与应用, 2002,(16): 198-200

2 方喜峰, 赵良才. 基于Oracle数据库的产品质量数据管理系统. 华东船舶工业学院学报, 2000,(4): 57-60

3 秦浦雄, 刘 鹏. ODBC技术在工程CDA系统中的应用. 计算机工程与应用, 1998,(12): 74-76

4 陈光伟, 齐玉杰, 王慧强. 铁路客票发售与预定系统数据库访问安全性研究. 中国铁道科学, 2000(3): 101-104

5 Zingirian N, Maresca M, Nalin S. Efficiency of Standard Software Architectures for Java-based Access to Remote Databases. Future Generation Computer Systems, 1999,(15): 417-424

6 Course T. Using ADAMS/Solver. Part number 120SOLG-01