基于非线性规划的包装箱组托优化设计

王秀宇
（内蒙古交通职业技术学院，赤峰 024005）

摘要：目的 以相同规格包装箱为研究对象，求解包装箱组托码放规则的最优化方案。**方法** 以 H2DPP 模型为基础，采用块启发式算法，建立非线性规划模型，利用 Lingo 软件求解最优码放数量及各决策变量值。**结果** 在托盘规格一定的情况下，包装箱规格越大，目标函数值越小；托盘长宽与包装箱长宽分别成整数比时，迭代的次数要远小于其他情况。**结论** 利用 Lingo 软件求解非线性规划组托模型，用时不超过 0.05 s，该方案更高效、准确地解决了流通过程中包装箱的组托问题，实现了组托方法的最优化。

关键词：包装箱组托；非线性规划；迭代
中图分类号：TB489 **文献标识码：**A **文章编号：**1001-3563(2016)19-0006-06

Optimization Design of Packing Box Group Support Based on Nonlinear Programming

WANG Xiu-yu
(Inner Mongolia Vocation and Technical College of Communications, Chifeng 024005, China)

ABSTRACT: With the same specifications of packing box as the research object, the work aims to solve the optimization scheme for packing group support stacking rules. Based on H2DPP model, a nonlinear programming model was built by heuristic algorithm. The optimal number of pieces and each decision variable values were solved by Lingo software. In the case of a certain size of the tray, the larger the size of the packing box, the smaller the value of the objective function. The number of iterations was much smaller than the others when the length and width of the pallet and the length and width of the packing box both formed a ratio of integers. The nonlinear programming model is solved by Lingo software. With use time no more than 0.05 s, this scheme more efficiently and accurately solves problems with group support of packing box in circulation process and realizes the optimization of the group support method.

KEY WORDS: support group of packing box; non-linear programming; iteration

在包装箱数量一定的情况下，提高单个托盘的码放数量，可以减少托盘的使用量、节省存储空间、降低货位的占用率，还能降低运输工具的承载空间。可见，合理的货物组托可以降低物流成本、提高工作效率，在企业资源分配和利用等方面具有重要的经济意义[1—2]。

组托的优劣体现在是否能高效利用托盘的使用面积和承载限额[3]。它属于组合优化范畴，是一个具有多约束条件的最优化求解问题。影响包装箱组托的因素有很多，受包装箱尺寸大小、堆码层数、托盘规格、仓储条件、运输工具等诸多因素的制约，组托方案也会有所不同，不同的组托方案所带来的经济效益不同。怎样更高效利用托盘的承载面积，使尽可能多的包装箱置于托盘承载面上，如何摆放这些包装箱，如何利用现代化的计算机技术，求解出包装箱组托的最优方案是文中主要探讨的问题。

收稿日期：2016-07-30
基金项目：内蒙古社会科学研究课题（16C10）
作者简介：王秀宇（1983—），女，内蒙古赤峰人，硕士，内蒙古交通职业技术学院讲师，主要研究方向为物流管理、包装技术。

1　相关研究

1.1　H2DPP 模型

H2DPP 是货物组托问题最具代表性的模型，该模型是以以下 3 个假设为基础：托盘的横向长度要大于等于托盘的纵向长度；统一规格的包装箱长边大于等于宽边，高度暂不考虑；不考虑托盘的承载能力。要将统一规格的包装箱尽可能多地码放到托盘承载面且不允许挤压产生重叠，需将包装箱沿托盘的边缘进行码放，且只允许包装箱的长边与托盘的长边平行或垂直，不可有其他放置的可能性。

考虑包装箱不超过托盘边缘且不出现重叠的情况下，利用托盘的 4 个角，将托盘的承载面积以逆时针旋转的方法来减少空白位置，且保证包装箱以水平或垂直的方向进行码放。把托盘长边设为横轴，宽设为纵轴。以托盘左下角为起点，沿横轴横向放置包装箱的列数为 x_1，每列数量为 y_1；以托盘右下角为起点，沿横轴纵向放置包装箱的列数为 x_2，每列数量为 y_2；以托盘右上角为起点，沿横轴横向放置包装箱的列数为 x_3，每列数量为 y_3；以托盘左上角为起点，沿横轴纵向放置包装箱的列数为 x_4，每列数量为 y_4。同一边上的 2 个角码放的包装箱相互垂直，而在对角线的 2 个角码放的包装箱相互平行[4—5]，见图 1。

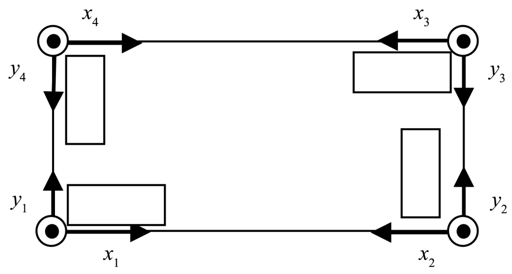


图 1　组托原理
Fig.1 Principle diagram of group support

H2DPP 最优解的迭代次数随着包装箱规格的减小而增大，导致计算时间几何级放大，因此实践过程中解决组托问题大多采用块启发式算法[6]。

1.2　块启发式算法

块启发式算法在解决包装箱堆码问题时，一般分为 1～8 块。块的数量与包装箱规格有关，包装箱规格越大，码放数量越小，可通过 1～2 块就能在较短时间求出相对最优解；反之包装箱规格越小，组托数量越大，就需要在第 2 块的基础上，用到 3～4 块模块才可在一定时间求出相对最优解。用到的模块越多，所需运算时间越长。根据 H2DPP 组托原理构建块启发式算法的数学模型[7]。

假定托盘和包装箱的长宽分别为 l_a，b_a，l_b，b_b。第 1 块运算逻辑：以托盘左下角为起点，包装箱的长度方向沿托盘横向放置直至剩余空间不可利用为止，码放数量为 x_1y_1；以托盘左下角为起点，包装箱的宽度方向沿托盘横向放置直至剩余空间不可利用为止，码放数量为 x_2y_2。第 2 块的运算逻辑：以托盘左下角为起点，以 0，1，…，(l_a/l_b)为列，以(b_a/b_b)为每列数量，沿托盘横向进行码放，码放数量为 x_1y_1；以托盘右下角为起点，以$[(l_a-l_bx_1)/b_b]$为列，以(b_a/l_b)为列数进行码放，码放数量为 x_2y_2；总的码放数量为 $x_1y_1+x_2y_2$。第 3 块、第 4 块的码放原理及码放数量的求解与第 2 块运算逻辑相同。迭代次数及最优解的取得受可行域上下限值的约束，下限值可通过第 1 块运算取小求得，上限值由$(l_ab_a)/(l_bb_b)$取整求得。以托盘承载面利用率最大为目标的目标函数为：

$$U_{max}=x_1y_1+x_2y_2+x_3y_3+x_4y_4$$

约束条件：

$$x_1\le[l_a/l_b]$$
$$y_1\le[b_a/b_b]$$
$$x_2=[(l_a-l_bx_1)/b_b]$$
$$y_2\le[b_a/l_b]$$
$$x_3=[(l_a-l_bx_1)/l_b]$$
$$y_3=[(b_a-l_by_2)/b_b]$$
$$x_4=[(l_a-l_bx_3)/b_b]$$
$$y_4=[(b_a-l_by_1)/b_b]$$
$$x_1y_1+x_2y_2+x_3y_3+x_4y_4\le(l_ab_a)/(l_bb_b)$$

式中：$x_i\ge0$，$y_i\ge0$ (i=1，2，3，4)，且为整数。

该块启发式的数学模型的使用存在一定的局限性，当对角线所码放的包装箱长度和或高度和超过托盘的长和宽时，无法继续迭代，故所求解未必是最优解。

1.3　非线性规划求解

非线性规划问题，其目标函数或约束条件中包含有非线性函数，求解过程相对线性规划要难得多，且可能要经过几万甚至几十万次迭代才能求出相对最优解，如果通过手工计算求解，几乎是不可能完成的[8—10]。利用 Lingo 软件对包装箱组托的非线性优化模型可快速求解，得出相对最优码放数量及码放规则。

2 组托优化的建模

2.1 建立非线性规划组托优化模型

2.1.1 参数设置

设置参数：l_a 为托盘表面的长度，即横轴；b_a 为托盘表面的宽度，即纵轴；h_a 为托盘的高度；l_b 为包装箱的长度；b_b 为包装箱的宽度；h_b 为包装箱的高度。

2.1.2 决策变量

决策变量：x_1，x_2，x_3，x_4，y_1，y_2，y_3，y_4。

2.1.3 目标函数

托盘承载面利用率最大，承载面积为 l_ab_a，包装箱底面积之和为$(x_1y_1+x_2y_2+x_3y_3+x_4y_4)l_bb_b$，则目标函数 $U_{max}=(x_1y_1+x_2y_2+x_3y_3+x_4y_4)l_bb_b/l_ab_a$。为了方便求解，在托盘规格及包装箱规格一定的情况下，还可以只求包装箱总数量 $x_1y_1+x_2y_2+x_3y_3+x_4y_4$ 的最大值，即目标函数转化为 $U_{max}=x_1y_1+x_2y_2+x_3y_3+x_4y_4$。

2.1.4 约束条件[11]

理论上每层托盘可放置包装箱最大数量可通过承载面积与包装箱底面积之商求得，即$(l_ab_a)/(l_bb_b)$取整求得，因此每层最多码放包装箱的数量要小于等于其上限值，即 $x_1y_1+x_2y_2+x_3y_3+x_4y_4\le(l_ab_a)/(l_bb_b)$；确保以托盘左下角为起点码放的包装箱，与以托盘右下角为起点码放的包装箱的尺寸和要小于托盘的横轴，但沿托盘边缘码放的剩余空间要大于 1 个包装箱的宽，即 $(l_a-b_b)<(l_bx_1+b_bx_2)<l_a$；确保以托盘右下角为起点码放的包装箱，与以托盘右上角为起点码放的包装箱的尺寸和要小于托盘的纵轴长度，但沿托盘边缘码放的剩余空间要大于 1 个包装箱的宽，即 $(b_a-b_b)<(l_by_2+b_by_3)<b_a$；确保以托盘右上角为起点码放的包装箱，与以托盘左上角为起点码放的包装箱的尺寸和要小于托盘的横轴长度，但沿托盘边缘码放的剩余空间要大于 1 个包装箱的宽，即 $(l_a-b_b)<(l_bx_3+b_bx_4)<l_a$；确保以托盘左下角为起点码放的包装箱，与以托盘左上角为起点码放的包装箱的尺寸和要小于托盘的纵轴长度，但沿托盘边缘码放的剩余空间要大于 1 个包装箱的宽，即 $(b_a-b_b)<(b_by_1+l_by_4)<b_a$[11—12]。

如果处于托盘对角线上的包装箱的长度和大于托盘长轴，即$(x_1+x_3)l_b>l_a$，则必须保证这两角所码放箱子的高度和要小于托盘纵轴，即$(y_1+y_3)b_b<b_a$；如果托盘另一对角线所码放包装箱的长度和大于托盘长轴，即$(x_2+x_4)b_b>l_a$，则必须保证这两角所码放箱子的高度和要小于托盘纵轴，即$(y_2+y_4)l_b<b_a$；如果处于托盘对角线上的包装箱的高度和大于托盘纵轴，即$(y_2+y_4)l_b>b_a$，则必须保证这两角所码放箱子的长度和要小于托盘长轴，即$(x_2+x_4)b_b<l_a$；如果处于托盘对角线上的包装箱的高度和大于托盘纵轴，即$(y_1+y_3)b_b>b_a$，则必须保证这两角所码放箱子的长度和要小于托盘长轴，即$(x_1+x_3)l_b<l_a$[13]。

得到非线性规划模型为：

$$U_{max}=x_1y_1+x_2y_2+x_3y_3+x_4y_4$$

约束条件为：

$$l_a-b_b<l_bx_1+b_bx_2\le l_a$$
$$b_a-b_b<b_by_3+l_by_2\le b_a$$
$$l_a-b_b<l_bx_3+b_bx_4\le l_a$$
$$b_a-b_b<b_by_1+l_by_4\le b_a$$

如$(x_1+x_3)l_b>l_a$，则$(y_1+y_3)b_b<b_a$

如$(x_2+x_4)b_b>l_a$，则$(y_2+y_4)l_b<b_a$

如$(y_2+y_4)l_b>b_a$，则$(x_2+x_4)b_b<l_a$

如$(y_1+y_3)b_b>b_a$，则$(x_1+x_3)l_b<l_a$

式中：$x_i\ge0$，$y_i\ge0$ (i=1，2，3，4)，且为整数。

2.2 lingo 软件求解组托优化方法应用

某物流配送中心，托盘规格 1200 mm×1000 mm×140 mm，货架层高 1500 mm，横梁高 120 mm，现有新产品的包装规格 450 mm×280 mm×250 mm，入库数量 100 箱，堆码层次极限为 4 层，可利用 Lingo 软件进行组托优化，准备入库货位。将各参数代入非线性规划模型可得约束条件[14]：

$$1200-280<450x_1+280x_2\le1200$$
$$1000-280<280y_3+450y_2\le1000$$
$$1200-280<450x_3+280x_4\le1200$$
$$1000-280<280y_1+450y_4\le1000$$

如$(x_1+x_3)450>1200$，则$(y_1+y_3)280<1000$

如$(x_2+x_4)280>1200$，则$(y_2+y_4)450<1000$

如$(y_2+y_4)450>1000$，则$(x_2+x_4)280<1200$

如$(y_1+y_3)280>1000$，则$(x_1+x_3)450<1200$

$x_i\geqslant0$，$y_i\geqslant0$ (i=1，2，3，4)，且为整数。

2.2.1 lingo 程序设置

lingo 程序的设置为：

MODEL：

Sets：!以命令行方式输入一个模型，因变量个数较多，因此以集合方式进行定义。

object/1..4/： f； !设置横轴变量集；

object1/1..4/：g； !设置纵轴变量集；

2.2.2　数据输入

输入的数据为：

Data：

L=1200；W =1000；a=450；b=280；!已知常量；

enddata

程序中：L 为 l_a；W 为 b_a；a 为 l_b；b 为 b_b。

2.2.3　目标函数

目标函数为 $U_{max}= x_1y_1+ x_2y_2+ x_3y_3+ x_4y_4$。

2.2.4　约束条件

约束条件的程序为：

```
!设定条件约束为以下变量；
f(1)=a*x1+b*x2；f(2)=a*x3+b*x4；
g(1)=a*y2+b*y3；g(2)=a*y4+b*y1；
f(3)=(x1+x3)*a；f(4)=(x2+x4)*b；
g(3)=(y1+y3)*b；g(4)=(y2+y4) *a；
!对条件约束集成员的约束；
@for(object(i):g(3)<@if(f(3)#ge#L,W,g(3)) ；
@for(object(i):g(4)<@if(f(4)#ge#L,W,g(4)))；
@for(object1(i):f(4)<@if(g(4)#ge#W,L,f(4))) ；
@for(object1(i):f(3)<@if(g(3)#ge#W,L,f(3)))；
!所求目标函数值必须小于等于组托上限值；
x1*y1+ x2*y2+ x3*y3+ x4*y4<L*W/a*b；
!未知数均为大于等于零的整数；
@GIN(x1)；@GIN(x2)；@GIN(x3)；@GIN(x4)；
@GIN(y1)；@GIN(y2)；@GIN(y3)；@GIN(y4)；
end
```

程序中：x1，…，x4 和 y1，…，y4 分别为 x_1，…，x_4 和 y_1，…，y_4；L 为 l_a；W 为 b_a；a 为 l_b；b 为 b_b。

2.2.5　运行结果

最后得到运行结果为"Objective value: 8.000000"，即最多码放 8 个包装箱。根据运行结果得，x_1 为 1，y_1 为 0，x_1y_1 为 0，即码放数量为 0；x_2 为 2，y_2 为 0，x_2y_2 为 0，即码放数量为 0；x_3 为 2，y_3 为 3，x_3y_3 为 6，即码放数量为 6；x_4 为 1，y_4 为 2，x_4y_4 为 2，即码放数量为 2。总的码放数量为 $x_1y_1+x_2y_2+x_3y_3+x_4y_4=8$。根据结果得出组托见图 2。

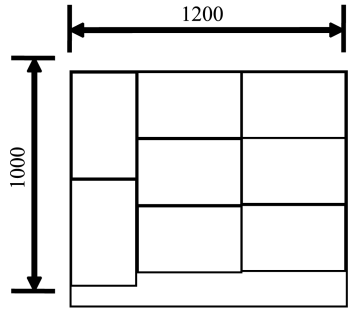
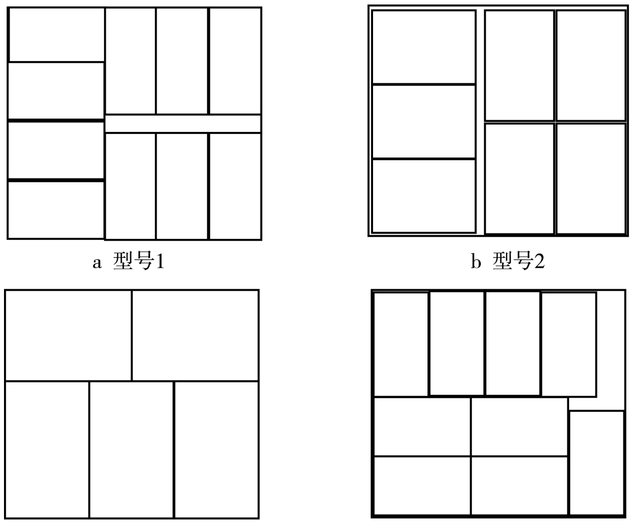


图 2　组托
Fig.2 Group support

考虑包装箱不超过托盘边缘和不出现重叠的情况下，此码放数量即为最大码放数量。每层货架的可利用空间求解方法为：货架高度—横梁高度—托盘高度—工作余量，工作余量根据操作人员的熟练程度取值一般为 90 ~ 150 mm 之间，文中按 90 mm 计算，即 1150 mm。可容纳包装箱层数约为 4 层（向下取整）与堆码层次极限对比，取小，文中堆码层数取 4。每个托盘可码放包装箱数量为 32，共需要托盘数量约为 4（向上取整），具体码放规则见图 3，因此货物入库前要准备 4 个货位。



a 型号1　　b 型号2

c 型号3　　d 型号4

图 3　各型号包装箱组托
Fig.3 Schematic diagram of each type of packing box

3　lingo 算法性能分析

若托盘规格一定，为 1200 mm×1000 mm×140 mm，待组托货物的包装箱型号见表 1，运行结果见表 2。根据运行结果，得出的各型号包装箱组托见图 3。

表 1 包装箱型号

Tab.1 Packing box model mm

型号	长	宽	高
1	455	245	200
2	480	320	200
3	595	395	375
4	460	260	230

表 2 决策变量值及求解时间

Tab.2 Decision variables and solution time

变量	型号 1	型号 2	型号 3	型号 4
x_1	1	1	0	2
x_2	3	2	3	1
x_3	1	1	2	0
x_4	3	2	0	4
y_1	4	3	2	2
y_2	2	2	1	1
y_3	0	0	1	2
y_4	0	0	0	1
Umax	10	7	5	9
求解花费时间	<0.05 s	<0.05 s	<0.05 s	<0.05 s

利用 lingo 软件进行非线性规划数学模型的求解，整个过程平均用时不超过 1 s，速度快且精准性高，能根据所得决策变量的值画出组托示意图。

求解最优化问题的常用软件还有 Matlab 或 WinQSB，但要熟练使用它们，需要学习者前期掌握的基础知识较多，且软件编码繁琐，容易出错。相较其他求解最优化软件，用 Lingo 软件求解非线性规划，编程更简单，求解速度更快，显示更直观[15]。这种非线性规划模型对不同型号包装箱组托的适用性尚需在 lingo 软件应用中进一步验证。

4 结语

lingo 软件界面友好、简单易用，求解组托问题快速而准确。它将传统人工实验求解的复杂过程简单化，为决策者在最短时间内做出最优组托决策提供了保障，为生产、流通、教学等环节提供了有力的支持。

参考文献：

[1] 秦健. 包装过程托盘堆码作业中常用优化方法的应用研究[J]. 物流工程与管理, 2016, 38(5): 33—37.
QIN Jian. Study on Application of Commonly Used Optimization Method in the Operation of Pallet Packaging Process[J]. Logistics Engineering and Management, 2016, 38(5): 33—37.

[2] 姚劼. 包装优化对物流成本的影响[J]. 江苏科技信息, 2015(2): 20—21.
YAO Jie. Impact of Packaging Optimization on Logistics Cost[J]. Jiangsu Science and Technology Information, 2015(2): 20—21.

[3] 刘建. 包装物流技术与物流成本优化[J]. 物流技术与应用, 2010(10): 118—121.
LIU Jian. Packaging Logistics Technology and Logistics Cost Optimization[J]. Logistics Technology and Application, 2010(10): 118—121.

[4] 武晓今, 朱仲英. 二维装箱问题的一种实现方法[J]. 研究与设计, 2003, 19(4): 20—23.
WU Xiao-jin, ZHU Zhong-ying. A Realization Method of Two Dimensional Packing Problem[J]. Research and Design, 2003, 19(4): 20—23.

[5] 唐冲. 基于 Matlab 的非线性规划问题的求解[J]. 计算机与数字工程, 2013, 41(7): 1100—1103.
TANG Chong. Solution of Nonlinear Programming Problem Based on Matlab[J]. Computer and Digital Engineering, 2013, 41(7): 1100—1103.

[6] 屈援, 王雪莲. 大型二维装箱问题及其禁忌算法研究[J]. 安徽大学学报, 2007, 31(5): 32—35.
QU Yuan, WANG Xue-lian. Research on Large Two-Dimensional Bin Packing Problem and Its Tabu Search Algorithm[J]. Journal of Anhui University, 2007, 31(5): 32—35.

[7] 程浩, 刘心报, 刘林, 等. 一种用遗传算法求解装箱问题的新编码方法[J]. 合服工业大学学报, 2006, 29(2): 144—147.
CHENG Hao, LIU Xin-bao, LIU Lin, et al. A New Coding Method of Genetic Algorithm for Bin Packing Problem [J]. Journal of Polytechnic University, 2006, 29(2): 144—147.

[8] 白春阳, 石东伟. 非线性规划在数学建模中的应用[J]. 科技信息, 2011(9): 167.
BAI Chun-yang, SHI Dong-wei. Application of Nonlinear Programming in Mathematical Modeling[J]. Science and Technology Information, 2011(9): 167.

[9] 桑杨阳, 朱万红, 但兵兵. 非线性规划建模与 LINGO 软件的编程应用[J]. 电脑知识与技术, 2012, 8(10): 2419—2422.
SANG Yang-yang, ZHU Wan-hong, DAN Bing-bing. Nonlinear Programming Modeling and Programming Application of LINGO Software[J]. Computer Knowledge and Technology, 2012, 8(10): 2419—2422.

[10] 戴彧虹, 刘新为. 线性与非线性规划算法与理论[J]. 运筹学学报, 2014, 18(1): 69—92.
DAI Yu-hong, LIU Xin-wei. Algorithm and Theory of Linear and Nonlinear Programming[J]. Operations Research Transactions, 2014, 18(1): 69—92.

[11] 王征, 胡祥培, 王旭坪. 带二维装箱约束的物流配送

车辆路径问题[J]. 系统工程理论与实践，2011, 31(12): 2328—2341.
WANG Zheng, HU Xiang-pei, WANG Xu-ping. Vehicle Routing Problem with Two Dimensional Bin Packing Constraints[J]. System Engineering Theory and Practice, 2011, 31(12): 2328—2341.

[12] 蒋兴波, 吕肖庆, 刘成城. 二维矩形条带装箱问题的底部左齐择优匹配算法[J]. 软件学报，2009, 20(6): 1528—1538.
JIANG Xing-bo, LYU Xiao-qing, LIU Cheng-cheng. Two Dimensional Rectangular Strip Packing Problem with Bottom Left Homogeneous Matching Algorithm [J]. Journal of Software, 2009, 20(6): 1528—1538.

[13] 隋树林, 邵巍, 高自友. 同一尺寸货物三维装箱问题的一种启发式算法[J]. 信息与控制，2005, 34(4): 490—494.
SUI Shu-lin, SHAO Wei, GAO Zi-you. A Heuristic Algorithm for Three Dimensional Packing Problem of the Same Size Cargo[J]. Information and Control, 2005, 34(4): 490—494.

[14] 曹亚群, 朱俊. 线性规划在物流工程中的应用[J]. 宿州学院学报, 2010, 25(11): 30—32.
CAO Ya-qun, ZHU Jun. Application of Linear Programming in Logistics Engineering[J]. Journal of Suzhou University, 2010, 25(11): 30—32.

[15] 方建斌. 基于 Matlab 的非线性规划问题的求解[J]. 科技资讯, 2013(2): 34—36.
FANG Jian-bin. Solution of Nonlinear Programming Problem based on Matlab[J]. Science and Technology Information, 2013(2): 34—36.