

基于线路优化算法的卷烟配送调度系统

孙壮志, 鄢烈虎, 要学玮

北京烟草物流中心, 北京市通州区九棵树西路198号101101

摘要: 为研究线路优化算法在卷烟配送中的应用, 以北京卷烟配送市场为背景, 结合配送条件的限制, 采用中心聚类、禁忌搜索、局部搜索等多种专业算法, 提出了一套适应性强、运算速度快、可实施性高的解决方案。同时, 通过选择性地取用原有固定线路的数据作为最终路顺的参考, 结合配送车型和配送员的工作负荷, 强化了动态调度结果的可实施性。方案实施测试效果良好, 月平均装载率达90%以上。

关键词: 线路优化; 中心聚类; 禁忌搜索; 局部搜索

doi:10.3969/j.issn.1004-5708.2014.05.021

中图分类号: TP315 **文献标志码:** A **文章编号:** 1004-5708 (2014) 05-0128-06

Cigarette distribution scheduling system based on route optimization algorithm

SUN Zhuangzhi, YAN Liehu, YAO Xuewei

Beijing Tobacco Logistics Center, Beijing 101101, China

Abstract: An adaptable, quick-responsive and feasible real-time solution was introduced by applying algorithms such as central clustering, tabu search and local search on the basis of cigarette distribution market in Beijing. Dynamic scheduling was fully achievable by taking original delivery map as reference and work load of delivery vehicles and working staff into full consideration. The solution was proved to be effective with monthly average loading rate exceeding 90%.

Keywords: route optimization; central clustering; tabu search; local search

随着物流相关技术的蓬勃发展, 物流行业正在经历一场深刻的变革。以先进的算法辅助实现物流配送线路优化是新技术应用的一项重要措施。

优化技术是一种以数学为基础的, 用于求解各种工程问题优化的应用技术。所谓优化算法, 就是一种搜索过程或规则, 它是基于某种思想和机制, 通过一定的途径或规则来得到满足用户要求的问题的解^[1]。聚类算法是一种数据简化技术, 把基于相似数据特征的变量和个案组合在一起^[2]。禁忌搜索是弥补局部搜索先天弱点的一种改进算法, 局部搜索算法的先天弱点在于会陷入局部最优, 所以设计好的跳离局部最优的启发策略是设计局部搜索算法的关键^[3]。禁忌搜索算法基本思想相当简单, 它采用一种“记忆”

装置驱使算法去探索搜索空间的新的区域^[4]。我们可以记住最近被检查过的解, 它们将成为下一个解的禁忌(被禁止)。贪婪法是一种对某些求最优解问题的更简单、更迅速的设计技术, 其特点是一步一步地按策略进行, 常以以前情况为基础根据某个优化测度作最优选择, 而不是考虑各种可能的整体情况^[5]。

论文以设计和开发一个以线路优化算法为基础的卷烟配送线路调度系统为目标, 根据需求分析, 进行线路优化算法的选择, 最终期望完成一个线路调度系统, 并对结果进行理论验证。

1 系统需求分析

1.1 实际业务需求情况

每天从访销中心接收访销数据, 包括订单日期、零售户信息、订货量。根据配送实际情况, 需设计一套算法, 根据北京市地理情况、配送车辆最大装载量限制、考虑配送员熟悉度, 在尽量短的时间内对当日

作者简介: 孙壮志(1971—), 高级工程师, 主要从事卷烟物流等方面的研究, Email:349890782@qq.com

收稿日期: 2014-03-12

订单进行配送线路优化, 为每位配送人员提供相对合理、科学的配送线路。同时要求计算结果线路可人工调整, 并使月平均装载率达到 90% 以上。

1.2 基础数据需求

根据实际业务需求, 调度系统需要以下 6 方面的基础数据支撑:

- 1) 城市道路基础信息: 北京市道路网络图, 即 GIS 地图;
- 2) 零售户信息: 零售户地理坐标点, 并将其投影在北京市道路网络图上; 订货量; 送货车型限制;
- 3) 车辆基础信息: 车型、最大装载量;
- 4) 人员信息: 司机以及配送员的基础信息; 他们在前一段时间的历史工作负荷;
- 5) 送货区域信息: 所属区域组; 是否进行融合; 区域内线路最大送货时间; 区域内可用配送资源 (指司机、配送员及车辆合成一个组合关系);
- 6) 原送货线路信息: 客户原属线路信息。

1.3 返回结果需求

根据业务需求, 在计算结果时需要返回的信息有以下 8 类: 每条线路的司机、配送员、送货车辆、包含的客户以及其送货顺序、预计里程、预计送货时间、送货量、装载率。

2 系统设计

2.1 整体设计

动态线路调度包含客户区域分配、区域识别、线路生成与区域间调整三个步骤。其中线路生成以及区域间调整被分成 2 条独立的路径, 一条采用以中心聚类为核心的算法进行计算, 另一条是采用线路串接的方式进行计算, 如图 1 所示。

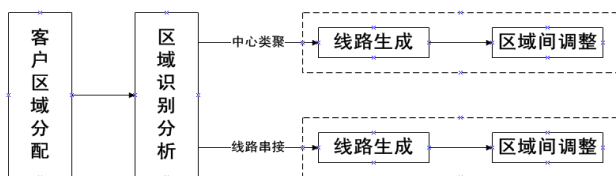


图 1 整体设计思路

Fig. 1 Integrated design map

2.2 详细设计

2.2.1 客户区域分配

获取营销订单后, 根据系统内划定的区域, 将订单分拆到各个区域内进行运算。

2.2.2 区域识别

线路优化针对不同的客户分布情况, 采用不同的

算法加以应对。

如果区块内的客户基本上是按照访销周期订货的, 都是捆绑得比较好的一整条的线路, 那么采用配送人员比较认可、熟悉度较高的线路进行串接, 将多条线路头尾相接串联到一起, 切割拆分生成多个车次。

如果区块内的客户是随机访问的, 七零八落地分布在各个线路中, 那么采用中心聚类的算法加以聚类分割, 生成多条线路。

分类判定算法是分析整线路的客户所占的比例。根据区块内已有的原线路, 将客户分配到各条原线路上去, 生成多个客户序列。如果序列内客户数最多的 8 个客户序列的总客户数占全部客户数的 80% 以上, 就说明该区域中被绑定在原有固定线路中的客户占所有客户的大多数。

2.2.3 线路生成与区域间调整

2.2.3.1 客户群生成

在每个区块中, 再次根据区块内已有的原线路, 将客户分配到各条原线路上去 (根据订单情况不同, 有可能出现大部分客户都集中于某几条线路, 也有可能出现客户分散于区块内的所有原线路)。分析各条原线路, 倘若发现其客户序列中, 前后两户不是相对接近的, 就进行分拆, 最终形成若干个连续并互相靠近的客户群。(有可能因为客户分散的关系, 大多数客户组都只包含了单个客户)。

2.2.3.2 线路装填

根据前面区域分析的结果, 用不同的算法 (聚类分析或者线路串接), 对生成的客户群进行合并以及拆分, 将彼此靠近的客户群根据车辆的容量, 以及最大送货时间, 合并为一条线路。归并的同时, 允许对那些“整个加进去太多, 不加又太少”的客户群进行分拆, 以最大程度提高装载率。根据当前客户的情况选择合适的车型进行配送, 避免无法配送到户的情况。同时在选择出车车辆时, 对司机的工作负荷进行一定的均衡 (因为车辆和司机是绑定的), 确保各人的劳动时间不会差异太大。

(1) 聚类分析算法

其基本策略是先对每个簇任意选择一个代表对象, 剩余的对象根据其代表对象的距离分配给最近的一个簇。然后反复利用非代表对象来代替代表对象, 以改进聚类的质量。聚类结果的质量用一个代价函数估算, 该函数度量对象与参照对象之间的平均相异度。为了判定一个非代表对象 O_{random} 是否是当前一个代表对象 O_j 的好的代表, 对于每一个非中心对象

p , 需要分为四种情况考虑: ① p 当前隶属于中心点对象 O_j 。如果 O_j 被 O_{random} 所代替, 且 p 离一个 O_i 最近, $i \neq j$, 那么 p 被中心分配给 O_i 。② p 当前隶属于中心点对象 O_j 。如果 O_j 被 O_{random} 所代替, 且 p 离 O_{random} 最近, 那么 p 被中心分配给 O_{random} 。③ p 当前隶属于中心点对象 $O_i, i \neq j$ 。如果 O_j 被 O_{random} 代替作为一个中心点, 且 p 仍离 O_j 最近, 那么对象的隶属不发生变化。④ p 当前隶属于中心点对象 $O_i, i \neq j$ 。如果 O_j 被 O_{random} 代替作为一个中心点, 且 p 离 O_{random} 最近, 那么 p 被重新分配给 O_{random} 。

计算时, 输入: 簇的数目 k 和包涵 n 个对象的数据。其中, 簇就是线路, 而对象就是客户群。输出: k 个簇, 使得所有对象与其最近中心点的相异度总和最小。

计算方法: 任意选择 k 个对象为初始的簇中心。指派每个剩余对象给离它最近的中心点所代表的簇。随机地选择一个非中心点的对象 O_{random} , 计算用 O_{random} 代替 O_j 的总代价 S , 如果 $S < 0$, 那么 O_{random} 代替 O_j , 形成新的 k 个中心点的集合。直到不发生变化。本算法以装载率尽量接近满载为代价函数 $f(x)$, 即 $f(x) = |Loading Rate - 1|$ 其中 $Loading Rate$ 为装载率。如果没装满送货时间就超标了, 则代价函数 $f(x)$ 改为时间上接近最大送货时间, 即 $f(x) = |Time - Max Time|$ 其中 $Time$ 为送货时间, $Max Time$ 为最大送货时间。

分割完毕后, 选择包含有计算起点的线路作为原始线路 (此线路内包含有多个客户群), 进入客户群拆分阶段。由于中心聚类不可能获得非常精密的结果, 上面的原始线路很有可能会出现超出最大装载量 (超出最大工作时间) 的情况。此时就需要对原始线路内的客户群进行排序, 按照顺序逐个将客户进行装车, 直到装不下为止。剩余的原始线路内客户重新组成客户群, 返还到待计算客户中, 与其他线路中的客户群一起回炉, 等待下一次计算, 这样就得到了一条线路的结果。

接下来, 对剩余待计算的客户群重新开始这一轮步骤: 首先预估所需的车辆, 再次取离区域中心最远的客户群作为计算起点, 继续中心聚类计算, 获取原始线路后, 再做客户群拆分。如此迭代, 直到只剩下 2 条线路为止。

(2) 线路串接算法

线路串接并非简单的把区域的客户群从头到尾串到一起, 然后根据车容量和工作时间来切分车辆, 它

是直达目标的。也就是说, 它并不是通过事先优化客户群的连接顺序来间接的优化最终的线路, 而是使用禁忌搜索尝试各种客户群排序的组合方式, 对此排序下的客户序列进行装车切割, 以实际得出的线路结果作为最终的评估对象。通过评估不同客户群排序组合下实际的切分装车结果, 来找出较优的客户群排序方式。

① 禁忌搜索: 为了得到更优的计算结果, 扩大计算范围, 减少计算量, 本算法引入了禁忌搜索。首先, 对置换问题定义一种邻域搜索结构, 如互换操作 (SWAP), 即随机交换两个点的位置, 则每个状态的邻域解有 $C_n^2 = n(n-1)/2$ 个其中 n 为点数。称从一个状态转移到其邻域中的另一个状态为一次移动 (move), 显然每次移动将导致适配值 (反比于目标函数值) 的变化。其次, 采用一个存储结构来区分移动的属性, 即是否为禁忌“对象”。需要指出的是, 由于当前的禁忌对象对应状态的适配值可能很好, 因此在算法中设置判断, 若禁忌对象对应的适配值优于“best so far”状态, 则无视其禁忌属性而仍采纳其为当前选择, 也就是通常所说的藐视准则 (或称特赦准则)。

② 评估参数: 评估的基本参数是所有线路的总工作时间。如果需要和其它区域连接, 那么还需要加上尾车的最后一户到下一个区域的连结点客户的时间, 总时间越短越好。此外, 如果有区域融合的要求, 还会附加上尽量能使尾车和与其它区域相连这个条件。

③ 对于特定客户群序的装车切割流程: 装车时, 默认方法为安排好的客户顺序装车, 假如碰到自己车子无法配送的客户, 将这一户剔出来, 继续装后面的, 这几户被剔除的客户留待后面的车子来尝试配送。倘若装车中碰到一个订货大户, 如果加上, 车子装不下, 如果不加, 车子装载率又太低, 这时候将跳过此户, 继续装后面订货量没这么大的, 保证装载率不会太低。

④ 车型匹配: 将起始客户初始化为第一个客户, 进行车型适配度的评估。其做法是, 尝试使用各种车型进行装车, 把生成的各个线路结果作为评估对象。生成线路的时候, 由于车型关系剔除的客户越多, 说明此车型的适配度越低。取适配度较好的几个车型作为待选车型, 从中选出一辆驾驶员历史累计送货时间较低的作为最终选择结果。另外, 如果可能的话, 区域内的每个驾驶员都需要至少出车一次。所以选择时, 优先从今天没有出过车的驾驶员里选择。

⑤ 标记: 一条线路生成后, 将线路中的客户作

上已计算标记,继续按照上面的算法,选择车型,选择车辆,计算剩下的客户序列中的客户,直到所有的客户都被计算完成。

2.2.3.3 路顺优化

线路生成后,分别以全体打乱重排的方式和保留客户群内客户顺序进行路顺计算。默认为保留客户群内的顺序,但如果默认结果与打乱重算的结果差很多,那么以打乱重排后的顺序为最终的线路路顺。打乱重排涉及局部搜索和禁忌搜索算法。

基本思想是在搜索过程中沿着某一方向搜索,该方向是从当前点到该点的所有邻居中离目标点最近的那一点的方向。

① 随机选择一个初始的可能解

$$x_0 \in D, x_b = x_0, P = N(x_b)$$

其中 D 是问题的定义域, x_b 用于记录到目标位置的最优解, P 为 x_b 的邻域。

② 如果不满足结束条件,则开始循环。

③ Begin

④ 选择 P 的一个子集 P' , x_n 为 P' 的最优解。 P' 可根据问题特点,选择适当大小的子集,也可按概率选择。本算法采用邻域 P 集合中的全部元素进行计算。

⑤ 如果 $f(x_n) < f(x_b)$, 则 $x_b = x_n$, $P = N(x_b)$, 转②, 其中 $f(x)$ 为指标函数。

⑥ 如果 $f(x_n) \geq f(x_b)$, 则 $P = P - P'$, 转②。

⑦ End

⑧ 输出计算结果。

而禁忌搜索是对局部搜索的扩展,是一种局部搜索能力很强的全局迭代寻优算法,其详细计算机制如前所述。与局部搜索相比具有以下特点:一是在搜索过程中可以接受劣解。二是新解不是在当前邻域中随机产生,而是优于“best so far”的解,或是非禁忌的最优解,因此选取优良解的概率远大于其它解。

2.2.3.4 尾车划拨

假如该区域(区块)开启了区域融合,那么首先要通过设置获取信息,判断该区域(区块)能和哪些其他区域(区块)交换客户。

接下来,查看本区域是否需要往外划拨的尾车,如果需要,那么查找距离这个尾车最近的可交换区域,将尾车的所有客户划拨给到那个区域去(或者抓取足够的客户填充)。假如需要将尾车交给其他区域,程序运算时会将尾车客户尽量放置在本区域的边缘地区,跟将要划拨到的邻居区域尽量接近。

如果找不到其他可划拨的区域了,尾车将独自成为一条线路。这种情形通常出现在几种情况下:这个

区域与其他的任何区域都很远,距离太远的话是不会划拨过去的;这个区域是所有可交换区域中,最后一个待计算的区域了;尾车的客户附近的其他车辆都无法配送。

2.3 曾使用的其它算法及遇到的问题

2.3.1 第一版算法

粗略预估一个月平均装载率大致能达到 90% 左右的车辆数,根据车辆列表获取所用的车辆。采用中心聚类,以装载率越平均越好为目标,对整个区域进行一次性的分割计算得到初步结果,然后使用渗透模拟的方式,在线路之间调整客户。

这种方式比单纯的中心聚类分割线路,稳定性要好不少。但由于仅仅考虑客户点的地理信息来安排路顺,不参考其原线路,所以路顺都是重新计算的。在基础信息不准确的情况下,路顺的计算结果往往不如人意。

另外,由于中心聚类天生的稳定性的缺陷,以及渗透模拟无法进行大范围调节的劣势,偶尔会可能出现一些穿插的很远的较劣结果。

2.3.2 第二版算法

参照原线路,把区域内的客户划分成一个个客户群,然后对客户群进行排序,生成一个很大的客户序列,之后,使用区域内车辆列表,对这个客户序列进行逐次装车,以车辆的最大装载量和最大送货时间为限制条件,分割成一条条线路。

分割完毕后,检查每个区域的尾车是否能和其他区域尾车进行合并,通过最大连接距离来限制尾车的合并范围,避免与过远的尾车连接。倘若无法简单合并,将尝试将尾车拆成多份,分配到附近区域的其他尾车上面去,以减少尾车数目,提高装载率。

这个做法在客户群较少的情况下效果较好,但如果客户群数量较多,通过简单的串接排序生成的线路,其聚团性较差,线路之间穿插较严重。另外,由于每个区域之间的尾车一般相距较远,尾车所能交互的对象并不多,所以其合并相对的有限。在开启全区合并的情况下,只能减少一半不到的尾车。

3 系统测试与结果分析

3.1 系统的测试环境

编程语言为 C#, 编译环境为 Visual Studio 2012, 基于 .NET Framework 4.5, 操作系统为 Win7 32bit。

部署软件环境,操作系统为 Windows Server 2008 R2 64bit, 安装 .NET Framework 4.5 框架。

部署硬件环境为 IBM X3650 M4 服务器（至强 CPU，内存 32G）。

3.2 测试结果

以 2014 年 2 月订货量为实验数据，计算原有送货模式和线路优化新模式下两种配送结果，以装载率

和出车趟次作为比较指标进行分析。选取 10 天的订单量进行测试，数据结果如表 1 所示。

由表 1 数据绘制图 2、3 的曲线图，其中横坐标为序号值，图 2 为装载率对比曲线图，图 3 为出车趟次对比曲线图。

表 1 2014 年 2 月测试数据结果
Tab. 1 Test results of February 2014

| 序号 NO. | 订单日期 Order Date (年月日) | 直配订单 Order Quantity / 条 | 原线路装载率 Old loading rate /% | 原线路趟次 Old department quantity / 趟 | 优化后装载率 New loading rate /% | 优化后趟次 New department quantity / 趟 |
|-----------|-----------------------------|-------------------------------|----------------------------------|--|----------------------------------|--|
| 1 | 20140207 | 723121 | 72.99 | 161 | 92.98 | 125 |
| 2 | 20140208 | 786851 | 72.99 | 175 | 94.36 | 134 |
| 3 | 20140210 | 798590 | 75.76 | 168 | 93.48 | 137 |
| 4 | 20140211 | 802142 | 73.21 | 178 | 93.26 | 138 |
| 5 | 20140212 | 747533 | 73.73 | 162 | 93.77 | 128 |
| 6 | 20140213 | 599799 | 70.37 | 140 | 92.41 | 106 |
| 7 | 20140214 | 622687 | 72.1 | 141 | 91.55 | 110 |
| 8 | 20140217 | 616549 | 71.94 | 137 | 92.59 | 107 |
| 9 | 20140218 | 630250 | 74.22 | 140 | 91.68 | 110 |
| 10 | 20140219 | 610982 | 71.22 | 138 | 93.55 | 105 |

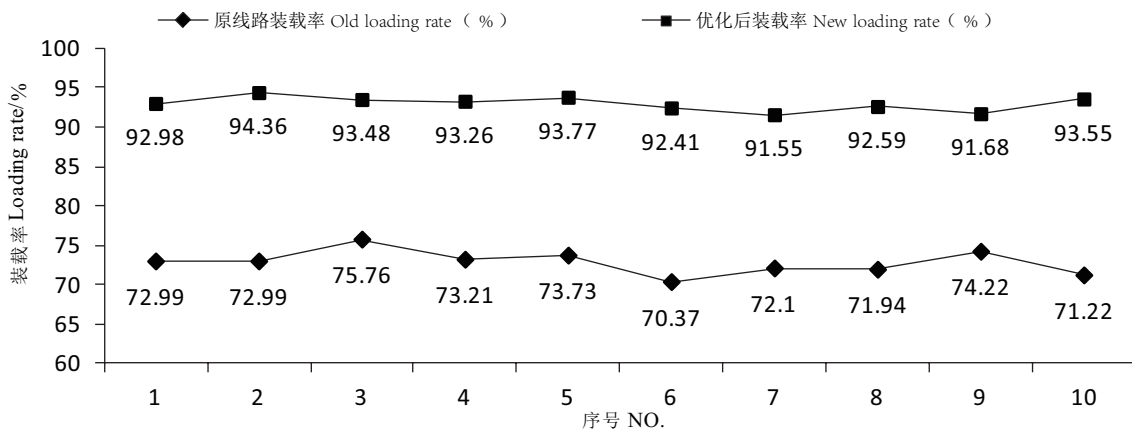


图 2 装载率对比曲线图

Fig.2 Comparison graph of loading rates

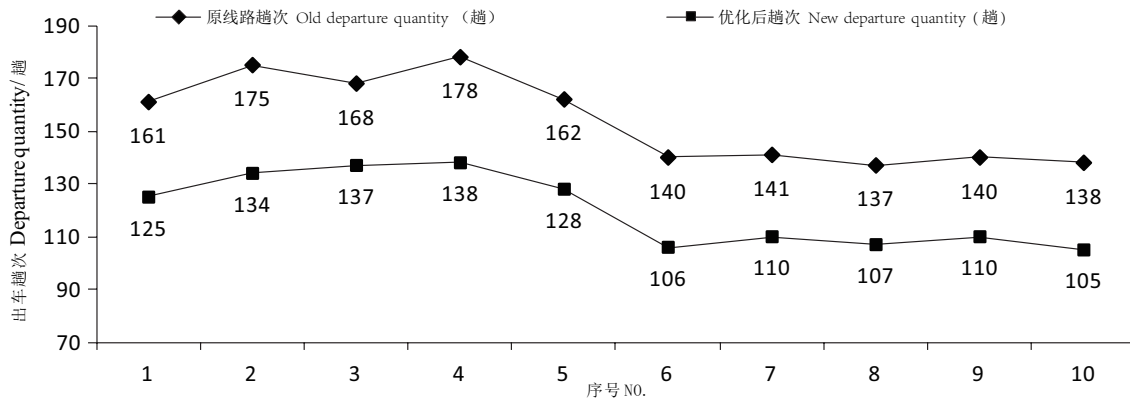


图3 出车趟次对比曲线图

Fig. 3 Comparison graph of department quantity

3.3 结果分析

(1) 如图2所示,在订单量一定的情况下,优化后装载率全部高于原有线路装载率,平均增加20.1%。优化后,平均装载率达到了90%以上的指标要求。

(2) 如图3所示,在订单量一定的情况下,优化后出车趟次全部少于原有线路出车趟次,平均减少超过20%。

(3) 在线路装填时,本算法通过评估不同客户群排序组合下,实际的切分装车结果,来找出较优的客户群排序方式。一般来说,很少有人会用这种系统资源巨大的方式来实现这个要求,但由于已经作了区域识别分析,能够确保区域内的客户基本上都是整线的,客户群的数目不会太多,计算负荷不至于无法接受。

(4) 本算法中心聚类算法与普通中心聚类不同,最大的区别是在于计算的次数。普通中心聚类,是一次性计算出所有的线路;而这个算法,每次只取包含有计算起点的线路作为结果,剩余的线路统统回炉重算,需要逐次计算N-1次,才能获取N条线路。保

证尾车的线路与其他区域很接近,方便进行区域间的客户调配。

4 结论

本算法满足了业务指标要求,突破了原有行政区划划分的界限限制,优化了配送调度结果,在大计算量的情况下仍能满足计算要求。对卷烟配送装载率和出车趟次具有良好的改善效果。同时,本算法具有普适性,在适应北京复杂路线的情况下仍能适应业务需求,具有较好的推广价值。

参考文献

- [1] 张晓辉. 禁忌搜索算法研究及其在电磁场优化问题中的应用 [D]. 硕士学位论文, 河北工业大学, 2003.
- [2] 刘志成, 文全刚. K-中心点聚类算法分析及其实现 [J]. 电脑知识与技术, 2005 (2): 20-24.
- [3] 朱建中. 求解可满足性问题的局部搜索算法 [D]. 硕士学位论文, 南京大学, 2004.
- [4] 郭娜. 基于节约算法和移动方向的禁忌搜索算法 [D]. 硕士学位论文, 大连理工大学, 2009.
- [5] 李少芳. 连续背包问题贪婪算法最优解的实现 [J]. 福建电脑, 2003 (11): 12-13.