# 基于 bi-level 迭代算法的物料循环配送研究

朱晓莹[1]，徐克林[1]，朱孝松[2]
（1. 同济大学 机械与能源工程学院，上海 201804；2. 北京科技大学 机械工程学院，北京 100083）

**摘　要：** 为解决车间物料配送费用高、配送效率低、灵活性差等问题，建立了基于 bi-level（双层）规划的车间物料循环配送模型。针对模型设计了基于 2-opt 改进的最大最小蚁群算法和双层迭代算法，引入线旁库存的限制，求得需求点的配送周期、配送量和配送路径，使库存和运输整合费用最小。通过数值算例求解，说明了该算法是有效的，也说明了该模型的实用价值和有效性。

**关键词：** 物料循环配送；双层规划；车辆路径问题；改进最大最小蚁群算法
**中图分类号：** TP391；TP301.6　　**文献标志码：** A　　**文章编号：** 1001-3695(2012)11-4176-04
doi：10.3969/j.issn.1001-3695.2012.11.045

# Study on material cycling distribution based on bi-level programming

ZHU Xiao-ying[1], XU Ke-lin[1], ZHU Xiao-song[2]
(1. *School of Mechanical & Energy Engineering, Tongji University, Shanghai* 201804, *China*; 2. *School of Mechanical Engineering, University of Beijing Technology, Beijing* 100083, *China*)

**Abstract:** To solve the high cost, low efficiency and poor flexibility of material distribution, this paper constructed a bi-level programming model of cycling material distribution. It used the model to minimize the inventory and transportation cost by finding the best distribution cycle strategy, distribution volume and distribution routing under the restriction of line besides inventory. In solving the model, it developed improved Max-min AS(IMMAS) based 2-opt, which were proved effective by solving numerical examples.

**Key words:** cycling material distribution; bi-level programming; vehicle routing problem; IMMAS

## 0　引言

物料配送过程中，库存和运输是费用耗费最大的两个因素。循环配送能有效地解决低库存和高成本运输之间的矛盾[1,2]。Rusdiansyah 等人[3]为解决设备配送供应链问题，建立多周期车辆路径模型，设计了两阶段结构式算法求解寻找配送策略。同时，若将库存和运输作为一个系统来研究，更能降低总费用[4]。王亮等人[5]对最小运输与库存费用的两级配送策略进行分析，构建了库存运输模型，运用启发式算法求解。Calvete 等人[6]为了解决由不同的决策者进行决策造成费用博弈的生产配送计划问题，分别建立了上层生产和下层配送的模型，并用蚁群算法进行了求解。然而，以上文献并没有设计一套完整的包含配送周期、配送量以及配送路径的策略。

本文以一家中型企业为背景进行物料配送研究，运用双层规划理论，针对物料循环配送运作模式建立模型，并设计了基于 2-opt 改进的 IMMAS 算法和双层迭代算法求解，最小化上下层模型的整合费用。

## 1　问题描述

在一个单物料中心点—多需求点的车间物流系统中，给定无向图 $G=(V,E)$。其中：$V$ 表示顶点集；$V_1$ 表示物料配送中心；$V_2$，$V_3$，…，$V_N$ 表示工位需求点；$E=\{(i,j),i,j\in V,i\neq j\}$ 表示边集。如图 1 所示，车间 $P$ 内有 $N$ 个工位，每个工位有确定的物料需求速率。给定计划期 $T^*$，所有的工位都需要物料中心（$i=1$）对其进行周期性配送。本文循环配送的目标是：对配送量、周期、路径进行决策，使得整体费用最小。

## 2　建立模型

### 2.1　假设条件

为便于建立模型，对问题作以下假设：

a）每个工位在整个计划期内的总物料需求量是确定的，且呈线性消耗；

b）每个工位在申请配送的周期内只由一辆小车服务，小车从配送中心出发，配送完回到中心；

c）配送小车分大小两种，视该周期需要配送的物料量而定，最大容量已知，每次运送不得超出最大容量；

d）工位不允许缺料，设置缺料罚数；

e）配送运输费用只与小车启用次数和距离有关；

f）每个工位为了防止缺货设有一定的安全库存，假设安全库存与时间无关。

### 2.2　模型的建立

上层模型是对循环配送过程中配送周期和配送量的优化，以期最小化物料被运至工位后的库存成本 $I$。

**收稿日期：** 2012-03-11；**修回日期：** 2012-04-27

**作者简介：** 朱晓莹（1987-），女，浙江人，硕士研究生，主要研究方向为物流配送、生产调度（laisefajia@163.com）；徐克林（1945-），女，上海人，博导，主要研究方向为工业工程；朱孝松（1988-），男，浙江人，硕士研究生，主要研究方向为车辆工程.

$$\min I=\sum_{k=1}^{K}\sum_{i=1}^{N}C_I\left[\frac{q_iX_{ki}T_i}{2}+S_i\right]+\sum_{i=1}^{N}D_i\max[0,\sum_{k=1}^{K}(\lambda_iT_i-q_iX_{ki}-S_i)] \tag{1}$$

$$\text{s.t. } K=T^*/T,\ T=\frac{1}{K}T^* \tag{2}$$

$$T_i=h_iT\quad h_i<K;h_i,K\in N^+ \tag{3}$$

$$\sum_{k=1}^{K}q_iX_{ki}=Q_i \tag{4}$$

$$\sum_{k=1}^{K}X_{ki}=T^*/T_i \tag{5}$$

$$\sum_{k=1}^{K}[q_iX_{ki}+S_i]\geqslant\sum_{k=1}^{K}\lambda_iT_iX_{ki} \tag{6}$$

$$X_{ki}(X_{ki}-1)=0 \tag{7}$$

其中:$C_I$ 为工位点的单位线旁库存费用;$q_i$ 为每次配送量;$T^*$ 为给定的计划期;$T_i$ 为工位 $i$ 的物料配送周期;$T$ 为基本配送周期,即每隔 $T$ 进行一次配送;$h_i$ 表示工位 $i$ 每隔 $h_i$ 个 $T$ 申请配送;$S_i$ 为工位点 $i$ 的安全库存;$D_i$ 为工位单位缺料损失费用;$\lambda_i$ 为工位 $i$ 的物料消耗速率;$K$ 为计划期内的总配送次数;$Q_i$ 为计划期 $T^*$ 内工位 $i$ 的物料总需求;$X_{ki}$ 是上层模型的决策变量:

$$X_{ki}=\begin{cases}1 & \text{第 } k \text{ 配送周期中给工位 } i \text{ 配送物料 } q_i\\ 0 & \text{otherwise}\end{cases}$$

$$i=1,\cdots,N;k=1,\cdots,K$$

上层模型中,目标函数是最小化库存成本,包括贮存费和缺货损失费。式(2)表示计划期 $T^*$ 与单位配送周期 $T$ 的关系;式(3)表示每个工位的派送周期 $T_i$ 与 $T$ 的关系;式(4)确保在计划期内,工位点 $i$ 的总需求量为 $Q_i$;式(5)是对每个工位配送次数的限制;式(6)表示不允许缺货;式(7)是对决策变量的 0-1 约束。

下层模型是对配送路径的优化,以期最小化物料在运输阶段增加的成本 $T$。根据上层模型求出的配送量,结合各距离以及车载量,建立多周期车辆路径问题(PVRP)模型。

$$\min T=C_{T1}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}Y_{kmij}d_{ij}+C_{T2}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{i=1}^{N}Y_{km1i} \tag{8}$$

$$\text{s.t. }\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{j=1}^{N}Y_{kmij}=\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{j=1}^{N}Y_{kmji}=T^*/T_i$$

$$i,j=1,\cdots,N;k=1,\cdots,K;m=1,\cdots,M \tag{9}$$

$$\sum_{m=1}^{M}\sum_{j=1}^{N}Y_{kmij}=\sum_{m=1}^{M}\sum_{j=1}^{N}Y_{kmji}=X_{ki}$$

$$i,j=1,\cdots,N;k=1,\cdots,K;m=1,\cdots,M \tag{10}$$

$$\sum_{i=1}^{N}Y_{kmji}=\sum_{i=1}^{N}Y_{kmij}\leqslant 1;i,j=1,\cdots,N;k=1,\cdots,K;m=1,\cdots,M \tag{11}$$

$$\sum_{j=1}^{N}Y_{kmji}-\sum_{j=1}^{N}Y_{kmij}=0;i,j=1,\cdots,N;k=1,\cdots,K;m=1,\cdots,M \tag{12}$$

$$\sum_{i=1}^{N}q_iY_{kmij}\leqslant m^*;i,j=1,\cdots,N;k=1,\cdots,K;m=1,\cdots,M \tag{13}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}Y_{kmij}=|V_m|-1,|V_m|\subseteq V/\{0\},|V_m|\geqslant 2$$

$$i,j=1,\cdots,N;m=1,\cdots,M \tag{14}$$

$$Y_{kmij}(Y_{kmij}-1)=0\quad i,j=1,\cdots,N \tag{15}$$

其中:$C_{T1}$ 为单位路径长度的运输费用;$C_{T2}$ 为单辆小车的单次启动费用;$d_{ij}$ 为工位 $i$ 和 $j$ 的距离;$M$ 为配送小车的数量;$m^*$ 为配送小车的容量;$Y_{kmij}$ 为下层模型的决策变量

$$Y_{kmij}=\begin{cases}1 & \text{第 } k \text{ 周期第 } m \text{ 辆小车经过}(i,j)\\ 0 & \text{otherwise}\end{cases}$$

$$i,j=0,1,\cdots,N;k=1,\cdots,K;m=1,\cdots,M$$

下层模型中,目标函数(8)最小化派送成本包括运输费和小车启动费用;式(9)是对工位配送次数的限制;式(10)表示在周期 $k$,若 $X_{ki}=1$,则周期 $k$ 中有辆车 $m$ 对 $i$ 进行配送;式(11)表示任一周期任一辆车必须从物料中心出发,再回到物料中心;式(12)表示任一周期任一辆车进入某工位点后必须离开此工位点;式(13)表示小车的容量约束;式(14)是次回路消除条件;式(15)是对决策变量的 0-1 约束。

## 3　模型求解

由于模型的复杂性以及库存和运输的费用博弈性,使得模型求解变得复杂。将上层模型的目标函数化简后,可以得出在计划期内物料中心对工位进行配送的频率越高,线旁库存费用就越小。但是从下层模型中可得,每个工位配送的次数越多,运输费用和小车启动费用就越大,配送过程很难达到系统最优。考虑到上下层模型的博弈性,本文将分别对上下层模型进行初步求解,再用双层循环迭代算法对其进行逐步迭代,使得循环配送系统的总费用 $I+T$ 达到整体最小。

### 3.1　上层模型求解

对于上层模型求解,给定一组 $T_i=[T_2,T_3,\cdots,T_N]$ 以及单位配送周期 $T$,便能求出目标值。由式(2)(3)可得:

$$T_i=h_iT=T^*h_i/K \tag{16}$$

为了避免在配送过程中工位需求点被遗漏或小车空运,令 $T$ 是 $T_i$ 数组的最大公约数,$K$ 是数组 $h_i$ 的最小公倍数。但是,若用于大规模问题,即使确定了以上关系,$T_i$ 的组合也是相当多。于是,需要确定 $T_i$ 的较优范围,减轻运算压力,作以下处理。

大体上,每个工位在计划期 $T^*$ 内的费用 $G$ 由储存费用和运输费用两部分组成。

$$G=\lambda_iT_i^2K_1+(T^*/T_i)D_iK_2 \tag{17}$$

其中:$K_1$ 是库存费用系数;$K_2$ 是运输费用系数;$D_i$ 是每次对工位 $i$ 进行派送小车需要行走的距离。由式(17)可知,欲求得总费用最小,若 $D_i$ 确定,那么 $T_i$ 就可以确定。本文近似地取 $D_i$ 为两点间距离的平均值。

$$D_i=\frac{1}{N}\sum_{j=1,\cdots,N}d_{ij}\quad i\in V \tag{18}$$

经过求导化简,按式(17)求出配送周期策略 $T_i$。

$$T_i=(T^*D_iK_2/2T_iK_1)\text{^}(1/3) \tag{19}$$

### 3.2　下层模型求解

PVRP 问题是 NP 难题,只有在需求点和路段数较少时才可能求其精确解[7]。因此,国内外很多学者用启发式算法对其求解,得到较优解或满意解。本文提出用 MMAS 来对问题进行求解,MMAS 是目前路径寻优效果较好的蚁群优化算法之一。但是,MMAS 存在寻优时间长、对局部寻优不理想等缺点[8]。为了增加解的多样性,加强局部搜索,本文引入 2-opt 邻域搜索来改进每次迭代得到的路径,从而扩大解空间,加快算法的运算效率。与传统的 ACO 算法相比,作如下改进。

1)信息素初始化

为了更加充分地进行寻优,各路径上的信息素初始浓度设置为一个较大数,本文设置

$$\tau_0=\tau_{\max} \tag{20}$$

这样,避免了算法的过早收敛,增大了搜索范围,使得算法有更好的发现较优解的能力。

2)信息素限定

为了避免算法过早地收敛于非全局最优解,将各条路径的信息素限制在$[\tau_{\min},\tau_{\max}]$,超过这个范围的值将被强制设置为$\tau_{\min}$或者$\tau_{\max}$。MMAS 算法可以有效地避免某条路径上的信息素浓度远远大于其余路径,使算法不再扩散;也有效避免了某些路径过早地被判"死刑",使算法出现停滞。$\tau_{\max}$和$\tau_{\min}$的设置如式(21)(22)。

$$\tau_{\max}=\left(\frac{1}{1-\rho}\right)\frac{1}{f^{\text{best}}} \tag{21}$$

$$\tau_{\min}=\frac{1-P^{\text{best}}(1/N)}{(N/2-1)\times P^{\text{best}}(1/N)}\tau_{\max} \tag{22}$$

其中:$f^{\text{best}}$是全局最优解,$P^{\text{best}}$是构造出最优解的概率。

3)状态转移规则

蚂蚁在选路径时,并不是随机乱选或只往概率大的方向选,而是对所有可行路径的转移概率进行轮盘赌选择。$t$ 时刻,蚂蚁 $m$ 在点 $i$ 选择点 $j$ 的转移概率[9]为

$$Pij^{m}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s\in \text{allowed}_m}\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)} & j\in \text{allowed}_{\text{m}}\\ 0 & \text{otherwise}\end{cases} \tag{23}$$

其中:$\text{allowed}_m=\{1,2,\cdots,n\}$表示蚂蚁 $m$ 下一步允许选择的城市;$\tau_{ij}$为路径$(i,j)$上的信息素浓度;$\eta_{ij}$为期望启发式因子;$\alpha$ 和 $\beta$ 为两个参数,分别反映了蚂蚁在运动过程中所积累的信息和启发信息在蚂蚁选择路径中的相对重要性。

4)2-opt 邻域搜索

在一次迭代结束后,对产生的路径进行邻域搜索,本文 2-opt 邻域搜索主要有三个步骤:a)重置,表示从某一条路径中取出一个需求点插入到另一条路径中(inter-route),或者插入自身的另一个位置中(intra-route);b)互换,表示分别取两条不同路径中的两个需求点,将它们互换;c)2-opt*,表示分别取两条不同路径的后半段,将它们互换。

具体的邻域搜索如图 2 所示。

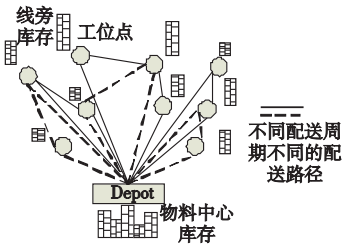



图1 物料循环配送问题描述

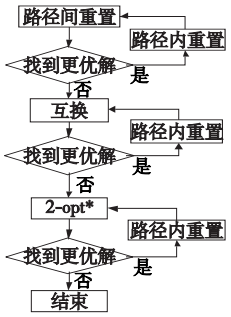



图2 2-opt邻域搜索算法流程

5)信息素更新

每次迭代结束,只有最优解所属路径上的信息素被更新。信息素更新如式(24)(25)所示。

$$\tau_{ij}(t+1)=\rho^{*}\tau_{ij}(t)+\Delta\tau_{ij}^{\text{best}}(t) \tag{24}$$

$$\Delta\tau_{ij}^{\text{best}}(t)=\begin{cases}\dfrac{1}{L^{\text{best}}} & \text{若边}(i,j)\text{在最优路径中}\\ 0 & \text{otherwise}\end{cases} \tag{25}$$

其中:$L^{\text{best}}$为最优路径长度,$\tau_{ij}(t)$为时刻 $t$ 边$(i,j)$上的信息素浓度。

### 3.3 双层模型求解

考虑到本文问题的特性,设计了双层迭代算法,主要思想:通过给出上层模型的初始解,代入下层模型进行求解,由于上下层的最优解之间相互影响,对此进行重复迭代,达到一个相对平衡,即系统的整体最优。具体的实现步骤如下:

a)设置迭代次数 $NC=0$,根据 3.1 节中的策略给一个初始的配送策略 $T_i^0=(T_2,T_3,\cdots,T_N)$。

b)根据 $T_i$ 计算配送量以及线旁库存费用 $I$,运用 3.2 节中的算法求出配送路径以及运输费用 $T$,得到上下层的总费用 $G$,$NC=NC+1$。

c)重复步骤 a)和 b),若 $NC<NC_{\max}$,则转步骤 a);否则转步骤 d)。

d)输出结果,配送周期策略、量、路径、费用 $I$、$T$ 和 $G$。

## 4 算例验证

现以 10 个需求点和一个物料中心组成的配送网络为例。涉及的参数为:$N=10$,$T^{*}=10$ h,$T=1$ h,$\lambda_i$ 取 1 到 10 的随机整数,$\alpha=1$,$\beta=4$,$\rho=0.95$,车容量 $m^{*}$ 有 200 和 50 两种,单位库存费用 $C_I=10$,单位运输费用 $C_{T1}$ 分别取 10、9、…、1。输入数据如表 1 所示。

表 1 算例输入数据

| 编号 | 坐标 | $\lambda_i$ | $Q_i$ | $S_i$ | 编号 | 坐标 | $\lambda_i$ | $Q_i$ | $S_i$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0,0 | — | — | — | 7 | 5,15 | 5 | 50 | 3 |
| 2 | -8,13 | 5 | 50 | 3 | 8 | 8,5 | 6 | 60 | 3 |
| 3 | 0,15 | 6 | 60 | 3 | 9 | -10,3 | 4 | 40 | 3 |
| 4 | 10,10 | 4 | 40 | 3 | 10 | 3,12 | 8 | 80 | 5 |
| 5 | -4,8 | 8 | 80 | 5 | 11 | 1,5 | 10 | 100 | 6 |
| 6 | 5,5 | 10 | 100 | 6 | | | | | |

本文算法用 MATALB 编程,迭代 500 次,当 $C_I/C_{T1}=10/10$ 时,最优结果如下:配送周期 $T_i=[5,5,5,5,5,5,5,5,10,5,5]$,路径为 1-5-2-3-10-7-4-1,1-8-6-11-1 和 1-9-1,前两条的配送周期是 5,频次为 2;后一条周期为 10,频次为 1,不同的线条表示不同的配送频次;总费用为 4 292.57。路径如图 3(a)所示。

为了说明文中算法的有效性以及问题的全面性,单位运输费用 $C_{T1}$ 分别取 10、9、…、1。它们的最优路径如图 3(b)~(d)所示。$C_I/C_{T1}$取不同比值时,循环配送双层规划结果的具体数据如表 2 所示。

表 2 $C_I/C_{T1}$取不同比值时循环配送双层规划结果

| $C_I/C_{T1}$ | 派送周期策略 | 频次 | 小车容量 | 派送路径 | 路径长度 | 装载率/% | 总长度 | 总费用 |
|---|---|---|---|---|---|---|---|---|
| 10/10 | $T_i=[5,5,5,5,5,5,5,5,10,5,5]$ | 2 | 200 | 1-5-2-3-10-7-4-1 | 74.188 | 90.0 | 148.38 | 4292.57 |
| | | | 50 | 1-8-6-11-1 | | 65.0 | | |
| | | 1 | 50 | 1-9-1 | 20.881 | 80 | 20.881 | |
| 10/7 | $T_i=[5,5,5,5,5,2,5,5,5,5,2]$ | 2 | 200 | 1-8-4-7-10-3-2-9-1 | 76.512 | 95 | 153.02 | 3737.118 |
| | | | 50 | 1-5-1 | | 80 | | |
| | | 5 | 50 | 1-11-6-1 | 16.17 | 80 | 80.85 | |
| 10/4 | $T_i=[5,5,5,2,2,5,5,5,2,2]$ | 2 | 200 | 1-9-2-3-7-4-8-1 | 55.775 | 75 | 111.55 | 3257.12 |
| | | | 50 | 1-5-10-1 | 45.546 | 64 | 227.73 | |
| | | 5 | 50 | 1-11-6-1 | | 80 | | |
| 10/1 | $T_i=[2,2,2,1,1,2,2,2,1,1]$ | 5 | 50 | 1-9-2-3-7-4-1 | 73.966 | 96 | 369.83 | 2183.26 |
| | | | 50 | 1-8-1 | | 24 | | |
| | | 10 | 50 | 1-11-5-10-6-1 | 33.343 | 72 | 333.43 | |

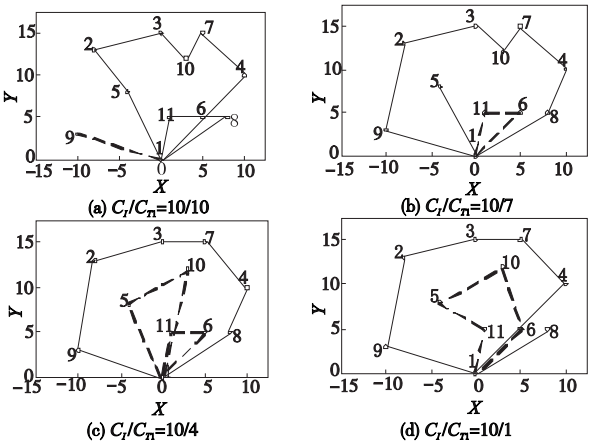



**图3　最优路径**

表 3　改进 MMAS 及本文算法结果比较

| $C_l/C_{Tl}$比 | 方案总费用 | | 费用节省/% |
|---|---|---|---|
| | 改进 MMAS | 本文算法 | |
| 10/10 | 5211.2 | 4292.57 | 17.63 |
| 10/7 | 4877.84 | 3737.118 | 23.39 |
| 10/4 | 4544.48 | 3257.12 | 28.33 |
| 10/1 | 4211.12 | 2183.26 | 48.15 |

由表 2 和表 3 可得,在相同 $C_l/C_{Tl}$ 比的情况下,将本文算法与仅仅用改进 MMAS 算法求解的结果相比,当 $C_l/C_{Tl}$ 取 10/10、10/7、10/4、10/1 时,总费用分别节省了 17.63%、23.39%、28.33%、48.15%;也可以由图 4 得,随着 $C_l/C_{Tl}$ 比值从 10/10 到 10/1 逐渐增大,本文算法节省的费用也逐渐上升。在实际运用中,根据不同公司、不同车间的 $C_l/C_{Tl}$ 比,可以求得不同的配送方案,高频次小批量配送或大批量低频次配送,令配送的整体费用最小。综上,本文的算法对于解决循环配送问题是有效的,结果优于改进的 MMAS。

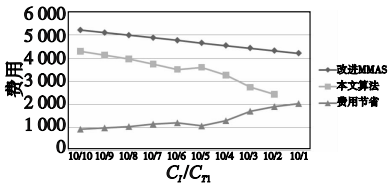



**图4　$C_l/C_n$比值对结果的影响**

## 5　结束语

本文主要研究了车间物料循环配送规划问题,建立基于 bi-level 规划数学模型,并以基于 2-opt 改进的 MMAS 的双层迭代算法求解物料循环配送量、周期和路径,针对不同公司或不同车间的具体情况,以库存费用和运输费用的比值为因子,对算法结果进行分析。相比于 IMMAS,本算法引进了线旁库存,首创性地提出了确定配送周期的策略,结合下层车辆路径模型进行迭代求解,结果表明费用节省显著,显示了本算法的优越性,为实际物料配送等提供了实用价值。进一步地,可以针对更大规模的问题,结合供应链上下游,快速解出最优配送周期、配送量和最优配送路径。

**参考文献:**

[1] PESCE L F, FRAZAO C D, CIVINSKAS J, *et al*. The next step for a lean production: milk-run(2000-01-3230)[R]. 2000.

[2] CHUAH K H. Optimization and simulation of just-in-time supply pickup and delivery systems[D]. Lexington: University of Kentucky, 2004.

[3] RUSDIANSYAH A, TSAO D B. An integrated model of the periodic delivery problems for vending-machine supply chains [J]. Journal of Food Engineering, 2005, 70(3): 421-434.

[4] 王旭,施朝春,葛显龙. 基于 Milk Run 模式的库存运输整合优化[J]. 计算机集成制造系统,2011,17(4):852-857.

[5] 王亮,孙绍荣,吴晓层. 最小化运输与库存费用的两级分销策略分析[J]. 系统工程理论与实践,2005,25(10):33-38.

[6] CALVETE H I, GALE C, OLIVEROS M J. Bi-level model for production-distribution planning solved by using ant colony optimization [J]. Computers and Operations Research, 2011, 38(1): 320-327.

[7] LIU Shu-chu, LEE W. A heuristic method for the inventory routing problem with time windows[J]. Expert Systems with Applications, 2001, 38(10): 13223-13231.

[8] 李哲,夏立,庄浩俊,等. MMAS-EC 算法求解旅行商问题[J]. 计算机工程与应用,2011,47(9):41-47.

[9] ZHANG Xiao-xia, TANG Li-xin. A new hybrid ant colony optimization algorithm for the vehicle routing problem[J]. Pattern Recognition Letters, 2009, 30(9): 848-855.

(上接第 4161 页)种基于参数化流形学习的压缩传感重构方法;最后通过实验实现了基于参数化流形学习的压缩传感重构方法。把提出的方法与已有的压缩传感重构方法进行了比较,充分验证了基于参数化流形学习的压缩传感重构方法的有效性。

**参考文献:**

[1] WANG Yao, LIN Shu-nan. Error-resilient video coding using multiple description motion compensation[J]. IEEE Trans on Circuits and Systems for Video Technology, 2002, 12(6): 438-452.

[2] BARANIUK R. Compressive sensing[J]. IEEE Signal Processing Magazine, 2007, 24(4): 118-121.

[3] VAISHAMPAYAN V A. Design of multiple description scalar quantizers[J]. IEEE Trans on Information Theory, 1993, 39(3): 821-834.

[4] MEKUZ N, BAUCKHAGE C, TSOTSOS J K. Subspace manifold learning with sample weights[J]. Image and Vision Computing, 2009, 27(1-2): 793-801.

[5] ZHANG Zhen-yue, ZHA Hong-yuan. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment[J]. SIAM Journal of Scientific Computing, 2004, 26(1): 313-338.

[6] WANG Jing, ZHANG Zhen-yue, ZHA Hong-yuan. Adaptive manifold learning[J]. Neural Information Processing Systems, 2005, 38(45): 1473-1480.

[7] 陈维桓. 微分流形初步[M]. 北京:高等教育出版社,2001:2-9.

[8] 练秋生,张红卫,陈书贞,等. 融合图像块低维流形特性与解析轮廓波稀疏性的压缩成像算法[J]. 电子与信息学报,2012,34(1):207-212.

[9] 赵凤. 半监督谱聚类特征向量选择算法[J]. 模式识别与人工智能,2011,24(1):148-156.

[10] NI J, TURAGA P. Example-driven manifold priors for image deconvolution[J]. IEEE Trans on Image Processing, 2011, 20(11): 3086-3096.

[11] MOHIMANI G H, BABAIE-ZADEH M, JUTTEN C. Complex-valued sparse representation base on smoothed L0 norm[C]//Proc of IEEE International Conference on Acoustics, Speech and Signal Processing. 2008: 3881-3884.

式(7)计算最近的聚类中心:

$$\begin{cases}\hat{i}=\arg\min_i \| p-u_i \| \\ \hat{\alpha}=V_i^{\mathrm{T}}(p-u_{\hat{i}})\end{cases} \tag{7}$$

投影算子通过式(8)进行建立:

$$\mathrm{Proj}_M(p)=u_{\hat{i}}+V_{\hat{i}}\hat{\alpha} \tag{8}$$

其中,$\hat{\alpha}$ 由式(7)计算得到。通过这种参数化流形学习的方法就可以求出每个图像块在其所在的流形空间上的投影。

### 3.2　重构算法实现

基于参数化流形学习的压缩传感重构算法实现步骤如下:

a)训练非线性流形 $M$,然后用相似度矩阵稀疏化的谱聚类方法对 $M$ 进行聚类,把 $M$ 分成 $J$ 个子流形(本实验中取 $J=100$),即有 $M=M_1\cup M_2\cup\cdots\cup M_J$。

b)对非线性流形 $M$ 进行随机投影,得到投影后的非线性流形 $M_P$,即有 $M_P=M_{P1}\cup M_{P2}\cup\cdots\cup M_{PJ}$。

c) 对要测试的图像 $X$ 进行分块处理,然后进行投影,得到每一块 $x$ 的观测值 $x_p$,根据参数化流形学习的方法,采用式(7)(8)把 $x_p$ 向非线性流形 $M_P$ 上进行投影,得到每个观测值在 $M_P$ 上的投影值 $x_p^1$。

d)将 $x_p^1$ 向集合 $C=\{x_p^1:\| y-\Phi x_p^1 \|_2^2\leqslant\varepsilon_1\}$ 投影得到

$$x_p^2=x_p^1+\Phi^{\mathrm{T}}(y-\Phi x_p^1)$$

e)把 $x_p^2$ 重新组合成新的图像 $X_1$,再对 $X_1$ 进行分块处理,把它分成具有重叠的图像块,得到每一块 $x_p^3$,用参数化流形学习的方法把 $x_p^3$ 向非线性流形 $M$ 上进行投影,得到投影值 $x_p^4$。

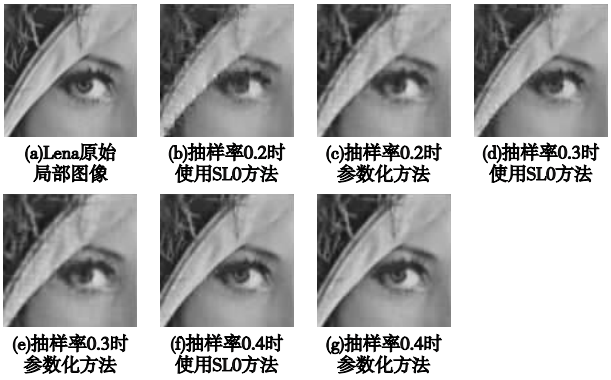
f)把 $x_p^4$ 重新组合成新的图像 $X_2$,再对 $X_2$ 进行分块处理得到 $x_p^5$,令 $x_p^5\rightarrow x_p^1$,转到 d)迭代循环。实验中设置循环次数为 120 次,最后得到重构的图像 $\bar{X}=X_2$。

## 4　实验结果

为了比较算法的性能,本章选择尺寸为 512×512 的标准二维图像 Lena 和尺寸为 256×256 的标准二维图像 house 进行实验。

图 2 是采用基于参数化流形学习的压缩传感重构方法(以下简称参数化方法)和压缩传感重构算法中效果比较好的 SL0 方法[11]分别对 Lena 图像进行重构的对比结果。实验中为了便于观察,只是显示了 Lena 图像局部区域的放大效果。比较图 2(b)(c)可以清楚地看到,在抽样率为 0.2 时,采用 SL0 方法重构的图像具有明显的块效应,重构的图像也比较模糊;采用参数化方法重构的图像的块效应得到明显的改善,同时也比采用 SL0 方法重构的图像清晰。比较图 2(d)(e),在抽样率为 0.3 时,随着抽样率的增加,重构图像变得清晰,应用 SL0 方法重构的图像依然具有块效应,采用参数化算法重构图像的块效应得到改善。在图 2(g)中,当抽样率为 0.4 时,采用参数化方法基本上消除了分块重构图像的块效应,同时重构的图像也变得非常清晰。

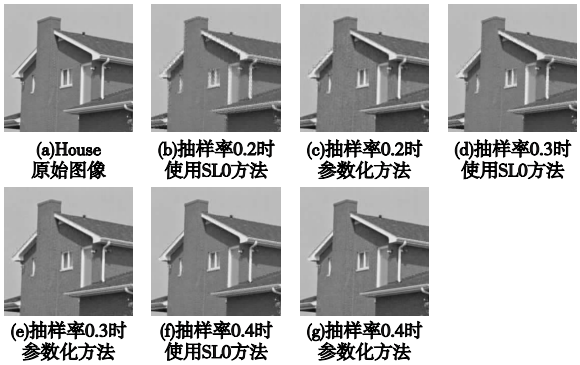
图 3 是对 house 图像采用参数化方法与 SL0 方法重构效果的比较。比较图 3(b)和(c)可以明显地看到,采用 SL0 方法重构的 house 图像的边缘区域非常模糊,具有明显的锯齿效应;而采用参数化方法重构的图像比较清晰,锯齿效应得到很大的改善,特别是对于屋顶的右侧部分边缘区域基本消除了锯齿效应。比较图 3(d)(e),在抽样率为 0.3 时也具有相似的效果。从图 3(g)中可以看到,当抽样率为 0.4 时,采用参数化方法基本上消除了重构图像边缘区域的锯齿效应,重构的图像变得非常清晰。



(a)Lena原始局部图像　(b)抽样率0.2时使用SL0方法　(c)抽样率0.2时参数化方法　(d)抽样率0.3时使用SL0方法

(e)抽样率0.3时参数化方法　(f)抽样率0.4时使用SL0方法　(g)抽样率0.4时参数化方法

**图2　Lena图像实验结果**



(a)House原始图像　(b)抽样率0.2时使用SL0方法　(c)抽样率0.2时参数化方法　(d)抽样率0.3时使用SL0方法

(e)抽样率0.3时参数化方法　(f)抽样率0.4时使用SL0方法　(g)抽样率0.4时参数化方法

**图3　House图像实验结果**

表 1 是实验中对 Lena 和 house 图像分别采用参数化方法和 SL0 方法进行重构,在不同抽样率下的重构图像的 PSNR。从表 1 中的数据可以清楚地看到,采用参数化方法重构的图像比采用 SL0 方法重构的图像的 PSNR 值提高了很多。以 Lena 图像为例,在抽样率为 0.2 时,PSNR 提高了 2 dB;在抽样率为 0.3 时,PSNR 提高了 1.34 dB;在抽样率为 0.4 时,PSNR 提高了 0.69 dB;在抽样率为 0.5 时,PSNR 提高了 0.74 dB。对于尺寸为 256×256 的 house 图像,在抽样率为 0.2 时,PSNR 提高了 2.49 dB;在抽样率为 0.3 时,PSNR 提高了 1.83 dB;在抽样率为 0.4 时,PSNR 提高了 1.13 dB;在抽样率为 0.5 时,PSNR 提高了 0.84 dB。在各种情况下,采用参数化方法与采用 SL0 方法相比,PSNR 值都得到了明显的提高,从而充分验证了参数化算法的有效性。

表 1　两种重构方法 PSNR 的比较　/dB

| 图像 | 方法 | 抽样率 | | | |
|---|---|---|---|---|---|
| | | 20% | 30% | 40% | 50% |
| Lena | SL0 | 29.65 | 32.60 | 34.43 | 36.18 |
| | 参数化方法 | 31.65 | 33.94 | 35.12 | 36.92 |
| house | SL0 | 29.77 | 32.73 | 34.66 | 36.90 |
| | 参数化方法 | 32.26 | 34.56 | 35.79 | 37.74 |

## 5　结束语

本文研究了图像块流形模型和图像块本身所具有的低维流形特性,然后对训练的图像块进行聚类,提出了图像块在流形上投影的新的计算方法,即参数化流形学习方法;并把参数化流形学习方法同压缩传感相结合,提出了一(下转第 4179 页)