

# 混合最大最小蚁群算法在VRPTW中的应用

苏红畏<sup>1</sup>,刘希玉<sup>1</sup>,王晓敏<sup>2</sup>

(1. 山东师范大学 管理与经济学院, 山东 济南 250014;

2. 山东师范大学 信息科学与工程学院, 山东 济南 250014)

**摘 要:**为解决有时间窗车辆路径问题,采用两个最大最小蚁群系统,一个蚁群最小化车辆数量,另一个蚁群最小化旅行距离。通过分析有时间窗车辆路径问题和旅行商问题的区别,改进了最大最小蚁群算法中状态转移策略,并增加与可用车辆相同数量的虚拟仓库,使这两个蚁群使用独立的信息素但通过分享全局最优解来协作,算法还结合了2-opt局部搜索,从而减少了算法的计算时间并避免过早收敛。仿真实验结果表明,该算法性能优良,能有效地求解有时间窗车辆路径问题。

**关键词:**最大最小蚁群算法;时间窗车辆路径问题;2-opt局部搜索

**中图分类号:**TP391

**文献标识码:**A

**文章编号:**1673-629X(2010)02-0090-05

## Hybrid Max - Min Ant System for Vehicle Routing Problem with Time Windows

SU Hong-wei<sup>1</sup>, LIU Xi-yu<sup>1</sup>, WANG Xiao-min<sup>2</sup>

(1. School of Management and Economics, Shandong Normal University, Jinan 250014, China;

2. Information Science and Engineering Institute, Shandong Normal University, Jinan 250014, China)

**Abstract:** Two ant colonies employing max-min ant colony system, one minimizes the number of vehicles while the other minimizes the traveled distances, have been designed to tackle the VRPTW (vehicle routing problem with time window). By analyzing the difference between the vehicle routing problem with time window and traveling salesman problem, the state transition strategy of the max-min ant colony system is improved and the depot is duplicated a number of times equal to the number of available vehicles. These two colonies work by using independent pheromone trails but collaborate by sharing the global best feasible solution, and integrating 2-opt local search method in order to decrease computing time and overcome early convergence. Simulation results show that the algorithm present is feasible and valid for VRPTW.

**Key words:** max-min ant system; vehicle routing problem with time window; 2-opt local search

## 0 引言

车辆路径问题(vehicle routing problem, VRP)最早由Dantzig和Ramser<sup>[1]</sup>于1959年提出。目前求解VRP的方法可以分为精确算法和启发式算法两大类,典型的启发式算法有禁忌搜索算法、模拟退火算法、遗传算法、神经网络算法、蚁群优化算法等。

蚁群算法由意大利学者Marco Dorigo<sup>[2]</sup>于1991年

在其博士论文中首次提出。为了克服基本蚁群算法搜索效率低和早熟停滞的不足,Thomas Stützle等人<sup>[3]</sup>提出的MMAS(Max-Min Ant System)在解决二次分配问题、大学课程时间表问题、装箱问题等组合优化问题<sup>[4]</sup>时取得了显著的效果。赵传信等<sup>[5]</sup>提出了将粒子群算法和禁忌搜索算法相结合的禁忌搜索粒子群算法,来解决车辆路径问题;刘志硕等<sup>[6]</sup>在分析车辆路径和中国邮递员问题的基础上构造了具有自适应功能的混合蚁群算法;万旭等<sup>[7]</sup>提出了改进的最大最小蚁群算法在VRPTW中的应用。但是这些算法都存在计算时间长和早熟停滞的缺点,笔者受Gambardella<sup>[8]</sup>的启发在分析VRPTW两个目标的基础上,利用两阶段构造策略并结合2-opt局部搜索提出了解决VRPTW的混合MMAS。

收稿日期:2009-05-31;修回日期:2009-08-14

基金项目:国家自然科学基金项目(60873058);山东省自然科学基金项目(Z2007G03);“泰山学者”建设工程专项经费资助项目(2005-2010)

作者简介:苏红畏(1982-),男,山东阳信人,硕士研究生,研究方向为进化计算与智能算法;刘希玉,“泰山学者”,教授,博士生导师,研究方向为人工智能与数据挖掘。

## 1 VRPTW 问题描述及模型

### 1.1 VRPTW 问题描述

VRPTW (vehicle routing problem with time windows) 即带有时间窗的车辆路径问题<sup>[9]</sup>是指:已知有一批客户,每个客户的地理位置坐标和货物需求已知,每辆车都从仓库出发完成若干客户点的货物分配任务后再回到仓库,车辆的负载能力给定。假设每个客户只能被访问一次,每辆车所访问客户的需求总和不能超过车辆的负载能力。每个客户  $i$  带有一个时间窗  $[e_i, l_i]$ ,  $e_i$  为客户  $i$  允许服务的最早开始时间,  $l_i$  为客户  $i$  允许的服务最迟开始时间,车辆必须在这个时间范围内对客户进行服务。若车辆在时间  $e_i$  之前到达客户点  $i$ ,则要等到  $e_i$  才开始服务。问题的目标是用最少的车辆数和总路长来完成所有任务。

### 1.2 VRPTW 数学模型

记每辆车的最大载重量为  $D$ , 客户  $i$  的需求为  $d_i$  ( $i \in V$ ),  $V$  为客户集合。 $se_i$  表示对客户  $i$  服务的开始时刻,  $sl_i$  表示服务终止时间,  $s_i$  表示完成任务所需的服务时间, 即  $s_i = sl_i - se_i$ ,  $t_{ij}$  为车辆从客户  $i$  到客户  $j$  的旅行时间,  $w_i$  表示开始为客户  $i$  服务所需的等待时间, 通常取  $t_{ij} = c_{ij}$ 。

设

$$x_{ijk} = \begin{cases} 1, & \text{若车辆 } k \text{ 经过路径 } (i, j) \\ 0, & \text{否则} \end{cases}$$

由此可得该问题的数学模型为:

$$\begin{aligned} \min \quad Z = & \left\{ \sum_i \sum_j \sum_k c_{ij} x_{ijk}, \sum_j \sum_k x_{0jk} \right\} \\ \text{s. t.} \quad & \begin{cases} \sum_i d_i y_{ki} \leq D, \forall k & (a) \\ \sum_k y_{ki} = 1, i \in V & (b) \\ \sum_i x_{ijk} = y_{kj}, j \in V, \forall k & (c) \\ \sum_j x_{ijk} = y_{ki}, i \in V, \forall k & (d) \\ \sum_{i,j \in S \times S} \sum_k x_{ijk} \leq |S| - 1, S \subseteq V & (e) \\ sl_i + t_{ij} \leq se_j, i, j \in V, \forall k & (f) \\ e_i \leq se_i \leq l_i, i \in V & (g) \\ x_{ijk}, y_{kj} \in \{0, 1\}, i, j \in V, \forall k \end{cases} \end{aligned}$$

其中,

约束(a)为车辆负载限制;

约束(b)保证每个车辆对每个客户只访问一次;

约束(c)~(e)保证形成可行回路;

约束(f)表示一条线路上两相邻任务存在的条件;

约束(g)表示时间窗约束;

目标为车辆数最少和总路长最短。

## 2 混合最大最小蚁群算法

### 2.1 改进的最大最小蚁群算法描述

在MMAS<sup>[3]</sup>基础上文中改进的最大最小蚁群算法如下。

Step 1: 状态转移规则采用下式:

$$\begin{aligned} & \text{if } q \leq q_0 \\ & \text{then } j = \arg \max \{ [\tau_{ij}]^a [\eta_{ij}]^\beta [r_{ij}]^\gamma [w_{ij}]^\theta \} \\ & \text{else } p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^a [\eta_{ij}]^\beta [r_{ij}]^\gamma [w_{ij}]^\theta}{\sum_{i \in N_i^k} [\tau_{iu}]^a [\eta_{iu}]^\beta [r_{iu}]^\gamma [w_{iu}]^\theta} & \text{if } j \in N_i^k \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (1)$$

其中,  $q_0$  为区间  $[0, 1]$  中的任意实数,  $q$  为  $[0, 1]$  中的随机数;  $p_{ij}^k$  表示位于结点  $i$  的车辆  $k$  选择结点  $j$  的概率;  $\tau_{ij}$  表示边  $(i, j)$  上信息素的值;  $\eta_{ij}$  表示由结点  $i$  选择  $j$  的期望值;  $r_{ij}$  为车辆从结点  $i$  到结点  $j$  的到达时间与时间窗上限的时间差;  $w_{ij}$  为车辆从结点  $i$  到结点  $j$  后, 在结点  $j$  的等待时间;  $\gamma$  和  $\theta$  分别为时间差和等待时间的相对重要性;  $N_i^k$  表示位于结点  $i$  的蚂蚁  $k$  可以直接到达的相邻结点的集合。

Step 2: 只有迭代最优  $\psi^b$  或全局最优  $\psi^{gb}$  的蚂蚁才允许释放信息素, 即

$$\tau_{ij} = (1 - \rho) \tau_{ij} + \Delta \tau_{ij}^{bs}, \text{ 对于 } \forall (i, j) \in \psi^{bs} \quad (2)$$

其中  $\rho$  为信息素的挥发系数,  $\Delta \tau_{ij}^{bs} = 1/f(\psi^{bs})$  或者  $\Delta \tau_{ij}^{bs} = 1/f(\psi^{gb})$ 。

Step 3: 为解决某些边上信息素增长过快以防止搜索停滞现象的出现, 把信息素的大小限制在  $\tau_{\max} = 1/\rho f(\psi^{bs})$  到  $\tau_{\min} = \tau_{\max}(1 - \sqrt[n]{0.05})/((n/2 - 1) \cdot \sqrt[n]{0.05})$  之间。

Step 4: 信息素的初始值设定为其取值范围的上界  $\tau_{\max}$ , 并与一个小的信息素蒸发率  $\rho$  相结合, 目的是使得算法可以在最初的搜索步骤中探索更多可能的路径。

### 2.2 混合最大最小蚁群算法在VRPTW上的应用

#### 2.2.1 HMMAS-VRPTW 算法

在HMMAS-VRPTW算法中采用两个蚁群, 第一个蚁群MMAS-VEI最小化使用的车辆数目; 第二个蚁群MMAS-TIME最优化由MMAS-VEI得到的可行解的距离。两个蚁群的信息素独立但两者通过分享由HMMAS-VRPTW求出的变量  $\psi^{gb}$  (全局最优解) 来协作。HMMAS-VRPTW算法的详细过程如下:

HMMAS-VRPTW()

Step 1:  $\psi^{gb} \leftarrow \psi^0$ , 其中  $\psi^0$  由最近邻启发式得到。

Step 2:

Repeat

将  $\psi^{pb}$  中车辆数赋给变量  $v$ ;

然后依次调用 MMAS-VEI( $v-1$ ) 和 MMAS-TIME( $v$ );

While MMAS-VEI 和 MMAS-TIME 没有结束

如果 MMAS-VEI 或 MMAS-TIME 可找到一个改进的解  $\psi'$ , 那么  $\psi^{pb} \leftarrow \psi'$ 。

如果  $\psi^{pb}$  中车辆数  $< v$ , 那么结束 MMAS-VEI 和 MMAS-TIME。

end While

Until a stopping criterion is met

## 2.2.2 MMAS-VEI 算法

MMAS-VEI( $s$ )

Step 1: 将含  $s$  个车辆的初始解赋给  $\psi^{MMAS-VEI}$ , 其中,  $s = v - 1$ , 变量  $\psi^{MMAS-VEI}$  保存本次迭代中含最高访问数量顾客的解(此解常常是不可行解)。

Step 2:

Repeat

对每只蚂蚁分别构建一个解  $\psi^k$ , 即调用 new\_active\_ant( $k, local\_search = FALSE, IN$ )。若顾客  $j$  未被蚂蚁  $k$  访问, 则变量  $IN_j++$ 。其中,  $IN_j$  表示顾客  $j$  未被插入到解中的次数。

如果存在一个蚂蚁  $k$ , 使得  $\psi^k$  中顾客数目  $> \psi^{MMAS-VEI}$  中顾客数目, 那么  $\psi^{MMAS-VEI} \leftarrow \psi^k$ , 否则, 清空向量  $IN$ 。如果  $\psi^{MMAS-VEI}$  是可行解, 那么, 将其保存到 HMMAS-VRPTW 中。

然后据公式(2)对信息素执行全局更新, 即  $\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{MMAS-VEI} \forall (i, j) \in \psi^{MMAS-VEI}$ ,  $\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho / f(\psi^{pb}) \forall (i, j) \in \psi^{pb}$ 。

如果  $\tau_{ij} \in [\tau_{min}, \tau_{max}]$ , 那么将信息素的值  $\tau_{ij}$  重新初始化到  $[\tau_{min}, \tau_{max}] \forall (i, j) \in \psi^{pb}$ 。

Until a stopping criterion is met

## 2.2.3 MMAS-TIME 算法

MMAS-TIME 的目标是使得由 MMAS-VEI 求出的路径尽可能短。在 MMAS-TIME 阶段让  $m$  只人工蚂蚁调用 new\_active\_ant() 来构造问题的解  $\psi^1, \dots, \psi^m$ , 这类类似于旅行商问题中路径的构建过程。然后将  $\psi^1, \dots, \psi^m$  与  $\psi^{pb}$  比较, 一旦有一个比  $\psi^{pb}$  更好的解, 设为  $\psi^k$ , 就将  $\psi^k$  加入 HMMAS-VRPTW 中。最后, 根据公式(2)和  $\tau_{min}, \tau_{max}$  执行信息素的全局更新。详细算法如下:

MMAS-TIME( $v$ )

Step 1: 用  $v$  来初始化信息素和数据结构。

Step 2:

Repeat

对每只蚂蚁分别构建一个解  $\psi^k$ , 即调用 new\_active\_ant( $k, local\_search = TRUE, 0$ ), 如果存在一个蚂蚁  $k$ , 使得  $\psi^k$  是可行解并且  $\psi^k$  比  $\psi^{pb}$  更优, 那么将  $\psi^k$  存入 HMMAS-VRPTW。

然后对信息素公式(2)执行全局更新, 即  $\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho / f(\psi^{pb}) \forall (i, j) \in \psi^{pb}$ 。

如果信息素更新后的值  $\tau_{ij} \in [\tau_{min}, \tau_{max}]$ , 那么将  $\tau_{ij}$  重新初始化到  $[\tau_{min}, \tau_{max}] \forall (i, j) \in \psi^{pb}$ 。

Until a stopping criterion is met

## 2.2.4 解的构建过程

new\_active\_ant( $k, local\_search, IN$ )

Step 1: 将蚂蚁  $k$  放到随机选择的虚拟仓库  $i$  上, 并置当前时刻和蚂蚁  $k$  的负载为 0, 即  $current\_time_k \leftarrow 0$ ,  $load_k \leftarrow 0$ 。

Step 2:

Repeat

当蚂蚁  $k$  位于结点  $i$  时, 计算其可行结点的集合  $N_i^k$ , 并对  $\forall j \in N_i^k$  计算启发式信息  $\eta_{ij}$  如下:

$Delivery\_time_j \leftarrow \max(current\_time_k + t_{ij}, b_j)$

$\Delta time_{ij} \leftarrow Delivery\_time_j - current\_time_k$

$distance_{ij} \leftarrow \Delta time_{ij} * (e_j - current\_time_k)$

$distance_{ij} \leftarrow \max(1.0, (distance_{ij} - IN_j))$

$\eta_{ij} \leftarrow 1.0 / distance_{ij}$

然后, 据公式(1)选择下一个可能的结点  $j$ , 并将  $j$  加入蚂蚁  $k$  的解  $\psi^k$  中。如果  $j$  是仓库, 那么  $current\_time_k \leftarrow 0, load_k \leftarrow 0$ 。

据公式(3)执行信息素局部更新, 即

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0 \quad (3)$$

然后置  $j$  为蚂蚁  $k$  的当前结点。

Until  $N_i^k = \emptyset$ , 即若无可行结点可用, 则结束 Step 3。

Step 3: 插入未访问的顾客来扩展路径  $\psi^k$ 。

Step 4: 如果参数  $local\_search = TRUE$  并且  $\psi^k$  是可行解, 那么使用局部搜索过程来优化可行解的路径, 即  $\psi^k \leftarrow 2-opt\_local\_search(\psi^k)$ 。

文中采用常用于旅行商问题中的 2-opt 局部搜索<sup>[6]</sup>来改进可行解的质量, 详细过程如下。

## 2.2.5 2-opt 局部搜索过程

设可行解  $\psi^k$  由  $p$  个子回路组成, 记为  $\{L_1, L_2, \dots, L_p\}$ , 更新可行解的过程实际上就是对各个子回路分别应用 2-opt。设  $W_k$  是路径  $L_k$  经过结点的有序排列,  $W_k = \{v_0, v_1, v_2, \dots, v_n, v_0\}$ ,  $v_i, v_j$  是从  $W_k$  中随机选择的 2 个结点, 交换这两点的位置得到路径  $L'_k$ ,  $C_N$  为

没有任何改进的最大循环次数,文中用到的 2-opt 局部搜索,即 2-opt\_local\_search( $\psi^k$ ) 其步骤如下:

- Step 1:初始化循环次数变量  $t = 1$ ,当前最优解  $\psi^* = \psi$ ,其长度为  $f(\psi^*)$ 。
- Step 2:在  $W_k$  中随机选择 2 个结点  $v_i, v_j, i < j$ ,且  $v_i$  和  $v_j$  不相邻。
- Step 3:求  $\Delta C_d = d(x_{i+1}, x_j) + d(x_i, x_{j+1}) - d(x_{i+1}, x_i) + d(x_j, x_{j+1})$ ,若  $\Delta C_d > 0$ ,则不交换,  $t++$ ,转到 Step 4,否则交换,更新  $W_k$ ,更新后的解为  $\psi'$ ,则最优解  $\psi^* = \psi', t = 1$ ,转到 Step 2。
- Step 4:若  $f(\psi^*)$  在最后  $C_N$  个循环中没有减少,则本算法结束;否则转到 Step 2。

3 仿真实验及分析

将文中提出的 HMMAS-VRPTW 算法应用于 Solomon 在文献[10]中所述的基准实例。每个实例都含有 100 个客户和一个中心仓库,并规定了车辆负载、服务时间和客户的地理位置坐标、货物需求量、时间窗等。实验是在 AMD Athlon(tm)64 X2 Dual, Core Processor 4000 + 2.10GHz,内存为 1G 的硬件环境,基于 Redhat Linux 9.0 平台在 GCC 3.2.2 编译器下用 C++ 语言实现。

3.1 参数设定及试验结果分析

实验开始将每个虚拟仓库都放置一个蚂蚁,蚂蚁个数与虚拟仓库的个数相同(各个实例蚂蚁个数  $n$  取值不同)。 $\alpha = 1.00, \beta = 2.00, \rho = 0.10, q_0 = 0.95, \gamma = 0.10, \theta = 0.10, \tau_0 = 1/\rho C^n$ ( $C^n$  由文献[11]获得)。对基准实例 R1 做 10 次实验,每次实验的时间限制为 60.00 秒,然后取全局最优解的平均值和最优解,见表 1。由表 1 可知,算法 HMMAS-VRPTW 的整体计算结果比文献[7]中结果要好,个别最优解如 R103 已经接近已知最优解。

表 1 文中平均值解、最好解与已知最优解的比较

实例	BestSoFar	MMAS-AVG	MMAS-BEST	HMMAS-AVG	HMMAS-BEST
R101	1650.80/19	1701.58/18	1660.71/18	1713.71/18	1676.23/18
R102	1486.12/17	1554.91/17	1503.47/17	1564.98/17	1519.45/17
R103	1292.68/13	1338.57/13	1307.74/13	1332.64/13	1301.28/13
R104	1007.31/9	1050.72/9	1032.29/9	1047.19/9	1028.75/9
R105	1377.11/14	1412.14/13	1374.03/13	1446.14/13	1378.91/13
R106	1252.03/12	1303.61/12	1266.74/12	1309.48/12	1276.64/12
R107	1104.66/10	1148.15/10	1121.94/10	1152.37/10	1116.74/10
R108	963.99/9	1002.17/9	979.64/9	1011.17/9	984.28/9
R109	1194.73/11	1247.32/11	1217.09/11	1238.12/11	1230.43/11
R110	1124.4/10	1161.25/10	1155.43/10	1179.42/10	1136.12/10
R111	1096.72/10	1133.94/10	1120.57/10	1132.41/10	1117.84/10
R112	982.14/9	1025.88/9	1004.71/9	1021.31/9	1001.72/9

表 1 说明:BestSoFar 是当前已知最优解,MMAS-AVG、MMAS-BEST 分别是文献[7]中平均值解和最优解,HMMAS-AVG、HMMAS-BEST 分别是文中 10 次实验全局最优解的平均值和最优解。

在文中算法中,实验开始将路径上的信息素初始化为上限值  $\tau_{max}$ ,这样可使该系统有更好的全局搜索的能力。以实例 r103 为例,图 1 描述了迭代次数和全局最优解之间的关系,其中横轴表示迭代次数,纵轴表示全局最优解的值。图例 MMAS 和 HMMAS 分别表示文献[7]、文中算法迭代次数和全局最优解之间的关系。由图 1 可知,迭代次数位于 1~170 时,算法 HMMAS 所求的解的质量明显要比算法 MMAS 解的质量低,迭代次数位于 171~380 时,算法 HMMAS 与 MMAS 所求解的质量相差不大,而在 381 次迭代之后算法 HMMAS 所求解的质量明显好。原因是迭代开始时文中算法将信息素的值设定为上限值  $\tau_{max}$ ,这可使算法 HMMAS 初始有更好的全局搜索的能力,但是所求解的质量会相对差一些,随着迭代次数的增加通过使用 2-opt 局部搜索机制来提高解的质量,算法 HMMAS 显示了较好的性能。文中算法每次实验运行时间限制为 60 秒,而文献[7]中的图 1 表明 MMAS 在运行时间为 350 秒时,总的运行距离的收敛才趋于平稳,这说明算法 HMMAS 在求解最优解时明显缩短了每次迭代的计算时间。

图 2 描述了算法运行时间和某路径上信息素的值之间的关系,其中横轴表示算法运行时间,纵轴表示信息素的值。由图 2 可知,随着时间和迭代次数的变化信息素的值出现了震荡并且被限定在  $[\tau_{min}, \tau_{max}]$  之间,这有利于避免某条路径上信息素聚集太多,而防止搜索过早陷入局部最优。

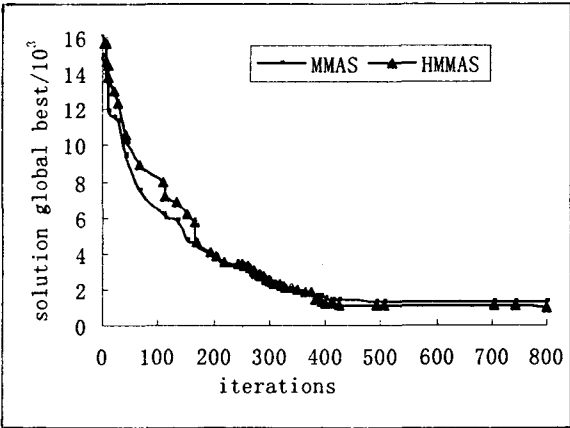


图 1 迭代次数和全局最优解之间的关系

4 结束语

针对有时间窗车辆路径问题中最小化车辆数和总

的旅行距离这两个目标,结合该问题本身的特点,受文献[8]的启发提出用信息素相互独立,但又通过交换信息来协作的混合蚁群算法。当 MMAS-VEI 算法找到可行解以后,通过使用 MMAS-TIME 算法并结合 2-opt 局部搜索来提高可行解的质量,在算法运行过程中信息素被限定在  $[\tau_{min}, \tau_{max}]$  之间,有利于防止某条路径信息素聚集过高,从而减少计算时间以避免过早收敛。通过使用含 2-opt 和不含 2-opt 局部搜索机制分别对 Solomon 的测试数据进行实验,并与文献[7]中实验做了对比,结果表明了文中算法的有效性。

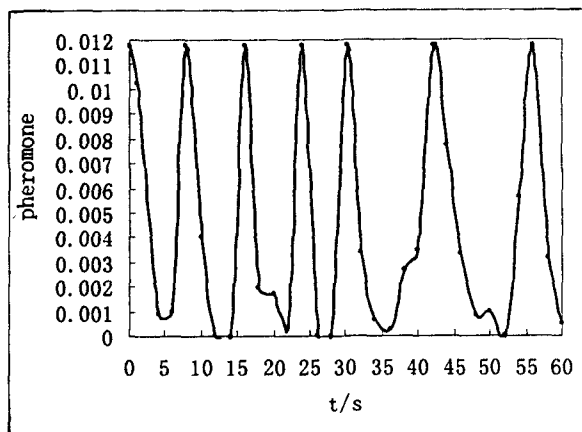


图 2 算法运行时间和信息素的值之间的关系

#### 参考文献:

- [1] Dantzig G, Ramser J. The truck dispatching problem[J].

(上接第 89 页)

数据集上明显优于 Adaboost 占 75%, 1 个数据集上持平占 5%。可以分析出 Ada-ens 在分类准确率、泛化性能、稳定性上都表现出较好的性能。

### 3 结束语

文中在选择性集成学习的集成上提出了多层次选择性集成算法 Ada-ens, 针对决策树与神经网络模型在 20 个标准数据集对集成学习算法 Ada-ens 进行了实验研究, 试验证明基于数据的集成学习算法的性能优于基于特征集的集成学习算法的性能, 有更好的分类准确率和泛化性能。

下一步工作中着重需要解决的问题: 与目前集成学习相似试验的参数均是认为设定的希望通过理论推导或试验分析得出最佳的参数设置和理由; 用数学理论佐证为什么选择性集成会更有效<sup>[8]</sup>, 在什么情况下更有效; 在真实的情况下测试 Ada-ens 的效果, 并依次对其进行改进和完善。

Management Science, 1959, 6(1): 80-91.

- [2] Dorigo M. Optimization, Learning and Natural Algorithms [D]. Milan, Italy: Dipartimento di Elettronica, Politecnico di Milano, 1992.
- [3] Stützle T, Hoos H H. MAX-MIN Ant System[J]. Future Generation Computer Systems, 2000, 16: 889-914.
- [4] Dorigo M, Stützle T. 蚁群优化[M]. 张军, 胡晓敏, 译. 北京: 清华大学出版社, 2007.
- [5] 赵传信, 张雪东, 季一木. 改进的粒子群算法在 VRP 中的应用[J]. 计算机技术与发展, 2008, 18(6): 240-242.
- [6] 刘志硕, 申金关. 车辆路径问题的混合蚁群算法设计与实现[J]. 管理科学学报, 2007, 10(3): 15-22.
- [7] 万旭, 林健良, 杨晓伟. 改进的最大-最小蚂蚁算法在有时间窗车辆路径问题中的应用[J]. 计算机集成制造系统, 2005, 11(4): 572-576.
- [8] Gambardella L M, Taillard E, Agazzi G. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows[M]. In: Corne D, Dorigo M, Glover F. New Ideas in Optimization. UK: McGraw-Hill, 1999: 63-76.
- [9] Flood M M. The Traveling Salesman Problem[J]. Operations Research, 1956(4): 61-75.
- [10] Solomon M. Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints[J]. Operations Research, 1987, 35: 254-365.
- [11] 马良, 朱刚, 宁爱兵. 蚁群优化算法[M]. 北京: 科学出版社, 2008.

#### 参考文献:

- [1] Hansen L K, Salamon P. Neural network ensembles[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12(10): 993-1001.
- [2] Zhou Z-H, Wu J, Tang W. Ensembling neural networks: many could be better than all[J]. Artificial Intelligence, 2002, 137(1-2): 239-263.
- [3] Dietterich T G. Ensemble methods in machine learning[C]// In: Proceedings of the First International Workshop on Multiple Classifier Systems. Cagliari, Italy: [s. n.], 2000: 1-15.
- [4] 唐耀华, 高静怀. 一种新的选择性支持向量机集成学习方法[J]. 西安交通大学学报, 2008(10): 1221-1225.
- [5] 蒋望东, 林士敏. 基于选择性集成遗传算法的 BNC 结构学习[J]. 计算机辅助工程, 2006(3): 46-50.
- [6] 王正群, 陈世福. 一种主动学习神经网络集成方法[J]. 计算机研究与发展, 2005(3): 375-380.
- [7] 王建敏, 李铁军. 基于神经网络集成学习的智能决策支持系统构建[J]. 电脑知识与技术, 2008(9): 2045-2050.
- [8] 李凯, 崔丽娟. 集成学习算法的差异性及性能比较[J]. 计算机工程, 2008(3): 35-37.