文章编号:1007－6735(2016)04－0367－06 DOI:10.13255/j.cnki.jusst.2016.04.011

求解无容量设施选址问题的混合蚁群算法

李 倩[1]， 张惠珍[1]， Cesar Beltran-Royo[2]
(1.上海理工大学 管理学院,上海 200093; 2.西班牙胡安卡洛斯大学 统计与运筹系,马德里)

摘要: 无容量设施选址(UFL)问题是经典的优化问题,属于NP难题,易于描述却难于求解.首先,介绍了UFL问题的数学模型,并对UFL问题的特点进行深入分析,得到其最优解所具有的基本特征;其次,针对UFL问题的最优解所具有的基本特征,设计了两种局部搜索策略,并将其与基本蚁群算法相结合,提出了一种用于求解UFL问题的混合蚁群搜索算法;最后,为了测试该算法的性能,分别利用混合蚁群算法和基本蚁群算法求解UFL问题基准问题库中的16个测试算例.计算结果表明,混合蚁群算法有效改进了基本蚁群算法求解UFL问题时易陷入局部最优、收敛速度慢等不足,该算法对求解UFL问题具有明显的可行性和有效性.

关键词: 无容量设施选址问题; 蚁群算法; 局部搜索
中图分类号: TP 183 **文献标志码**: A

Hybrid Ant Colony Algorithm for the Uncapacitated Facility Location Problem

LI Qian[1], ZHANG Huizhen[1], Cesar Beltran-Royo[2]
(1. *Business School, University of Shanghai for Science and Technology, Shanghai 200093, China*;
2. *Statistics and Operations Research, Rey Juan Carlos University, Madrid, Spain*)

Abstract: Uncapacitated facility location problem (UFL) is a classic NP hard problem, easy to describe but difficult to solve. Combined with two local search strategies, a hybrid ant colony algorithm was proposed for solving the UFL problem. By solving 16 typical instances of UFL problem, the basic ant colony algorithm and the hybrid ant colony algorithm were tested. The numerical results prove the feasibility and effectiveness of the hybrid algorithm for solving the UFL problem. The hybrid algorithm performs better in terms of local optimum and rate of convergence.

Keywords: *uncapacitated facility location problem*; *ant algorithm*; *local search*

无容量设施选址问题(uncapacitated facility location,UFL)是从没有限定容量大小的设施位置集合中选择要开放的设施,使其以最小的代价服务于给定的所有客户.无容量设施选址问题具有广泛

收稿日期: 2015－11－06
基金项目: 国家自然科学基金资助项目(71401106);高等学校博士学科点专项科研基金联合资助课题(20123120120005);上海市教育委员会科研创新项目(14YZ090);上海市高校青年教师培养资助计划(slg12010)
第一作者: 李 倩(1990－),女,硕士研究生.研究方向:智能优化. E-mail:liucan_qian@163.com
通信作者: 张惠珍(1979－),女,副教授.研究方向:运筹学、智能优化. E-mail:zhzzywz@163.com

的实际应用背景，生活中许多实际的问题均可被抽象为UFL问题进行求解，如银行选址、网络设计、聚类分析、证券投资管理等.

目前，用于求解UFL问题的算法大致可以分为3类：近似算法、精确算法和智能优化算法. 在多项式时间内能够求得问题的一个解，并且其目标函数值与最优解的目标函数值之比不超过一个常数的算法被称为近似算法[1]. 1997年，Williamson等[2]给出了求解设施选址问题的第一个常数近似度算法；2013年，Li[3]提出了求解UFL问题的1.488－近似算法. 求解UFL问题的精确算法主要有：割平面算法、列生成算法、Erlenkotters算法[4]、分支定界法[5－6]等. 这些算法能够求得问题的最优解，但适用于规模较小的问题，且计算速度慢. 虽然智能优化算法搜索效率高，适用于求解大规模的问题，但是这些算法容易陷入局部最优，一般情况下只能求得问题的满意解，不一定是最优解. 为了克服单一智能优化算法易陷入局部最优、收敛速度慢等不足之处，近年来求解UFL问题的混合智能优化算法得到了长足的发展，如：混合多层启发式算法[7]、并行多粒子群优化算法[8]、优化启发式算法[9]、启发式并行局部搜索算法[10]等.

蚁群算法(ant colony algorithm，ACA)具有鲁棒性强、可以进行分布式计算、易与其他算法有效结合等优点，但其容易陷入局部最优[11]. 针对UFL问题的具体特点，将两种局部搜索策略与基本蚁群算法相融合[12－13]，提出了一种求解UFL问题的混合蚁群算法，并通过求解系列经典UFL问题验证该算法的求解性能.

1 无容量设施选址问题

给定建造无容量限制的设施位置集 $M=\{1,\cdots,m\}$，客户集 $N=\{1,\cdots,n\}$，对于任意给定的 $i\in N$ 和 $j\in M$，c_{ij} 表示客户 i 与设施 j 之间的运输费用，$f_j>0$ 表示设施 j 的开放费用. 要求每个客户必须选择一个且只能选择一个设施来满足其需求，同时使总费用最少. UFL问题的数学模型可被描述为

$$\min\ z=\sum_{i=1}^{n}\sum_{j=1}^{m}c_{ij}x_{ij}+\sum_{j=1}^{m}f_jy_j \tag{1}$$

$$\text{s.t.}\quad \sum_{j=1}^{m}x_{ij}=1,\forall\, i\in N \tag{2}$$

$$x_{ij}\leqslant y_j,\forall\, i\in N,\forall\, j\in M \tag{3}$$

$$x_{ij}\in\{0,1\},\forall\, i\in N,\forall\, j\in M \tag{4}$$

$$y_j\in\{0,1\},\forall\, j\in M \tag{5}$$

式中：z 为目标函数值，即总费用；y_j 表示设施 j 是否开放，如果设施 j 选择开放，则 $y_j=1$，否则 $y_j=0$；x_{ij} 表示客户 i 是否选择设施 j 为其提供服务，如果客户 i 选择设施 j 为其提供服务，则 $x_{ij}=1$，否则 $x_{ij}=0$.

记 X^* 为UFL问题(1)～(5)的最优解集，分析UFL问题的具体特点，可以得到定理1.

定理1 设 $(x^*,y^*)\in X^*$ 为最优解集中的一个解，记 $J=\{j\mid y_j^*=1,j\in M\}$，对于 $\forall\, j\in J$，记 $I_j=\{i\mid x_{ij}^*=1,i\in N\}$，则存在：

a. $\sum\limits_{i\in I_j}c_{ij}+f_j=\min\limits_{j'\in M}\Big\{\sum\limits_{i\in I_j}c_{ij'}+f_{j'}\Big\}$

b. 给定 $\forall\, i\in I_j$，则 $c_{ij}=\min\limits_{j'\in J}\{c_{ij'}\}$.

证明 **a.** 假定存在 $j_0\in J$，$I_{j_0}=\{i\mid x_{ij}^*=1,i\in N\}$，使得下式成立：

$$\sum_{i\in I_{j_0}}c_{ij_0}+f_{j_0}>\min_{j'\in M}\Big\{\sum_{i\in I_{j_0}}c_{ij'}+f_{j'}\Big\}$$

则可定义给定的UFL问题的解 $(\tilde{x},\tilde{y})$ 为

$$\begin{cases}\tilde{x}_{ij}=x_{ij}^*, & i\in N,j\neq j_0\\ \tilde{x}_{ij}=0, & i\in N,j=j_0\end{cases}$$

$$\begin{cases}\tilde{y}_j=y_j^*, & j\neq j_0\\ \tilde{y}_j=0, & j=j_0\end{cases}$$

计算使下式成立的 j''：

$$\sum_{i\in I_{j_0}}c_{ij''}+f_{j''}=\min_{j'\in M}\Big\{\sum_{i\in I_{j_0}}c_{ij'}+f_{j'}\Big\}$$

并令 $\tilde{y}_j''=1$ 和 $\tilde{x}_{ij''}=1(i\in I_{j_0})$，可得

$$\sum_{i=1}^{n}c_{ij_0}x_{ij_0}^*+f_{j_0}y_{j_0}^*>\sum_{i=1}^{n}c_{ij''}\tilde{x}_{ij''}+f_{j''}\tilde{y}_{j''}\Rightarrow$$

$$\sum_{i=1}^{n}c_{ij_0}x_{ij_0}^*+f_{j_0}y_{j_0}^*+\sum_{i=1}^{n}c_{ij''}x_{ij''}^*+f_{j''}y_{j''}^*>$$

$$\sum_{i=1}^{n}c_{ij''}\tilde{x}_{ij''}+f_{j''}\tilde{y}_{j''}=\sum_{i=1}^{n}c_{ij_0}\tilde{x}_{ij_0}+$$

$$f_{j_0}\tilde{y}_{j_0}+\sum_{i=1}^{n}c_{ij''}\tilde{x}_{ij''}+f_{j''}\tilde{y}_{j''}\Rightarrow\sum_{i=1}^{n}\sum_{j=1}^{m}c_{ij}x_{ij}^*+$$

$$\sum_{j=1}^{m}f_jy_j^*>\sum_{i=1}^{n}\sum_{j=1}^{m}c_{ij}\tilde{x}_{ij}+\sum_{j=1}^{m}f_j\tilde{y}_j$$

这与 $(x^*,y^*)\in X^*$ 相矛盾.

b. 对于给定的 $j_0\in J$，假定存在 $i_0\in I_{j_0}$，使得 $c_{i_0j_0}>\min\limits_{j'\in J}\{c_{i_0j'}\}$. 计算使下式成立的 j''：

$$c_{i_0j''}=\min_{j'\in J}\{c_{ij'}\}$$

并定义给定的UFL问题的解 $(\tilde{x},\tilde{y})$ 为

$$\begin{cases}\widetilde{x}_{ij}=0, & i=i_0 \text{且} j=j_0\\ \widetilde{x}_{ij}=1, & i=i_0 \text{且} j=j'', \widetilde{y}_j=y_j^*(j\in M)\\ \widetilde{x}_{ij}=x_{ij}^*, & \text{其他}\end{cases}$$

显然，$c_{i_0j_0}x_{i_0j_0}>c_{i_0j''}x_{i_0j''}\Rightarrow\sum_{i=1}^{n}\sum_{j=1}^{m}c_{ij}x_{ij}^*+\sum_{j=1}^{m}f_jy_j^*>\sum_{i=1}^{n}\sum_{j=1}^{m}c_{ij}\widetilde{x}_{ij}+\sum_{j=1}^{m}f_j\widetilde{y}_j$，这与$(x^*,y^*)\in X^*$相矛盾.

(x^*,y^*)为所求 UFL 问题的任一最优解，I_j为该最优解中设施j所服务的客户集合，定理 1 说明了 UFL 问题的最优解所具有的两个特点：a. 设施j为集合I_j中所有客户的服务费用之和$\left(\sum_{i\in I_j}c_{ij}+f_j\right)$小于或等于其他任一设施$j'(j'\in M)$为集合$I_j$中的客户提供服务的费用之和$\left(\sum_{i\in I_j}c_{ij'}+f_{j'}\right)$；b. 任一客户始终选择与其运输费用最小的已开放设施.

根据定理 1，可将给定 UFL 问题的任一可行解(x,y)进行如下改进：a. 该解中，若已开放的设施j为集合I_j中所有客户的服务费用之和大于设施$j'(j'\in M)$为集合I_j中的客户提供服务的费用之和，则可关闭设施j，而开放设施j'，并令j'服务I_j中的所有客户；b. 设施j和j'均为开放设施，且设施j服务客户i，即$x_{ij}=1$，若$c_{ij}>c_{ij'}$，则可将客户i的服务设施j改变为设施j'，即令$x_{ij}=0, x_{ij'}=1$. 记将(x,y)经过这两种改进后的可行解为$(\widetilde{x},\widetilde{y})$，显然，$(\widetilde{x},\widetilde{y})$所对应的目标函数值小于$(x,y)$所对应的目标函数值.

2　混合蚁群算法的设计

2.1　基本蚁群算法

蚁群算法是一种源于大自然中生物世界的新的仿生类算法，属于随机型搜索算法. 其基本原理来自于昆虫学家们对生物界中蚂蚁搜索食物源的过程的观察：蚂蚁在搜索食物时，能在其经过的路径上释放一种蚂蚁特有的分泌物——信息素，使得其他蚂蚁可以感知并且影响其行为. 当选择某些路径的蚂蚁越来越多时，该路径上累积的信息素就越多，以致后来蚂蚁选择该路径的概率也越高，也就增加了该路径的吸引强度，依靠这种生物机制，蚂蚁最终可以找到一条到达食物源的最短路线. 本文将这种蚂蚁群体寻优思想作一些修改和引申，以便符合所要求解的 UFL 问题.

将目标函数值作为每只蚂蚁k的评价值. 对于客户 1，将$[1,m]$区间内产生的一个随机整数作为蚂蚁k为客户 1 所选择的设施. 对于其他客户$i$$(1<i\leqslant n)$，在$[0,1]$区间内产生随机数$q$，若$q\leqslant q_0(0<q_0<1)$，则蚂蚁$k$在客户$i$处所选择的设施$j$由下述式(6)决定；否则，利用式(7)计算蚂蚁$k$选择设施$j$的概率$P_{ij}^k$及其累积概率，并采用轮盘赌法确定所选设施.

$$j=\arg\max_{j\in M}[(\tau_{ij})^{\alpha}(1/\eta_{ij}^k)^{\beta}] \tag{6}$$

$$P_{ij}^k=\frac{(\tau_{ij})^{\alpha}(1/\eta_{ij}^k)^{\beta}}{\sum_{j\in M}(\tau_{ij})^{\alpha}(1/\eta_{ij}^k)^{\beta}} \tag{7}$$

式中：$M=\{1,\cdots,m\}$为设施位置集合；η_{ij}^k为启发函数，表示蚂蚁k在客户i处选择设施j的期望程度，通过蚂蚁k给前$(i-1)$个客户提供服务所开放的设施费用及配送费用之和来计算；τ_{ij}为客户i与设施j之间的信息素；α为信息素重要程度因子，其值越大，表示信息素的浓度在选择过程中所起的作用越大；β为启发函数重要程度因子，其值越大，表示启发函数在选择过程中所起的作用越大.

表 1 表示了蚂蚁k为客户i选择设施时，10 个设施处的概率P_{ij}^k和累积概率. 当$q>q_0$时，将随机数q作为选择指针来确定蚂蚁k在客户i处的被选设施. 如：$q_0=0.15$和$q=0.63$时，蚂蚁k在客户i处的被选设施为 5.

表 1　轮盘赌法

Tab. 1　Roulette method

设施	1	2	3	4	5	6	7	8	9
P_{ij}^k	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03
累计概率	0.18	0.34	0.49	0.62	0.73	0.82	0.89	0.95	0.98

随着时间的推移，蚂蚁在路径上留下的信息素逐渐衰减. 记信息素衰减系数为$\rho(0<\rho<1)$，蚂蚁完成一次循环后，利用式(8)更新信息素为

$$\tau_{ij}^{\text{new}}=\rho\tau_{ij}^{\text{old}}+\sum_{k=1}^{b}\Delta\tau_{ij}^k \tag{8}$$

式中，$\Delta\tau_{ij}^k$表示蚂蚁k在本次循环过程中留在客户i和设施j之间的信息素. 选择应用 Ant－Cycle 模型计算$\Delta\tau_{ij}^k$为

$$\Delta\tau_{ij}^k=\begin{cases}Q/z_k, & \text{若}(i,j)\text{在最优路径上，}\\ & z_k\text{为目标函数值}\\ 0, & \text{其他}\end{cases}$$

式中：Q为常数；z_k为蚂蚁k在本次循环中的目标函数值.

2.2　局部搜索

在定理 1 的基础上，设计了两种求解 UFL 问题

的局部搜索策略，并将其嵌入蚁群优化算法，以有效改进由蚁群算法所求得的UFL问题的可行解. 下面通过一个例子来对两种局部搜索策略进行简要介绍.

例 给定一个 $m=5$ 和 $n=5$ 的UFL问题，该问题中客户与设施之间的运输费用矩阵 $\boldsymbol{C}$ 和设施安装费用向量 $\boldsymbol{F}$ 分别为

$$\boldsymbol{C}=\begin{bmatrix}1\,696 & 1\,666 & 1\,354 & 1\,224 & 192\\ 1\,309 & 1\,980 & 1\,749 & 132 & 1\,439\\ 1\,488 & 1\,227 & 1\,240 & 121 & 1\,375\\ 1\,235 & 1\,783 & 1\,831 & 1\,340 & 112\\ 1\,621 & 1\,135 & 108 & 1\,004 & 1\,149\end{bmatrix},$$

$$\boldsymbol{F}=[130,199,139,127,103]$$

该问题的最优目标函数值为1 034. 假设蚂蚁算法中 l 只蚂蚁搜索一次所得的满意解为：$x_{14}=1$，$x_{23}=1$，$x_{32}=1$，$x_{42}=1$，$x_{55}=1$，$y_1=0$，$y_2=1$，$y_3=1$，$y_4=1$，$y_5=1$. 即：开放的设施集合为{2,3,4,5}；客户1选择设施4；客户2选择设施3；客户3和4选择设施2；客户5选择设施5；总费用为 $199+139+127+103+1\,224+1\,749+1\,227+1\,783+1\,149=7\,700$.

利用两种局部搜索策略对上述蚂蚁算法的求解结果改进如下：

策略1 以设施2所服务的客户3和4为例，客户集{3,4}分别被各个设施服务的费用为

设施1 $c_{31}+c_{41}+f_1=1\,488+1\,235+130=2\,853$；

设施2 $c_{32}+c_{42}+f_2=1\,227+1\,783+199=3\,209$；

设施3 $c_{33}+c_{43}+f_3=1\,240+1\,831+139=3\,210$；

设施4 $c_{34}+c_{44}+f_4=121+1\,340+127=1\,588$；

设施5 $c_{35}+c_{45}+f_5=1\,375+112+103=1\,590$.

由上述计算结果及定理1可知，若客户3和4选择为其服务费用最小的设施4，则可使蚂蚁算法的求解结果得以改进，则令 $x_{32}=0$，$x_{42}=0$，$x_{34}=1$，$x_{44}=1$，$y_2=0$，$y_4=1$. 改进后的结果为 $x_{14}=1$，$x_{23}=1$，$x_{34}=1$，$x_{44}=1$，$x_{55}=1$，$y_1=0$，$y_2=0$，$y_3=1$，$y_4=1$，$y_5=1$，目标函数值计算结果为 $v=5\,952$.

策略2 蚂蚁算法的求解结果经策略1改进后，开放的设施集合为{3,4,5}. 由定理1知：给定开放的设施集合，每个客户应选择运输费用最小的开放设施为其服务.

客户1 由 $\min\{c_{13},c_{14},c_{15}\}=\min\{1\,354,1\,224,192\}=192$ 得设施5应为客户1提供服务，则令 $x_{14}=0$，$x_{15}=1$；

客户2 由 $\min(c_{23},c_{24},c_{25})=\min(1\,749,132,1\,439)=132$ 得设施4应为客户2提供服务，则令 $x_{23}=0$，$x_{24}=1$；

客户3 由 $\min(c_{33},c_{34},c_{35})=\min(1\,240,121,1\,375)=121$ 得客户3的服务设施保持不变，仍为设施4，即 $x_{34}=1$；

客户4 由 $\min(c_{43},c_{44},c_{45})=\min(1\,831,1\,340,112)=112$ 得设施5应为客户4提供服务，则令 $x_{44}=0$，$x_{45}=1$；

客户5 由 $\min(c_{53},c_{54},c_{55})=\min(108,1\,004,1\,149)=108$ 得设施3应为客户5提供服务，则令 $x_{55}=0$，$x_{53}=1$.

经策略2改进后的结果为 $x_{15}=1$，$x_{24}=1$，$x_{34}=1$，$x_{45}=1$，$x_{53}=1$，$y_1=0$，$y_2=0$，$y_3=1$，$y_4=1$，$y_5=1$，$v=1\,034$. 显然，蚂蚁算法的计算结果经策略1和策略2得到了明显改进.

2.3 混合蚁群算法

将基本蚁群算法和两种局部搜索策略相结合，提出了求解UFL问题的混合蚁群算法. 算法具体步骤为：

Step 1 初始化蚂蚁个数 b、反映信息素重要程度的因子 α、反映启发函数重要程度的因子 β、信息素衰减因子 ρ、常数 q_0 和 Q、最大迭代次数 N 和信息素矩阵 $\boldsymbol{T}=(\tau_{ij})_{N\times M}$，令迭代次数 $nc=1$，蚂蚁编号 $k=1$ 和满意解的目标函数值 $v=+\infty$；

Step 2 将[1,m]区间内产生的一个随机整数作为蚂蚁 k 为客户1所选择的设施，根据式(6)和式(7)为蚂蚁 k 的其他客户选择一个设施；

Step 3 若 $k<b$，则令 $k=k+1$，并转Step 2；否则，转Step 4；

Step 4 运用两种局部搜索方法对蚂蚁的本次搜索结果进行改进；

Step 5 更新当前满意解及其所对应的目标函数值 v；

Step 6 根据式(8)更新信息素矩阵；

Step 7 令 $\Delta\tau_{ij}=0(i\in N,j\in M)$ 和 $nc=nc+1$；

Step 8 若 $nc\leqslant N$，则转Step 2；否则，结束搜索，并输出结果.

3　算例分析

为验证混合蚁群算法的可行性及其求解性能，采用两组 UFL 基准问题库中的 16 个算例（每个算例的设施数等于其客户数，即 $m=n$）进行求解测试，并对基本蚁群算法和混合蚁群算法的计算结果进行对比分析.

实验环境：Intel(R) Core(TM) i7－3770 CPU @ 3.40 GHz，3.41GB 内存，操作系统为 Windows 7，数据处理由 Matlab 2010 完成.

表 2 给出了基本蚁群算法和混合蚁群算法中的相关参数. 表 3 给出了所求算例的规模、文献[14]所给的（已知）最优目标函数值、基本蚁群算法和混合蚁群算法的计算结果. 计算结果包括：迭代 200 次的运行总时间、找到所输出的满意解时的迭代时间（迭代时间）、满意解的目标函数值（目标函数值）、满意解的目标函数值与文献[14]所给的（已知）最优目标函数值之间的差距（Gap）. 其中，所有时间均以 s 为单位，Gap 的计算公式如下：

$$Gap=\frac{(\text{目标函数值}-(\text{已知})\text{最优目标函数值})}{((\text{已知})\text{最优目标函数值})}$$

表 2　相关参数

Tab. 2　Related parameters

参数含义	混合蚁群算法		基本蚁群算法	
	参数	值	参数	值
蚂蚁个数	b	20	b	20
信息素重要程度因子	a	1	a	1
启发函数重要程度因子	β	5	β	5
衰减系数	ρ	0.1	ρ	0.1
常数 1	q_0	0.15	q_0	0.15
最大迭代次数	N	200	N	200
常数 2	Q	100	Q	100
局部搜索选择蚂蚁个数	A	30%	—	—
局部搜索选择客户个数	B	50%	—	—

表 3　计算结果

Tab. 3　Computational results

算例	规模	（已知）最优目标函数值	基本蚁群算法				混合蚁群算法			
			运行总时间/s	迭代时间/s	目标函数值	Gap/%	运行总时间/s	迭代时间/s	目标函数值	Gap/%
gs00250al	250	257 964	141.91	54.46	383 702	48.74	142.80	129.85	258 875	0.35
gs00250bl	250	276 761	142.24	44.21	572 121	106.72	144.48	141.59	351 571	27.03
gs00500al	500	511 229	537.60	90.03	774 936	51.58	551.53	491.96	518 357	1.39
gs00500bl	500	537 931	538.21	113.87	1 167 231	116.99	542.51	501.93	709 514	31.90
gs00750al	750	763 671	1 273.92	269.43	1 173 403	53.65	1 295.36	1 108.78	782 013	2.40
gs00750bl	750	797 026	1 276.63	426.17	1 775 518	122.77	1 334.50	1 274.51	1 077 446	35.18
ga00250al	250	257 957	153.47	8.55	380 598	47.54	168.76	159.20	257 989	0.01
ga00250bl	250	276 339	159.87	92.84	578 149	109.22	167.62	134.82	350 951	27.00
ga00500al	500	511 422	551.22	382.75	768 197	50.21	571.65	362.16	520 780	1.83
ga00500bl	500	538 060	548.96	102.07	1 178 861	119.09	578.93	526.32	731 632	35.98
ga00750al	750	763 576	1 243.59	698.79	1 162 992	52.31	1 278.58	1 077.40	780 901	2.27
ga00750bl	750	796 480	1 245.33	386.17	1 780 709	123.57	1 281.73	1 281.73	1 076 357	35.14
500－1 000	500	99 169	527.73	116.39	2 461 168	2 381.79	566.88	343.20	105 759	6.65
1 000－1 000	1 000	220 560	2 273.38	432.93	5 179 321	2 248.26	2 327.98	990.49	275 911	25.10
1 500－1 000	1 500	334 962	5 720.58	1 746.70	7 841 836	2 241.11	5 837.52	5 581.62	359 925	7.45
2 000－1 000	2 000	437 686	11 497.83	179.54	10 667 234	2 337.19	11 567.02	9 037.72	449 198	2.63
平均值	687.50	461 299.56	1 739.53	321.56	2 365 373.50	638.17	1 772.37	1 446.46	537 948.88	15.14

由表 3 可知：混合蚁群算法和基本蚁群算法求解 16 个算例的平均运行总时间分别为 1 740 s 和 1 773 s. 虽然混合蚁群算法中加入两种局部搜索策略，致使其运行总时间略高于基本蚁群算法的运行总时间，但是混合蚁群算法的求解结果明显优于基本蚁群算法的求解结果，混合蚁群算法求解 16 个算例的平均目标函数值为 537 948，其明显小于由基本蚁群算法计算得到的平均目标函数值 2 365 374. 对于一半以上的算例而言，利用混合蚁群算法求解的 Gap 值均小于 8%，表明利用混合蚁群算法求解的满意解非常接近于文献[14]给出的已知最优解. 然而，这些算例利用基本蚁群算法计算的最小 Gap 值

为47.54%，最大的 Gap 值竟达2 381.79%.

此外，表3的计算结果表明，相对于基本蚁群算法而言，混合蚁群算法的收敛性能明显得到改善. 如：利用基本蚁群算法和混合蚁群算法求解算例ga00250a1的运行总时间分别为153.47 s和168.76 s，并且混合蚁群算法在迭代时间为159.20 s时求得所输出的满意解，但基本蚁群算法在迭代时间为8.55 s时就已求得所输出的满意解，剩余144.92 s的搜索迭代对所求得的满意解没有任何改进.

4 结论

首先，深入分析了UFL问题的最优解所具有的特点，并在此基础上提出了两种适用于UFL问题求解的局部搜索策略，然后将这两种局部搜索策略和基本蚁群算法相结合，设计了求解UFL问题的混合蚁群算法. 求解UFL基本问题库中的16个测试算例的结果表明，这两种局部搜索策略明显改进了基本蚁群算法的计算性能和求解结果.

结合模型特点，给出的这两种求解UFL问题的局部搜索策略，不仅可以与基本蚁群算法相结合，而且可与其他智能优化算法相结合构成求解UFL问题的混合智能优化算法，这对改进求解UFL问题的智能优化算法的求解性能是非常有益的.

参考文献：

[1] 堵丁柱，葛可一，胡晓东. 近似算法的设计与分析[M]. 北京：高等教育出版社，2011.

[2] WILLIAMSON D P，HALL L A，HOOGEVEEN J A，et al. Short shop schedules[J]. Operations Research，1997，45(2)：288－294.

[3] LI S. A 1.488 approximation algorithm for the uncapacitated facility location problem[J]. Information and Computation，2013，222：45－58.

[4] JANÁČEK J，BUZNA L. An acceleration of Erlenkotter-Körkel's algorithms for the uncapacitated facility location problem[J]. Annals of Operations Research，2008，164(1)：97－109.

[5] CAPRARA A，GONZÁLEZ S. A branch-and-cut algorithm for a generalization of the uncapacitated facility location problem[J]. Top，1996，4(1)：135－163.

[6] 李翼，赵茂先，李岳佳. 无容量限制设施选址问题的分支定界法[J]. 山东理工大学学报(自然科学版)，2012，26(1)：70－73.

[7] RESENDE M G C，WERNECK R F. A hybrid multistart heuristic for the uncapacitated facility location problem[J]. European Journal of Operational Research，2006，174(1)：54－68.

[8] 王大志，闫杨，汪定伟，等. 基于OpenMP求解无容量设施选址问题的并行PSO算法[J]. 东北大学学报(自然科学版)，2008，29(12)：1681－1684.

[9] KLINCEWICZ J G，LUSS H，ROSENBERG E. Optimal and heuristic algorithms for multiproduct uncapacitated facility location[J]. European Journal of Operational Research，1986，26(2)：251－258.

[10] CURA T. A parallel local search approach to solving the uncapacitated warehouse location problem[J]. Computers & Industrial Engineering，2010，59(4)：1000－1009.

[11] 王欣盛，马良. 工件排序的改进蚁群算法优化[J]. 上海理工大学学报，2011，33(4)：362－366.

[12] 李煜，马良. 用量子蚁群算法求解大规模旅行商问题[J]. 上海理工大学学报，2012，34(4)：355－358.

[13] 李俊青，潘全科，王法涛. 求解混合流水线调度问题的离散人工蜂群算法[J]. 运筹与管理，2015，24(1)：157－163.

[14] BELTRAN-ROYO C，VIAL J P，ALONSO-AYUSO A. Semi-Lagrangian relaxation applied to the uncapacitated facility location problem[J]. Computational Optimization and Applications，2012，51(1)：387－409.

（编辑：丁红艺）