求解TSP问题的一种改进的十进制MIMIC算法

郝承伟[1] 高慧敏[2]

(太原科技大学计算机学院 太原 030024)[1] (嘉兴学院机电学院 嘉兴 314001)[2]

摘 要 十进制MIMIC算法是基于MIMIC二进制编码算法思想的可用来求解TSP的离散分布估计算法。着重考虑该算法在较大规模TSP问题上的算法缺陷,对其编码方式和概率模型进行了改进,提出了新的个体生成策略,在初始化种群阶段使用了贪心算法,在进化过程中引入了杂交算子、变异算子、映射算子、优化算子等演化算子,采用了动态调整方法来确定优势群体的规模。以上改进使得算法在小种群解大规模TSP问题的情况下仍可保持种群的多样性。实验结果表明,改进算法在求解规模、求解质量和寻优速度上都有明显提高。

关键词 MIMIC算法,旅行商问题,分布估计算法,概率矩阵

中图法分类号 TP18 **文献标识码** A

Modified Decimal MIMIC Algorithm for TSP

HAO Cheng-wei[1] GAO Hui-min[2]

(School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China)[1]

(Mechanical and Electrical Engineering College, Jiaxing University, Jiaxing 314001, China)[2]

Abstract Modified decimal MIMIC algorithm is a kind of discrete estimation of distribution algorithm, which is based on binary MIMIC algorithm and convenient to solve traveling salesman problem. Considering the drawbacks of the MIMIC algorithm while solving larger-scale TSP, this paper improved the encoding mode and probability model, proposed new individual strategy, introduced greedy algorithm at the initial phase of the probability matrix, and adopted crossover operator, mutation operator, etc. during the process of evolution, employed dynamic adjusted method to determine the population size. These modifications gurantee the population diversity even in small population and for larger-scale TSP. Experiment results show that problem scale, solution quality and speed of the optimization are improved significantly.

Keywords MIMIC algorithm, TSP, Estimation of distribution algorithm, Probability matrix

1 引言

TSP(Traveling Salesman Problem)已被证明具有NPC(Non-deterministic Polynomial Complete)计算复杂度,而任何NP问题都可归约为NPC问题,因此TSP问题的求解为众多领域所关注。该问题描述如下:设有M个节点的集合$P=\{1,2,3,\cdots,M\}$,两节点i,j之间的距离记为$d(i,j)$,求P集合全排列中的一个$(p_1,p_2,p_3,\cdots,p_M)$,使得该排列节点构成的环形距离D最小。

$$D=d(p_M,p_1)+\sum_{i=1}^{M-1}d(p_i,p_{i+1}) \quad (1)$$

虽然TSP问题描述简单,但该问题中所有可能的环路数量是随着节点数量M呈指数级增长的,当问题规模M增大到一定程度时,计算量将极大超出计算机所允许的极限,即所谓的组合爆炸。现唯一求解NPC问题理论上可确保求得最优解的算法是穷举法,但以$M=50$的问题规模为例,使用每秒计算1亿次的计算机按穷举法求解,需要计算5 * 1048年。因此,寻找该类问题的近似求解方法具有重要的意义。目前近似求解TSP问题常用的算法有蚁群算法[1]、人工神经网络[2]、模拟退火算法[3]、人工免疫算法[4]、遗传算法[5]、粒子群算法[6]等。分布估计算法[7,8](Estimation of Distribution Algorithms, EDA)是基于统计学习的随机优化搜索算法,它作为一种新的进化模型,为求解TSP问题开辟了新的思路。

十进制MIMIC(Mutual Information Maximization for Input Clustering)算法[9]的思想源自MIMIC[7,10]。它是一种双变量相关的离散分布估计算法。不同于遗传算法,在种群数量足够大时,它有着不易陷入局部最优的优点,但其运行效率较低,且不能有效求解较大规模TSP问题。本文在文献[9]算法的基础上,提出一种更高效的搜索算法,它改进了原有概率模型、群体生成策略,引入贪心算法、杂交算子、变异算子。实验证明,新的算法寻优速度显著增快,对TSPLIB中提供的城市规模在100以内的实例进行了实验,其均能得到最优解或近似解路径。

到稿日期:2011-09-13 返修日期:2011-11-23 本文受山西省自然科学基金(2009011011-3),山西省回国留学人员科研项目(2011-078)资助。

郝承伟(1984—),男,硕士生,主要研究方向为智能计算;高慧敏(1970—),男,博士,教授,主要研究方向为复杂系统的建模、仿真及优化调度,E-mail:humorgao@gmail.com(通信作者)。

2 分布估计算法求解 TSP

2.1 分布估计算法

分布估计算法基本原理：分布估计算法是一种基于概率模型的进化算法，使用概率模型表述变量间的相互关系，通过概率模型的学习更新和采样操作使得群体逐代向着优势个体进化。从进化方式上看，遗传算法倾向于个体间微观的基因变化，而分布估计算法是对生物群体基因整体分布规律的建模与模拟。因此相比于传统遗传算法，其是解决非线性、变量耦合的高维问题的更好选择，并且在适当群体规模保证的情况下，不易陷入局部最优。根据优化问题的复杂性，按照概率模型中变量间不同的关系，分布估计算法分为变量无关、双变量相关和多变量相关3种。

2.2 求解 TSP 的十进制 MIMIC 算法

十进制 MIMIC 算法是一种双变量相关分布估计算法。它借鉴了 MIMIC 算法的思想，使用十进制编码建立概率模型，用以表达 TSP 问题中两两节点之间的关系。概率模型的核心为路径概率分布矩阵(以下简称概率矩阵)。该算法使用式(1)作为适应度计算函数，进化过程中不断学习更新不同节点间的两两关系，从而最终逼近或达到全局最优解。

十进制 MIMIC 算法的主要思想：

编码方式：对个体编码采用自然编码的方式，每个个体为十进制节点序号排列构成的一条路径。例如，有7个节点 $P=\{1,2,3,4,5,6,7\}$，则(1 3 6 5 7 4 2)表示 TSP 环路 1→3→6→5→7→4→2→1。

概率矩阵：进化过程中每代选取优势个体数 S，统计所有优势个体路径中两两节点之间邻接的次数，得到分布矩阵 A，故矩阵每行、每列之和均为优势个体总数 S。矩阵行号 i 和列号 j 都表示节点序号，矩阵中元素 A_{ij} 表示在优势个体中节点 j 在节点 i 之后的次数，因此概率分布矩阵 $C=A/S$ 的第 i 行元素代表 i 节点后续节点的概率分布。

首节点概率向量：统计优势个体中首节点的概率分布，构造首节点概率向量 $FC=(c_1,c_2,\cdots,c_M)$。在生成每个新个体时，首先根据首节点概率向量 FC 采样第一个节点序号，然后根据概率矩阵 A/S 采样后续节点序号，直到完成整个路径。

修正参数：概率矩阵 C 中主对角线全部为零，但在优势群体实际取样时，某些非对角线上元素也会被置零，为避免最优邻接关系在进化过程中丢失，对此种情况进行修正。修正参数为 α，统计概率矩阵中非对角线零元素个数为 t，并将其置为 α/t，其余非零元素置为 $(1-\alpha)*C_{ij}$。

2.3 改进的十进制 MIMIC 算法

本文在十进制 MIMIC 算法基础上，对算法做了部分改进，并引入遗传算子，提高了算法的收敛速度。

2.3.1 改进的十进制 MIMIC 算法基本思想

(1)个体编码方式

原算法中(1365742)与(5742136)虽然起点不一样，但是所对应的 TSP 环路是相同的。为了避免这种环路相同而导致的编码重复[11]，以及减少计算量，设定编码起点都是1。因此新算法中舍弃了首节点概率向量。

(2)概率模型

根据 TSPLAB 中的定义，TSP 表示距离对称问题，ATSP (Asymmetric Traveling Salesman Problem)表示距离非对称问题，即存在两节点 i,j 使得 $d(i,j)\neq d(j,i)$。原算法概率模型严格定义了两节点的顺序关系，即概率矩阵中，C_{ij} 与 C_{ji} 含义不同，统计得到的概率也不同。经过实验证明，这种模型更适合 ATSP 问题。因此，选用忽略两节点顺序关系的对称概率矩阵作为新算法的概率模型。

(3)新个体生成策略

原算法在使用概率矩阵采样新个体下一节点时，采用“采样-舍弃”的方法，即已采样的节点仍然在可被采样的范围内，如果当前采样节点已在路径内，则舍弃该节点重新采样。这种方法非常耗费计算机资源，所以在新算法中引入辅助矩阵 C'，用以存储概率矩阵 C 的备份。在采样时，将已选用的节点序号对应的列置零，这样每次采样得到的序号都是未取的，不必再舍弃，待每条新的路径产生完毕，再使用备份将概率矩阵恢复。例如对于4节点 TSP 问题，设优势群体规模为10，一条新路径已取节点{1,3}，则下一个节点选取按照矩阵 C 中 C_{32} 与 C_{34} 的比例1:1取2节点或4节点，其对应概率矩阵 C 及辅助矩阵 C' 如下：

$$C=\begin{pmatrix}0 & 4/10 & 0 & 4/10\\ 0 & 0 & 0 & 2/10\\ 0 & 4/10 & 0 & 4/10\\ 0 & 2/10 & 0 & 0\end{pmatrix}$$

$$C'=\begin{pmatrix}0 & 4/10 & 2/10 & 4/10\\ 4/10 & 0 & 4/10 & 2/10\\ 2/10 & 4/10 & 0 & 4/10\\ 4/10 & 2/10 & 4/10 & 0\end{pmatrix}$$

(4)使用贪心算法初始化种群

文献[12]中的贪心算法每次采样均与当前节点距离最短，且尚未被采样的节点作为下一个节点。以 M 规模 TSP 问题为例，若两两节点间距离均不等，使用这种方法采样得到的不同路径只有 M 条，使得种群缺乏多样性，造成初始概率矩阵失衡。而要保证算法的全局收敛性，必须维持群体的个体多样性，避免有效节点联系的丢失。鉴于此，本文在如上贪心算法的基础上进行改进，即增加了对不同距离节点选取的概率，采样下一个节点不再固定选取剩余节点中距离最近的，而是按距离从近到远，节点被采样的几率逐渐递减。设剩余节点数为 r，r 个节点按照距当前节点的距离升序排列，i 表示排列位置从1到 r，则 i 位置节点被采样比例为 $(r/i)\wedge 2$。这样，使用改进后的贪心算法得到的群体较优且不失多样性。

(5)引入新的演化算子

随着 TSP 问题规模的增大，原十进制 MIMIC 算法在求解时，所需群体总量也急剧增大，计算会变得异常缓慢。为克服这一缺点，在以概率矩阵为模型生成群体的基础上，引入了杂交算子[5,14]、变异算子[5,14]来增加种群多样性，且引入映射算子[13]、优化算子[13]来加快算法寻优速度。这里的杂交算子与变异算子涉及到的逆转操作均为文献[14]中的环形逆转，逆转后进行调整，保证首节点仍为1。映射算子、优化算子可使好的基因片段能让更多的染色体所享有，以进一步增加优势基因的竞争能力。每一代群体保留微小比例个体由演化算子生成，从而使得算法在较小规模群体情况下也能保持种群多样性，提升算法寻优速度。

杂交算子：在概率矩阵生成的群体中随机选取两条路径 P,P'，在 P 中随机选取节点 X，在 P' 中指定 X 的下一个节点

为X'。对P中X的下一个节点与X'间的节点实施逆转操作，若P中X'是与X相邻的节点，则不进行逆转。例如，随机选取两条路径P(1 3 6 5 7 4 2)，P'(1 4 6 2 7 3 5)，随机选取节点$X=7$，则$X'=3$，而后对P中片段(4 2 1 3)实施逆转操作，调整首节点为1后得到新个体(1 2 4 6 5 7 3)。

变异算子：在概率矩阵生成的群体中随机选取一条路径P，在P中随机地选取两节点X，X'，对X的下一个节点与X'间的所有节点(包括X')进行逆转操作。例如，随机选取一条路径P(1 3 6 5 7 4 2)，若随机选取两节点$X=4$，$X'=6$，则逆转片段(2 1 3 6)，调整首节点为1后得到新个体(1 2 5 7 4 6 3)；若随机选取两节点$X=6$，$X'=4$，则逆转片段(5 7 4)，得到新个体(1 3 6 4 7 5 2)。.

映射算子：同文献[13]中的映射算子。在概率矩阵生成的群体中随机选取两条路径P和P'，比较两条路径适应度，不妨设P'较优，从P中随机选择一段基因片断ΔP，判断P'中是否存在这样一段基因$\Delta P'$，它与ΔP长度相同且第一位节点编号相同。如果存在$\Delta P'$，则将染色体P中的基因片断ΔP置换为$\Delta P'$，剩余基因按部分映射进行调整。

优化算子：同文献[13]中的优化算子。在概率矩阵生成的群体中随机选择两条路径P，P'，比较两条路径适应度，不妨设P'较优，将P作为父体，随机选择P中一段路径ΔP，然后判断P'中是否存在这样一段路径$\Delta P'$，它与ΔP中的基因长度相同且有相同的节点编号，只是基因节点的排列顺序不同；判断ΔP和$\Delta P'$的基因片路径长度，如果$\Delta P'$的路径短，则用$\Delta P'$置换掉P的ΔP片段。

(6)动态调整优势群体选取规模

进化过程实际是不断优选并缩小搜索空间的过程。进化初期必须保证足够大的选取规模，以免过早进入局部搜索空间，随着进化的进行，搜索空间逐渐分裂并减小。基于此推想，本文算法尝试动态调整选取优势个体规模，优势个体选取规模$S=fix(N*(SP-G*MP/GL))$(SP为初始优势群体选择比率，MP是优势群体递减比率，G为当前代数，GL为进化代数上限，N为种群数量)。经实验证明，在SP可保证寻优效果的情况下，适当增大MP可有效加快演化速度。当优势群体进化至最优和最差个体的适应值相等时，停止进化并输出结果。

(7)修正参数

使用原文的方法修正，会减慢概率矩阵的进化速度，从而减慢寻优速度，所以本文仅将概率矩阵非对角线零元素置为$1/S$，其他非零元素不做修正。

2.3.2 *改进的十进制MIMIC算法流程*

Step1 使用改进的贪心算法初始化种群，生成N条路径，统计样本的中邻接点的关联生成初始概率矩阵。初始化各参数：N表示群体规模，优势个体选择比例SP，其他遗传算子计算比例ND，优势群体规模$S=N*SP$。

Step2 通过概率矩阵生成L个个体(第一代$L=N$，否则$L=N(1-2*ND)-S$)，每个个体是一条遍历所有节点的路径，每生成一个个体后使用$\boldsymbol{C}'$将$\boldsymbol{C}$恢复。计算每个个体的适应度$Fitness$。

Step3 变异算子：从种群L中随机选取$P*ND$个样本，变异生成同数量子代，计算子代适应度$Fitness$，并将其加入到种群。

Step4 杂交算子：从种群L中随机选取两条路径作为父体杂交生成子个体，重复此步，直到产生$P*ND$个子个体，计算子代适应度$Fitness$，并将其加入到种群。

Step5 映射算子：从种群L中随机选取两条路径，根据如上映射条件更新父路径，重复此步$P*ND$次。

Step6 优化算子：从种群L中随机选取两条路径，随机选取基因片段，如果符合优化条件，则更新父路径，重复此步$P*ND$次。

Step7 将Step2－Step6中生成的种群N进行适应度排序，选取适应度最小且各异的S条路径作为当代优势群体。

Step8 使用优势群体S生成新一代概率矩阵$\boldsymbol{C}$，备份概率矩阵$\boldsymbol{C}'$并在下一代种群N中保留S。

Step9 更新下一代优势群体规模$S=fix(N*(SP-G*MP/GO))$。

Step10 重复执行Step2－Step10，直到满足停止准则(优势群体S中最优个体与最差个体适应度相等，或进化代数达到上限)，并输出找到的最优路径。

本文程序中将贪心算法、杂交算子、变异算子、映射算子、优化算子、适应度计算各自封装成独立的函数供主程序调用。

3 实例计算

文献[9]的MIMIC十进制算法使用VC++6.0实现。为了对比改进算法与原算法的性能，在MATLAB平台上实现了这两种算法。本文的所有实验数据均在Intel Pentium Dual CPU T2370 1.73Ghz的PC机上运行得到，操作系统为Windows 7，以下是部分实验数据。

为了更好地说明改进算法的有效性，我们选用了国际上最通用的TSP测试库中的多个实例进行测试。测试10次，结果见表1(TSPLIB提供的最短路径是将两两节点之间距离去小数后计算路径总长，而TSPLIB最优路径计算结果指精确计算路径长度再后取有效数字)。部分实例实验数据收敛，见图1－图3。

表1 改进算法对TSP问题测试结果统计

测试实例	TSPLIB提供的最短路径	TSPLIB最优路径计算结果	实验所得最佳结果	实验结果平均值
burma14	3323	3330.607	3330.607	3334.496
ulysses16	6859	6866.956	6866.956	6868.754
ulysses22	7013	7023.935	7023.935	7026.654
gr24	1272	1272	1272	1273.9
gr48	5046	5046	5046	5065.4
eil51	426	429.9833	429.5303	431.5473
Eil76	538	545.3876	545.3876	549.8657
Gr96	55209	55256.04	55256.04	55308.23
krod100	21294	21294.29	21294.29	21357.07
Eil101	629	642.3095	642.3095	654.9325
Gr120	6942	6942	7042	7131.5
Ch130	6110	6110.861	6314.631	6367.255

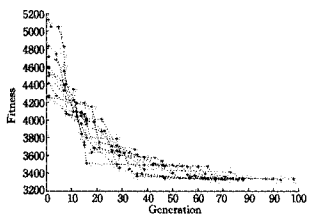

图1 实例burma14实验数据收敛图

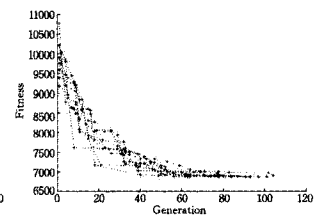

图2 实例ulysses16实验数据收敛图

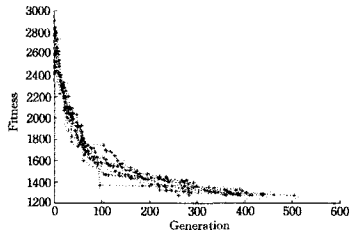

图 3　实例 gr24 实验数据收敛图

对于实例 eil51，TSPLIB 中提供的最短路径为 426，而 TSPLIB 中最优路径(见图 4)长度为 429.9833，改进算法得到的最短路径(见图 5)长度为 429.5303，优于 TSPLIB 中提供的最优路径。

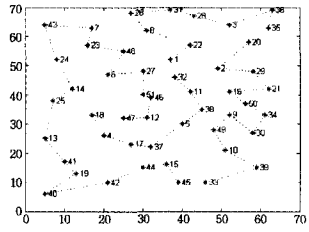

图 4　TSPLIB 提供的 eil51 最优路径

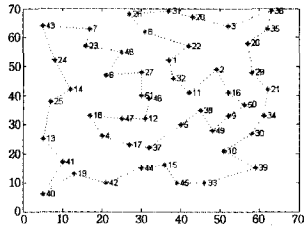

图 5　本文算法得到 eil51 的最优路径

为比较改进算法与原十进制 MIMIC 算法求解 TSP 的性能，在两种算法都可有效求解的 TSP 规模前提下，对两者进行了均值、标准差以及算法平均每次耗时的比较实验(20 次实验)，结果见表 2。实验中原算法使用的种群规模、进化代数、修正参数均为文献[9]的数据。此外，除 30 城市之外的其他实例在两种算法适应度计算之前，均对两两节点距离做取整处理。改进算法中 SP 取 0.7～0.8，MP 取 0.2～0.3，ND 取 0.02～0.04(参数选取据仿真实验经验得出)。对于不同实例，参数需做些许调整。

表 2　改进算法与文献[9]算法效率对比

测试实例		种群规模	平均进化代数	最优结果	平均结果	结果标准差	平均耗费时间(秒)
10 城市	原算法	500	100	323	326.3	4.9447	1.6022
	改进算法	150	44.75	323	323.5	2.2361	0.10481
burma 14	原算法	500	100	3323	3708.25	99.726	2.3124
	改进算法	150	68.4	3323	3334.25	6.27	0.26764
Ulysses 16	原算法	500	100	6859	7021.4	389.034	2.7795
	改进算法	200	79.35	6859	6871.6	11.146	0.31054
Gr17	原算法	500	200	2085	2456.5	85.145	11.431
	改进算法	300	159.05	2085	2087.75	2.5521	1.2746
Gr21	原算法	1000	200	2707	2945.5	343.68	25.442
	改进算法	300	297.25	2707	2731.25	43.906	3.1812
Ulysses 22	原算法	1000	200	7013	8143.5	407.87	23.279
	改进算法	300	279.3	7013	7030.5	31.905	3.5954
gr24	原算法	2000	500	1272	1284	57.985	87.308
	改进算法	300	409.5	1272	1273.9	3.7543	5.1205
30 城市	原算法	10000	500	424.8693	434.4765	76.408	554.76
	改进算法	500	1120.2	424.8693	426.0967	2.0844	25.123
gr48	原算法	10000	3000	5046	5234.8	376.43	1354.5
	改进算法	1000	2449.6	5046	5065.4	14.432	113.42
eil51	原算法	20000	3000	428	475.90	443.56	2043.5
	改进算法	1000	2387.5	426	429.5	17.669	154.49

结束语　与文献[9]算法相比，改进算法有以下特点：

(1)本文在概率矩阵初始化阶段运用了贪心算法，使得算法初期的搜索效率得以提高。

(2)以概率矩阵为主，其它优化算子为辅，同时增加了一些其他控制机制，使得算法能够更快速地收敛。

(3)改进算法结合了分布估计算法不易陷入局部最优与遗传算法对种群规模要求低的优点，在保证解的质量的前提下，提高了算法的运行速度。

如上实验结果表明，改进算法在求解速度和求解能力方面都大大超过了原有十进制 MIMIC 算法，在解的质量、稳定性、求解问题规模 3 个方面也都有明显的提高。

从表 1、表 2 可以看出，改进算法求解 100 规模以下的 TSP 问题非常有效，大部分实例均可搜索到最优解，甚至可以搜索出比 TSPLIB 中提供的最优路径更好的解。改进算法不仅提高了算法的求解质量，而且增强了算法对最优解命中的稳定性，较好地解决了算法的收敛速度与解质量之间的矛盾。

参 考 文 献

[1] 许昌，常会友，徐俊，等. 一种新的融合分布估计的蚁群优化算法[J]. 计算机科学，2010，32(2)：186-211

[2] 阎平凡，张长水. 人工神经网络与模拟进化计算[M]. 北京：清华大学出版社，2005：398-630

[3] 闫利军，李宗斌，卫军胡. 模拟退火算法的一种参数设定方法研究[J]. 系统仿真学报，2008，20(1)：245-247

[4] 吴建辉，章兢，张小刚，等. 一种求解 TSP 问题的分层免疫算法[J]. 计算机科学，2010，37(6)：256-264

[5] Guo Tao, Michalewicz Z. Evolutionary algorithms for the TSP [C]// Eiben A E, et al., eds. Proceedings of the 5th parallel Problem Solving from Nature Conference. Lecture Notes in Computer Science 1498. Berlin: Springer, 1998: 803-812

[6] 曹平，陈盼，刘世华. 改进的粒子群算法在旅行商问题中的应用[J]. 计算机工程，2008，34(11)：217-221

[7] Larranaga P, Lozano J. A Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation [M]. Dordrecht: Kluwer Academic Publishers, 2002

[8] 周树德，孙增圻. 分布估计算法综述[J]. 自动化学报，2007，33(2)：113-121

[9] 黄宝珠，肖菁. 改进的 MIMIC 算法求解旅行商问题[J]. 计算机工程与设计，2010，31(16)：3650-3657

[10] Jevemy S, Bonet D, Charles L, et al. MIMIC: Finding optima by estimating probability densities[C]// Advances in Neural Information Processing Systems Cambridge. MIT Press, 1997: 424-430

[11] 陈智军. 一种改进的求解 TSP 问题的遗传算法[J]. 软件导刊，2011，10(2)：52-54

[12] 胡晓辉，李晓阳，陈俊莲. 基于贪心策略的混合遗传算法在 TSP 中的实现[J]. 兰州交通大学学报，2009，28(3)：58-61

[13] 蔡之华，彭锦国，高伟，等. 一种改进的求解 TSP 问题的演化算法[J]. 计算机学报，2005，28(5)：823-828

[14] 王劲飞，陈琎，魏巍，等. 基于改进郭涛算法的 TSP 问题求解[J]. 计算机工程与设计，2006，27(5)：774-751