

# 一种基于逆转算子的求解 TSP 问题的改进演化算法

苏劲松<sup>1,2</sup>, 周昌乐<sup>2</sup>, 蒋旻隽<sup>1,2</sup>

(1. 厦门大学 软件学院, 福建 厦门 361005;

2. 厦门大学 人工智能研究所, 福建 厦门 361005)

**摘要:**使用逆转算子求解 TSP 的演化算法具有很强全局搜索能力,在求解 TSP 问题中显示了巨大的优势。但是,该算法同样存在执行效率低、最终得到的最优个体整体质量不高等缺陷。在对算法和 TSP 问题进行分析的基础上,对算法进行三方面的改进:就近选择;动态变异概率;基于较优个体的贪婪搜索。实验结果表明:经过改进的算法提高了执行效率,能够改善算法得到的最优个体的整体质量。

**关键词:**旅行商问题;演化算法;逆转算子

**中图分类号:**TP301.6

**文献标识码:**A

**文章编号:**1673-629X(2007)07-0094-04

## An Improved Evolutionary Algorithm for Traveling Salesman Problem Based on Inver - Over Operator

SU Jin-song<sup>1,2</sup>, ZHOU Chang-le<sup>2</sup>, JIANG Min-jun<sup>1,2</sup>

(1. Software School of Xiamen University, Xiamen 361005, China;

2. Institute of Artificial Intelligence of Xiamen University, Xiamen 361005, China)

**Abstract:** The evolutionary algorithm using inver - over operator for the traveling salesman problem (TSP) has great ascendancy, because its ability in global searching for optimal individual is powerful. However, it has the some limitations: low executive efficiency and the dissatisfactory average individuals obtained by algorithm. To avoid these limitations, improves the algorithm mentioned above in three aspects: close - by visit method, dynamic mutation probability and greedy search based on preferable individuals. A desirable result is obtained. It is showed by the experiment that the algorithm can be executed with high efficiency and the average quality of the optimal individual of the algorithm is improved.

**Key words:** traveling salesman problem; evolutionary algorithm; inver - over operator

### 0 引言

TSP(traveling salesman problem)问题也称为旅行商问题、货郎担问题。该问题的提出可以追溯到 18 世纪的欧拉时代,但直到 20 世纪中叶由于优化技术的兴起,特别是当它被证明是 NP - Hard 问题,并且与 VLSI 制造、输油管道铺设、电路布线等实际问题密切相关后,人们才对该问题进行深入的研究。TSP 问题可以简单描述为:设有  $N$  个城市并已知各个城市之间的距离,要求找一条路径,使得走遍所有城市,且每个城市只能经过一次,最后回到原处,该路径总距离最短。

TSP 的数学描述如下:有  $N$  个城市  $C_1, C_2, \dots, C_N$  每个城市  $C_i$  的坐标为  $(x_i, y_i)$ , 两点间的距离为:

$d(C_i, C_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ , 求一个城市序列:  $\{C_{\pi(1)}, C_{\pi(2)}, \dots, C_{\pi(N)}\}$ , 使得  $\sum_{i=1}^{N-1} d(C_{\pi(i)}, C_{\pi(i+1)}) + d(C_{\pi(N)}, C_{\pi(1)})$  最小。

TSP 是一个经典的组合优化问题,所有的路线的组合数为  $(N - 1)!/2$ , 其搜索空间随着城市数增大而迅猛增大,这就产生了组合爆炸问题。目前求解 TSP 问题的方法主要有贪婪法、分支定界法和动态规划法等精确算法以及最近邻法、模拟退火算法和遗传算法等近似算法。由于 TSP 问题是 NP - Hard 问题,因此,贪婪法、分支定界法和动态规划法等精确算法,在城市数  $N$  较大时计算量太大而无法实现。几百年来,将生物界所提供的答案应用于实际问题求解已被证明是一个成功的方法,而演化计算正是基于这种思想而发展起来的一种通用的问题求解方法,近年来,利用演化算法的全局寻优能力来求解 TSP 问题的研究相当活跃,演

收稿日期:2006-09-25

作者简介:苏劲松(1982-),男,福建泉州人,硕士研究生,研究方向为人工智能;周昌乐,教授,博士生导师,CCF 会员,主要从事人工智能领域的教学和研究工作。

化算法已经显示了巨大的优势。

## 1 逆转算子

在遗传学中有一种称为倒位的现象,就是在染色体中存在着两点,在这两点之间的基因位置是倒转的,从而使得那些在父代中离得很远的基因位在后代中紧靠在一起。仿造此现象,Holland 提出了一种“逆转算子”,并将其应用于演化算法,使得染色体位串上的重要基因更加紧凑,更不容易被交叉算子分裂。

逆转算子首先在个体串上随机地选择 2 点,位串被这 2 点分成 3 段,然后将中间段的左右顺序倒转过来并与另两段相连,形成新的个体位串。以(1 2 3 4 5 6 7 8)为例,其中 4 和 7 为倒位点,则经过逆转算子作用后,形成新的位串(1 2 3 7 6 5 4 8)。

使用逆转算子求解 TSP 问题的算法<sup>[1]</sup>可以描述如下:

第一步,随机产生若干条路径形成初始群体。

第二步,对群体中的每个个体  $S_i$  按照下面规则来选择倒位点所对应的两个城市:

(1) 杂交,对  $S_i$  中的某个城市  $C$ ,随机选择另一个个体  $S_j$ ,找出  $S_j$  中城市  $C$  的下一个城市  $C'$ ;

(2) 变异,对  $S_i$  中的某个城市  $C$ ,在  $S_i$  中随机选择下一个城市  $C'$ 。

在选定  $C$  和  $C'$  之后,在  $S_i$  中,以城市  $C$  的下一个城市以及  $C'$  为倒位点,进行逆转操作。如果逆转后得到的新路径长度小于原路径长度,则用新路径取代原路径,否则保持原路径,对群体反复进行演化,直到满足终止条件。

很多专家学者利用逆转算子来求解 TSP 问题<sup>[2~5]</sup>,取得了良好的效果。文中以此为基础,分析现有逆转算子算法中存在的不足,并加以改进,以此来提高算法的执行效率,改善实验得到的最优个体的整体质量。

## 2 算法的改进

### 2.1 就近选择

通过对 TSP 问题的求解过程进行分析发现,算法得到的较优个体中,绝大多数的城市都是和靠近它的几个城市中的某一个相连,而当两个城市距离非常远时,得到的个体质量往往都比较差。因此,文中采用就近选择的方法。求出与每个城市  $C_i$  距离最近的 numB 个城市,存放入数组 nearCity[ $C_i$ ] 中,并用该数组来限制产生初始种群和进行逆转算子操作时选择下一个城市的范围。

产生初始种群时选择下一个城市过程如下:把城市的  $C_i$  的下一个城市限制在 nearCity[ $C_i$ ] 中产生,如果其中所有城市都已经被选定,再去选取 nearCity[ $C_i$ ] 外的城市。这样产生的初始群体具有较高的质量。

为了保持群体的多样性,在演化过程中要通过逆转算子操作来完成杂交和变异。和前面相同,通过逆转算子操作得到的个体中, $C_i$  和  $C'$  相邻,如果  $C_i$  和下一个城市  $C'$  距离非常远的时候,那么该个体质量一定非常差。在此,把选定下一个城市的范围限制在 nearCity[ $C_i$ ] 之中,如果选择的下一个城市超出了这个范围,则重新进行选择。这样,避免了大量无效的逆转算子操作,能够使得算法前期快速收敛,提高了算法的执行效率。

### 2.2 动态变异概率

变异操作在演化计算中起着非常重要的作用。种群在收敛过程中,容易陷入局部最优,特别是当迭代次数到达一定程度时候,这种现象就表现得更为明显,这时就只有通过变异操作,使得种群个体跳出局部最优。但是,如果刚开始变异概率设置的比较大,那么就不容易保留演化过程中个体的优良片段,种群个体不容易收敛,反而降低算法的执行效率。由此,提出动态变异概率的方法,刚开始时,变异概率设置的不高,使得种群个体能以较快的速度收敛,到演化后期时,种群个体较为稳定,随着迭代次数的增加逐步提高变异概率,使得种群个体能有更高概率跳出局部最优。

### 2.3 基于较优个体的贪婪搜索

演化计算执行到后期时,种群往往都较为稳定。根据前面所述,文中将变异概率动态增大,以此来提高得到较优个体的概率。但是,变异操作并不是启发式搜索,具有很强的随机性,要得到一个较优个体,往往需要大量的运算时间,甚至最后得到的最优个体质量并不高。对此,结合贪婪算法的思想,提出在算法前面步骤得到的较优种群基础上,对每个个体进行贪婪搜索,以使得在一定迭代次数内,能够得到更优个体,提高算法得到的最优个体的整体质量。搜索过程如下:

已知经过算法前面两个步骤得到群体个体  $S:(\dots C_{pre}, C_i, C_{next} \dots)$ ,其中  $C_{pre}$  和  $C_{next}$  分别为  $C_i$  在路径  $S$  中的前一个城市和下一个城市,判断是否存在  $C_a$  和  $C_b$  使得:

1)  $C_a$  和  $C_b$  同属于 nearCity[ $C_i$ ];

2)  $C_a$  在路径  $S$  中的下一个城市是  $C_b$ ,  $C_a$  不等于  $C_{pre}$ ,且  $C_b$  不等于  $C_{next}$ ;

3)  $d(C_{pre}, C_i) + d(C_i, C_{next}) + d(C_a, C_b) > d(C_{pre}, C_{next}) + d(C_a, C_i) + d(C_i, C_b)$ 。

对每个个体都进行上述搜索计算,选择使得路程减少量最大的  $C_i, C_a$  和  $C_b$ , 用  $S'$ : ( $\dots C_{pre}, C_{next} \dots C_a, C_i, C_b \dots$ ) 代替原来的  $S$ , 同样,对种群的所有个体进行相同的贪婪搜索,并求出其中的最优个体。当种群的最优个体连续几代都不改变时,则搜索结束。

上述变化过程如图 1 所示。

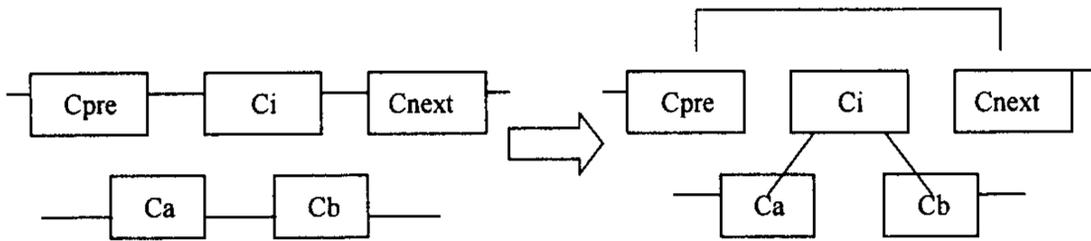


图 1 基于较优个体的贪婪搜索时的个体变化  
文中通过以上 3 方面对逆转算子求解 TSP 问题的算法进行了改进,算法的基本结构如图 2 和图 3 所示。

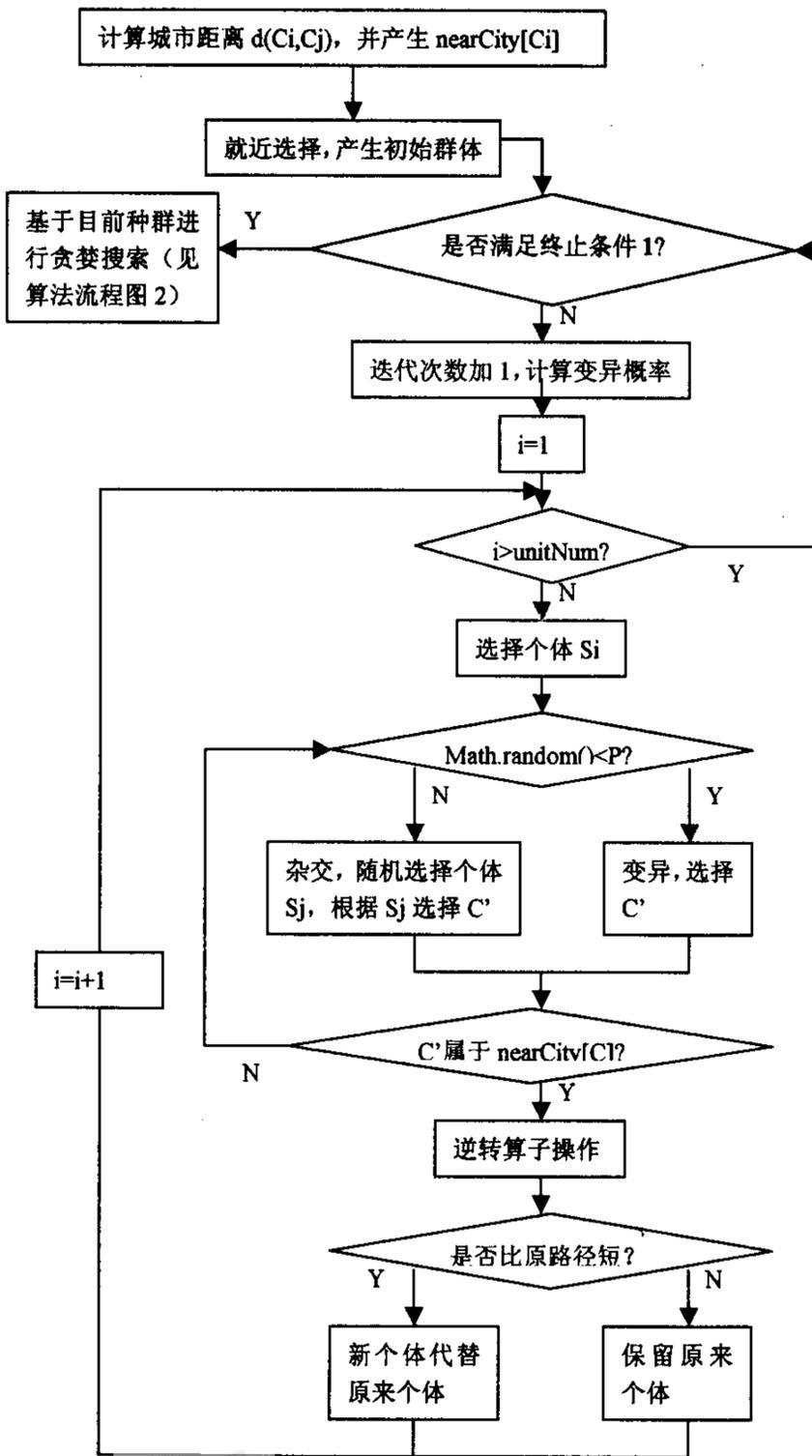


图 2 算法流程图 1

为了证明对算法的改进是有效的,文中用原算法和改进后的算法对 CHN144 实例分别进行求解。设定

距离都是 Double 型,以原算法和改进后的算法分别执行 100 次,根据平均数值来对比算法改进前后的迭代次数、执行时间、最优个体的整体质量和所求到的最优个体。

实验条件如下:

CPU:P4 1.8G 内存:512M

参数设置见表 1。

实验结果见表 2。

1) 分别取定 38000, 35000, 32000, 31000 四个点,计算当算法的目前最优个体小于每个点时的平均迭代次数和执行时间,观察算法改进前后的执行效率。

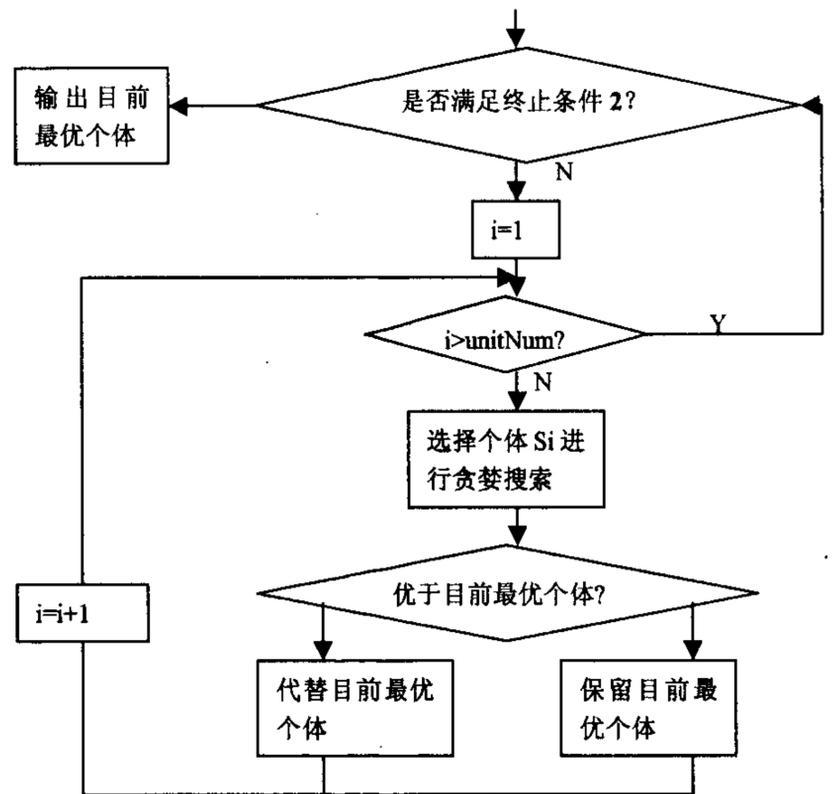


图 3 算法流程图 2

表 1 实验参数设置

	种群个数	变异概率	相邻城市个数	终止条件 1	终止条件 2
原算法	100	0.1	—	最优个体保持 10000 代	—
改进后的算法	100	$p = 0.1, \text{genNum} < = 10000$ ; 反之, $p = \text{genNum} / 10000$	25	最优个体保持 10000 代	最优个体保持 20 代

表 2 实验结果 1

	平均迭代次数		平均运行时间(ms)	
	原算法	改进后算法	原算法	改进后算法
38000	1930.00	1095.83	4600.50	3504.17
35000	2399.87	1445.53	5213.10	4079.67
32000	3545.33	2435.37	6682.87	5268.20
31000	6144.10	4306.33	10278.13	8057.27

由以上实验数据可以看出经过改进的算法比起原算法前期收敛速度更快,执行效率更高,因而文中所提出“就近选择”对算法的改进是有效的。

2) 分别对改进前后算法所求出来的最优个体的总路径进行平均计算,以此来衡量算法最优个体的整体质量(见表 3)。目前公布的最好结果为 30354.3<sup>[2]</sup>。

表3 实验结果2

	实验结果平均值	实验结果最优值	目前最优值
原算法	30525.04	30355.12	30354.3
改进后的算法	30468.25	30355.12	30354.3

由以上实验数据可以看出原算法和改进后的算法所能搜索到的最优个体是一样的,但是改进后算法得到的最优个体的整体质量更优,因而文中所提出的“动态变异概率”和“基于较优个体的贪婪搜索”对算法整体结果的改进是有效的。如图4所示。

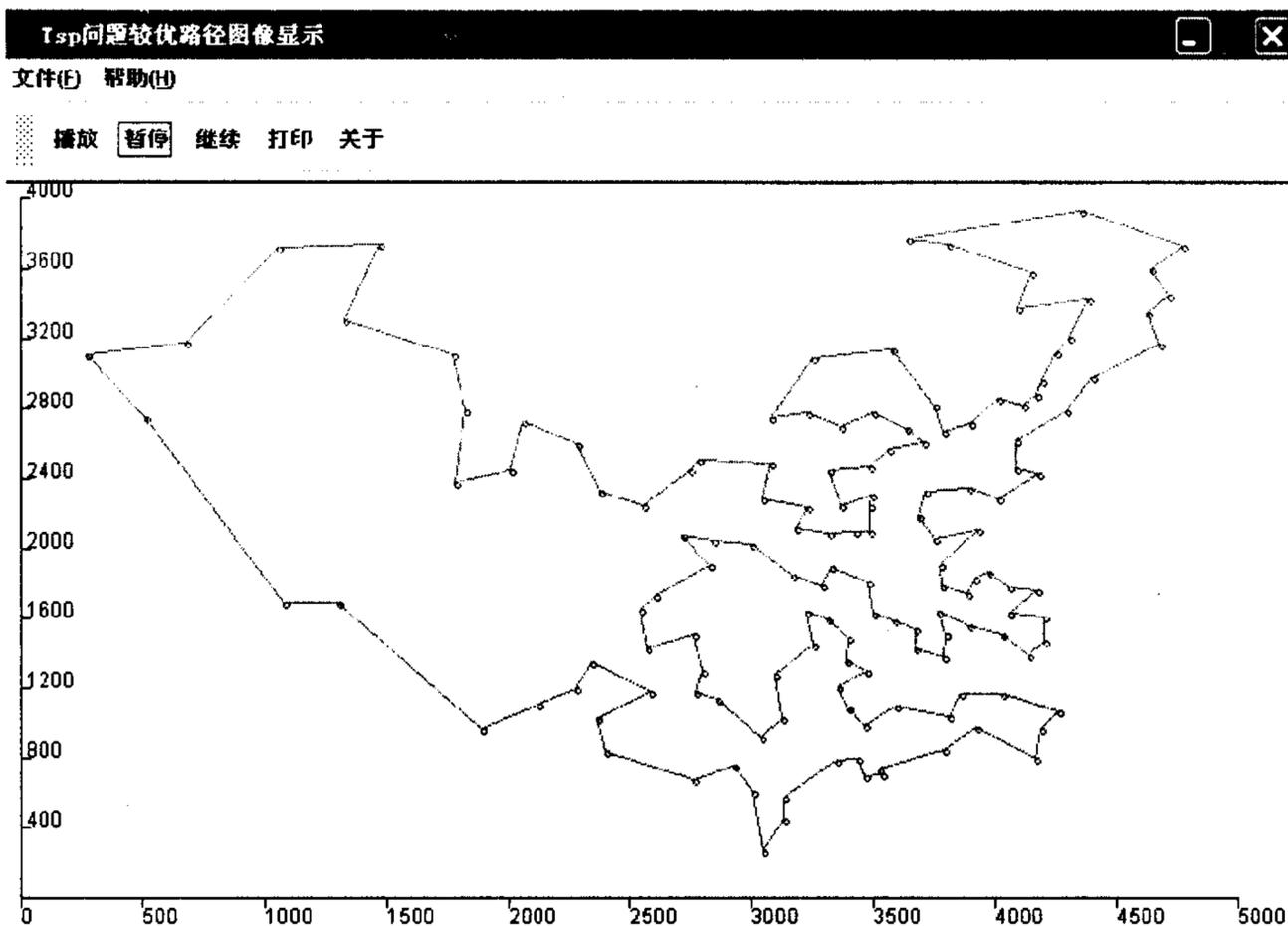


图4 实验最优路径

### 3 结论

文中对使用逆转算子求解 TSP 的算法进行了改进,提出了就近选择、动态变异概率和基于较优个体的贪婪搜索的方法,并通过求解经典 CHN144 城市实例证明了以上方面对算法的改进都是有效的,使得算法在保证执行效率的同时,提高了最优个体的整体质量。

#### 参考文献:

- [1] Michalewicz T G. Inver-over operator for the TSP[C]// Parallel Problem Solving form Nature, PPSN V. Berlin, Gemany: Springer - Verlag, 1998:803 - 812.
- [2] 吴 斌,史忠植.一种基于蚁群算法的 TSP 问题分段求解算法[J]. 计算机学报, 2001, 24(12): 1228 - 1233.
- [3] 潘正君,康立山,陈毓屏. 演化计算[M]. 北京:清华大学出版社,2004.
- [4] 康立山,谢 云. 非数值并行算法[M]. 北京:科学出版社,1997.
- [5] 宋 丹,傅 明,朱亨荣,等. 一种改进的遗传算法及其在 TSP 中的实现[J]. 微机发展, 2004, 14(6): 21 - 23.

(上接第 93 页)

基于分类的思想组织资源,具有较优的负载均衡能力。

首先,采用结构化的 P2P 技术,消除了集中式的索引服务器,实现各 peer 间的资源共享,很大程度上解决了系统可扩展性问题,避免了完全随机搜索的低效,具有很高的路由效率和很强的负载均衡能力。

其次,分布式与层次式结合的资源组织方式,可以在同一个资源组中进行任务转移,如将部分负载转移到轻载节点,实现相似类型资源间的负载均衡。

### 4 总结

文中对网格资源发现及管理的相关技术及已有构架进行了探讨和分析。在此基础上,提出了一个在资源组织方式、资源发现机制、负载均衡等多方面性能进行优化的网格资源发现模型。该模型结构主要采用集中式与分布式相结合的方式、P2P 技术、对资源服务进行分类的思想进行构建,符合资源自配置、自主管理、

动态资源发现和容错等要求,并具有高效的资源定位能力和良好的强壮性。

#### 参考文献:

- [1] 董方鹏,龚奕利,李 伟,等. 网格环境中资源发现机制的研究[J]. 计算机研究与发展, 2003, 40(12): 1749 - 1755.
- [2] 都志辉,陈 渝,刘 鹏. 网格计算[M]. 北京:清华大学出版社,2002.
- [3] 黄道颖,黄建华,庄 雷,等. 基于主动网络的分布式 P2P 网络模型[J]. 软件学报, 2004, 15(7): 1081 - 1089.
- [4] Iamnitchi A. Resource discovery in large resource sharing environments[D]. Chicago: University of Chicago, 2003.
- [5] Iamnitchi A, Foster I. On Fully Decentralized Resource Discovery in Grid Environments[C]//In: Lee C A. Grid Computing - Grid 2001. Germany: Springer, 2001.
- [6] 朱 承,张维明,刘 忠,等. 一种基于资源类型的网格资源发现方法[J]. 计算机研究与发展, 2004, 41(12): 2156 - 2163.