

# 一种新型异构分布式混合遗传算法<sup>\*</sup>)

熊忠阳 张 磊

(重庆大学计算机学院 重庆400044)

**摘 要** 本文结合混合遗传算法和异构分布式遗传算法两种思想提出了一种异构分布式混合遗传算法。实验表明,该算法在时间性能和优化性能上优于上述两种算法,并且提供了一种在最佳搜索空间的快速搜索和找到更好的最佳解的能力。

**关键词** 模拟退火,禁忌搜索,混合遗传算法,异构分布式遗传算法

## A New Heterogeneous Distributed Hybrid Genetic Algorithms

XIONG Zhong-Yang ZHANG Lei

(Department of Computer Science and Engineering, Chongqing University, Chongqing 400044)

**Abstract** Hybrid genetic algorithm and heterogeneous distributed genetic algorithm are well combined in this paper, and a new algorithm is derived, that is the heterogeneous distributed hybrid genetic algorithms. The experiments results show that not only is this algorithm better than two algorithms above on time performance and optimization performance, but also it provides a method to search the optimal search space quickly and find out the better solution.

**Keywords** Simulated annealing, Tabu search, Hybrid genetic algorithms, Heterogeneous distributed, Genetic algorithms

## 1 引言

遗传算法(genetic algorithm,简称GA)最早由美国密执安大学的J. H. Holland教授提出,起源于20世纪60年代对自然和人工自适应系统的研究<sup>[1]</sup>。它是模仿自然界生物遗传和进化过程中“物竞天择、适者生存”的原理而开发出的一种全局优化随机搜索算法。虽然理论上已经证明,遗传算法能从概率的意义上以随机的方式寻求到问题的最优解,但实际应用时收敛条件往往难以保证。大量研究表明,传统的遗传算法具有易早熟、局部寻优能力较差、算法参数敏感等缺点,这些缺点严重影响了遗传算法的运行效率。另一方面,模拟退火(Simulated Annealing,简称SA)算法、禁忌搜索(Tabu Search或Taboo Search,简称TS)算法具有较强的局部搜索能力,并能使搜索过程避免陷入局部最优解,但却对整个搜索空间缺乏了解,不利于搜索最有希望的区域,从而各种结合SA、TS和GA的混合遗传算法(简称GASATS)应运而生<sup>[2]</sup>。应用实践表明,这种结合能提高算法的搜索能力,避免早熟,但时间性能较GA有较大的下降。

除了上述方法外,多种群并行遗传进化也是一种改进遗传算法性能的有效途径<sup>[3]</sup>。本文结合两种思想提出了一种异构分布式混合遗传算法(简称HDHGAs)。实验表明,HDHGAs表现出两个明显的优点:1)在时间性能上较GASATS有较大的改善;2)在优化性能上较GASATS和HDGAs有较大改善,具有找到更好最佳解的能力。

## 2 引入模拟退火和禁忌搜索机制的混合遗传算法(GASATS)

标准遗传算法存在着局部搜索能力差的缺陷,将遗传算

法和其它搜索方法相结合构成混合遗传算法的主要目的是改善标准遗传算法的局部搜索能力,进一步提高优化质量和搜索效率,以弥补单一优化方法的某些不足之处<sup>[2]</sup>。GA与SA、TS结合方式多种多样,本文所采用的GASATS其优化流程图1。

针对单纯GA局部搜索能力弱的缺陷,GASATS增强了GA的局部搜索能力。与单纯TS相比,GASATS拓展了TS的局部搜索范围,增强了TS的全局搜索能力。实践表明,GASATS有效地克服了过早收敛现象,加快了收敛速度,解的质量也有提高<sup>[2]</sup>。

## 3 异构分布式遗传算法(HDGAs)

最初人们为了避免早熟现象,将单一种群划分为几个独立运行的子种群,产生了分布式遗传算法(简称DGAs)。每个子种群可以具有不同的控制参数、种群规模和遗传算子,这种类型的分布式遗传算法称为异构分布式遗传算法(简称HDGAs)<sup>[4]</sup>。HDGAs形式多种多样,本文所采用的HDGAs各子种群的优化流程图2。

如图3所示为四个子种群并行进化的例子,子种群1的交叉概率 $P_c$ 、变异概率 $P_m$ 较大,退温速率 $C$ 较小,称为探测子种群,用于在进化过程中不断提供新的搜索空间;子种群4的 $P_c$ 、 $P_m$ 较小,退温速率 $C$ 较大,称为开发子种群,用于在局部范围内寻找优秀个体;子种群2、3的 $P_c$ 、 $P_m$ 、 $C$ 值在1和4之间,称为探测开发子种群,兼具上述两个子种群的功能<sup>[3]</sup>。各个子种群具有不同的交叉算子,用于保持搜索的多样性。每个子种群独自进化五代,然后向下一个子种群传播当前最优的 $M$ 个个体,如此往复,直至满足收敛条件为止。

<sup>\*</sup>)本文研究得到重庆市科委应用基金项目(No. 6968)资助。熊忠阳 副教授,主要研究方向为数据库,数据挖掘和并行计算。张 磊 硕士研究生,研究方向为软件复用,混合优化策略和并行计算。

HDGAs 是解决早熟收敛问题的一种有效方法,使得算法的全局搜索和局部搜索能力得到了充分的结合。

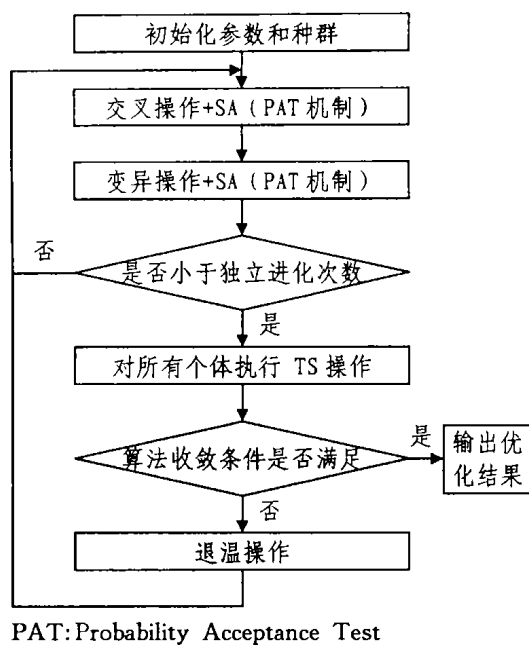


图1 GASATS 优化流程图

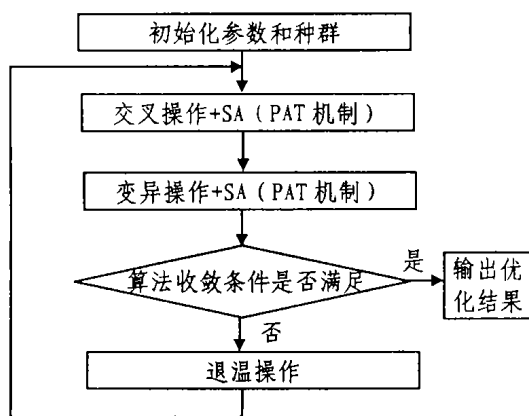


图2 HDGAs 各子种群优化流程图

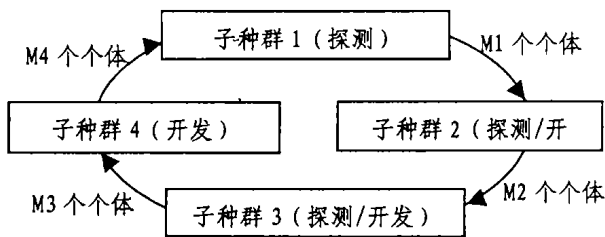


图3 HDGAs 结构图

#### 4 异构分布式混合遗传算法 (HDHGAs) 的基本思想

本文结合上述两种算法的思想提出了一种异构分布式混合遗传算法,它由一个根处理器和多个处理单元组成,各处理单元执行异构的 GASATS,用于产生一个当前最佳搜索空间。所谓异构是指每个处理单元具有不同的交叉算子,不同的交叉概率,不同的退温速率,不同的 TS 领域函数。根处理器上执行禁忌搜索算法,用于对当前最佳搜索空间进行局部搜索。经过一定的间隔,各处理单元将各自的当前最优个体组发

送给根处理器,根处理器对其进行处理后,返回其最优个体组到各个处理单元,如此往复,直到满足收敛条件为止。

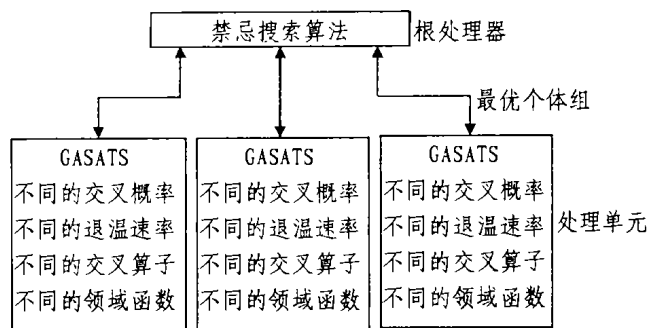


图4 HDHGAs 结构图

图4所示为1个根处理器和3个处理单元并行运行的例子。算法利用 SA 的 PAT 机制<sup>[5]</sup>,降低了算法对控制参数敏感性。各个处理单元上的子种群分别执行不同的交叉操作,这样的多交叉操作使得搜索行为具有明显的多样性,增强了算法的全局搜索能力。各个处理单元上的子种群在禁忌搜索阶段分别执行不同的领域函数,这样的机制使得禁忌搜索不仅可以多点同时进行,而且还增加了搜索行为的多样性,增强了算法的局部搜索能力。根处理器执行禁忌搜索算法对当前最佳搜索空间进行搜索,增强了整个算法的局部搜索能力和收敛速度。

#### 5 异构分布式混合遗传算法 (HDHGAs) 的基本步骤

在根处理器上执行如下操作:

- (1)初始化控制参数:子种群个体数  $M_0 = nproc \times M$ , 独立搜索次数  $V_0$ , 禁忌长度  $L_0$ , 禁忌列表长度  $LIST_0$ ;
- (2)算法初始化,随机产生  $nproc$  个处理单元,并将各处理单元运行所需的参数值发送到各个处理单元上;
- (3)随机产生初始种群,设定进化代数  $Loop = 0$ ;
- (4)若当前最优个体满足收敛条件,则终止各处理单元的执行,算法结束,输出最优解;否则,继续以下步骤;
- (5)找出当前最差的  $M$  个个体,并对其执行禁忌搜索操作;
- (6)若当前最优个体满足收敛条件,则终止各处理单元的执行,算法结束,输出最优解;否则,继续以下步骤;
- (7)找出当前最优的  $M$  个个体,将其发送给各个处理单元;
- (8)等待接收来自各个处理单元的最优个体组;
- (9) $Loop \leftarrow Loop + 1$ ,若  $Loop > \text{最大进化代数 } MaxLoop$ ,则终止各处理单元的执行,算法失效结束;否则,转向步骤(4);

各处理单元执行如下操作:

- (1)接收来自根处理器的控制参数:子种群个体数  $M$ , 交叉概率  $P_c$ , 变异概率  $P_m$ , 退温速率  $C$ , 禁忌长度  $L$ , 禁忌列表长度  $LIST$ , 每次发送的个体数  $M$ , 其中  $1 \leq i \leq nproc$ ;
- (2)随机产生初始子种群  $P$ , 计算退火初始温度  $T$ , 设定进化代数  $Loop = 0$ ;
- (3)设定计数变量  $K = 0$ ;
- (4)等待接受来自根处理器的最优个体组,替换掉  $P$  中最差的  $M$  个个体;
- (5)对  $P$  中的个体以随机配对方式组成  $M/2$  对个体组

$(x_j, x_k)$ , 以交叉概率  $P_c$  进行与  $i$  对应的交叉操作, 产生新个体  $x_j'$  和  $x_k'$ , 并计算适应度函数值  $F(x_j')$  和  $F(x_k')$ ; 若  $\min\{1, \exp(-(F(x_j) - F(x_j'))/Ti)\} \geq \text{Random}[0, 1]$ , 则接受新个体  $x_j'$ , 否则保留原个体  $x_j$ ; 若  $\min\{1, \exp(-(F(x_k) - F(x_k'))/Ti)\} \geq \text{Random}[0, 1]$ , 则接受新个体  $x_k'$ , 否则保留原个体  $x_k$ ;

(6) 对交叉后的个体按变异概率  $P_m$  进行变异操作, 按(5)中的方法决定是否接受新个体;

(7) 若  $K < \text{独立进化次数 } S$ , 则  $K \leftarrow K + 1$ , 转向步骤(5); 否则, 继续以下步骤;

(8) 对  $P_t$  中的每一个个体执行禁忌搜索操作;

(9) 从  $P_t$  中选出最优的  $M$  个个体组成最优个体组, 并将其发送到根处理器;

(10)  $Loop \leftarrow Loop + 1$ , 修改  $P_t$  的退火温度,  $T_i \leftarrow C \cdot T_i$ , 转向步骤(4)。

## 6 实验及结果分析

为了验证 HDHGAs 的性能, 我们选择了基于 windows 2000 的 PVM 并行计算平台和 TSP 问题。

### 6.1 基于 windows 2000 的 PVM 并行计算平台

PVM 最初设计的目的是使异构的 Unix 计算机网络能作为一个大型的消息传递并行机来使用。随着流行程度的增加, PVM 也被移植到 windows 平台上。针对目前运行 windows 系统的 PC 机占主流的现状和 windows 系统上丰富的资源<sup>[6]</sup>, 我们采用了基于 windows 平台的 PVM3.4.3 和 windows 2000 作为并行计算平台。

### 6.2 TSP 问题

旅行商问题(Traveling Salesman Problem, 简称 TSP)是典型的 NP 完全问题, 并且是一个典型的组合优化问题<sup>[1]</sup>。该问题可描述为: 给定  $n$  个城市, 要求确定一条经过各城市仅一次, 最后回到出发城市的最短路线。TSP 问题在实际应用中具有很重要的意义, 例如可用来解决分配问题、路径问题、车辆调度问题、网络问题等。TSP 问题的描述虽然简单, 但用传统优化算法很难精确地求出其最优解, 其主要原因就是所谓的“组合爆炸”。因此, 近十年来, 基于遗传算法求解 TSP 问题的研究相当活跃。用 HDHGAs 求解一些有代表性的 TSP 问题是对其性能的一个极好的检验。

### 6.3 实验结果及分析

GASATS 算法主要控制参数如下: 种群规模均为 20; 交叉概率为 0.5; 变异概率为 0.9; 独立化次数为 20; 退温速率为 0.9; 初温取  $-\Delta_{\max}/\ln P_t$ ; 禁忌长度为 5; 禁忌列表长度为 10; 候选解集大小为 6; 独立搜索次数取 10; 以最优值连续 20 代不变或达到参考最优值为收敛条件。

HDGAs 算法主要控制参数如下: 种群规模分别为 80, 70, 60, 50; 交叉概率分别为 0.9, 0.5, 0.7, 0.3; 变异概率分别为 0.9, 0.3, 0.1, 0.05; 退温速率分别为 0.8, 0.85, 0.9, 0.95; 初温均取  $-\Delta_{\max}/\ln P_t$ ; 发送个体数分别为 8, 7, 6, 5; 以最优值连续 150 代不变或达到参考最优值为收敛条件。

HDHGAs 根处理器上算法主要控制参数如下: 种群规模为 20; 禁忌长度为 5; 禁忌列表长度为 10; 候选解集大小为 6; 以最优值连续 20 次不变或达到参考最优值为收敛条件。

HDHGAs 各处理单元上主要控制参数如下: 种群规模均为 20; 交叉概率分别为 0.7, 0.5, 0.3; 变异概率均为 0.9; 独立进化次数均为 20; 退温速率分别为 0.85, 0.9, 0.95; 初温均取  $-\Delta_{\max}/\ln P_t$ ; 禁忌长度均为 5; 禁忌列表长度均为 10; 候选解集大小均为 6; 独立搜索次数取 10; 每次向根处理器发送个体

数均为 10。

(1) 实验一。为了检验 HDHGAs 的时间性能, 我们采用了 TSP 研究中常用的 Eilon 等 1969 年提出的 75 城市 TSP 问题<sup>[7]</sup>。实验以 Fogel(1993) 给出的最佳优化结果 549.18 作为参考最优解, GASATS 是在 1 台 Pentium 4 2.4G PC 机上完成, HDGAs 和 HDHGAs 是在 4 台 PC 机(Celeron 1.7G、1.8G、1.7G 和 Pentium 4 2.4G) 构成的机群上完成, 约定各算法均随机运行 10 次。

性能指标	算法		
	GASATS	HDGAs	HDHGAs
最佳解	543.908	545.942	535.780
最劣解	564.964	565.277	547.784
平均值	550.421	552.981	543.890
平均运算时间	624秒	60秒	43秒
相对误差	-0.009600	-0.005896	-0.024401
波动率	0.002259	0.006921	-0.009632

其中, 相对误差 = (算法多次运行所得的最佳优化值 - 问题的最优值) / 问题的最优值; 波动率 = (算法多次运行所得的平均值 - 问题的最优值) / 问题的最优值。

(2) 实验二。为了检验 HDHGAs 的优化性能, 我们选取了 TSPLIB<sup>[8]</sup> 库中的 KroA100、Bier127 和 Pr152 三个问题进行计算, 以公布的最短距离作为参考最优值。算法在上述机群上完成, 随机运行 5 次。

算法	性能指标	KroA100	Bier127	Pr152
HDHGAs	已知最短距离	21282	118282	73682
	最佳解	21136	116601	71278
	平均值	21198	117554	72672

(3) 结果分析。由实验结果可见, HDHGAs 的时间性能和优化性能较 GASATS 有较大的提高, 而且优于 HDGAs; HDGAs 虽然在时间性能上大大优于 GASATS, 但在最优化性能方面 GASATS 优于 HDGAs, HDHGAs 结合了两者的优点, 而且每次所得的最优解均好于目前公布的最优解; GASATS 和 HDGAs 波动性较大, HDHGAs 在这方面体现出了较大的优越性。

我们认为 HDHGAs 之所以表现出很强的优化能力的原因是, 利用模拟退火的爬山性能提高了遗传算法的收敛速度, 同时赋予搜索过程避免局部极小的能力; 利用禁忌搜索的强大的局部搜索能力弥补了遗传算法局部搜索能力的不足; 采用多交叉操作和多邻域函数不仅增强了全空间的有效搜索, 而且也能增加种群的多样性; 根处理器上执行禁忌搜索算法, 进一步增强了算法的收敛速度和优化性能, 有利于进化过程的发展。

**结束语** 本文分析了 GASATS 和 HDGAs 存在的不足, 结合这两种思想提出了一种异构分布式混合遗传算法, 实验表明, HDHGAs 在时间性能和优化性能上优于 GASATS 和 HDGAs, 达到了预期的目的。未来的工作包括: 1) 进一步验证算法的性能; 2) 调整算法的控制参数使算法的性能达到最优; 3) 将算法运用到其他优化问题。

## 参考文献

- 1 周明, 孙树栋著. 遗传算法原理及应用. 国防工业出版社, 1999
- 2 陈国良, 王煦法, 庄镇泉, 王东生. 遗传算法及其应用. 人民邮电出版社, 1996

- 1) 取出属性个数最小的 REDUCT, 如只有一个, 则该 REDUCT 即为最终所求的 REDUCT
- 2) 属性个数最小的 REDUCT 有多个时, 计算每个 REDUCT 的任意两个属性之间的条件信息熵(参见定义10)
- 3) 对于任意的 REDUCT, 计算平均条件信息熵
- 4) 平均信息熵最小的 REDUCT 即为最终所求的 REDUCT

## 6 举例

如表1, 令预处理门限值为60%, 经过非一致性数据欲处理后, 得到表2。

表2的区别矩阵为

表3

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_1$							
$x_2$							
$x_3$	$b, c, d$	$b, c$					
$x_4$	$b$	$b, d$	$c, d$				
$x_5$	$a, b, c, d$	$a, b, c$		$a, b, c, d$			
$x_6$	$a, b, c, d$			$a, b, c, d$			
$x_7$		$a, b$	$c, d, a, b$	$cd$	$c, d$		

从表3可知,  $CORE = \{b\}$ ,  $Q = c \vee d$

得到  $RED_1 = \{b, c\}$ ,  $RED_2 = \{b, d\}$

$U/b = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5, x_6, x_7\}\}$

$U/c = \{\{x_1, x_2, x_4, x_7\}, \{x_3, x_5\}, \{x_6\}\}$

$U/d = \{\{x_1, x_4, x_7\}, \{x_2, x_3, x_5, x_6\}\}$

$H(c|b) = 0.302, H(d|b) = 0.221$

取信息熵最小的  $H(d|b)$ , 可知, 最终所求的 REDUCT =  $\{b, d\}$ 。

## 7 实验

KAN LI, YU-SU LIN 提出的算法<sup>[1]</sup>, 不能对包含有非一致性元素的数据进行处理, 在进行运算前, 必须确保论域中数据为一致性的。本算法与之相比, 能对非一致性数据进行预处理, 然后进行特征选择。而且本算法与普通的算法比, 也较为简洁易懂。

以下选择 UCI 标准数据集进行测试, 并与文[1]中算法结果进行比较, 文[1]中算法明确的指出不可以处理非一致性数据, 故而文[1]算法只取一致性数据的试验结果。其中非一致性数据预处理门限值为60%。

从以上比较可看出, 本算法扩展了计算数据集的范围, 对于一致性数据取得和文[1]算法大致相同的效果, 而本算法还可以对包含非一致性数据的数据集进行有效的处理。显然, 本论文所提出的算法要优越。

**结论** 本论文提出了一种具有非一致性数据预处理的粗糙集特征选择算法。利用非一致性数据处理得到一致性论域后, 用粗糙集方法得到 CORE 和多个可能的 REDUCT。然后用信息论知识计算各 REDUCT 中各元素的相关性, 平均信息熵最小说明各元素的相关性小, 取该 REDUCT 作为最后的属性集。通过实验, 证明本算法具有合理的非一致性数据处理方法, 选取 REDUCT 具有较好的效果。

数据集	实例个数	非一致性实例个数	属性个数	REDUCT 个数	
				本算法	文[1]算法
Ballon(1)	16		4	2	2
Ballon(2)	16		4	2	3
Ballon(3)	16		4	2	2
Ballon(4)	16		4	4	4
tic-tac-toe	958		9	8	8
Chess(kr vs kp)	3196		36	29	32
Balance scale	625	190	4	4	
Mushroom	8124	2684	22	12	
Postoperative Patient	90	6	8	8	
Restricted (primary-tumor)	339	27	17	16	
restricted (breast-cancer)	286	6	9	8	

在计算 REDUCT 时, 虽然方法简单, 但是当实例特别多或者属性个数比较多时, 从区别矩阵计算 REDUCT 仍然有计算量大的问题, 采用新的方法, 比如启发式等别的方法是今后研究的重点<sup>[4]</sup>。

## 参考文献

- 1 LI K, LIN Y S. Rough set based attribute reduction approach in data mining. In: Proc. of the First Intl. Conf. on Machine Learning and Cybernetics, Beijing, China, Nov. 2002
- 2 Miao Duoqian, Wang Jue. Information-based algorithm for reduction of knowledge. In: 1997 IEEE Intl. Conf. on Intelligent Proceeding Systems, Beijing, China, Oct. 1997
- 3 Questier F, Arnaut-Rollier I, Walczak B, Massart D L. Application of rough set theory to feature selection for unsupervised clustering. Chemometrics and Intelligent Laboratory Syatems, 2002, 63: 155~167
- 4 Zhong N, Dong J, Ohusuga S. Using Rough Sets with Heuristics for Feature Selection. Journal of Intelligent Information Systems, 2001, 16: 199~214
- 5 Pawlak Z. Rough Sets. International Journal of Computer and Information Sciences, 1982, 11: 341~356
- 6 Zheng Zheng, Wang Guoying, Yu Wu. Object's Combination Based Simple Computation of Attribute Core. In: Proc. of the 2002 IEEE Intl. Symposium on Intelligent Control, Vancouver, Canada, Oct. 2002
- 7 张祥德, 张巍, 刘玉蓉. 数据挖掘分类问题的贪婪粗糙集约简算法. 东北大学学报(自然科学版), 2001, 22(5)
- 8 杨华军, 苏德富. 基于 Windows 95/NT 的 PVM 并行计算平台. 计算机工程, 1999, 25(2): 24~25
- 9 王凌著. 智能优化算法及其在应用. 清华大学出版社, 2001
- 10 http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/
- 11 Grefenstette J, et al. Genetic Algorithms for the Traveling Salesman Problem. In: Proc. of 1st Intl. Conf. on Genetic Algorithms and Their Applications, Lawrence Erlbaum Associates, 1985. 154~159
- 12 Xiong Shengwu, Li Chengjun. A Distributed Genetic Algorithm to TSP. Intelligent Control and Automation, In: 2002. Proc. of the 4th World Congress on, Volume: 3, June 2002. 1827~1830

(上接第195页)

- 3 吴浩扬, 常炳国, 朱长纯, 刘君华. 基于模拟退火机制的多种群并行遗传算法. 软件学报, 2000, 11(3): 416~420
- 4 Herrera F, Lozano M. Heterogeneous Distrubuted Genetic Algorithms Based on the Crossover Operator. Genetic Algorithms In Engineering Systems: Innovations And Applications, 1997. GALEZIA 97. Second Intl. Conf. On (Conf. Publ. No. 446), Sept. 1997. 203~208
- 5 Fung C C, Chow S Y, Wong K P. Solving the Economic Dispatch Problem with an Integrated Parallel Genetic Algorithm, Power System Technology, 2000. In: Proc. PowerCon 2000. Intl. Conf. on, Volume: 3, Dec. 2000. 1257~1262