基于函数复杂度的自适应模拟退火和禁忌搜索新算法

许鹏飞[1],苗启广[1],李伟生[2],张军英[1]
(1.西安电子科技大学计算机学院,陕西西安 710071; 2.重庆邮电大学计算机学院,重庆 400065)

摘 要: 在求解多峰复杂函数的过程中,传统的模拟退火算法和禁忌搜索算法经常出现算法快速收敛于局部最优解、后期收敛速度变慢和搜索能力变差等问题.为解决这些问题,本文给出函数复杂度的定义,并提出基于函数复杂度的自适应模拟退火和禁忌搜索算法.该算法首先根据函数复杂度自适应调整步长控制参数,然后根据调整后步长求得函数的粗糙解,在此基础上再使用初始步长求得全局最优解.实验表明,该算法不仅可以跳出局部最优解的限制,并且减少了迭代次数,有效地提高了全局和局部搜索能力.

关键词: 函数复杂度; 模拟退火算法; 禁忌搜索算法; 函数优化
中图分类号: TP301.6 **文献标识码:** A **文章编号:** 0372-2112 (2012) 06-1218-05
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.06.025

Adaptive Simulated Annealing Algorithm and Tabu Search Algorithm Based on the Function Complexity

XU Peng-fei[1], MIAO Qi-guang[1], LI Wei-sheng[2], ZHANG Jun-ying[1]
(1. *School of Computer, Xidian University, Xi'an, Shaanxi* 710071, *China*;
2. *College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing* 400065, *China*)

Abstract: In the process of applying traditional simulated annealing algorithm and tabu search algorithm to solving multi-peak complex functions, the following problems often occur: particles converge to the local optimal solution too fast, the late converge slows down and search ability turns poor. In order to solve these problems, the definition of function complexity is proposed, and adaptive simulated annealing algorithm and tabu search algorithm are presented based on the function complexity. In these algorithms, the step length control parameters are adaptively adjusted according to the function complexity; then rough solution of the function is obtained in terms of regulated step length; finally, and the original step length is used to acquire the global optimal solution. Experiments show that the proposed method cannot only transcend the limits of the local optimal solution, but also reduce iteration of the above algorithms, and efficiently improve local and global search ability.

Key words: function complexity; simulated annealing algorithm; tabu search algorithm; function optimization

1 引言

模拟退火算法[1,2](Simulated Annealing, SA)最早由 Kirkpatrick 等应用于组合优化领域.标准 SA 算法[3,4]由某一高温度开始,利用具有概率突变特性的 Metropolis 抽样策略,在解空间随机搜索,随着温度的不断下降,重复抽样过程,最终得到问题的全局最优解.但是 SA 算法对整个搜索空间的状况了解不多,不能使搜索过程进入最有希望的搜索区域,容易陷入局部最优.

禁忌搜索[5]算法(Tabu Search, TS)是 Glover 提出的一种启发式全局搜索算法.在标准 TS 算法[6,7]中,邻域、禁忌表、禁忌长度、特赦规则、终止规则等成为算法设计的关键.算法引入灵活的存储结构和相应的禁忌准则以避免重复搜索.但同时存在以下两个问题:(1)算法容易陷入局部最优解;(2)算法收敛速度比较慢.

目前对上述算法的改进[8,9]基本都是针对算法本身,几乎没有考虑到目标函数的复杂情况对算法有效性的影响.本文针对 SA 和 TS 算法在求解多峰复杂函数中存在的问题,提出基于函数复杂度的自适应模拟退火和禁忌搜索新算法,新算法根据函数复杂度自适应调整步长控制参数,既可以使算法求得全局最优解,又能保证最优解的精度.

收稿日期:2011-07-03;修回日期:2011-11-23
基金项目:国家自然科学基金项目(No.61072109, No.61142011);中央高校基本科研业务费专项资金(2012 年度);西安市科技局计划项目(No.CXY1133(1), No.CXY1119(6))

2 函数复杂度和函数优化算法

2.1 函数复杂度

传统 SA 和 TS 算法存在很多不足,相应的改进算法[10,11]相继被提出,这些方法主要是在新解产生的步骤中加入随机扰动或扩大局部搜索范围[12~17].但是改进算法中步长控制参数的调整没有可靠的依据,遇到较复杂的函数时,容易陷入局部最优.为解决这一问题,本文提出函数复杂度概念的定义.

定义 1 函数复杂度是指一个函数的函数值和波峰波谷变化的剧烈程度,用函数值跨越大小和函数波峰波谷变化频率来表示.其表达式可以用式(1)表达:

$$\rho = \log_{10}(f - g) + (m + n) \tag{1}$$

其中 ρ 是函数复杂度,f 是最大函数值,g 是最小函数值,m 是函数波峰数,n 是函数波谷数.$(f - g)$较大时,传统算法容易提前收敛,$(f - g)$经常处于较高数量级,需使用对数降低数量级的影响;$(m + n)$越大时,传统算法搜索过的局部最优点越多,陷入局部最优的可能性就越大.

2.2 基于函数复杂度的算法步长自适应调整

SA 和 TS 算法在函数优化的应用中一般选用较小的步长控制参数,目的是求取更精确的全局最优解.但同时容易陷入局部最优或提前收敛.

由大量实验发现函数自身的复杂情况对算法相关参数,特别是步长控制参数的选择有较大的影响.据此,提出基于函数复杂度进行自适应调整步长控制参数的方法,以避免算法陷入局部最优或提前收敛.新步长控制参数可以用式(2)确定.

$$l' = \rho l \tag{2}$$

其中 l' 是调整后的步长控制参数,ρ 是函数复杂度,l 是 SA 或 TS 算法的步长控制参数初始值.

2.3 步长控制参数和解的精确度的权衡

根据函数复杂度自适应调整步长控制参数可以使算法跳出局部最优解,避免提前收敛,但随着步长控制参数的增大,求得的函数最优值的准确性也随之下降.为此,本文算法先根据函数复杂度自适应调整步长控制参数 l',基于 l' 应用 SA 或 TS 算法求得函数全局最优解所在的区域;然后以“初步最优解”为初始解,基于初始步长控制参数 l 应用相应算法(SA 或 TS)求得高精度的函数全局最优解.此方法既可以使算法避免局部最优和提前收敛的情况出现,又能保证解的精度,而且算法能够较快找到“初步最优解”,整体上提高了算法的收敛速度.

2.4 函数优化算法步骤

(1)在目标函数的定义域内随机产生一个点群.计算点群对应的目标函数的最大值 f 和最小值 g 之差.

(2)根据目标函数的等高线估计函数波峰数 m 和波谷数 n;

(3)应用式(1)计算目标函数的函数复杂度 ρ;

(4)基于 ρ,根据式(2)自适应调整相应算法步长控制参数,得到 l';此处步长控制参数 l' 的自适应调整指的是针对不同函数进行的自适应调整,而不是针对单个函数在寻优过程中自适应调整步长;

(5)基于 l' 采用相应的算法求得目标函数的初步最优解;

(6)基于初始步长控制参数 l,以“初步最优解”为初始解,使用相应算法对目标函数进行寻优,最终得到全局最优解.

3 仿真实验与分析

根据上述算法,选用 5 个目标函数(见表 1)分别对基于函数复杂度的 SA 算法(SAF)和基于函数复杂度的 TS 算法(TSF)的有效性进行验证.其中函数 1 是较简单函数,只需要使用初始步长控制参数就可以求得全局最优解.函数 2~5 相对于函数 1 复杂度逐渐增大.对比实验算法选用 SA 算法[6,7](SA)、TS 算法[8,9](TS)、模拟退火遗传算法[15](SA-GA)和改进的 TS 算法[17](ITS).具体的实验数据及其分析如下:

3.1 仿真实验

实验 1

实验 1 是对算法 SAF 有效性的验证,与 SA 和 SA-GA 算法进行对比.表 2 是目标函数的函数复杂度以及由式(2)确定的新步长控制参数,表 3 是 SAF 和 SA-GA 算法的实验数据.

表 1 函数列表

序号	函数	定义域	最小值	精度
1	$f = 0.5 \times x(:)' \times P1 \times x(:)$	$P1 = [1\ 0; 0\ 2]$, $x(1) \in [-2,2]$, $x(2) \in [-2,2]$	0.0000	1e-05
2	$f = (1.5 - x(1) \times (1 - x(2)))^2 + (2.25 - x(1) \times (1 - x(2)^2))^2 + (2.625 - x(1) \times (1 - x(2)^3))^2$	$x(1) \in [-3,4]$, $x(2) \in [-1,1.5]$	0.0000	1e-05
3	$f = 100 \times (x(2) - x(1)^3)^2 + (x(1) - 1)^2$	$x(1) \in [-2,2]$, $x(2) \in [-2,2]$	0.0000	1e-05
4	$f = 0.5 \times \max(\mathrm{abs}(x)) + \min(\mathrm{abs}(\mathrm{round}(x) - x))$	$x(1) \in [-2,2]$, $x(2) \in [-2,2]$	0.0000	1e-05
5	$f = \mathrm{sum}(\mathrm{abs}(x).^{0.8} + 5 \times \sin(x.^3))$	$x(1) \in [-2,2]$, $x(2) \in [-2,2]$	−7.7515	1e-05

表 2 函数复杂度及 SAF 算法中步长控制参数

函数	$f-g$	$m+n$	$\log_{10}(f-g)$	函数复杂度	步长控制参数				
					初始步长 0.01	0.03	0.05	0.17	0.2
1	5.9975	1	0	1	√				
2	2.12e+02	1	2	3		√			
3	1.00e+04	1	4	5			√		
4	1.1837	17	0	17				√	
5	21.0486	19	1	20					√

表 3 SAF 和 SA-GA 算法的实验比较

		SAF						SA-GA	
函数	实验次数	初始解	粗糙解	粗糙函数值	最优解	最优函数值	迭代次数	最优函数值	迭代次数
1	1	(−0.7824, −0.1977)			(0.0033,0.0039)	1.10e-07	78	5.56e-06	102
	2	(1.2396, −0.5571)			(0.0040, −4.91e-005)	7.21e-07	67	1.77e-05	102
	3	(0.1333,1.3017)			(8.07e-005,0.0043)	1.89e-07	73	3.52e-07	103
	4	(−1.7242,1.4215)			(0.0031,0.0014)	2.16e-07	99	1.36e-05	103
2	1	(0.6664, −0.7762)	(3.0088,0.4859)	3.61e-05	(3.0093,0.5133)	8.07e-07	54	1.58e-05	104
	2	(−0.1812,0.9048)	(3.0124,0.4951)	1.03e-06	(2.9967,0.5107)	7.60e-07	52	1.24e-06	102
	3	(3.0523, −0.9320)	(3.0058,0.4955)	1.89e-05	(3.0038,0.4560)	8.47e-06	73	8.67e-05	105
	4	(−1.591, −0.0336)	(3.9841,0.4939)	1.08e-05	(2.9982,0.5037)	2.47e-07	68	6.75e-05	110
3	1	(0.3780, −1.3356)	(0.9740,0.9263)	6.60e-04	(0.9966,1.0284)	5.39e-05	83	1.83e-05	104
	2	(0.7352, −1.4756)	(0.9629,0.9241)	4.00e-04	(0.9973,0.9692)	7.07e-05	55	1.02e-05	102
	3	(−1.4273,1.1198)	(0.9878,0.9794)	1.54e-04	(0.9863,0.9882)	1.63e-05	92	1.74e-06	109
	4	(1.3517, −0.7909)	(1.0096,0.9740)	7.27e-04	(1.0055,1.0129)	5.00e-07	45	9.74e-05	426
4	1	(−1.4732, −1.4806)	(0.0426, −0.0033)	0.0035	(0.0052, −0.0011)	0.0003	19	0.0002	150
	2	(1.4392, −1.8179)	(0.0011, −0.0293)	0.0036	(0.0003, −0.0038)	0.0003	57	0.0003	150
	3	(1.5891,1.4655)	(−0.0411, −0.0697)	0.0007	(−0.0042,0.0011)	0.0006	12	0.0002	139
	4	(−1.6974,1.6227)	(−0.0205, −0.0280)	0.0031	(−0.0027,0.0015)	0.0005	43	0.0001	137
5	1	(−0.9554,0.8563)	(−1.1675, −1.2254)	−7.7503	(−1.151, −1.1524)	−7.7515	57	−7.7515	107
	2	(0.6232, −1.2418)	(−1.1135,1.0719)	−7.7509	(−1.1537, −1.1529)	−7.7515	30	−7.7514	110
	3	(0.3154, −0.2420)	(−1.1553, −1.1969)	−7.75	(−1.1548, −1.1569)	−7.7515	60	−7.7515	108
	4	(0.4872,0.9317)	(−1.2028, −1.1483)	−7.7485	(−1.1505, −1.1476)	−7.7515	66	−7.7515	105

SA 算法只能对较简单的函数(如第 1 个函数)找到全局最优解,当函数的复杂度逐渐增大时(如第 4、5 个函数),算法就容易陷入局部最优解或提前收敛;从表 2 和表 3 的数据可以看出,SAF 和 SA-GA 算法对 5 个函数都可以找到全局最优解;并且最优解的精度相差不大,但是由于 SA-GA 算法中 GA 算法的运算效率不高和计算量大的缺点,使得 SA-GA 算法的迭代次数明显要大于 SAF 算法,而且 SA-GA 算法在 GA 过程中计算了函数定义域内的大量坐标点,计算量远远大于 SAF 算法.

实验 2

实验 2 是对 TSF 算法有效性的验证,与 TS 和 ITS 进行对比.表 4 是新步长控制参数.表 5 是 TSF 和 ITS 算法的实验数据.

从表 4 和表 5 的数据可以看出,TSF 和 ITS 算法对 5 个函数都可以找到全局最优解;但是 TSF 算法比 ITS 算法求得的解更准确,而且 ITS 算法在求解过程中的迭代次数明显要大于 TSF 算法.

通过对实验数据的分析,发现仿真对比表 3 和表 5 中本文新算法对函数 4 优化结果的精度不如 SA-GA 算法和 ITS 算法.其主要原因在于算法迭代次数的不足导致得到的最优解没有达到较高的精度,为解决这一问题,可以增加算法的迭代次数.实验数据如下:

从实验数据可知,迭代次数的增加主要在算法求取函数精确解的步骤中,虽然迭代次数在原有基础上明显增加,但是整个算法在达到甚至超过 SA-GA 算法和 ITS 算法得到解的精度时,本文算法的迭代次数仍然比 SA-GA 和 ITS 算法少,其主要原因是本文算法使用基于函数复杂度调整后的步长能够快速找到初步最优解,较大程度上减少了迭代次数.

表 4 函数复杂度及 TSF 算法中步长控制参数

函数	$f-g$	$m+n$	$\log_{10}(f-g)$	函数复杂度	初始步长 0.3	0.9	1.5	5.1	6
1	5.9975	1	0	1	√				
2	2.12e+02	1	2	3		√			
3	1.00e+04	1	4	5			√		
4	1.1837	17	0	17				√	
5	21.0486	19	1	20					√

表 5　TSF 和 ITS 算法的实验比较

TSF								ITS	
函数	实验次数	初始解	粗糙解	粗糙函数值	最优解	最优函数值	迭代次数	最优函数值	迭代次数
1	1	(1.6001,-1.8682)			(0.0072,0.00090)	2.70e-05	31	0.0004	77
	2	(1.8513,1.8309)			2.59e-005,-6.56e-006	3.80e-10	9	4.16e-05	75
	3	(-1.8673,1.8762)			(-2.457e-005,-6.06e-006)	3.37e-10	11	2.60e-05	73
	4	(-1.9099,-1.8598)			(-2.85e-005,7.33e-006)	4.61e-10	16	0.00012	70
2	1	(-2.3457,-0.4481)	(3.0733,0.5199)	0.0009	(2.9775,0.49406)	8.54e-05	27	7.38e-05	69
	2	(2.3710,1.3295)	(2.8944,0.4705)	0.0021	(3.0041,0.499)	9.63e-05	15	6.90e-05	78
	3	(-0.6398,0.8166)	(-0.2113,-0.2761)	0.0003	(3.0237,0.5054)	9.16e-05	13	0.0003	85
	4	(3.3315,1.3490)	(2.9746,0.4761)	0.0068	(2.9843,0.4965)	4.51e-05	38	8.47e-05	79
3	1	(-1.3699,-1.5598)	(0.9298,0.8006)	0.0060	(0.9818,0.9463)	0.0003	13	0.0190	75
	2	(-0.0598,-1.0871)	(1.0312,-1.0953)	0.0011	(1.0006,1.0039)	0.0004	14	3.8610	69
	3	(-1.3865,1.458)	(0.8346,0.5584)	0.0825	(0.9826,0.9467)	0.0008	21	0.1034	75
	4	(1.6324,-1.3511)	(0.9213,0.7812)	0.0063	(0.9716,0.9174)	0.0008	18	0.0056	74
4	1	(-1.5569,-1.6134)	(-0.1939,-0.1333)	0.2302	(0.0894,0.0216)	0.0663	8	0.0042	70
	2	(1.4659,-1.5682)	(-0.0091,0.0308)	0.0245	(0.0706,0.0646)	0.0999	12	0.0019	70
	3	(1.5900,1.5141)	(0.3932,0.0993)	0.2959	(-0.0510,-0.0678)	0.0844	9	0.0054	72
	4	(-1.4791,1.4798)	(0.0057,-0.2807)	0.1972	(-0.0145,-0.0611)	0.04502	9	0.0992	68
5	1	(-1.6063,0.1980)	(-1.1741,-1.0775)	-7.5421	(-1.1538,-1.1537)	-7.7514	7	-7.7501	73
	2	(-0.2906,0.9866)	(-11.0734,-1.0677)	-7.3022	(-1.1527,-1.1527)	-7.7515	5	-7.7498	74
	3	(0.9675,0.4892)	(-1.0324,-1.1811)	-7.2738	(-1.1527,-1.1527)	-7.7515	9	-7.2039	71
	4	(1.1945,-1.1383)	(-1.1062,-1.0816)	-7.3326	(-1.1522,-1.1531)	-7.7515	7	-7.1326	70

表 6　增加迭代次数后 SAF 和 SA-GA 算法的实验比较

SAF								SA-GA	
函数	实验次数	初始解	粗糙解	粗糙函数值	最优解	最优函数值	迭代次数	最优函数值	迭代次数
4	1	(1.1757,1.093)	(-0.0010,0.0524)	0.0076	(0.0005,-0.0020)	3.47e-05	14+68=82	0.0002	150
	2	(-1.1052,-1.0486)	(0.0682,0.0627)	0.0222	(-0.0019,-0.0047)	6.47e-06	16+86=92	0.0003	150
	3	(1.1814,-0.5617)	(0.08899,-0.0405)	0.0110	(-0.0013,-0.0005)	8.38e-05	14+50=64	0.0002	139
	4	(-0.7447,0.9409)	(-0.0308,-0.0562)	0.0060	(0.0004,0.0034)	2.89e-05	17+69=86	0.0001	137

表 7　增加迭代次数后 TSF 和 ITS 算法的实验比较

TSF								ITS	
函数	实验次数	初始解	粗糙解	粗糙函数值	最优解	最优函数值	迭代次数	最优函数值	迭代次数
4	1	(1.1350,-1.2158)	(-0.9869,-0.3961)	0.0417	(-0.0027,-0.0094)	0.0003	7+11=18	0.0042	70
	2	(-1.2744,1.2242)	(-0.9487,-1.0265)	0.1666	(-0.0430,-0.0140)	0.0007	6+21=27	0.0019	70
	3	(1.3470,1.5774)	(-0.5971,-0.0635)	0.0145	(-0.0037,-0.0358)	0.0005	7+32=39	0.0054	72
	4	(-1.2616,-1.3473)	(-1.0120,-0.8090)	0.1443	(-0.0144,0.0439)	0.0006	5+25=30	0.0992	68

3.2　实验分析

对于较简单函数,SA 和 TS 算法可以找到函数的全局最优解,对于较复杂函数,标准算法容易陷入局部最优解或提前收敛.已有改进算法在一定程度上使标准算法能够跳出局部最优解,但这是以算法的计算量、迭代次数和解的精度为代价.本文提出的改进算法,充分考虑目标函数自身的复杂情况对步长控制参数的影响,根据函数复杂度自适应调整步长控制参数,增强算法的全局和局部搜索性能,提高算法求解的速度,减少计算量和迭代次数,并保证求解精度.

4　总结

本文给出了函数复杂度的定义,并提出基于此的 SAF 和 TSF 新算法,新算法具有较好的全局和局部搜索能力,避免早熟收敛和陷入局部最优解.实验数据表明,SAF 和 TSF 算法在求解复杂程度不同的函数时,特别是在多峰复杂函数的求解过程中,能够保证求解的精度,并且大幅度提高了算法的求解速度,减少了算法的计算量.但是函数复杂度这一概念目前还处于初步研究阶段,可能函数的某些其他因素同样在不同程度上影响着函数复杂度,有关函数复杂度这一概念在一

定程度上可能还不是很完善,这个问题也是后期进一步要研究的内容.

参考文献

[1] 周明,孙树栋.遗传算法原理及应用[M].北京:国防工业出版,2005.125－127.

[2] 张火明,陆慧娟,等.混合离散变量模拟退火方法及其应用[J].中国计量学院学报,2006,17(1):44－49.

ZHANG Huoming, LU Huijuan, et al. Improvement and application of annealing simulation method for hybrid discrete variables[J]. Journal of China Jiliang University, 2006, 17(1): 44－49. (in Chinese)

[3] 谢云.模拟退火算法综述[J].计算机应用研究,1998(5):7－9.

[4] 康立山,谢云.非数值并行计算——模拟退火算法[M].北京:科学出版社,1998.22－55

[5] GLOVER F. Tabu search: Part Ⅱ[J]. ORSA Journal on Computing, 1990, 2(1): 4－32.

[6] GLOVER F. Tabu search: Part I[J]. ORSA Journal on Computing, 1990, 1(3): 190－206.

[7] BATTITI R, TECCHIOLLI G. The reactive tabu search[J]. ORSA Journal on Computing, 1994, 6(2): 126－140.

[8] 宋炜,刘强.基于模拟退火算法的过程挖掘研究[J].电子学报,2008,36(4A):35－139.

SONG Wei, LIU Qiang. Business process mining based on simulated annealing[J]. Acta Electronica Sinica, 2008, 36(4A): 35－139. (in Chinese)

[9] 李铁,王航宇,等.目标表面 BRDF 统计建模中的遗传模拟退火算法[J].量子电子学报,2008,25(4):489－492.

LI Tie, WANG Hangyu, et al. Application of genetic simulated annealing algorithm in BRDF statistical modeling[J]. Chinese Journal of Quantum Electronics, 2008, 25(4): 489－492. (in Chinese)

[10] 雍正正,罗萍,等.一种进化模拟退火矢量量化图像编码新算法[J].电子学报,2001,29(5):1－4.

YONG Zhengzheng, LUO Ping, et al. A new vector quantization image coding algorithm based on evolutionary simulated annealing[J]. Acta Electronica Sinica, 2001, 29(5): 1－4. (in Chinese)

[11] 张昊,陶然,等.基于 KNN 算法及禁忌搜索算法的特征选择方法在入侵检测中的应用研究[J].电子学报,2009,37(7):1628－1632.

ZHANG Hao, TAO Ran, et al. A research and application of feature selection based on KNN and tabu search algorithm in the intrusion detection[J]. Acta Electronica Sinica, 2009, 37(7): 1628－1632. (in Chinese)

[12] 蓝海,王雄,等.复杂函数全局最优化的改进遗传退火算法[J].清华大学学报(自然科学版),2002,42(9):1237－1240.

LAN Hai, WANG Xiong, et al. Improved genetic-annealing algorithm for global optimization of complex functions[J]. J Tsinghua Univ (Sic &. Tech), 2002, 42(9): 1237－1240. (in Chinese)

[13] 王登刚,刘迎曦,等.最优化问题全局寻优的混合遗传算法[J].力学学报,2002,34(3):0471－0473.

[14] 陈龙,朱宏.改进的模拟退火遗传算法在函数优化中的应用[J].计算机与数学工程,2010,38(7):13－16.

CHEN Long, ZHU Hong. Application of improved simulated annealing genetic algorithm in function optimization[J]. Computer & Digital Engineering, 2010, 38(7): 13－16. (in Chinese)

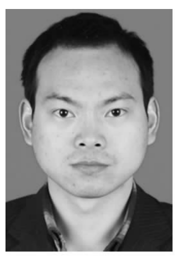
[15] 石运序,范红梅.改进的遗传模拟退火算法在多峰值函数优化中的应用[J].烟台大学学报(自然科学与工程版),2008,21(3):209－212.

[16] Young-Jae Jeon, Jae-Chul Kin. Application of simulated annealing and tabu search for loss minimization in distribution systems[J]. Electrical Power and Energy Systems, 2004, 26(1): 9－18.

[17] 张晓菲,张火明.基于连续函数优化的禁忌搜索算法[J].中国计量学院学报,2010,21(3):251－256.

ZHANG Xiaofei, ZHANG Huoming. Improved tabu search algorithm for continuous problems[J]. Journal of China University of Metrology, 2010, 21(3): 251－256. (in Chinese)

作者简介



许鹏飞 男,1987 年出生于安徽省巢湖市,2009 获西安电子科技大学管理学学士学位,2009 年秋进入西安电子科技大学计算机学院攻读硕士研究生,2010 年秋进入西安电子科技大学计算机学院攻读博士学位.现主要研究方向是模式识别,数字图像处理.

E-mail: xpf1987071500@126.com

苗启广 男,1972 年生于山东青岛,博士,西安电子科技大学计算机学院教授,博士生导师;CCF 高级会员.主要从事智能图像处理与模式识别研究.主持和参与国家自然科学基金、省自然科学基金、国防预研重点项目等 10 项;2011、2008 年分别获西安电子科技大学"十佳师德标兵"称号;在国内外重要学术期刊、国际会议上发表论文 50 余篇.

E-mail: qgmiao@mail.xidian.edu.cn