图的 Steiner 最小树问题的混合遗传算法

赵礼峰,王小龙
(南京邮电大学 理学院,江苏 南京 210023)

摘 要:图的 Steiner 最小树问题是经典的组合优化问题,在通信网络和电路设计中有广泛应用。文中在遗传算法的基础上,对交叉率 p_c 和变异率 p_m 采用自适应过程,构造一种新的确定 p_c 和 p_m 的公式,有效解决了参数选取对最终结果的影响问题。再与模拟退火算法相结合,提出了一种解决 Steiner 最小树问题的混合遗传算法。该算法克服了遗传算法易早熟和收敛性能差的缺点,有效地增强了算法的进化能力。通过对 OR-Library 的部分实例进行计算结果表明,在大多数情况下混合遗传算法比遗传算法有更好的性能。

关键词:Steiner 最小树;遗传算法;自适应;混合遗传算法

中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2014)10-0110-05

doi:10.3969/j.issn.1673-629X.2014.10.026

Hybrid Genetic Algorithm of Graphical Steiner Tree Problem

ZHAO Li-feng,WANG Xiao-long
(College of Science,Nanjing University of Posts and Telecommunications,Nanjing 210023,China)

Abstract:The Graphical Steiner tree Problem (GSP) is a classical combinatorial optimization problem,which has been widely used in communication network and the circuit design. In this paper,on the basis of genetic algorithm,apply the adaptive process for the crossover rate and mutation rate,to construct a new formula determining the crossover rate and mutation rate,effectively solving the problem of parameters selection on the final result. Combined with the simulated annealing algorithm,propose a hybrid genetic algorithm to solve the problem of minimum Steiner tree. The algorithm overcomes the faults of genetic algorithm is easy to premature and poor convergence performance,effectively enhancing the capacity of the evolution of the algorithm. By the OR-Library test case calculation results show that,in most cases,the hybrid genetic algorithm has better performance than genetic algorithm.

Key words:Steiner minimal tree problem;genetic algorithm;self-adaption;hybrid genetic algorithm

0 引 言

图的 Steiner 最小树问题(GSP)是求图中连接给定顶点最小树长的问题,最早是在 1971 年由 Hakimi 提出[1],1972 年 Karp 证明了图的 Steiner 最小树在一般情况下是 NP-完全问题[2],所以不存在多项式时间的最优解。Steiner 最小树能够解决多播最佳路由选择问题,在通信网络和电路设计中有广泛的应用。

目前有许多解决 Steiner 最小树问题的精确解法,但是这些精确算法计算量随着问题规模的增大而呈指数级增长,所以并不适用。启发式算法并不能保证给出最优解,但是其优点在于算法的实现较简单,算法的复杂度不高。有许多求解 Steiner 最小树问题的启发式算法[3-4],其中比较著名的算法有 KMB 算法、MPH 算法、ADH 算法等。

近年来,智能优化算法[5]得到快速发展,由于其在求解 NP 难问题时具有实用性、通用性、灵活性、高效性等特点,因此许多学者用这些算法求解 Steiner 最小树问题。如遗传算法(GA)[6]、蚁群算法[7]、禁忌算法[8]。但是对于每一种具体的算法,其又有自身的劣势。如遗传算法局部搜索能力差,容易出现“早熟”现象。蚁群算法一般需要较长的搜索时间,并且容易出现停滞现象。禁忌算法对初始解的依赖性较强,并且是串行算法不能进行并行搜索。

文中首先根据 Steiner 最小树的数学性质[9],对图中的非正则点进行分类。找出那些肯定在最后所求树中的点,排除不在 Steiner 最小树中的点,剩下的非正

收稿日期:2013-12-04 **修回日期**:2014-03-13 **网络出版时间**:2014-07-28

基金项目:国家自然科学基金资助项目(61070234,61071167)

作者简介:赵礼峰(1959-),男,教授,硕士研究生导师,研究方向为图论及其在通信中的应用;王小龙(1989-),男,硕士研究生,研究方向为图论及其在通信中的应用。

网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20140728.1226.032.html

则点作为候选点。把求解 Steiner 最小树问题转化为寻找 Steiner 点的组合优化问题。给出了一种确定交叉率 p_c 和变异率 p_m 的公式,结合遗传算法和模拟退火算法各自的优点,提出了一种解决该问题的混合遗传算法(GSA)。

1　问题的定义与性质

1.1　问题的定义

给定无向连通图 $G(V,E)$,其中 V 为顶点集,E 为边集,定义权值函数 $f:E\rightarrow R_+$,构成网络 $N(G,f)$。给定子集 $P\subseteq V$,要求在网络 $N(G,f)$ 上寻找一棵子树 $T=(Y,U)$, st $P\subseteq Y\subseteq V$, $U\subseteq E$,且 $\sum_{e\in U(T)} f(e)$ 最小。称 T 为图 G 关于 P 的 Steiner 最小树。点集 P 称为原点(或正则点),不属于 P 的点称为非正则点。点集 S 称为 Steiner–点,简称 s–点,其中 $S\subseteq V\backslash P$, $Y=P\cup S$。记 $|V|=n$, $|E|=l$, $|P|=m$, $r=m-n$。

1.2　性　质

为了降低所求问题的规模以及群体初始化的需要,下面讨论一些关于图中正则点与非正则点的数学性质。

性质 1:度为 1 的非正则点,必定不在 Steiner 最小树中。

证明:(反证法)假设图 G 关于点集 P 的 Steiner 最小树为 T,且 T 中包含度为 1 的非正则点,不妨设其中一个为 v_3,在 T 中与 v_3 相连的边记为 e_3。由于 v_3 的度为 1,所以树 T 去除点 v_3 和边 e_3 后仍为一棵树记为 T',树 T' 包含点集 P。因为 e_3 的权值 $f(e_3)$ 为正值,所以 $\sum_{e'\in U(T')} f(e') < \sum_{e\in U(T)} f(e)$,这与假设 T 为 Steiner 最小树相矛盾。故假设不成立,定理得证。

性质 2:与度为 1 的正则点相邻的顶点,必定在 Steiner 最小树中。

证明:与度为 1 的正则点相关联的边只有一条,且只有一个相邻的顶点,由于 Steiner 最小树 T 包含所有的正则点,所以度为 1 的正则点也在树 T 中。故与度为 1 的正则点所关联的边也在树 T 中,相邻的顶点也在树 T 中,否则不满足树的连通性,定理得证。

性质 3:度为 2 的非正则点 v_2 所关联的顶点为 v_x 和 v_y,若边 (v_x,v_y) 存在,且 $f(v_2,v_x)+f(v_2,v_y)>f(v_x,v_y)$,则非正则点 v_2 必定不在 Steiner 最小树中。

证明:(反证法)假设 Steiner 最小树包含点 v_2,则边 (v_2,v_x) 和 (v_2,v_y) 都在树中。因为如果顶点 v_x 和 v_y 只有一个在树中,则边 (v_2,v_x) 和 (v_2,v_y) 只有一条在树中,此时根据性质 1 知,可以删除 v_2;如果两个顶点都不在树中,则 v_2 也不在树中。因为 $f(v_2,v_x)+f(v_2,v_y)>f(v_x,v_y)$,所以可以删除顶点 v_2 和边 (v_2,v_x),(v_2,v_y) 用更短的边 (v_x,v_y) 代替,此时仍为 Steiner 最小树且总权值更小。与假设矛盾,故 v_2 不在 Steiner 最小树中。

性质 4[10]:如果 Steiner 最小树存在,则 Steiner 点的数目最多不超过正则点的数目减 2,即 $|S|\leqslant \min(m-2,r)$。

2　遗传算法和 KMB 算法

2.1　基本遗传算法

遗传算法(Genetic Algorithm,GA)是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化概率搜索算法[11]。基本遗传算法的实现流程为:首先,随机产生初始种群,并设定交叉率、变异率、终止条件等参数;然后计算种群中每个个体的适应度值,根据适应度值进行选择、交叉、变异等操作,产生新一代种群,判断是否满足终止条件,如果满足则结束进化过程,并输出当前最优个体,如果不满足则从计算适应度值开始,继续进行种群进化,直到满足终止条件为止。

遗传算法在进行种群进化的过程中,选择和交叉的作用是使适应度值较高的个体得到保留,并且数量随着进化而不断增加,最后可能在种群中出现大量的“近亲个体”,产生“近亲繁殖”,使进化停止过早收敛于局部值。而变异操作可以增大搜索空间,扩大搜索范围,有利于找到全局最优解,但是会影响收敛的速度。

2.2　遗传算法改进的必要性

一方面由于遗传算法的交叉率 p_c 和变异率 p_m 都是固定不变的,当 p_c 设置过大时,新个体产生的速度就越快,而具有较高适应度的个体结构被破坏的概率也越大。但如果 p_c 设置过小时,搜索过程会变得非常缓慢,以至于停滞不前。当 p_m 设置过大时,遗传算法就会变成随机搜索算法。而当 p_m 设置过小时,又不容易产生新的个体结构。因此遗传算法中参数如何选取才能得到最优结果一直是很难解决的问题。另一方面,遗传算法虽然有较强的全局搜索能力,但是局部搜索能力较差,并且容易产生早熟现象。综合上述存在的问题,对算法的改进是有必要的。

2.3　KMB 算法

由于 KMB 算法[12]可以把所选择的节点都添加到 Steiner 树中,不会产生不可行解,在进行评价时不需要加惩罚函数,避免了对不可行解进行考虑所带来的问题。并且算法简便,易于实现,因此文中选用 KMB 算法来计算个体的目标值。KMB 启发式算法的步骤如下:

Step1:构造图 G 的完全距离图 $G'=(V',E')$,其中

$\forall (u,v) \in E'$ 是图 G 中节点 u 到节点 v 的最短路径;

Step2:在 G' 中求包含 W 的子图 G_1;

Step3:求 G_1 的最小生成树 T_1;

Step4:把 T_1 中的每一条边,依据 G 中的最短路扩充为 G 的一个子图 G_2;

Step5:求 G_2 的最小生成树 T_2;

Step6:删除 T_2 中非正则点的叶子节点,最后得到所求的树为 T_{DNH}。

3 求解 GSP 的混合遗传算法

3.1 改进策略

为了克服遗传算法中 p_c 和 p_m 不容易确定的缺点,对 p_c 和 p_m 采用自适应过程[13],构造了一种新的确定 p_c 和 p_m 的公式。由于模拟退火算法具有较强的局部搜索能力,能使搜索过程避免陷入局部最优解,因此将遗传算法与模拟退火算法相结合,能够有效地克服遗传算法局部搜索能力较差的缺点,同时还能发挥遗传算法并行搜索能力强的优势。

3.1.1 交叉率 p_c 和变异率 p_m 的确定

根据交叉操作和变异操作在群体进化过程中的作用,设计了如下确定 p_c 和 p_m 的规则:

(1)当个体适应度值低于平均适应度值时,采用较高的交叉率和变异率,使个体有更大的概率进化成优良个体;

(2)当个体适应度值高于平均适应度值时,采用较低的交叉率和变异率,这样保证了优良个体不容易被改变结构;

(3)群体中最大适应度值个体的交叉率和变异率不为零,这样可以使种群中表现优良的个体不会处于一种近似于停滞的状态。

根据上述规则给出了确定 p_c 和 p_m 的公式:

$$p_c = \begin{cases} p_{c1} - p_{c2}\dfrac{f' - f_{avg}}{f_{max} - f_{avg}} & f' \geqslant f_{avg} \\ p_{c1} & f' < f_{avg} \end{cases}$$

$$p_{c1} = 0.8, p_{c2} = 0.4$$

$$p_m = \begin{cases} p_{m1} - p_{m2}\dfrac{f' - f_{avg}}{f_{max} - f_{avg}} & f' \geqslant f_{avg} \\ p_{m1} & f' < f_{avg} \end{cases}$$

$$p_{m1} = 0.01, p_{m2} = 0.001$$

其中,f' 表示要交叉的两个个体中较大的适应度值;f_{avg} 表示每代群体平均适应度值;f_{max} 表示群体中最大的适应度值。

交叉算子是采用单点交叉的方式以交叉概率 p_c 进行,变异算子是采用离散变异算子的方法,对个体的每一个基因用变异概率 p_m 进行变异。

3.1.2 对最优个体进行模拟退火操作

对最优个体进行模拟退火操作,可以使最优个体以一定的概率向周围进行搜索,增强局部搜索的能力。在遗传进化过程的每一代中,完成交叉和变异后,找出最优个体 f,对其进行模拟退火操作,按照邻近个体集合生成策略 2-opt 映射产生一个邻近个体 f',f' 以概率 p 取代 f。降温方式采用快速降温策略 $t_k = \dfrac{t_0}{1+k}$,接受概率 p 采用 Metropolis 准则:

$$p = \begin{cases} 1 & E(f) \leqslant E(f') \\ \exp\left(\dfrac{E(f) - E(f')}{t_k}\right) & E(f) > E(f') \end{cases}$$

其中,E 为个体的适应度;t_k 为退火温度;t_0 为初始温度;k 为迭代次数。

3.2 编码方法和群体初始化

编码:对个体基因,采用二进制编码方法,使用的编码符号由二进制符号 0 和 1 组成,它所构成的个体基因型是一个二进制编码符号串。把候选点集中的点按标号从小到大的顺序排列。

解码:对于个体基因,如果某位置的基因为 1,表示候选点集中该位置的点在所要寻找的 Steiner 最小树中;如果为 0,表示候选点集中该位置的点不在寻找的 Steiner 最小树中。

初始化方法:根据性质 1 和性质 3 排除不在 Steiner 最小树中的非正则点,根据性质 2 确定一定在 Steiner 最小树中的点,剩下的点作为候选点集。把候选点集的长度作为染色体的长度,用 L 来表示。M 表示群体中个体的数量,随机产生 M 个长度为 L 的个体,其中的元素为 0 或 1,判断每一个染色体基因型中 1 的个数是否满足性质 4,如果不满足则把相应的染色体剔除,再随机产生一个个体,直到 M 个个体都满足性质 4。

3.3 适应度计算和选择种群

适应度的计算方法:首先对于目标函数 $\min \sum_{e \in U(T)} f(e)$,根据 KMB 算法求出树 T_{DNH},把树 T_{DNH} 的权值作为目标值。其次对目标函数值进行降序排序,把最不适应的个体放置在目标函数值列表的第一个位置,最适应个体放置在最后一个位置上。然后按照公式(1)对目标函数值进行线性排序,选择压差为 2 来估算适应值。

$$\mathrm{FF}(\mathrm{pos}) = 2 - \mathrm{dp} + 2 \times (\mathrm{dp} - 1) \times \frac{\mathrm{pos} - 1}{M - 1} \tag{1}$$

其中,pos 表示目标函数值所在的排序中位置;M 表示种群中个体的数量;dp 表示压差,此处取 dp=2。

选择种群:基于种群中个体的适应度值,来表明每

个个体被选择的预期概率,适应度越高的个体,被选中的概率就越大。采用随机遍历抽样的方法,从当前种群中选择优良个体,并将选择的个体返回到新的种群中,种群中个体的总数量不变。

3.4　重插入子代到种群

把经过交叉、变异的子代和经过模拟退火操作的个体重新插入到父辈中,基于每个个体适应度的大小,确定被选中的概率,对子代进行选择,用子代代替父代中适应度最小的个体,增强种群的整体适应度。

3.5　算法步骤

Step1:根据性质 1 ~4,进行群体初始化;

Step2:对群体中的每一个个体进行适应度计算;

Step3:根据适应度选择种群;

Step4:自适应交叉操作;

Step5:自适应变异操作;

Step6:对最优个体进行模拟退火操作;

Step7:重新插入子代到种群;

Step8:判断是否达到终止条件(使用迭代次数来控制进化算法的结束),如果达到终止条件,输出最优个体的适应值,否则转 Step2。

4　实例测试

为了验证算法的有效性,上述算法使用谢菲尔德(Sheffield)大学推出的基于 Matlab 遗传算法工具箱[14],用 Matlab 在 Win7 平台下进行编程实现[15]。并对 STEINLIB 标准数据集中[16]的部分数据进行实验。为了表述方便引入下述记号:opt 为当前已知的最优解;best 为实验最优解;avrg 为实验平均解;avrgit 为最优解第一次出现的平均代数;avrgtime 为平均运行时间。

对于测试用例 B 中的 18 个测试用例分别用 GA 和 GSA 方法运行 15 次,计算出 best、avrg、avrgtime、avrgit。其中有 16 个测试用例使用 GA 和 GSA 方法得到的 best 和 avrg 都等于 opt,在此不再列出,只列出剩余两个 B_{14} 和 B_{18} 的测试结果,见表 1。对于测试用例 C 和 D 中的部分用例分别用 GA 和 GSA 方法运行 10 次,计算出 best、avrg、avrgtime、avrgit,测试结果见表 1。

图 1 和图 2 给出了 GA 和 GSA 方法平均迭代次数的对比,可以得出,对所有的测试用例,GSA 寻找到最优解所用的平均迭代次数比 GS 要少,说明了 GSA 算法通过加强局部搜索,有效地减少了寻找到最优解所用的迭代次数。图 3 和图 4 给出了 GA 和 GSA 方法平均时间之比,从图中可以看到只有三个点的值是大于 1 的,也就是说只有三个测试用例 GSA 运行的时间比 GA 长。对 90% 左右的测试用例,GSA 运行的时间比 GA 短,说明 GSA 算法显著地提高了搜索的速度。

表 1　实验数据与结果对比

测试用例	图的规模 $n/l/m$	opt	GAavrg	GSAavrg	GAbest	GSAbest
B_{14}	100/125/25	235	235.8	235.2	235	235
B_{18}	100/200/50	218	218.2	218	218	218
C_{01}	500/625/5	85	85	85	85	85
C_{02}	500/625/10	144	144	144	144	144
C_{03}	500/625/83	754	756.6	755.8	754	754
C_{04}	500/625/125	1 079	1 083.2	1 085.6	1 079	1 081
C_{06}	500/1 000/5	55	55	55	55	55
C_{07}	500/1 000/10	102	102	102	102	102
C_{08}	500/1 000/83	509	514.2	512.8	511	509
C_{14}	500/2 500/125	323	328.2	325.2	323	323
D_{01}	1 000/1 250/5	106	106	106	106	106
D_{05}	1 000/1 250/500	3 250	3 360.5	3 325.4	3 287	3 262
D_{07}	1 000/2 000/10	103	108.8	106.2	107	105
D_{13}	1 000/5 000/167	500	518.8	504.2	510	502

由表 1 的测试结果对比可以发现,绝大多数的测试用例,GSA 的最优解和平均最优解都等于或者小于 GA 的最优解和平均最优解,说明 GSA 算法有更强的全局和局部搜索能力,并且搜索的解的稳定性更高。

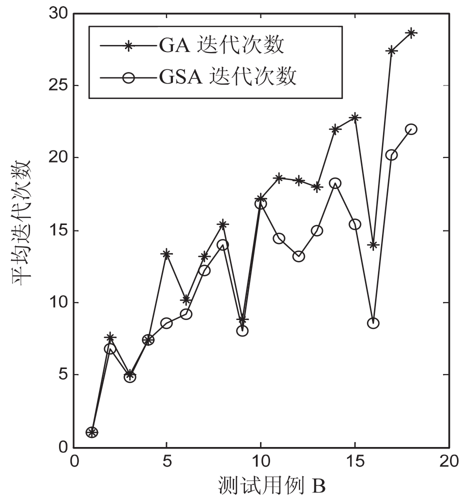

图 1　用例 B 迭代次数比较

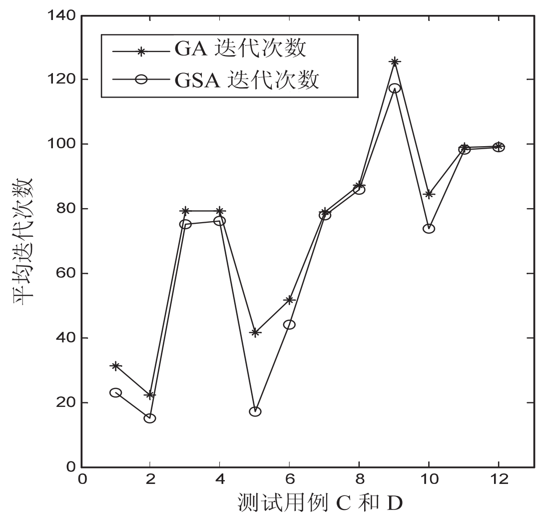

图 2　用例 C 和 D 迭代次数比较

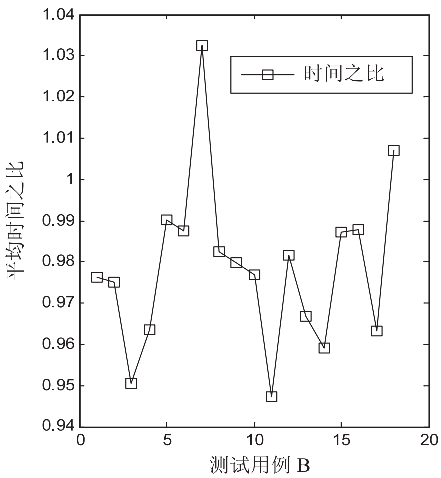

图 3　用例 B 时间比较

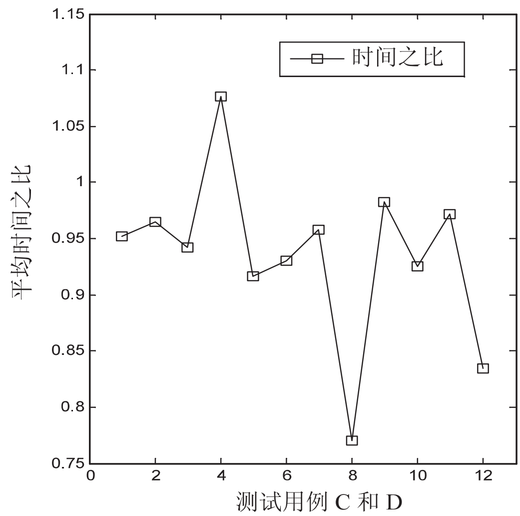

图 4　用例 C 和 D 时间比较

5　结束语

文中通过对基本遗传算法进行改进,在群体初始化时用相关的数学性质排除一些点,减少了候选点的规模,降低了染色体的长度。在个体进行交叉、变异时采用自适应方法,使进化过程可以得到有效的控制,对最优个体进行模拟退火操作,提出了求解 Steiner 最小树的混合遗传算法(GSA)。通过实验数据的对比表明,在大多数情况下 GSA 算法比 GA 算法在搜索速度上更快,得到的最优解更接近于精确解,并且解的稳定性更高。

参考文献:

[1] Hakimi S L. Steiner's problem in graphs and its implications[J]. Networks,1971,1(2):113-133.

[2] Karp R M. Reducibility among combinatorial problems,complexity of computer computations[M]. New York:Plenum Press,1972.

[3] Hwang F K, Richards D S. Steiner tree problems[J]. Networks,1992,22(1):55-89.

[4] Winter P. Steiner problem in networks:a survey[J]. Networks,1987,17(2):129-167.

[5] 梁　旭,黄　明. 现代智能优化混合算法及其应用[M]. 北京:电子工业出版社,2011.

[6] Esbensen H. Computing near-optimal solutions to the Steiner problem in a graph using a genetic algorithm[J]. Networks,1995,26(4):173-185.

[7] 杨文国,郭田德. 求解最小 Steiner 树的蚁群优化算法及其收敛性[J]. 应用数学学报,2006,29(2):352-361.

[8] Ribeiro C C,Souza M C. Tabu search for the Steiner problem in graphs[J]. Networks,2000,36:138-146.

[9] 熊小华,刘艳芳,宁爱兵. 图的 Steiner 最小树的竞争决策算法[J]. 上海理工大学学报,2012,34(5):461-465.

[10] Lawler E L. Combinatorial optimization:networks and matroids[M]. New York:Holt,Rinehart and Winston,1976.

[11] 周　明,孙树栋. 遗传算法原理及应用[M]. 北京:国防工业出版社,1999.

[12] Kou L,Markowsky G,Berman L. A fast algorithm for Steiner trees[J]. Acta Informatica,1981,15(2):141-145.

[13] 栾庆林,卢辉斌. 自适应遗传算法优化神经网络的入侵检测研究[J]. 计算机工程与设计,2008,29(12):3022-3025.

[14] 雷英杰. MATLAB 遗传算法工具箱及应用[M]. 西安:西安电子科技大学出版社,2005.

[15] 王海英. 图论算法及其 MATLAB 实现[M]. 北京:北京航空航天大学出版社,2010.

[16] Beasley J E. OR-library:distributing test problems by electronic mail[EB/OL]. 1989. http://people.brunel.ac.uk/~mastjjb/jeb/orlib/steininfo.html.

(上接第 109 页)

2012:44-50.

[11] 李向阳,李玲娟,陈建新,等. 面向情境感知的不确定性数据融合策略[J]. 计算机技术与发展,2012,22(2):127-130.

[12] 汤效琴,戴汝源,徐　琪. 数据挖掘中变量聚类方法的应用研究[J]. 计算机工程与应用,2004,40(24):171-173.

[13] Chen A. Context-aware collaborative filtering system:predicting the user's preferences in ubiquitous computing environment[C]//Proc of the LoCA 2005. [s. l.]:[s. n.],2005:244-253.

[14] 徐风苓,孟祥武,王立才. 基于移动用户上下文相似度的协同过滤推荐算法[J]. 电子与信息学报,2011,33(11):2785-2789.

图的Steiner最小树问题的混合遗传算法

作者： 赵礼峰， 王小龙， ZHAO Li-feng， WANG Xiao-long
作者单位： 南京邮电大学 理学院,江苏 南京,210023
刊名： 计算机技术与发展ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2014(10)