

文章编号: 1001-0920(2011)12-1895-05

基于测点必要度的模拟电路测点优选方法

黄以锋, 景博, 周宏亮

(空军工程大学 工程学院, 西安 710038)

摘要: 针对模拟电路的测点选择问题, 根据测点的必要程度, 提出了“测点必要度”的概念, 同时结合“故障隔离度”的概念, 建立了新的测点优选方法, 给出了新算法的详细计算过程和应用实例. 实验分析结果表明, 新算法的计算效果优于熵算法和基于故障隔离度的测点选择算法, 可用于复杂模拟电路的测点选择.

关键词: 模拟电路; 故障诊断; 故障字典; 测点选择

中图分类号: TP306

文献标识码: A

Test point selection method for analog circuits based on essential degree

HUANG Yi-feng, JING Bo, ZHOU Hong-liang

(Engineering College, Air Force Engineering University, Xi'an 710038, China. Correspondent: HUANG Yi-feng, E-mail: huangyiff@126.com)

Abstract: The problem of selecting the optimum test points of analog circuits is considered. The test point essential degree is proposed to identify how the test point is needed. Then combined with fault isolation degree, a new test point selection method is presented. The detailed calculating process of the new algorithm is given and two examples are proposed. The experiment results show that the new algorithm is better than entropy algorithm and the algorithm based on fault isolation degree, and can be applied to test point selection for complicated analog circuits.

Key words: analog circuit; fault diagnosis; fault dictionary; test point selection

1 引言

由于模拟电路的输入激励和输出响应是连续量, 元件参数存在容差、电路中存在非线性现象和反馈回路、故障模型比较复杂等因素, 模拟电路的故障诊断技术发展较慢, 应用得较少. 故障字典法^[1-3]是具有使用价值的一种模拟电路故障诊断方法, 该方法测点的选择是一个重要环节, 其目的是在故障可隔离的前提下选择测点数量最小的测点集. 由于模拟电路的元件存在容差, 同一状态的测点电压不是一个固定值, 而是在某一范围内变化. 该问题可以通过划分模糊集和构建整数编码故障字典来解决. 整数编码故障字典的测点选择问题是 NP-hard 问题^[5], 全局最优算法不适用于较大规模的系统. 目前, 国内外学者提出了多种局部最优算法. Prasad 等人^[4]提出了 3 种包含法和 1 种排除法. Starzyk 等人^[5]提出了一种基于熵的方法. 汪鹏等人^[6]提出了基于故障隔离度的测点选择方法. 文献 [7-8] 使用现代优化算法求取模拟电路的最小测点集, 但这些算法模型较为复杂, 计算时间较长.

本文对各个故障对的可隔离测点集进行了分析, 提出了“测点必要度”的概念, 同时结合“故障隔离度”的概念, 提出了基于测点必要度的测点优选方法. 该方法综合考虑各个测点的必要度和故障隔离度, 缩小了测点搜索的范围, 提高了效率.

2 预备知识

2.1 模糊集和整数编码故障字典

由于模拟电路中的元件存在容差, 电路在正常状态下和各种故障状态下测点的电压是一个连续的小区间, 这些小区间可能重叠, 而使该测点无法区分重叠的多个状态. 为了解决这一问题, 根据各个状态的测点电压区间将电压分为若干个模糊域. 测点电压区间落入某个模糊域的所有故障状态 (包括正常状态) 组成一个模糊集, 并依次用整数对模糊集进行编码. 对所有测点的模糊集进行划分、编码后, 根据故障状态、测点和模糊集间的关系建立整数编码故障字典. 用 $F = \{f_0, f_1, f_2, \dots, f_m\}$ 表示故障状态集 (包

收稿日期: 2010-06-22; 修回日期: 2011-05-14.

基金项目: 航空科学基金项目(20101996012).

作者简介: 黄以锋(1982-), 男, 博士生, 从事可测性设计、故障诊断的研究; 景博(1965-), 女, 教授, 博士生导师, 从事故障预测与健康管理、可测性设计等研究.

括正常状态), $T = \{t_1, t_2, \dots, t_n\}$ 表示可用测点集, d_{ij} 表示模糊集的整数编码. 其中: i 为对应的故障状态的编号, j 为对应的可用测点.

2.2 故障隔离度

从故障状态集 F 中任意选出两个故障状态 f_i 和 f_j 构成一个故障对 (f_i, f_j) . 设 P_F 为 F 中所有故障状态生成的故障对所构成的集合, 则有 $P_F = \{(f_i, f_j) | 0 \leq i, j \leq m \text{ 且 } i \neq j\}$. 对于可用测点集中的任意测点 t_k , 若故障字典中的整数编码 d_{ik} 和 d_{jk} 不相等, 则表明测点 t_k 可以区分故障对 (f_i, f_j) .

定义 1^[6] 设 $P_F = \{p_{F_1}, p_{F_2}, \dots, p_{F_{m_1}}\}$ 为故障对集, $T = \{t_1, t_2, \dots, t_n\}$ 为可用测点集, 可用测点在故障对集中可区分的故障对数称为该测点的故障隔离度, 用 I 表示. 假设可用测点 t_j 可区分的故障对集分别为 D_j , 则 t_j 的故障隔离度为 D_j 中元素的个数, 即

$$I_j = |D_j|. \quad (1)$$

定义 2^[6] 设 $T_1 = \{t_1, t_2, \dots, t_{n_1}\}$ 为已选的优化测点集, $P_{F_1} = \{p_{F_1}, p_{F_2}, \dots, p_{F_{m_2}}\}$ 为剩余未隔离的故障对集, 则测点 $t_k (t_k \notin T_1)$ 在故障对集 P_{F_1} 中可区分的故障对数称为 t_k 相对于 T_1 的条件故障隔离度, 记为 $I_k(T_1)$.

2.3 测点必要度

从故障状态集 F 中任意选出两个故障状态 f_i 和 f_j 构成一个故障对 (f_i, f_j) . 将可用测点集中所有能隔离该故障对的测点组成一个集合, 称为故障对 (f_i, f_j) 的可隔离测点集. 在故障对集 P_F 中, 每个故障对都有自己的可隔离测点集, 若集合中只有一个测点, 则该测点一定会被选中, 否则故障状态将不能全部隔离. 若可隔离测点集中有两个测点, 则这两个测点各有一半的概率被选中. 所以, 可隔离测点集中元素的个数包含着测点的必要性信息.

定义 3 设 $P_F = \{p_{F_1}, p_{F_2}, \dots, p_{F_{m_1}}\}$ 为故障对集, $T = \{t_1, t_2, \dots, t_n\}$ 为可用测点集, $T_F = \{t_{F_1}, t_{F_2}, \dots, t_{F_{m_1}}\}$ 为与故障对相对应的可隔离测点集的集合. 则测点 t_i 的必要度可表示为

$$B_i = \frac{1}{\min_j |t_{F_j}| |t_i \in t_{F_j}|}, \quad (2)$$

其中 $|t_{F_j}|$ 为集合中的元素的个数.

定义 4 设 $T_1 = \{t_1, t_2, \dots, t_{n_1}\}$ 为已选的优化测点集, $P_{F_1} = \{p_{F_1}, p_{F_2}, \dots, p_{F_{m_2}}\}$ 为剩余未隔离的故障对集, $T_{F_1} = \{t_{F_1}, t_{F_2}, \dots, t_{F_{m_2}}\}$ 为与故障对相对应的可隔离测点集的集合, 则测点 $t_k (t_k \notin T_1)$ 在 P_{F_1} 和 T_{F_1} 条件下的必要度称为 t_k 相对于 T_1 的条件必要度, 记为 $B_k(T_1)$.

3 测点优选算法

3.1 熵算法

文献 [5] 提出了一种基于熵的测点选择方法, 基于测点提供的信息量越大, 越有利于故障隔离的原理, 通过不断选择信息量最大的测点来组成优化的测试集. 算法过程如下: 首先利用所有可用测点将故障状态集划分为多个模糊集, 分别计算各个测点的熵, 选择熵最小的测点; 然后更新已选择测点划分的各个子集的可用测点集和故障状态集; 再计算各个子集, 选择熵最小的测点; 不断循环计算, 直至所有故障状态均被隔离. 熵算法详细计算过程参见文献 [5].

3.2 WP 算法

文献 [6] 提出了一种基于故障隔离度的测点选择算法 (本文简称 WP 算法). 该算法将测点可区分的故障对数定义为故障隔离度, 用故障隔离度和条件故障隔离度来描述测点的故障区分能力, 选择隔离度较大的测点组成优化测点集. 算法过程如下: 首先根据整数编码故障字典生成故障对集 P_F , 计算每个可用测点的故障隔离度, 选择隔离度最大的测点放入优化测点集; 然后更新故障对集和可用测点集; 再次计算每个可用测点的故障隔离度, 选择隔离度最大的测点放入优化测点集; 不断重复计算, 直至故障对集为空集. WP 算法详细计算过程参见文献 [6].

WP 算法通过不断选择可隔离故障对最多的测点来组成优化测点集, 但该优化测点集并不一定是最优的. 假设故障字典中可用测点 t_1, t_2 和 t_3 对应的可区分故障对集分别为 D_1, D_2 和 D_3 , 有

$$D_1 = \{p_{F_1}, p_{F_2}, p_{F_3}, p_{F_4}, p_{F_5}\},$$

$$D_2 = \{p_{F_1}, p_{F_2}, p_{F_6}\},$$

$$D_3 = \{p_{F_3}, p_{F_4}, p_{F_5}, p_{F_7}\}.$$

按照 WP 算法, 先选择测点 t_1 , 剩下可用测点 t_2 和 t_3 , 对应的可区分故障对集分别更新为 $D_2 = \{p_{F_6}\}$ 和 $D_3 = \{p_{F_7}\}$, 最后得到的优化测点集为 $\{t_1, t_2, t_3\}$. 实际上, 测点 t_2 和 t_3 即可隔离所有的故障状态.

3.3 基本算法

在第 3.2 节的例子中, 如果考虑测点的必要度, 则会得到不同的结果. 假设各故障对所对应的可隔离测点集分别为 $t_{F_1}, t_{F_2}, \dots, t_{F_7}$, 分析可得

$$t_{F_1} = \{t_1, t_2\}, t_{F_2} = \{t_1, t_2\}, t_{F_3} = \{t_1, t_3\},$$

$$t_{F_4} = \{t_1, t_3\}, t_{F_5} = \{t_1, t_3\}, t_{F_6} = \{t_2\}, t_{F_7} = \{t_3\}.$$

根据必要度定义, t_2 和 t_3 的必要度均为 1, 表明 t_2 和 t_3 是必选的, 否则故障对 p_{F_6} 和 p_{F_7} 将不能隔离. 所以, 首先将 t_2 和 t_3 加入优化测试集, 经计算可知, t_2 和 t_3 已将故障对全部隔离, 优化测点集为 $\{t_2, t_3\}$. 测

点集中仅有两个测点, 优于 WP 算法的计算结果。

基本算法考虑了测点的必要度和故障隔离度, 优先选择必要度较大的测点, 在必要度相同时选择故障隔离度较大的测点。基本算法的计算过程如下:

Step 1: 根据整数编码故障字典生成故障对集 P_F , 计算出与故障对相对应的可隔离测点集的集合 T_F , 设定优化测点集 T_1 为 $\{\emptyset\}$, 最小测点集中的元素个数 K_a 为 1。

Step 2: 计算 T_F 每个测点集中元素的个数 $K_{t_{F_i}}$, 若某个测点集中元素个数为 0, 则转至 Step 6。

Step 3: 比较是否存在 $K_{t_{F_i}} = K_a$, 若不存在, 则令 $K_a = K_a + 1$, 转至 Step 3; 若存在, 则再判断 K_a 是否为 1, 为 1 则测点 t_i 的必要度为 1, 是必要测点, 将测点 t_i 放入 T_1 ; 不为 1 则计算集合 t_{F_i} 中各测点相对应 T_1 的条件隔离度, 选择条件隔离度最大的测点放入 T_1 ; 若有多个最大测点则选择序号最小的测点。

Step 4: 从可用测点集 T 中删除新选测点, 从 P_F 中删除新选测点能隔离的故障对, 从 T_F 中删除对应的测点集。

Step 5: 判断 P_F 中元素的个数是否为 0, 若为 0, 则计算结束; 否则, 判断 T 中的元素个数是否为 0, 若为 0, 则转至 Step 6, 否则转至 Step 2。

Step 6: 故障对集不可隔离, 计算结束。

在实际计算过程中, 必要度计算较为复杂, 这里通过优先在最小的可隔离测点集中选择优化测点来体现对条件必要度的考虑。

3.4 改进算法

在基本算法中, 虽然考虑了测点的必要度和故障隔离度, 但实验分析表明算法只有在系统存在必要度为 1 的测点, 优势才较为明显。因此, 本文在基本算法的基础上提出了一种改进算法, 改进算法同样优先选择必要度高的测点, 但在必要度相同的情况下, 选择测点的方法不同。改进算法计算过程如下:

Step 1: 根据整数编码故障字典生成故障对集 P_F , 计算出与故障对相对应的可隔离测点集的集合 T_F , 设定优化测点集 T_1 为 $\{\emptyset\}$, 最小测点集中的元素个数 K_a 为 1。

Step 2: 计算 T_F 每个测点集中元素的个数 $K_{t_{F_i}}$, 若某个测点集中元素个数为 0, 则转至 Step 6。

Step 3: 比较是否存在 $K_{t_{F_i}} = K_a$, 若不存在, 则令 $K_a = K_a + 1$, 转至 Step 3; 若存在, 则再判断 K_a 是否为 1, 为 1 则将测点 t_i 放入 T_1 , 不为 1 则进行如下操作:

1) 对于集合 t_{F_i} 中每个测点 (如 t_k), 假设将 t_k 放入 T_1 , 更新 T , P_F 和 T_F (从可用测点集 T 中删除新选

测点, 从 P_F 中删除新选测点能隔离的故障对, 从 T_F 中删除对应的测点集), 用 WP 算法得出优化测试集。

2) 选择集合 t_{F_i} 中优化测试集最小的测点放入 T_1 , 若有多个最大的测点则选择序号最小的测点。

Step 4: 从可用测点集 T 中删除新选测点, 从 P_F 中删除新选测点能隔离的故障对, 从 T_F 中删除对应的测点集。

Step 5: 判断 P_F 中元素的个数是否为 0, 若为 0, 则计算结束; 否则, 判断 T 中的元素个数是否为 0, 若为 0, 则转至 Step 6, 否则转至 Step 2。

Step 6: 故障对集不可隔离, 计算结束。

改进算法同样通过优先在最小的可隔离测点集中选择优化测点来体现对条件必要度的考虑, 同时使用 WP 算法来实现对故障隔离度的考虑, 充分利用了故障隔离度信息。

4 应用实例

4.1 实例 1

图 1 是某飞机燃油管理系统中的一个稳压电路。电路中设定 5 个测试点, 用 Electronics Workbench 软件对电路进行仿真, 得到各种故障状态下每个测点的大概电压值; 然后将电压区间分为若干个模糊域, 并用整数进行编码, 得到电路的整数编码故障字典如表 1 所示。

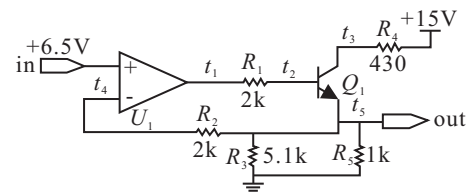


图 1 稳压电路

表 1 稳压电路的整数编码故障字典

故障状态	可用测点				
	t_1	t_2	t_3	t_4	t_5
f_0 (正常)	2	3	2	3	3
f_1 (R_1 开路)	3	0	4	0	0
f_2 (R_2 开路)	3	4	2	0	4
f_3 (R_3 短路)	3	0	0	0	0
f_4 (R_3 开路)	2	3	3	3	3
f_5 (R_4 短路)	2	3	4	3	3
f_6 (R_4 开路)	3	2	1	2	2
f_7 (Q_1b 开路)	3	6	4	0	0
f_8 (Q_1c 开路)	3	2	4	2	2
f_9 (Q_1e 开路)	3	5	4	0	0
f_{10} (U_1P 开路)	1	1	4	1	1
f_{11} (U_1N 开路)	3	4	2	4	4
f_{12} (U_1O 开路)	0	0	4	0	0

用基本算法进行计算, 首先对 P_F , T_F , T_1 和 K_a 进行初始化; 然后计算各测点集中元素的个数; 再进

行 Step 3 (如图 2 所示), 故障对 $\{f_0, f_4\}$ 对应的可隔离测点集中只有一个测点 t_3 , 将其放入 T_1 ; 更新相关参数, 经过 Step 5 后转至 Step 2, 重复计算. 图 2 展示了每个计算周期的计算结果, 最终得到的优化测试集为 $\{t_3, t_2, t_1, t_4\}$. 在实例 1 中, 优化测试集中的每个测点都是必要测点, 改进算法和基本算法的过程基本一致, 结果也一样, 此处不再详述.

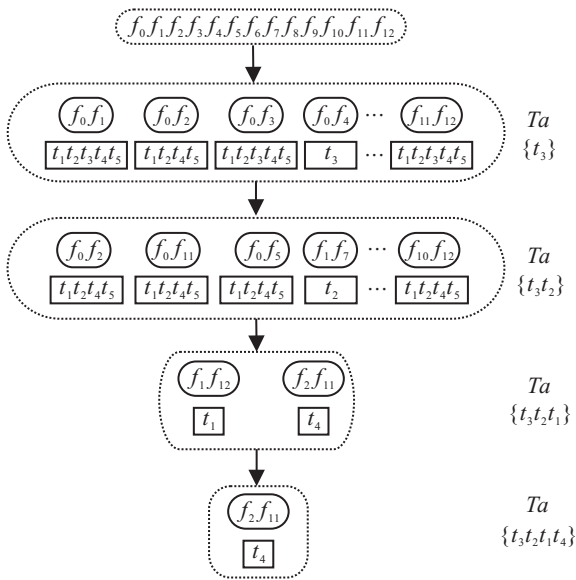


图 2 例 1 的计算结果

4.2 实例 2

实例 2 是一个滤波器电路, 电路图和整数编码故障字典参见文献 [4]. 用基本算法对实例 2 进行计算, 结果如图 3 所示. 从图 3 中可以看出, t_{11}, t_1, t_5 和 t_9 依次选入优化测点集, 最终得到的优化测试集为 $\{t_{11}, t_1, t_5, t_9\}$, 这 4 个测点也均为必要测点, 改进算法和基本算法的过程基本是一致的, 计算结果相同.

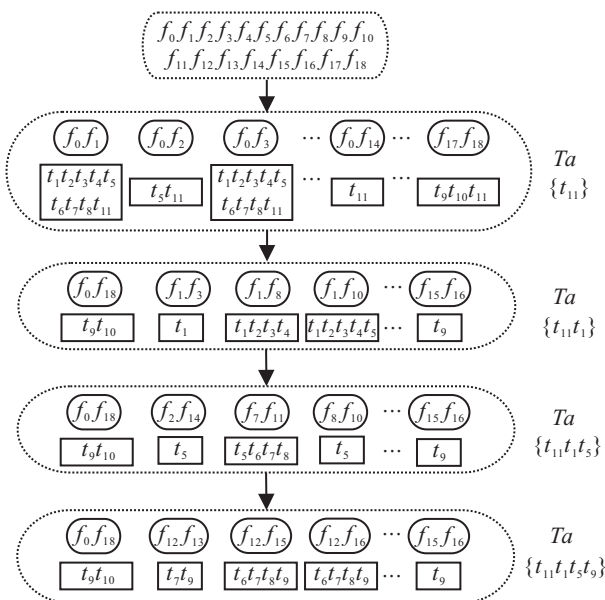


图 3 例 2 的计算结果

5 实验分析

为了比较算法的计算效果, 本文编写了 WP 算法、熵算法、基本算法和改进算法的 Matlab 程序. 以下计算数据均在英特尔双核处理器 E2140, 内存 1 GB, windows XP, Matlab 7.1 环境下得到.

5.1 实际电路计算效果分析

分别用 WP 算法、熵算法、基本算法和改进算法对实例进行了计算, 结果如表 2 所示. 其中: 实例 1 和实例 2 为第 4 节中的两个例子, 实例 3 为文献 [4] 中的例 1, 实例 4 为文献 [3] 中的表 1. 从表 2 可以看出, 熵算法的计算时间最快, 但计算结果最差; WP 算法和本文提出的两种算法计算结果一致, 但本文两种算法在计算时间上具有优势.

表 2 实际电路的计算结果

	故障字典	熵算法	WP 算法	基本算法	改进算法
实例 1	测点数量	4	4	4	4
	计算时间/s	0.0018	0.0149	0.0074	0.0073
实例 2	测点数量	4	4	4	4
	计算时间/s	0.0022	0.0903	0.0261	0.0260
实例 3	测点数量	4	3	3	3
	计算时间/s	0.0010	0.0061	0.0062	0.0064
实例 4	测点数量	8	7	7	7
	计算时间/s	0.0039	0.1213	0.0354	0.0359

5.2 随机数据计算效果分析

第 5.1 节中几个实际电路的规模都不大, 且必要测点较多, 为了检验其他情况下各种算法的计算效果, 随机生成 100 个具有 60 个故障状态、20 个测点、4 个模糊集的整数字典故障字典. 分别用 WP 算法、熵算法、基本算法和改进算法对 100 个故障字典进行计算, 并对平均计算时间和每种算法得到的优化测点集中的测点个数进行统计, 结果如表 3 所示.

表 3 随机数据的计算结果

算法	优化测点集中的测点个数						平均计算时间/s	
	4	5	6	7	8	9		10
熵算法	0	0	2	15	36	25	22	0.0069
WP 算法	17	83	0	0	0	0	0	2.4871
基本算法	10	90	0	0	0	0	0	3.6724
改进算法	51	49	0	0	0	0	0	8.3240

从表 3 中可以看出, 熵算法的平均计算时间最短, 但计算结果最差; WP 算法的计算时间比基本算法小, 计算结果略优于基本算法; 改进算法的计算时间要多几秒, 但计算结果最好, 有 51 组优化测点集中的测点个数为 4 个, 远多于 WP 算法中的 17 组.

综合表 2 和表 3 的数据可知, 基本算法只在具有

必要测点的系统中有优势, 而改进算法的计算结果一直是最优的。

6 结 论

故障字典法是模拟电路的一种常用的故障诊断方法, 测点选择是该方法中的一个重要环节, 其目的是在故障都可隔离的前提下, 选择测点数量最小的测点集。本文针对最小测点集问题, 提出了“测点必要度”的概念, 同时结合“故障隔离度”的概念, 提出了基于测点必要度的测点优选方法。实验分析表明, 本文提出的改进算法计算效果较好, 可用于复杂模拟电路的测点选择, 同时对模拟电路的可测试性设计具有一定的参考价值。

参考文献(References)

- [1] 杨士元. 模拟系统的故障诊断与可靠性设计[M]. 北京: 清华大学出版社, 2001.
(Yang S Y. Fault diagnosis and reliability design for analog system[M]. Beijing: Tsinghua University Press, 2001.)
- [2] Yang Cheng-lin, Tian Shu-lin, Long Bing. Application of heuristic graph search to test points selection for analog fault dictionary techniques[J]. IEEE Trans on Instrumentation and Measurement, 2009, 58(7): 2145-2158.
- [3] 杨成林, 田书林, 龙兵. 多值故障字典的测点选择与序测试设计[J]. 系统工程与电子技术, 2009, 31(9): 2271-2275.
(Yang C L, Tian S L, Long B. Design of test points selection and test sequencing problem of multivalued fault dictionary[J]. Systems Engineering and Electronics, 2009, 31(9): 2271-2275.)
- [4] Prasad V C, Badu N S C. Selection of test nodes for analog fault diagnosis in dictionary approach[J]. IEEE Trans on Instrumentation and Measurement, 2000, 49(6): 1289-1297.
- [5] Starzyk J A, Liu D, Liu Z H, et al. Entropy-based optimum test points selection for analog fault dictionary techniques[J]. IEEE Trans on Instrumentation and Measurement, 2004, 53(3): 754-761.
- [6] 汪鹏, 杨士元. 模拟电路故障诊断测试节点优选新算法[J]. 计算机学报, 2006, 29(10): 1781-1785.
(Wang P, Yang S Y. New algorithm for test node selection for analog circuits diagnosis[J]. Chinese J of Computers, 2006, 29(10): 1781-1785.)
- [7] Golonek T, Rutkowski J. Genetic-algorithm-based method for optimal analog test points selection[J]. IEEE Trans on Circuits and Systems: Express Briefs, 2007, 54(2): 117-121.
- [8] Zhang Chao-jie, He Guo, Liang Shu-hai. Test point selection of analog circuits based on fuzzy theory and ant colony algorithm[C]. IEEE Autotestcon 2008. Salt Lake City, 2008: 8-11.
-
- (上接第1894页)
- [4] 邹亮, 徐建闽, 朱玲湘. 遗传算法在动态路径诱导系统中的应用[J]. 交通运输系统工程与信息, 2007, 7(3): 45-48.
(Zou L, Xu J M, Zhu L X. Application on genetic algorithm in dynamic route guidance system[J]. J of Transportation Systems Engineering and Information Technology, 2007, 7(3): 45-48.)
- [5] 孙霞, 黄席樾, 杨祖元, 等. 基于改进遗传算法的城市交通动态最优路径求解[J]. 计算机工程与应用, 2007, 43(30): 245-248.
(Sun X, Huang X Y, Yang Z Y, et al. Improved GA-based dynamic optimal routing in urban traffic system[J]. Computer Engineering and Application, 2007, 43(30): 245-248.)
- [6] 靳凯文, 李春葆, 秦前清. 基于蚁群算法的最短路径搜索方法研究[J]. 公路交通科技, 2006, 23(3): 128-134.
(Jin K W, Li C B, Qin Q Q. Study on shortest path search method based on ant algorithm[J]. J of Dighway and Transportation Research and Development, 2006, 23(3): 128-134.)
- [7] Ahn C W, Ramakrishna R S. Shortest path routing algorithm using hopfield neural network[J]. Electronics Letters, 2001, 37(19): 1176-1178.
- [8] Bin R. Toward a class of link travel time functions for dynamic assignment models on signalized networks[J]. Transportation Research-B, 1997, 31(4): 277-290.