◎研发、设计、测试◎

改进的 SAMPSO 的软件测试数据自动生成

魏付强,姜淑娟
WEI Fu-qiang,JIANG Shu-juan

中国矿业大学 计算机科学与技术学院,江苏 徐州 221008
School of Computer Science and Technology,China University of Mining and Technology,Xuzhou,Jiangsu 221008,China
E-mail:wwwnwfq@163.com

WEI Fu-qiang,JIANG Shu-juan.Automated test data generation by improved Simple and Adaptive Mutation Particle Swarm Optimization algorithm.Computer Engineering and Applications,2009,45(32):57-60.

Abstract: An improved Simple and Adaptive Mutation Particle Swarm Optimization(SAMPSO) algorithm is proposed here based on the combination of simple particle swarm optimization and adaptive mutation particle swarm optimization for automated software test data generation.During the run time,the mutation probability for the current best particle is determined by two factors: the variance of the population fitness and the current optimal solution.The mutation operator is designed to enhance the global search capability of PSO algorithm at starting.The particle velocity is discarded.The evolutionary process is only controlled by the variables of the particle position.Test examples show that it is better than basic particle swarm optimization algorithm and can improve the efficiency of automated test data generation.
Key words: software testing;test data generation;simple and adaptive mutation;Particle Swarm Optimization(PSO)

摘 要:针对软件测试数据的自动生成提出了一种简化的自适应变异的粒子群算法(SAMPSO)。该算法在运行过程中根据群体适应度方差以及当前最优解的大小来确定当前最佳粒子的变异概率,变异操作增强了粒子群优化算法前期全局搜索能力,去掉了粒子群优化(PSO)算法中进化方程的粒子速度项,仅由粒子位置控制进化过程,避免了由粒子速度项引起的粒子发散而导致后期收敛变慢和精度低问题。实验结果表明该算法在测试数据的自动生成上优于基本的粒子群算法,提高了效率。
关键词:软件测试;测试数据生成;简化的自适应变异;粒子群算法
DOI:10.3778/j.issn.1002-8331.2009.32.018 **文章编号**:1002-8331(2009)32-0057-04 **文献标识码**:A **中图分类号**:TP311.56

测试数据生成是软件测试过程中必不可少的重要环节,通常可分为功能测试数据生成和结构测试数据生成。结构测试是针对程序内部逻辑的测试,要求对程序的结构特性做到一定的覆盖,即在被测程序的输入参数域中搜索能使程序的控制流按预定的程序路径执行的目标参数值[1-2]。为了解决此问题人们提出了许多种方法:迭代松弛法[3]、符号执行法[4]和启发式算法。用于面向路径的测试数据生成的启发式算法包括模拟退火[5]、遗传算法[6]和蚁群算法[7]。文献[8-9]提出了基于粒子群优化算法的软件结构测试数据自动生成的方法,取得了一定的效果。

粒子群优化(Particle Swarm Optimization,PSO)算法是Kennedy 和 Eberhart 于 1995 年提出的一种基于群体智能的进化优化算法,其思想来源于对鸟群和鱼群群体觅食运动行为的模拟[10-11]。粒子群优化算法采用实数求解,并且需要调整的参数较少,易于实现。因此,算法一提出就得到众多学者的重视,并且已经在神经网络训练、函数优化和模糊系统控制等领域取得了大量的研究成果。

然而,粒子群优化算法根据全体粒子和自身粒子的搜索经验向着最优解的方向发展。在进化前期容易陷入局部最优解,进化后期收敛速度变慢,同时算法精度不高。针对面向路径的测试数据生成特点,提出了基于简化的自适应变异的粒子群(Simple and Adaptive Mutation Particle Swarm Optimization,SAMPSO)算法的测试数据自动生成。

1 PSO 算法及其改进

PSO 算法是一种群体智能算法,它利用 m 个粒子组成的粒子群在 D 维目标搜索空间中以迭代的方式寻找最优解。在每步迭代中,第 i 个粒子状态更新操作如下:

$$v_{id}(t+1)=\omega v_{id}(t)+c_1r_{1d}(t)(p_{id}-x_{id}(t))+c_2r_{2d}(t)(p_{gd}-x_{id}(t)) \quad (1)$$

基金项目:教育部科学技术研究重点项目(the Key Project of Chinese Ministry of Education under Grant No.108063);江苏省自然科学基金(the Natural Science Foundation of Jiangsu Province of China under Grant No.BK2008124);中国矿业大学科学研究基金(the Science Research Foundation of China University of Mining and Technology No.OD080310)。
作者简介:魏付强(1980-),男,硕士研究生,主要研究方向:软件测试,程序理解和分析,嵌入式应用;姜淑娟(1966-),女,教授,博士生导师,研究方向:程序设计语言,软件分析与测试,编译技术。
收稿日期:2009-06-03 **修回日期**:2009-07-27

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \quad (2)$$

其中，$i=1,2,\cdots,m$，$d=1,2,\cdots,D$，D 维向量 $\boldsymbol{x}_i(t)$ 与 $\boldsymbol{v}_i(t)$ 分别为粒子 i 在 t 时刻的位置与速度；p_i 为粒子 i 当前搜索到的最优位置；p_g 为整个粒子群当前搜索到的最优位置；ω 称为惯性权重；c_1、c_2 为非负常数，称为学习因子；r_{1d} 和 r_{2d} 是介于[0,1]之间均匀分布的随机数。当学习因子在一定范围内变化时，惯性权重 ω 越大，则算法的全局搜索能力越强；ω 越小，则算法的局部搜索能力越强。利用这个特点，文献[2]提出了惯性权重线性递减策略，t 时刻的惯性权重可表示为：

$$\omega(t)=(\omega_{\max}-\omega_{\min})\frac{(T_{\max}-t)}{T_{\max}}+\omega_{\min} \quad (3)$$

其中，$T_{\max}$ 为最大进化代数；$\omega_{\min}$ 为初始惯性权重值；$\omega_{\max}$ 为最大代数时惯性权重值。Angeline[12]等人借鉴遗传算法思想提出杂交 PSO 算法概念，提高了算法的收敛速度和精度。

下面对基本 PSO 中粒子速度项进行分析：

迄今为止，基本 PSO 及其改进算法都是基于粒子“位置”和“速度”这两个关键概念，从而在进化方程中都包含位置变量和速度变量，仔细分析基本 PSO 的生物模型和进化迭代方程式（1）和方程式（2）可以发现：在 PSO 中，粒子速度概念不是必需的。从基本 PSO 模型角度来看，粒子位置 x_1 代表当前问题的解，优化的最终结果是使 x_1 无限逼近最优解 X^*，因此，只需要考虑 x_1 的直接变化。粒子速度 v_1 代表粒子移动的快慢程度，粒子移动速度的大小并不代表粒子能够有效趋近最优解位置，反而可能造成粒子偏离正确的进化方向，出现粒子“发散”现象，从而有可能造成后期收敛缓慢，收敛精度低。另外，从式（1）（2）来看，位置与速度直接进行加法运算，而没有粒子运动时间概念，这也不符合现实生活中的运动规律 $x=vt$。

定理 1 基本 PSO 进化过程与粒子速度无关。

证明 除 p_{id} 和 p_{gd} 对搜索空间各维的联系以外，每维的更新相互独立。故不失一般性，证明过程可以简化到一维进行，下标 d 可以省略。进一步地，假设种群中除第 i 个粒子外其余粒子保持不动，下标 I 可以省略。再令 $\varphi_1=r_1c_1$，$\varphi_2=r_2c_2$，$\varphi=\varphi_1+\varphi_2$，$\rho=\frac{\varphi_1p_0+\varphi_2p_g}{\varphi_1+\varphi_2}$。为方便理解，将式（1）和式（2）中变量符的上标移到变量符后的括号中，则式（1）和式（2）可以变为：

$$v(t+1)=\omega v(t)+\varphi(\rho-x(t)) \quad (4)$$

$$x(t+1)=x(t)+v(t+1) \quad (5)$$

将式（4）和式（5）迭代可以得到式（6）：

$$x(t+2)+(\varphi-\omega-1)x(t+1)+\omega x(t)=\varphi\rho \quad (6)$$

式（6）是不含速度项的经典二阶微分方程（假设粒子的位置移动为连续过程）。

定理 1 的重要性在于说明基本 PSO 算法可以没有粒子速度的概念，避免了人为确定参数$[-v_{\max},v_{\max}]$而影响粒子的收敛速度和收敛精度。

文献[13]结合标准 PSO 进化过程与粒子速度无关和分析了收敛于局部极值的原因提出了简化的粒子群优化算法（SP-SO），其公式如下：

$$x_{id}(t+1)=\omega x_{id}(t)+c_1r_{1d}(t)(p_{id}-x_{id}(t))+c_2r_{2d}(t)(p_{gd}-x_{id}(t)) \quad (7)$$

详细证明请参考文献[13]。

定义 1 设粒子群的粒子数目为 n，f_i 为第 i 个粒子的适应度，f_{avg} 为粒子群目前的平均适应度，φ^2 为粒子群的群体适应度方差，则 φ^2 可以定义为：

$$\varphi^2=\sum_{i=1}^{n}\left(\frac{f_i-f_{avg}}{f}\right)^2 \quad (8)$$

其中，f 是归一化定标因子，其作用是限制 φ^2 的大小。f 可以取任意值，只需注意两个条件：（1）归一化后，整个粒子群$|f_i-f_{avg}|$的最大值不大于 1；（2）f 随算法的进化而变化。在该文算法中，f 的取值采用如下公式：

$$f=\begin{cases}\max\{|f_i-f_{avg}|\} & \max\{|f_i-f_{avg}|\}>1\\ 1 & \text{其他}\end{cases} \quad (9)$$

定义 1 表明：群体适应度方差 φ^2 反映的是粒子群中所有粒子的“收敛”程度。φ^2 越小，则粒子群趋于收敛；反之，粒子群则处于随机搜索阶段。

定义 2 设粒子群中某个粒子在 t 时刻的位置为 $x(t)$，p 为搜索空间内的任意位置，则粒子收敛定义如下[14]：

$$\lim_{x\to\infty}x(t)=p \quad (10)$$

该定义表明，粒子的收敛是指粒子最终停留在搜索空间内某一固定位置 p。

文献[15]将粒子群的适应度方差作为全局最优化变异条件，提出自适应变异的粒子群优化算法。

定理 2 如果粒子群优化算法陷入早熟收敛或者达到全局收敛，粒子群中的粒子将聚集在搜索空间的一个或几个特定位置，群体适应度方差 φ^2 等于零。

文献[15]给出了详细证明，这里不再赘述。

定理 2 给出了粒子群优化算法收敛状态与群体适应度方差之间的关系。显然，仅凭群体适应度方差等于零不能区别早熟收敛与全局收敛，还须进一步判断算法此时得到的最优解是否为理论全局最优解或是期望最优解 f_d。如果此时已经得到全局最优，则可认为算法达到全局收敛；反之，则表明算法陷入局部最优。

考虑到粒子在当前 p_{gd} 的作用下可能发现更好的位置，因此新算法将变异操作设计成一个随机算子，即对满足变异条件的 p_{gd} 按一定的概率 p_m 变异。p_m 的计算公式如下：

$$p_m=\begin{cases}k & \sigma^2<\sigma_d^2 \text{且} f(p_{gd})>f_d\\ 0 & \text{其他}\end{cases} \quad (11)$$

对于 p_{gd} 的变异操作，采用增加随机扰动的方法，设 p_{gdk} 为 p_{gd} 的第 k 维取值，η 是服从 Gauss（0，1）分布的随机变量，则：

$$p_{gdk}=p_{gdk}(1+0.5*\eta) \quad (12)$$

2 简化的自适应变异的粒子群算法

针对粒子群优化算法前期的早熟收敛和后期收敛速度慢、算法精度不高的问题，对粒子群优化算法进行了改进，提出了一种新的基于简化的自适应变异的粒子群优化（SAMPSO）算法。其算法流程如下：

（1）随机初始化粒子群中粒子的位置 X。

（2）将粒子的 p_{id} 设置为当前位置，p_{gd} 设置为初始群体中最佳粒子的位置。

（3）判断算法收敛准则是否满足，如果满足，转向（9）；否则，执行（4）。

（4）对于粒子群中的所有粒子，执行如下操作：

①根据式（3）和式（7）更新粒子的位置。

②如果粒子适应度优于 p_{id} 的适应度，p_{id} 设置为新位置。

③如果粒子适应度优于 p_{gd} 的适应度，p_{gd} 设置为新位置。

(5)根据式(8)与式(9)计算群体适应度方差 σ^2,并计算 $f(p_{gd})$。

(6)根据式(11)计算变异概率 p_m。

(7)产生随机数 $r\in[0,1]$,如果 $r<p_m$,按式(12)执行变异操作;否则,转向(8)。

(8)判断算法收敛准则是否满足,如果满足,执行(9);否则,转向(4)。

(9)输出生成的数据,算法运行结束。

3 基于SAMPSO算法的软件测试数据生成

3.1 基于PSO算法的软件测试数据生成系统

对于采用程序直接执行方式,文献[16]给出了将遗传算法作为搜索策略来生成测试数据的系统模型。针对粒子群算法的特点,采用了将粒子群算法作为搜索策略来生成测试数据的系统模型。如图1所示。

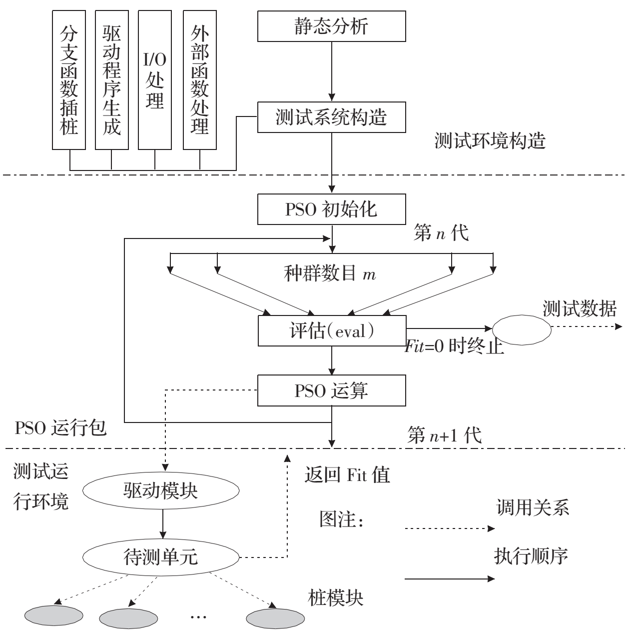

图1 基于粒子群算法的测试数据生成系统

该模型可分为三部分:测试环境构造,粒子群算法包和测试运行环境。

第一部分测试环境构造是整个系统的基础,它主要利用静态分析提供的基本单元信息并借助于各种插装技术来自动构造相应的测试运行环境(包括驱动模块、测试单元及桩模块)。

第二部分PSO运行包是系统的核心,它主要是随机初始化粒子群中粒子的位置和速度(或者只初始化位置),然后通过对该种群进行反复的PSO运算,从而引导种群不断地向目标值进化直到最终找到解或达到限定的运行代数为止。

在第二部分中需要对种群中的每一个个体的优劣进行评估,评估由第三部分来完成,它主要是通过实时地调用并运行插装后测试系统来返回一个 *Fit* 值供PSO使用。

3.2 适应度函数的设计

适应度函数的构造方法是检测条件语句的真假值关系[17],若能满足给定的真假值,则目标函数值为0;否则进行如表1所示的运算。

文中在"分支函数"基础上,采用"分支函数叠加法"。分支函数是一个分支谓词到实际值的映射,可以量化地反映在测试

表1 适应值函数值计算方法

表达式	适应值函数值
布尔表达式	0;(表达式为TRUE) k;(表达式为FALSE)
$a=b$	0;(abs($a-b$)=0) abs($a-b$)+k;(其他情况)
$a!=b$	0;(abs($a-b$)!=0) k;(其他情况)
$a<b$	0;($a-b<0$) ($a-b$)+k;(其他情况)
$a\leqslant b$	0;($a-b\leqslant 0$) ($a-b$)+k;(其他情况)
$a>b$	0;($b-a<0$) ($b-a$)+k;(其他情况)
$a\geqslant b$	0;($b-a\leqslant 0$) ($b-a$)+k;(其他情况)
$a\vee b$	min(*fit*(a),*fit*(b))
$a\wedge b$	max(*fit*(a),*fit*(b))

数据的驱动下,被测试程序的实际执行路径对选定路径的覆盖程度。具体做法是(假定选定路径上有 m 个分支点,n 个参数):扫描被测试程序,在决定路径转移的语句自动插入一段代码,同时返回一个目标函数值。若一条路径所经过的条件判断语句不止一个,则将每个目标函数值进行累加,最后的和为此个体的目标函数值。

$$\varphi_1=f_1(x_1,x_2,\cdots,x_n),\varphi_2=f_2(x_1,x_2,\cdots,x_n),\cdots,\varphi_m=f_m(x_1,x_2,\cdots,x_n)$$

最后适值函数值为:

$$F=\Psi(\varphi_1)+\Psi(\varphi_2)+\cdots+\Psi(\varphi_m)$$

$$\Psi(x)=\begin{cases}0 & x\leqslant 0\\ x & x>0\end{cases}$$

4 实验与结果分析

针对软件测试领域经典的三角形分类问题(Tri)以及扩展三角形分类问题(增加直角三角形判别,Tri-a)采用简化的自适应变异的粒子群优化算法(SAMPSO)分别进行了测试,并与使用基本遗传算法(GA)、基本粒子群优化算法(BPSO)进行了性能上的比较,结果如图2、图3所示。

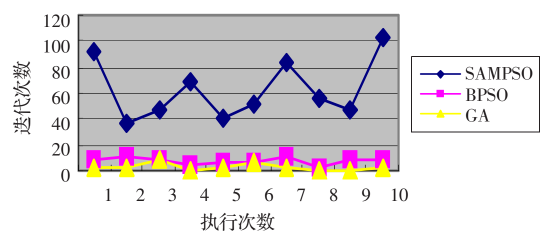

图2 等边三角形迭代次数曲线

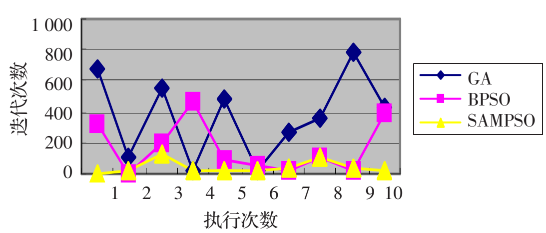

图3 直角三角形迭代次数曲线

其中实验设置的参数如下:GA算法的交叉概率为0.85,变异概率为0.15。BPSO算法的参数:粒子数目/种群大小为100,$c_1=c_2=2$,$\omega_{max}=0.7$,$\omega_{min}=0.3$。SAMPSO算法的参数:粒子数目/种群

大小为 100,$c_1=c_2=2$,$\omega_{max}=0.9$,$\omega_{min}=0.1$。

可见,对于等边三角形分类问题来说,应用 GA 一般在 70 代左右产生适应度最优个体。应用 BPSO 一般在 8 代左右产生适应度最优个体,应用 SAMPSO 最优个体提前 2~3 代就可以产生,稳定性进一步提高。而对于直角三角形分类问题来说,应用 GA 平均 250 代左右产生适应度最优个体,但是很不稳定。应用 BPSO 平均 60 代产生适应度最优个体,但是迭代次数的幅度也很大。而应用 SAMPSO 后平均可以提前 10 代左右,并且趋于稳定。说明简化的自适应变异的粒子群算法从进化的代数和稳定性上都可以得到令人满意的结果,证明提出的基于简化的自适应变异的粒子群算法的测试数据自动生成是合理有效的。

5 结束语

采用简化的自适应变异的粒子群算法增强了前期全局搜索能力,并克服了后期收敛变慢和精度低的问题。应用于测试数据自动生成, 与传统的遗传算法或者基本的粒子群算法相比,其产生最优解的代数和稳定性都有明显的改进。该文的进一步工作是将 SAMPSO 算法的测试数据模型与切片算法相结合,使之只关注与兴趣点相关的语句,可以应用于回归测试、覆盖分析和完整性测试等。

参考文献:

[1] Pargas R P,Harrold M J.Test data generation using genetic algorithms[J].Journal of Software Testing,Verification and Reliability,1999,9(4):263–282.

[2] Korel B.Automated software test data generation[J].IEEE Transactions on Software Engineering,1990,16(8):870–879.

[3] Gupta N,Mathur A P,Soffa M L.Automated test data generation using an iterative relaxation method[C]//Proc of the ACM SIGSOFT Sixth International Symposium on Foundations of Software Engineering,Florida,United States,1998:231–244.

[4] 单锦辉.面向路径的测试数据自动生成方法研究[D].国防科学技术大学,2002.

[5] 傅博.基于模拟退火遗传算法的软件测试数据自动生成[J].计算机工程与应用,2005,41(12):82–84.

[6] 高海昌,冯博琴,侯芸.测试数据自动生成的研究进展[C]//中国控制与决策学术年会论文集,2006:460–464.

[7] 傅博.基于蚁群算法的软件测试数据自动生成[J].计算机工程与应用,2007,43(12):97–99.

[8] Windisch A,Wappler S,Wegene J.Applying particle swarm optimization to software testing[C]//Proc Conf on Genetic and Evolutionary Computation(GECCO 2007),2007:1121–1128.

[9] 李爱国,张艳丽.基于 PSO 的软件结构测试数据自动生成方法[J].计算机工程,2008,34(6):93–97.

[10] Kennedy J,Eberhart R C.Particle swarm optimization [C]//Pro IEEE International Conference on Neural Networks. [S.l.]:IEEE Press,1995,4:1942–1948.

[11] Shi Y,Eberhart R C.A modified swarm optimizer[C]//IEEE International Conference of Evolutionary Computation.Anchorage,Alaska:IEEE Press,1998.

[12] Angeline P J.Evolutionary optimization versus particle swarm optimization:Philosophy and performance differences[C]//Proc of the 7th Annual Conf on Evolutionary Programming.Berlin:Springer–Verlag,1998:601–610.

[13] 胡旺,李志蜀.一种更简化而高效的粒子群优化算法[J].软件学报,2007,18(4):861–868.

[14] Bergh F.An analysis of particle swarm optimizers[D].South Africa:Department of Computer Science,University of Pretoria,2002:81–83.

[15] 吕振肃,侯志荣.自适应变异的粒子群优化算法[J].电子学报,2004,32(3):416–420.

[16] 汪浩,谢军凯,高仲仪.遗传算法及其在软件测试数据生成中的应用研究[J].计算机工程与应用,2001,37(12):64–68.

[17] 邢恺,伦立军.测试数据自动生成方法[J].计算机技术与发展,2006,16(9):53–55.

(上接 52 页)

比较这两个三角形的两条权较小的边权之和有 $bc+ce<cf+ce$,则选择 $b-c-e$ 为通路,并将源点由 a 变为 f。

(2)此时操作已进入 e 点只有顶点 e、d、f 没有经过,则删除最大权边 $f-e$ 回到源点 f。

(3)将得到的边权和相加 2+3+4+5+8+9=31 即得到结果。经检验这个结果是最小回路。

5 结语

图论和矩阵的关系是密不可分的,在算法中用类似三角剖分有关矩阵乘法的办法完全可以找到权和最小的三角形,而以后的步骤也可以用矩阵的有关运算实现, 因此算法是可行的。旅行售货员问题是一个典型的易于描述却难以处理的 NP 完全难题,其可能的路径数目与城市数目是呈指数型增长的,求解非常困难,所以能够快速找到解决这个问题的办法有很高的理论价值和实用价值。相对引言中的算法,该文算法用图论的办法得到一种较新的算法,相对文献[6]所提算法该文算法有更好的精度(上例由文献[6]所提方法得到结果为 2+4+5+7+10+9=37),原因是该文算法每次经过的边都能保证是最小的边,而文献[6]中的算法在中间过程中就不能保证总是最小的边。相对文献[1–3]所提算法该文算法要容易理解得多。

参考文献:

[1] 马良.旅行推销员问题的算法综述[J].数学的实践与认识,2000,30(2):156–165.

[2] 马良.多目标旅行售货员问题的蚂蚁算法[J].系统工程理论方法应用,1999,8(4):23–25.

[3] 任小康.基于禁忌搜索的旅行售货员问题[J].佳木斯大学学报,2005,23(3):333–345.

[4] 左孝凌.李为鉴,刘永才.离散数学[M].上海:上海科学技术文献出版社,2004:271–312.

[5] 伍庆成.论欧拉图.哈密顿图的判定及应用[J].中国高新技术企业,2007,7(3):89–92.

[6] 邱伟星,沈金龙.旅行商问题的一个近似算法[J].南京邮电学报,1998,18(1):106–108.

[7] 阎克俭.旅行售货员问题解法的再探讨[J].烟台师范学院学报,1992,8(3):11–14.

[8] Gregory G,Abraham P.The traveling salesman problem and its variations[M].Dordrecht Hardbound:Kluwer Academic Publishers,2002:38.

[9] Dang Jian–wu,Chen Yi–xing.Study on a polynomial time evolution algorithm for the traveling salesman problem[J].Journal of Lanzhou Railway University:Natural Sciences,2001,20(1):49–53.

[10] 杨晋吉,苏开乐.SAT 问题中局部搜索法的改进[J].计算机研究与发展,2005,42(1):60–65.