第41卷 第11期 西 安 交 通 大 学 学 报 Vol.41 №11
2007年11月 JOURNAL OF XI′AN JIAOTONG UNIVERSITY Nov. 2007

一种具有跟踪替代特征的小世界算法

陈煜聪[1]，杨斌[1]，杜海峰[1]，邵韶[2]，庄健[1]
（1.西安交通大学机械工程学院，710049，西安；2.俄克拉荷马大学动物系，73019，美国诺曼）

摘要：针对简单小世界算法在优化复杂函数时出现的停滞现象，提出对搜索进行跟踪、对停滞节点进行更替的策略.对每个搜索节点，从搜索的第1代开始进行跟踪，记录节点在每个传递位置停留的次数，当停滞次数超出设定值时便认为该节点进入停滞状态，在搜索空间中随机生成一个节点替代该停滞节点，以保证搜索的高效性.仿真试验表明，改进算法有效地克服了原算法的停滞现象，与原算法相比，改进算法种群多样性好、优化效率高、鲁棒性强，并具备解决更复杂工程优化问题的潜能.

关键词：小世界算法；停滞；跟踪；替代
中图分类号：O224 **文献标识码**：A **文章编号**：0253-987X(2007)11-1360-04

Small-World Algorithm with the Replacing and Tracking Characters

Chen Yucong[1], Yang Bin[1], Du Haifeng[1], Shao Jie[2], Zhuang Jian[1]
(1. School of Mechanical Engineering, Xi′an Jiaotong University, Xi′an 710049, China;
2. Department of Zoology, University of Oklahoma, Norman 73019, USA)

Abstract: When simple small world algorithm is adopted to optimize complex functions, the searching nodes are prone to be trapped in these local optimums. Aiming at the above, the following strategy is suggested: adding the tracking and replacing mechanism to every searching node; counting the stagnation times of a node at a certain station in its searching route; if the times oversteps the allowable value, a new node randomly arises in the solution space to replace the stagnating one for improving the searching efficiency. The improved method is tested via a few benchmark test functions in a simulation and the corresponding results show the efficiency for solving stagnation. Compared with the simple small world algorithm, the modified small world algorithm is endowed with better robustness and faster convergence to solve complex optimization problems.

Keywords: small world algorithm; stagnation; tracking; replacing

智能优化算法在信息科学、计算机科学、系统科学等领域以及工程实践中有着广泛的应用[1].“仿生”和“拟物”是已有智能算法发展的2个主要途径，与生物和物理这两类系统相比，人类社会是一个更加复杂的大系统，一些社会现象同样为发展新的优化算法提供了思路.受社会网络中的六度分离现象[2]启发，依据社会网络是一种中间网络形态的理论[3]和最短路径理论[4-5]，文献[6]将社会网络中的搜索机理用于解决优化问题，提出了一种新的“仿社会”优化算法——小世界算法.小世界算法主要由局部搜索算子和随机长连接搜索算子构成，由于小世界算法强调邻域搜索，因此在对复杂的多极值问题进行优化时，邻域搜索一旦失去作用将会出现严重的停滞现象，导致算法搜索效率明显下降，影响算法的优化性能.针对小世界算法的上述缺陷，借鉴文献[2]中Stanley Milgram有关小世界现象社会学试

收稿日期：2007-04-03. 作者简介：陈煜聪(1981～)，女，硕士生；杜海峰(联系人)，男，副教授. 基金项目：国家自然科学基金资助项目(50505034).

验的措施，本文提出一种基于跟踪替代策略的小世界算法（M-SWA），算法对每一个搜索节点进行全程跟踪，一旦发现节点在某个位置出现停滞现象，便用搜索空间中随机生成的一个节点替换该停滞节点，以提高算法的搜索效率. 本文将文献[6]中的小世界算法称为简单小世界算法（S-SWA）.

1　S-SWA 算法及其与遗传算法的比较

1.1　S-SWA 算法的基本原理

在 S-SWA 算法中，根据求解精度对优化对象的解空间进行网格化处理，每个网格节点代表一个潜在的可行解，随机生成初始节点集合，在搜索的每一步，节点通过执行局部搜索和随机长连接搜索来找寻下一个最优传递目标（节点），从而实现节点的更新和信息传递. 节点集合进行反复的寻优迭代，直到搜索到最优解为止.

采用文献[6]中的定义将节点编码记为 s_i；节点集合记为 S，$(s_i \in S)$；S 中的节点个数记为 n；$\| s_i - s_j \|$ 表示海明距离；节点 s_i 的 ℓ 邻域节点集合 $\zeta^{\ell}(s_i)$ 定义为 $\zeta^{\ell}(s_i)=\{s_j \mid 0<\| s_i-s_j \| \leqslant \ell, s_j \in S\}$；非 ℓ 邻域节点集合 $\overline{\zeta^{\ell}(s_i)}$ 定义为 $\overline{\zeta^{\ell}(s_i)}=\{s_j \mid \ell<\| s_i-s_j \|, s_j \in S\}$. 局部搜索算子 Ψ 的作用是将信息从节点 $s_i(k)$ 传递给 $\zeta^{\ell}(s_i(k))$ 中距离目标节点最近的节点 $s_i'(k+1)$，作为备选替代节点，记为 $s_i'(k+1) \leftarrow \Psi(s_i(k))$. 其中，$s_i(k)$ 表示 S 中第 i 个节点的第 k 步搜索，随机长连接搜索算子 Γ 在 $\overline{\zeta^{\ell}(s_i(k))}$ 中随机选择一点 $s_i''(k+1)$ 作为备选替代节点，记为

$$s_i''(k+1) \leftarrow \Gamma(s_i(k))$$

1.2　S-SWA 算法与遗传算法的异同

受 Milgram 有关社会学试验和 Kleinberg 二维网格小世界网络模型的启发，文献[6]提出的小世界算法与已经比较成熟的进化算法有一定的相似之处. 首先，两类算法具有相同的算法结构，即要经过“初始种群的产生→评价标准计算→种群间个体信息交换→新种群产生”这一循环过程；其次，两种算法都是采用群体搜索策略，算法本质上都具有并行性和搜索变化的随机性；最后，二者处理的对象都是待求解参数的编码串，而非参数本身，具备弱方法通用性强的特点. 但是，小世界算法不是进化算法的简单改进，小世界算法的基本思想来源于社会学，强调类似社会现象中“熟人关系”的局域搜索，而非全局搜索. 在新个体的产生策略中没有采用全局选择操作，而是代以局部选择，强调了小世界效应中的局部信息传递特性，从而更好地保持了种群多样性和搜索中信息传递的高效性.

2　S-SWA 算法的不足及改进

S-SWA 算法中的每个节点都具有局域搜索能力和一定的全局搜索能力，算法的结构简单，算子和需要设定的参数少，容易操作和实现，同时 S-SWA 算法在快速性、多样性和稳定性方面表现出优于遗传算法的性能[6]. 但是，S-SWA 算法仍然存在一些不足之处，由于该算法强调类似于社会网络中熟人关系的局部搜索操作，如果在传递过程中出现局部搜索失效的情况，节点搜索将出现停滞问题，因此造成传递的中断. 当网络中节点的适应度分布情况非常复杂时，如在优化空间存在多个局部极值，或者最优解在优化空间分布比较特殊（如边界点）等，传递很容易陷入停滞状态，造成 S-SWA 算法搜索性能的下降.

为了防止 S-SWA 算法的搜索节点在优化复杂问题时出现停滞现象，受 Milgram 的社会学研究试验[2]启发，提出的改进思路是对所有搜索节点进行全程跟踪，对节点传递路径中的每一个位置记录其在该位置停留的次数. 当判断节点陷入停滞状态时，在可行解空间中随机生成一个节点来替代原节点重新进行搜索，以保证传递不间断，提高算法的执行效率，具体的改进方法如下.

步骤 1　在节点的搜索过程中，加入一种跟踪机制，即对每个节点从初始位置开始对其整个搜索过程进行跟踪，统计节点在搜索路径中某个位置连续停留的次数，若次数超出设定值（如 g），则认为该节点已经陷入局部最优点，并对该节点进行步骤 2 的操作.

步骤 2　判断节点是否是目前搜索到的最优节点，如果是，则不进行替换操作，以保证改进后算法的稳定性和收敛性. 否则，在解空间中随机生成一个备选节点来替换该节点.

在步骤 1 中，一般取 $g=l/n_i$，其中 l 代表节点的编码长度，n_i 为节点 i 临时传递网络的节点数目. 考虑到少数随机长连接搜索的存在，应用中应将 g 值稍微加大一些. 本文的仿真试验采用二进制编码，改进后的二进制编码小世界算法的伪代码（以最小化某一目标函数为例）如下：将初始化的参数设置为节点集规模 n、l、局部搜索概率 p_c、ℓ、n_i、最大允许搜索次数 $k_{\max}$、求解精度 ε 和 g.

```
Begin
产生初始节点集 S(0)={s1(0),s2(0),…,sn(0)};
    初始化变量 gi:gi=0(i=1,2,…,n)
```

$k\leftarrow 0$;
while $k<=k_{max}$ and $|f(e^{-1}(s^*(k)))-f^*|>\varepsilon$;
$S'(k)\leftarrow S(k)$;
for 每一个 $s_i'(k)\in S'(k)$
repeat n_i 次以下操作
依概率 p_c 进行：
局部搜索：$s_i'(k+1)\leftarrow\Psi(s_i'(k))$
或
随机长连接搜索：$s_i''(k+1)\leftarrow\Gamma(s_i'(k))$
end repeat
从以上操作得到的 n_i 个备选节点中选择一个最优节点 $s_i^*(k)$.
if$(e^{-1}(S_i^*(k))<f(e^{-1}(s_i(k))$
$s_i(k+1)\leftarrow s_i^*(k)$
end
$s_i(k+1)\leftarrow s_i^*(k)$
if $g_i<=g$
if $s_i(k+1)==s_i(k)$
$g_i=g_i+1$;
else
$g_i=0$;
end if
else
if $s_i(k)\neq s^*(k)$
随机产生一节点 s_i''，并对$s_i(k)$进行替换，
$s_i(k+1)\leftarrow s_i''$;
end if
$g_i=0$;
end if
end for
从更新后的节点集合中找出适应度最高的节点 $s^*(k)$
$k=k+1$;
end while
end

其中 g_i 为节点在局部最优解连续停留的次数，$s^*(k)$为第 k 代搜索到的最优节点，f^* 为测试函数的已知最优解，$s_i(k+1)\leftarrow s_i^*(k)$表示用较优节点 $s_i^*(k)$代替原节点 $s_i(k+1)$.

3 试验与讨论

3.1 测试函数与试验参数的选取

采用 Rosenbrock 函数[7] f_1、Camel 函数[7] f_2、Rastrigrin 函数[8] f_3、Shekel's Foxholes 函数[9] f_4、Schaffer's f_6 函数[8] f_5、Schwefel's 函数[10] f_6、Shubert 函数[11] f_7 和 Hansen 函数[11] f_8 这 8 个典型测试函数进行试验，测试函数的基本特征如表 1 所示，测试函数的具体表达式、三维空间特性，以及算法改进前后对其他函数的测试结果等详细内容可查阅文献[12]. 试验使用 Matlab 7.0 在 PIV3.06 G、1 G 内存的 PC 机上完成.

试验分别使用 S-SWA 算法和 M-SWA 算法对每个测试函数进行 50 次独立随机试验，除 M-SWA 算法中的 g 取 30 外，其他参数在 S-SWA 算法和 M-SWA 算法中的取值都一样，均以测试函数的最小值作为适应度函数. 将 S 的规模 n 取为 20，n_i 取为 3，节点邻域的 ℓ 取为 1，单个变量的二进制编码长度定为 20，p_c 为 0.8，最大迭代次数定为 1 000. 由这些参数设定值可知，试验中每迭代一次的函数值评价次数为 40.

3.2 试验结果

表 2 列出了上述试验的统计结果，对于不能次次找到最优值的函数，仅对其找到的情况进行相关项的统计. 表中的 O_{best}、N_{max}、N_{mean}、N_{std} 分别表示 S-SWA 算法在 50 次试验中搜索到最优解的次数、搜索到最优解所需的最大迭代次数、平均迭代次数和迭代次数之间的标准差. O'_{best}、N'_{max}、N'_{mean}、N'_{std} 分别表示 M-SWA 算法在 50 次试验中得到的最优解次数、最优解所需的最大迭代次数、平均迭代次数和迭代次数之间的标准差.

表 1 测试函数特征

函数	变量取值区间	最优适应度值	精度	函数	变量取值区间	最优适应度值	精度
f_1	$[-5.12,5.12]^2$	0.000 0	0.001	f_5	$[-10,10]^2$	0.00	0.000 1
f_2	$[-5.12,5.12]^2$	−1.031 6	0.001	f_6	$[-500,500]^2$	−837.96	0.500 0
f_3	$[-5.12,5.12]^2$	0.000 0	0.010	f_7	$[-10,10]^2$	−24.06	0.010 0
f_4	$[-65.536,65.536]^2$	0.998 0	0.001	f_8	$[-10,10]^2$	−176.54	0.100 0

表 2　函数仿真试验的结果统计

函数	O_{best}	O'_{best}	N_{max}	N'_{max}	N_{mean}	N'_{mean}	N_{std}	N'_{std}
f_1	7	28	945	928	533.57	428.39	356.56	259.28
f_2	50	50	777	141	63.12	44.90	110.15	28.80
f_3	48	50	798	322	183.29	97.26	205.92	74.46
f_4	50	50	947	361	103.20	65.16	191.65	54.25
f_5	50	50	931	435	145.40	117.28	164.01	104.59
f_6	50	50	450	235	90.30	75.68	91.72	49.88
f_7	45	50	988	754	238.64	180.28	246.90	168.55
f_8	47	50	648	473	172.51	130.00	162.32	108.50

3.3　分析与讨论

从表 2 中可以看出，M-SWA 算法与 S-SWA 算法相比，性能有了明显的提高. 首先，通过对比 O_{best} 和 O'_{best} 可以看出，除了 Rosenbrock 函数，对于其他 7 个测试函数 M-SWA 算法总能次次搜索到最优解，且对于 Rosenbrock 函数，M-SWA 算法搜索到最优解的次数明显大于 S-SWA 算法，充分说明了 M-SWA 算法的有效性. 其次，通过 N_{max} 和 N'_{max} 以及 N_{mean} 和 N'_{mean} 的比较可以看出，对于 8 个测试函数，除 Rosenbrock 函数外，M-SWA 算法搜索到最优解所需的最大迭代次数明显小于 S-SWA 算法，且对于每个测试函数 M-SWA 算法搜索到最优解所需的平均迭代次数均小于 S-SWA 算法，即 M-SWA 算法的快速性优于 S-SWA 算法. 最后，通过 N_{std} 和 N'_{std} 的对比可以看出，M-SWA 算法的稳定性要优于 S-SWA 算法.

4　结论与展望

本文在 S-SWA 算法的基础上，针对该算法中搜索节点陷入局部极值的问题，提出了对节点搜索寻优过程进行跟踪、对停滞节点进行替换操作的改进策略. 通过 8 个测试函数的优化仿真试验，将 S-SWA 算法和 M-SWA 算法进行了性能对比分析，各项数据的比较表明，M-SWA 算法可以有效地避免搜索节点陷入停滞状态，提高算法的搜索效率和鲁棒性，体现出了较好的解决复杂优化问题的潜能.

参考文献：

[1] 王正志，薄涛. 进化计算[M]. 长沙：国防科技大学出版社，2000：6.

[2] Watts D. Six degrees：the science of a connected age [M]. New York：WW Norton & Company，2004：19-100.

[3] Watts D J，Strogatz S H. Collective dynamics of small-world networks [J]. Nature，1998，393(4)：440-442.

[4] Kleinberg J M. The small-world phenomenon and decentralized search [J]. SIAM News，2004，37(3)：1-2.

[5] Kleinberg J M. Navigation in a small world [J]. Nature，2000，406(8)：845.

[6] 杜海峰，庄健，张进华，等. 用于函数优化的小世界优化算法 [J]. 西安交通大学学报，2005，39(9)：1011-1015.
Du Haifeng，Zhuang Jian，Zhang Jinhua，et al. Small-world phenomenon for function optimization [J]. Journal of Xi′an Jiaotong University，2005，39(9)：1011-1015.

[7] Price K V. Differential evolution：a fast and simple numerical optimizer [C]//Smith M H，Lee M A，Keller J，et al. 1996 Biennial Cenference of the North American Fuzzy Information Processing Society. Piscataway，USA：IEEE Press，1996：524-527.

[8] Eberhart R C，Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization[C]//Proceedings of the Evolutionary Computation. Piscataway，USA：IEEE Press，2000：84-88.

[9] 焦李成，杜海峰，刘芳，等. 免疫优化：计算、学习与识别 [M]. 北京：科学出版社，2006：402.

[10] Leung Y W，Wang Yuping. An orthogonal genetic algorithm with quantization for global numerical optimization [J]. Evolutionary Computation，2001，5(1)：41-53.

[11] Madsen K. Non-linear approximation and engineering design [EB/OL]. Lyngby，Denmark：Informatric and Mathematical Modeling Institute. [2006-12-21]. http：//www2. imm. dtu. dk/～km/GlobOpt/testex/testproblems. html.

[12] 杜海峰. 小世界优化算法测试函数及实验结果[EB/OL]. 西安：西安交通大学机电系统与测控技术研究所. [2007-02-15]. http：//202. 117. 58. 58/website/testData. htm.

（编辑　管咏梅）