用于 TSP 的自适应贪婪 GA 算法

陈张和[1] 洪 龙[1,2] 钱建屹[1]

（南京邮电大学计算机学院 南京 210003）[1] （软件开发环境国家重点实验室 北京 100191）[2]

摘 要 TSP 问题是一个典型的组合优化问题，很多现实生活中的问题都可以归结为 TSP 问题，GA 算法是一种典型的优化算法。通过对 GA 算法要点的分析，提出了一种自适应贪婪 GA 算法，以解决 TSP 问题。自适应适应度函数的各种定义、定理，确保了算法的正确性。通过平均复制的方法进行选择操作，使得算法不会过早地陷入局部最优。通过建立基于哈密顿回路的双向环贪婪插入算子进行交叉操作，确保了算法收敛的高效性。最后通过实例的计算分析及与传统 GA 算法的比较，说明了所提出的自适应贪婪 GA 算法在 TSP 研究中能够更好地发挥作用。

关键词 自适应适应度函数，平均复制，双向环贪婪插入

中图法分类号 TP301.6 **文献标识码** A

Adaptive Greedy GA Algorithm for TSP

CHEN Zhang-he[1] HONG Long[1,2] QIAN Jian-yi[1]

(College of Computer, Nanjing University of Posts & Telecommunications, Nanjing 210003, China)[1]

(State Key Laboratory of Software Development Environment, Beijing 100191, China)[2]

Abstract TSP is a typical combinatorial optimization problem, and many real life problems can attributed to the TSP. GA is a typical optimization algorithm. Analyzing GA's important points, a adaptive greedy GA was proposed to solve TSP. Definitions and theorems on adaptive fitness function ensure the correctness of the algorithm. Algorithm does not prematurely fall into local optimum because of average replication method for select operations. Algorithm can be efficiently converged by establishing bidirectional ring greed insert operator based on Hamiltonian two-way loop circuit for cross operation. Finally, the calculation and analysis of the example and the comparison with the traditional GA algorithm show that the proposed GA algorithm can play better role in TSP study.

Keywords Adaptive fitness function, Average copy, Bidirectional greed insert

1 引言

遗传算法是美国 Michigan 大学的 J. Holland 教授于 1975 年提出的。它是一种进化算法，其基本原理是仿效生物界中的“物竞天择、适者生存”的演化法则。遗传算法的具体步骤是把问题参数编码为染色体，再利用迭代的方式进行选择、交叉和变异等运算来交换种群中的染色体信息，最终生成符合优化目标的染色体。实践证明，遗传算法对于解决 TSP 等优化问题具有较好的寻优性能[1]，但其在解决优化问题中也存在不足，即在适应度函数选择不当的情况下其有可能收敛于局部最优，不能达到全局最优。

旅行商问题（Traveling Salesman Problem, TSP）是求加权完全无向图中访问每个顶点恰好一次且返回出发点的总权数最小的闭路，又称为最优哈密顿回路[8]。问题等价于求完全加权图中总权数最少的哈密顿回路。若选定出发点，对 n 个顶点进行排列，第二个顶点有 $n-1$ 种选择，第三个顶点有 $n-2$ 种选择，依此类推，共有 $(n-1)!$ 条哈密顿回路。考虑以相反顺序来经过一条哈密顿回路，需要检查 $(n-1)!/2$ 条哈密顿回路，并从中找出权和最小的一条[2]。

TSP 同时具有实践和理论的重要性，人们已经投入了巨大的努力来设计解决它的有效算法，但目前还没有实现。现已证明它是一个 NPC 问题[3]。不过可以通过一些智能算法进行近似的求解，比较典型的有遗传算法、神经网络、蚂蚁算法、模拟退火等。城市管道铺设优化、物流等行业中的车辆调度优化、制造业中的切割路径优化等现实生活中的优化问题都可以归结为 TSP 来求解，因此快速有效地求解 TSP 有着重要的意义。遗传算法作为一种典型的智能算法，能够很好地用来解决 TSP[4]。

2 自适应贪婪 GA 算法的主要思想

2.1 自适应适应度函数确定

遗传算法是以达尔文进化论中的“适者生存”作为依据建立的进化算法，所以适应度函数的确定，也就是“适者生存”中的个体对环境适应程度的度量，在整个算法中显得最为关键。

到稿日期：2011-07-24 返修日期：2011-11-24 本文受软件开发环境国家重点实验室开放课题（SKLSDE-2011KF-04）及国家高技术研究发展计划（863 计划）（2009AA043303）资助。

陈张和（1987－），男，硕士生，主要研究方向为智能算法、非经典逻辑及应用，E-mail：chenzhanghe2012@gmail.com；**洪 龙**（1952－），男，博士，教授，主要研究方向为人工智能、非经典逻辑及应用、计算机系统结构；**钱建屹**（1987－），男，硕士生，主要研究方向为计算机网络。

从生物界的繁衍发展来看，在物种进化的初始阶段，单个生物体之间对环境的适应度区别很大；随着种群进化，这种对环境适应度的区别将慢慢变小。这种现象在人类社会中表现得尤为明显。遗传算法中，由于算法的开始阶段种群是随机产生的，因此其中较差染色体的比例可能较高，优秀的基因片段可能比较少，这时我们希望能够适当加大区分以便能够更快淘汰掉较差的染色体，而保留优秀的染色体。当算法的迭代次数越来越大，种群中优秀染色体的比例变高，大部分的染色体中都含有很优秀的基因片段时，我们希望减小区分来更多地保留优秀的基因片段，而把淘汰较差染色体的工作更多地交给算法的其他操作[6]。

参考生物进化模型并综合考虑到遗传算法的要求，本文对适应度函数做出了定义。

记 $D(x_i,x_j)$ 为点 x_i 到点 x_j 的距离。设 N 个点之间存在哈密顿回路。显然，哈密顿回路的长度为

$$\sum_{1}^{N-1}D(x_i,x_{i+1})+D(x_N,x_1)$$

注意，此处点的序号应随哈密顿回路的不同而变化，换言之，每个确定点的下标不是固定的。

定义 1 记回路长度集合为 G，称 $f_m:G\rightarrow R$ 为关于 G 的适应度函数。若 $g,g_{\min},g_{\max}\in G$，则

$$f_m=\eta(g_{\max}-g_{\min})-(g-g_{\min})$$

式中，η 为调节系数。为了使适应度函数能够随着种群进化代数 t 自适应地做出调整，即上文所阐述的在进化的初始阶段加大区分、进化的成熟阶段减少区分，η 必须与 t 直接相关。同时考虑到算法进化完全时淘汰率变为 0 的要求，给出了 η 的具体形式，即 $\eta=1+(1/2)(t/T)^{\alpha}(\alpha\geqslant1)$，$t$ 表示当前的进化代数，T 通过人为设定，代表最大的进化代数，t/T 表示算法的进化程度；α 为调节因子，可以通过设定 α 的大小来调节种群淘汰率的变化快慢，淘汰率为 P。

定理 1 $f_m\geqslant0$。

证明：

$$\begin{aligned}f_m&=[1+(1/2)(t/T)^{\alpha}](g_{\max}-g_{\min})-(g-g_{\min})\\&=g_{\max}-g_{\min}-g+g_{\min}+(1/2)(t/T)^{\alpha}(g_{\max}-g_{\min})\\&=g_{\max}-g+(1/2)(t/T)^{\alpha}(g_{\max}-g_{\min})\ (\alpha\geqslant1)\end{aligned}$$

式中，$g_{\max}-g\geqslant0$；$(1/2)(t/T)^{\alpha}(g_{\max}-g_{\min})\geqslant0$，所以有 $f_m\geqslant0$。

定理 2 若 $g_{\min}=g_{\max}$，则 $f_m=0$。

证明：当 $g_{\min}=g_{\max}$ 时，有 $g=g_{\min}=g_{\max}$，显然，$f_m=\eta(g_{\max}-g_{\min})-(g-g_{\min})=0$。

定理 2 说明只有一条哈密顿回路时，不存在适应问题。

定义 2 设 $u(f_m)=f_m/((1/n)\sum f_m)$。若所有染色体的 f_m 不全为 0，则称 u 为染色体的淘汰判断函数。

定义 3 设 $x=Ran(f_m)$。若 $u(x)<\beta$，则称 $u(x)$ 为 f_m 所对应染色体的 β 淘汰。一般取 $\beta=0.5$。

定理 3 若进化代数 $t=0$，则淘汰率 $P=1/4$。

证明：令 $C_{\max}=g_{\max}-g_{\min}$，$C=g-g_{\min}$；

当 $t=0$ 时

$$\begin{aligned}f_m&=[1+(1/2)(t/T)^{\alpha}](g_{\max}-g_{\min})-(g-g_{\min})\\&=C_{\max}-C\end{aligned}$$

C 在 $[0,C_{\max}]$ 区间上服从均匀分布；易得概率密度函数 $F(x)=x/C_{\max}$；$f_m/((1/n)\sum f_m)<0.5$ 将 $f_m=C_{\max}-C$ 代入，解得 $C\in[(3/4)C_{\max},C_{\max}]$ 时会被淘汰；则淘汰率 $P=(C_{\max}-(3/4)C_{\max})/C_{\max}=1/4$；即 $t=0$ 时，种群的淘汰率 $P=1/4$。

定理 4 若进化代数 $t=T$，则淘汰率 $P=0$。

证明：令 $C_{\max}=g_{\max}-g_{\min}$，$C=g-g_{\min}$。

当 $t=T$ 时

$$\begin{aligned}f_m&=[1+(1/2)(t/T)^{\alpha}](g_{\max}-g_{\min})-(g-g_{\min})\\&=(3/2)C_{\max}-C\end{aligned}$$

C 在 $[0,C_{\max}]$ 区间上服从均匀分布；易得概率密度函数 $F(x)=x/C_{\max}$；$f_m/((1/n)\sum f_m)<0.5$ 将 $f_m=C_{\max}-C$ 代入，解得 $C>C_{\max}$ 时会被淘汰，而 $C\leqslant C_{\max}$，则淘汰率 $P=0$；即 $t=T$ 时，种群的淘汰率 $P=0$。

定理 5 种群淘汰率 P 是关于 t 的递增函数，且淘汰率 P 的变化率为 $-t/T$。

证明：令 $C_{\max}=g_{\max}-g_{\min}$，$C=g-g_{\min}$；文中取 $\alpha=2$。

$$\begin{aligned}f_m&=[1+(1/2)(t/T)^2](g_{\max}-g_{\min})-(g-g_{\min})\\&=[1+(1/2)(t/T)^2]C_{\max}-C\end{aligned}$$

$f_m/((1/n)\sum f_m)<0.5$ 将 $f_m=[1+(1/2)(t/T)^2]C_{\max}-C$ 代入上式，解得 $C\in((1/2)((3/2)+(1/2)(t/T)^2)C_{\max},C_{\max}]$ 时会被淘汰；可以得到关于 t 的概率函数 $P(t)=1-(1/2)((3/2)+(1/2)(t/T)^2)$。

在 $t>0$ 的区间内，这是一个单调递减函数，随着 t 变大，淘汰率变小；淘汰率的变化速度 $dP(t)/dt=P(t)'=-t/T$。

通过上述定理，能够确保在算法进化过程中：

(1)定理 1 能够确保所有染色体的适应度值不会出现负值，以便后续操作。

(2)定理 3、定理 5 能够确保在算法的开始阶段，每个染色体的适应值区分比较大，这样可以确保在较早的迭代过程中能够更快淘汰掉较差的染色体。虽然有可能淘汰一些优秀的基因片段，但综合考虑到效率，我们认为这样的牺牲是值得的，而且很有可能在后面的交叉操作中重新得到这些优秀的基因片段。

(3)定理 4、定理 5 能够确保随着算法的迭代，基本上每个染色体都可能拥有很优秀的基因片段。这些优秀的基因片段能够保留下来，那么我们设计的自适应适应度函数就有了用处：当进化代数较大的时候，淘汰率会比较低，这样有利于种群的多样性，能够保留更多的优秀基因片段。

2.2 平均复制选择策略

选择操作是建立在群体中个体的适应度值上的，即评价值大的个体能够被复制的概率较大；评价值小的个体能够被复制的概率较小。

参考人类社会的进化模型：当今的人类社会，即使特别精英的个体也不会非常大量地繁殖自己的后代，所以本文采取了一种平均复制策略，即越优秀的个体，拥有优先复制和多后代复制的优势；较弱势的个体也可能被复制，只是要等到优秀个体复制结束后再来判定是否被复制。平均复制策略兼顾了淘汰效率和种群多样性，能够使种群有一定的进化效率而又不使算法过早地陷入局部最优解。此复制策略和本文设计的交叉操作能够很好地结合。

2.3 双向环贪婪交叉操作

交叉操作是 GA 中起核心作用的操作。本文设计的双向环贪婪插入算法采用回路内局部贪婪策略进行抉择。对于两个待交叉的染色体，将一个作为基本哈密顿回路，另一个作为参考哈密顿回路。按照在参考哈密顿回路中节点的邻接顺

序，利用距离差函数作为评价标准，运用贪婪策略，逐步改变基本哈密顿回路的编码顺序，然后将基本哈密顿回路与参考哈密顿回路互换角色，改变参考哈密顿回路的编码顺序，从而提高经过交叉的两个子代哈密顿回路的适应度。

下面以两个染色体为例来阐述具体的过程。设两个染色体分别为 $X_i=(4,2,1,0,3,5)$，$X_j=(3,2,5,1,0,4)$，6 个节点之间的距离如表 1 所列。

表 1 6 节点之间的距离图

	P0	P1	P2	P3	P4	P5
P0	0	6	4	6	7	4
P1	6	0	6	4	4	5
P2	4	6	0	3	4	6
P3	6	4	3	0	2	4
P4	7	4	4	2	0	3
P5	4	5	6	4	3	0

将上述两个染色体所对应的哈密顿回路画出，如图 1 所示。

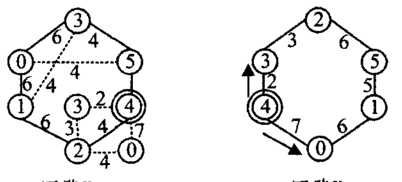

图 1 回路 X_i 和回路 X_j 的哈密顿回路

以回路 X_j 作为参考回路对回路 X_i 进行交叉操作：

(1)选取交叉操作开始点，从 X_i 回路的任何一个节点开始都可以，本文从 4 节点开始，在参考回路 X_j 中也找到编号为 4 的节点。

(2)因为 X_j 表示的是一个无向哈密顿回路，所以它可以向两个方向前进，在 X_j 中 4 节点的下一个节点可以是 3，也可以是 0。

(3)计算节点 3 和节点 0 在插入 X_i 的 4 节点后的距离差函数 $G_{43}(X_i)$ 和 $G_{40}(X_i)$ 将节点 3 虚拟插入 X_i 回路，此时原 X_i 回路中 3 节点就要删除，这样 0 节点和 5 节点直接连接。通过比较插入后回路的路程总和与未插入时的路程总和来确定是否插入，因为在插入过程中只影响了 4 节点和 3 节点附近的边，所以计算距离函数

$$G_{43}(X_i)=[D(4,3)+D(3,2)-D(4,2)]-[D(3,0)+D(3,5)-D(0,5)]$$

代入表格中的距离可以计算得 $G_{43}(X_i)=[2+3-4]-[6+4-4]=-5$。

这里的 -5 代表插入 3 节点后，距离总和比未插入时减少了 5。同样，计算出

$$G_{40}(X_i)=[D(4,0)+D(2,0)-D(4,2)]-[D(0,1)+D(0,3)-D(1,3)]$$

代入表格中的距离可以计算得 $G_{40}(X_i)=[7+4-4]-[6+6-4]=-1$。

这里的 -5 和 -3 说明插入节点后整个回路距离总和变小了，因为只需插入一个节点，所以选择最优的那个节点进行插入。可以看到插入节点 3 后距离总和减少了 5，比插入 0 节点后距离总和减少得多，所以选择插入节点 3。

插入策略：距离差函数 $G(X_i)$ 为负值，则选择距离差函数 $G(X_i)$ 最小的那个节点进行插入，其他情况下保持原来哈密顿回路不变。

(4)插入节点 3 后的 X_i 回路继续进行交叉操作，此时交叉操作的点变为了 X_i 中的 3 节点。依此顺序往下进行交叉操作，待所有的结点都完成至少一次的交叉操作后整个交叉过程完成。

(5)完成整个交叉过程后的 X_i 回路为 $X_i=(4,2,1,0,3,5)$。计算完成交叉操作的 X_i 的距离和 $\sum D(X_i)_2=23$；未进行插入操作的 X_i 距离和 $\sum D(X_i)_1=29$。比较发现，整个回路距离和变小了，染色体正朝着更优的方向进化。

(6)再将回路 X_i 作为参考回路对回路 X_j 进行交叉操作，最后使得整个种群染色体都进行交叉操作。

双向环贪婪插入算法的优势在于充分利用了父代染色体中优秀的基因片段，采取局部贪婪策略使得环向更优的方向进化。该算法既克服了一般交叉算法的随机性，又克服了局部贪婪算法的局限性，从而能够很好地避免算法陷入局部最优解。通过与平均复制选择算法相结合运用可以达到最好的效果。该插入算法可以使多个染色体同时进入匹配池，对一个染色体的交叉操作用多个参考染色体，采用并行计算使得整个算法计算更快。

3 算法设计和流程

遗传算法求解问题一般经过 7 个步骤，即染色体编码、种群初始化、适应度函数的确定、终止条件确定、选择操作、交叉操作和变异操作[5]。

算法的流程如图 2 所示。

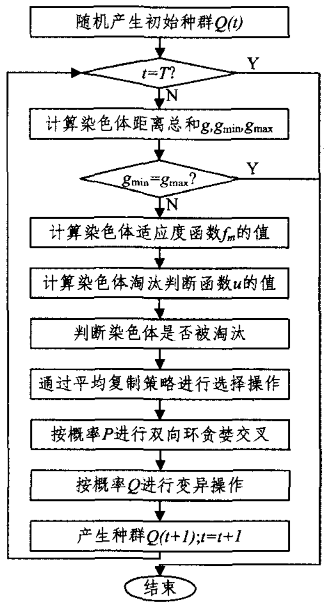

图 2 自适应贪婪 GA 的算法流程

算法采用字符编码，用字符来表示城市顶点，如 0 代表第 1 个顶点，9 代表第 10 个顶点，A 代表第 11 个顶点，依此类推。种群的数量视城市的规模而定，其取值在 50～200 之间。其中 T 是设置的最大迭代次数。

4 实例分析

以美国中部一个规模为 10 个城市的 TSP 问题为例进行实验，具体数据如表 2 所列。

表 2　美国 10 个城市的距离

城市	0	1	2	3	4	5	6	7	8	9
0 芝加哥	0	960	1050	500	410	860	460	290	560	700
1 达拉斯	960	0	780	490	940	510	640	630	410	370
2 丹佛	1050	780	0	600	840	610	540	860	760	510
3 塔萨斯	500	490	600	0	450	350	200	260	170	180
4 明尼阿波利斯	410	940	840	450	0	800	360	550	590	640
5 俄克拉荷马	860	510	610	350	800	0	460	500	280	80
6 奥马哈	460	640	540	200	360	460	0	450	370	300
7 圣路易	290	630	860	260	550	500	450	0	210	450
8 斯普林菲尔德	560	410	760	170	590	280	370	210	0	250
9 卫奇塔	700	370	510	180	640	80	300	450	250	0

通过最近邻法的求解，最优路径为(5,1,2,6,4,0,7,8,3,9)，距离和为 3530；通过本文设计的 GA 算法求解，最优路径为(0,7,3,8,1,9,5,2,6,4)，距离和为 3500。比较发现，本文设计的 GA 算法更好，且 3500 确实是最优路径。

下面的表格记录的是种群规模为 50 时，传统 GA 方法和本文设计的 GA 方法的结果比较，程序运行了 10 次。

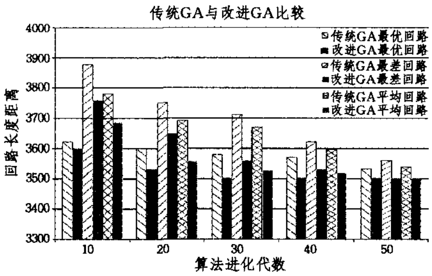

图 3　传统 GA 与自适应贪婪 GA 的比较

通过上面的实验数据，很容易发现本文设计的 GA 方法解决 TSP 是十分有效的。当进化代数变大时，本文 GA 算法的最优回路和最差回路长度都在变小，这说明算法是在朝着一个更优的方向前进，这也是 GA 方法所希望看到的。当算法进化到 50 代时，本文设计的 GA 方法已经找到了最优解，也就是回路长度为 3500 的那组解。

通过与传统 GA 方法的比较可以发现，本文设计的 GA 方法在执行 10 个城市的 TSP 时的表现要明显优于传统 GA 方法。当进化代数达到 50 时，本文的 GA 方法最优回路和最差回路已经趋同，而传统 GA 方法还未完全收敛，这说明本文设计的 GA 方法的收敛速度要快于传统 GA 方法，即当进化代数一定时本文的 GA 方法能更大可能地找到最优解；本文的 GA 方法还能更好地利用局部信息，并且能够很好地继承父代的优秀基因片段。

结束语　通过实验可以看出，本文提出的改进 GA 方法在求解 TSP 上相对于传统 GA 方法有着很大的优势。遗传算法是一种随机算法，但算法的每一步操作并不能被精确地控制。遗传算法能够计算的基础是适应度的确定，应该还能找到一些更好的确定适应度的方法。

参 考 文 献

[1] 陈国良. 遗传算法及其应用[D]. 北京：人民邮电出版社，1996

[2] 王桂平，王衍，任嘉辰. 图论算法理论、实现及应用[D]. 北京：北京大学出版社，2011

[3] 陈国良. 遗传算法及其应用[D]. 北京：人民邮电出版社，1996

[4] 玄光男，程润伟. 遗传算法与工程优化[D]. 北京：清华大学出版社，2004

[5] 王文杰，叶世伟. 人工智能原理及应用[M]. 北京：人民邮电出版社，2004

[6] 王小平，曹立明. 遗传算法：理论、应用与软件实现[M]. 西安：西安交通大学出版社，2002

[7] Chan K C C，Lee V，Leung H. Generating Fuzzy Rules for Target Tracking Using a Steady-State Genetic Algorithm[J]. IEEE Transactions on Evolutionary Computation，1997，1(3)：189-200

[8] Holland J H. Adaption in Natural and Artificial Systems[M]. The University of Michigan Press，Ann Arbor，1975

(上接第 178 页)

个对考生有较高指导价值的高考志愿在线模拟填报系统。

参 考 文 献

[1] 张巧. 商务智能发展现状与趋势分析[J]. 中国证券期货，2009，12(02)：14-17

[2] 胡翠华，陈登科. 商务智能在我国的发展现状、问题及其对策[J]. 科技管理研究，2007(10)：50-52

[3] 殷员分，张自力，蔡海敏，等. 数据仓库与 OLAP 技术在高考志愿数据分析中的应用[J]. 计算机科学，2010(5)：162-164

[4] 蔡海敏，张自力，曾铮，等. 基于数据仓库与联机分析技术的高考加分政策评估[J]. 计算机科学，2010，37(6)：223-225

[5] 曾铮，张自力，殷员分，等. OLAP 技术在高考志愿填报方式分析评估中的应用[J]. 西南师范大学学报：自然科学版，2010，35(3)：239-242

[6] Han Jia-wei，Kamber M. 数据挖掘：概念与技术(第 2 版)[M]. 范明，孟小峰，译. 北京：机械工业出版社，2007

[7] 吴远红. ETL 执行过程的优化研究[J]. 计算机科学，2007，34(1)：81-83

[8] Ralph K，Joe C. The Data WareHouse Toolkit：Practical Techniques for Extracting，Cleaning[M]. Wiley，2004：29-48

[9] Melome E. SQL Server 2005 Analysis Services 标准指南：中文版[M]. 武桂香，等译. 北京：电子工业出版社，2008：56-61

[10] George S，Sivakumar H. MDX 解决方案(第 2 版)[M]. 李仁见，董霖，等译. 北京：清华大学出版社，2008：10-18

[11] 吴亮奎. 寻找失落的基础：基础教育"基础"的反思[J]. 天津师范大学学报：基础教育版，2010(3)：1-4

[12] Larson B. Microsoft SQL Server 2005 商业智能实现[M]. 赵志恒，武海峰，译. 北京：清华大学出版社，2008：468-469

[13] MacLennan J，Crivat B，Tang Zhao-hui. Data mining with Microsoft SQL server 2008[M]. Indianapolis，IN：Wiley Pub.，2009

[14] 万永生. 高考加分政策的现状及思考[J]. 教学与管理，2009，36(10)：75-77

[15] 赵岩. 数据挖掘中的关联规则技术研究[D]. 西安：西安电子科技大学，2008

[16] Noyes B. Smart client deployment with ClickOnce：deploying Windows Forms applications with ClickOnce[M]. Upper Saddle River，NJ：Addison-Wesley，2007