

Ad Hoc 网络中基于遗传蚁群算法的 QoS 多播路由算法

邬长安, 邵 罕, 孙艳歌

WU Chang-an, SHAO Han, SUN Yan-ge

信阳师范学院 计算机与信息技术学院, 河南 信阳 464000

College of Computer and Information Technology, Xinyang Normal University, Xinyang, Henan 464000, China

E-mail: sunyg1982@yahoo.com.cn

WU Chang-an, SHAO Han, SUN Yan-ge. QoS multicast routing based on genetic algorithm and ant colony algorithm for Ad Hoc networks. *Computer Engineering and Applications*, 2008, 44(20): 107-110.

Abstract: Genetic algorithm has the ability of doing a global search quickly and stochastically. Ant colony algorithm has the ability of distributed parallel processing, and has good feedback capacity. The combination of both the algorithms can make full use of each advantage. In this paper, a hybrid algorithm GAAC is presented for solving QoS multicast routing problem in Ad Hoc networks. Firstly, it adopts genetic algorithm to give information pheromone to distribute. Secondly, it makes use of the ant colony algorithm to give the precision of the solution. In addition, the algorithm can judge the congestion by the average queue length quickly and avoid it. Simulation results show the algorithm has a better performance than the algorithm only based on ant colony algorithm.

Key words: Ad Hoc networks; QoS multicast routing; Genetic Algorithm and Ant Colony Algorithm (GAAC); congestion avoidance

摘 要: 利用遗传算法的快速全局搜索能力和蚁群算法的正反馈收敛机制, 把蚁群算法和遗传算法融合起来, 提出了一种遗传蚁群算法(GAAC)来解决 Ad Hoc 网络中 QoS 路由问题。首先利用遗传算法生成信息素分布, 然后用蚁群算法求精解, 优势互补, 期望获得优化性能和时间性能的双赢。并针对算法应用于 Ad Hoc 网络 QoS 路由由普遍产生的拥塞问题, 采用拥塞回避的策略, 从而实现网络业务流负载均衡。仿真表明该算法比其它单一采用蚁群算法进行路由选择更适合于动态 Ad Hoc 网络环境。

关键词: Ad Hoc 网络; QoS 多播路由; 遗传蚁群算法; 拥塞回避

DOI: 10.3778/j.issn.1002-8331.2008.20.033 文章编号: 1002-8331(2008)20-0107-04 文献标识码: A 中图分类号: TP393

1 引言

近年来, 随着各类多媒体业务及网络信息业务的普及和推广, 要求网络在带宽、时延等方面提供保障, 在 Ad Hoc 网络中提供 QoS 支持越来越重要。然而考虑到 Ad Hoc 网络中节点是移动的, 网络拓扑结构不断变化, 同时这些节点的计算能力和存储容量较低、能量受限, 如何设计出满足多个 QoS 约束条件的多播路由是一个具有挑战性的课题。

Ad Hoc 网络的 QoS 多播路由是一个满足多约束的问题, 是 NP-完全问题, 传统的路由算法难以有效解决, 可采用启发式算法来解决。蚁群算法是由意大利学者 M. Dorigo 通过模拟自然界蚂蚁寻径行为, 提出的一种全新启发式算法^[1]。该算法不依赖于具体问题的数学描述, 具有全局优化能力和本质上的并行性, 是解决 NP-完全问题的有效方法, 已被用于解决 QoS 路由问题^[2-4]。

遗传算法^[5]是以达尔文的生物进化论“适者生存、优胜劣汰”和孟德尔的遗传变异理论为基础, 模拟生物界进化过程, 它具有大范围快速全局搜索能力。

实验研究表明, 遗传算法在搜索的初期具有较高向最优解收敛的速度, 但对于系统中的反馈信息利用却无能为力, 当求解到一定范围时往往做大量的冗余迭代, 使得求精确解效率降低。而蚁群算法在搜索的初期由于缺乏信息素, 使得搜索速度缓慢, 但当信息素积累到一定的强度之后, 向最优解收敛的速度迅速提高。

基于以上分析, 本文提出了 (Genetic Algorithm and Ant Colony Algorithm, GAAC) 用于解决 Ad Hoc 网络中的 QoS 多播路由问题。算法的基本思想是利用遗传算法的快速全局搜索能力和蚂蚁算法的正反馈收敛机制, 初期采用遗传算法过程生成信息素分布, 后期利用蚂蚁算法正反馈求精确解, 从而获得优化性能和时间性能的双赢。

2 Ad Hoc 网络 QoS 多播路由模型

为便于分析问题, 特作以下一般性定义:

定义 1 网络拓扑可表示为加权图 $G=(V, E)$, 其中 V 为网络节点的集合, E 是节点间双向链路集合。对于任意链路 $e \in$

基金项目: 国家自然科学基金(the National Natural Science Foundation of China under Grant No.10671166)。

作者简介: 邬长安(1961-), 男, 副教授, 硕士生导师, 研究方向: 模式识别, 数字图像处理; 邵罕(1981-), 男, 助教, 研究方向: 计算机网络; 孙艳歌(1982-), 女, 硕士研究生, 研究方向为 Ad Hoc 网络, 智能计算。

收稿日期: 2008-01-31 修回日期: 2008-04-23

E , 有 $e=(v_i, v_j), v_i \in V, v_j \in V, i \neq j, v_i, v_j$ 为相邻节点。为适应网络拓扑动态变化, 节点周期性地交换 Hello 信息进行邻居探测, 维护网络拓扑。

定义 2 对任意链路 $e \in E$, 可用四元组 $(\text{delay}(e), \text{bandwidth}(e), \text{delay_jitter}(e), \text{cost}(e))$ 表示 QoS 特征值, 分别为延迟函数、带宽函数、延迟抖动函数和费用函数。对任意节点 $n \in V$, 用四元组 $(\text{delay}(n), \text{packet_loss}(n), \text{delay_jitter}(n), \text{cost}(n))$ 表示其 QoS 特征值, 分别为延迟函数、包丢失率函数、延迟抖动函数和费用函数。

定义 3 若 $p(s, d)$ 表示从源节点 s 到目标节点的一条路径, $T(s, M)$ 表示多播树, $s \in V$ 为一棵多播树的源节点, $M \subseteq \{V - \{s\}\}$ 为该多播树的端节点或叶节点集。则有以下关系: 实际上, QoS 多播路由问题是寻找一棵多播树 $T(s, M)$ 满足以下约束条件:

$$\begin{aligned} \text{delay}(p(s, d)) &= \sum_{e \in p(s, d)} \text{delay}(e) + \sum_{n \in p(s, d)} \text{delay}(n) \\ \text{bandwidth}(p(s, d)) &= \min\{\text{bandwidth}(e)\} \\ \text{delay_jitter}(p(s, d)) &= \sum_{e \in p(s, d)} \text{delay_jitter}(e) + \sum_{n \in p(s, d)} \text{delay_jitter}(n) \\ \text{packet_loss}(p(s, d)) &= 1 - \prod_{e \in p(s, d)} (1 - \text{packet_loss}(e)) \\ \text{cost}(T(s, M)) &= \sum_{e \in T(s, M)} \text{cost}(e) + \sum_{n \in T(s, M)} \text{cost}(n) \end{aligned}$$

实际上, QoS 多播路由问题是寻找一棵多播树 $T(s, M)$ 满足以下约束条件:

- (1) 延迟约束: $\text{delay}(p(s, d)) \leq D, D$ 为最大时延。
- (2) 带宽约束: $\text{bandwidth}(p(s, d)) \geq B, B$ 为瓶颈带宽。
- (3) 延迟抖动约束: $\text{delay_jitter}(p(s, d)) \leq DJ, DJ$ 为最大时延抖动。
- (4) 包丢失率约束: $\text{packet_loss}(p(s, d)) \leq L, L$ 为最大包丢失率。

并使得多播树 $T(s, M)$ 的总费用 $\text{cost}(T(s, M))$ 最小。

3 基于遗传蚁群算法的 QoS 多播路由算法

GAAC 算法的基本思想是首先由遗传算法产生较优解, 较优的路径留下信息素, 其它路径不改变; 然后在有一定初始信息素分布的情况下再按照蚁群算法, 充分利用蚁群算法并行性、正反馈性求精解。这样, 既能发挥遗传算法与蚁群算法在寻优搜索中各自的优势, 又能克服遗传算法在搜索到一定阶段时最优解搜索效率低以及蚁群算法初始信息素匮乏的问题。

GAAC 算法能够有效地选择一条最优路径, 但忽视了实际网络的最优路径一旦形成所有的数据都从最优路径传输的问题。这样一来, 当负载较大时发生堵塞的可能性就比较大。一旦堵塞发生, 消除它需花很长时间, 采用堵塞回避的策略可以获得较好的效果, 从实现网络的负载均衡。

3.1 GAAC 中的遗传算法规则

3.1.1 遗传编码

文献[2]中 Xiang 等提出一种通用的遗传算法, 但算法采用 $N \times N$ 的一维二进制编码机制(其中 N 为网络节点数), 这种编码模式使得算法编码、解码过程复杂, 并且算法的搜索空间随网络规模增大而急剧增大, 算法效率很低。本文根据网络路由的特点和遗传算法的编码原则, 采用节点序列编码, 即对给定

的路径, 以该路径经过的物理节点标识(ID 号)序列作为该路径的编码值。例如: 开始节点为 2, 终止节点为 9 的一条路径编码为(2 4 8 3 1 0 9), 并保持不变。这种编码方式自然、直观。

3.1.2 适应度函数

适应度函数可简单地等价于蚁群算法过程中的目标函数, 采用罚函数的形式:

$$f(p(s, d)) = \frac{1}{\text{cost}(p(s, d))} (Af_d + Bf_{dj} + Cf_{pl})$$

$$f_d = \Phi_d(\text{delay}(p_k(r, s)) - D)$$

$$f_{dj} = \Phi_{dj}(\text{delay_jitter}(p_k(r, s)) - DJ)$$

$$f_{pl} = \Phi_{pl}(\text{packet_loss}(p_k(r, s)) - PL)$$

$$\Phi_d(Z) = \begin{cases} 1 & Z \leq 0 \\ r_d & \text{否则} \end{cases} \quad \Phi_{dj}(Z) = \begin{cases} 1 & Z \leq 0 \\ r_{dj} & \text{否则} \end{cases}$$

$$\Phi_{pl}(Z) = \begin{cases} 1 & Z \leq 0 \\ r_{pl} & \text{否则} \end{cases}$$

其中, A, B, C 分别为 f_d, f_{dj}, f_{pl} 的加权系数, 表示延时、延时抖动和包丢失率在目标函数中所占比重。 $\Phi_d(Z), \Phi_{dj}(Z)$ 和 $\Phi_{pl}(Z)$ 分别是时延度量、时延抖动度量和包丢失率的惩罚函数, 当路由满足约束条件时值为 1, 否则值分别为 r_d, r_{dj} 和 r_{pl} , 是 $(0, 1)$ 范围内的值。 r_d, r_{dj}, r_{pl} 的大小决定惩罚的程度, 在本算法中取值都为 0.5。

3.1.3 初始群体的生成

采用随机方法从中选择若干个体组成初始种群, 其具体做法是: 首先精简网络, 删除不满足 QoS 约束条件的节点及与之相连的链路, 再删除不满足带宽要求的链路, 得到一个新的网络拓扑。基于此拓扑图进行搜索, 从源节点开始, 随机地选择一个与之相关联的节点, 将两点相连; 然后从选择的节点继续随机地选择与其关联的下一个节点, 在连接时要判断是否有回路, 如有回路则重新进行节点的选择, 一直这样做下去直至搜索到目的节点。

3.1.4 选择算子

本遗传算法采用最佳保留选择机制, 即首先将当前的解群体中适应度最高的个体结构完整地复制到下一代群体中, 然后按照轮盘赌选择机制执行选择功能。

(1) 交叉算子

针对节点序列编码, 采用如图 1 所示的路径交叉算子。其中 s 为源节点, d 为目的节点。

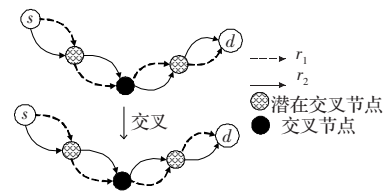


图 1 路径交叉算子

路径交叉操作如下:

- ① 将路径 r_1 和 r_2 共同经过的节点(源节点和目的节点除外)作为备选交叉节点;
- ② 从备选交叉节点中随机选取一点作为此次路径交叉的交叉节点;
- ③ 将 r_1 和 r_2 位于交叉节点后的子路径进行交换得到新的 r_1 和 r_2 路径。

当两条路径之间没有备选交叉节点时, 则不做路径交叉,

否则, 将破坏已有的模式, 使遗传算法的收敛速度变慢。

(2) 变异算子

采用图 2 所示路径变异算子, 其中 r_1 为经过变异节点的原始路径, r_2 为变异节点的前驱和后继节点之间的最短路径(只考虑 $cost$ 权值)。



图 2 路径的变异算子

路径变异操作如下:

① 从原始路径的中间节点随机选取一变异节点 n_m ;

② 将变异节点的前驱和后继节点以适应度函数为指标, 将使适应度值最大的路径作为新的变异路径。只有当适应度值有提高的才接受下来, 否则无效。

3.2 GAAC 中的蚁群算法规则

规则 1 路径选择规则。

在节点 i 的第 k 只蚂蚁选择下一跳节点 j 的规则:

$$j = \begin{cases} \arg \max_{j \in N_i} \{ [\tau_{i,j}(t)]^\alpha [\eta_{i,k}(t)]^\beta \} & \text{若 } q \leq q_0 \\ \text{根据公式(2)选择 } j & \text{否则} \end{cases} \quad (1)$$

其中, q 是 $(0, 1)$ 中均匀分布的随机数, 在选择下一跳之前由随机实验产生。 $q_0 \in [0, 1]$ 是一个确定选择最佳个体的概率, 取 q_0 为 0.7。该策略增强了搜索的多样性, 以避免过早陷于搜索停滞。

$$p_{i,j}^k(t) = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha [\eta_{i,k}(t)]^\beta}{\sum_{s \in N_i} [\tau_{i,s}(t)]^\alpha [\eta_{i,s}(t)]^\beta} & j \in N_i \\ 0 & \text{否则} \end{cases} \quad (2)$$

其中, N_i 表示 v_i 的邻居节点的集合; $\tau_{i,j}(t)$ 表示 v_i 到 v_j 链路的信息素量; α 为信息启发因子; β 为期望启发式因子; $\eta_{i,j}$ 为启发函数, 构造为 $\eta_{i,j}(t) = 1/cost(i,j)$, 即表示节点倾向于选择费用小的链路。

规则 2 信息素的初值设置规则。

通过遗传算法得到了一定的路径信息素, 因此把信息素的初值设置为:

$$\tau_s = \tau_c + \tau_g \quad (3)$$

其中, τ_c 是一个根据具体求解问题规模给定的一个信息素常数, 相当于 MMAS 算法中的 τ_{max} ; τ_g 是遗传算法求解结果转换的信息素值。

规则 3 信息素更新规则。

采用局部更新和全局更新相结合的策略, 局部更新可避免蚂蚁在上次最好的路径有限相邻区域内搜索; 全局更新时对历史最优解的路径上的信息素进行更新。

当蚂蚁 k 成功地完成从 v_i 到 v_j 的一跳, 信息素按如下公式进行更新:

$$\tau_{i,j}(t+1) = (1-\rho)\tau_{i,j}(t) + \rho\Delta\tau_{i,j}(t) \quad (4)$$

$$\Delta\tau_{i,j}(t) = \sum_{k=1}^m \Delta\tau_{i,j}^k(t)$$

$$\Delta\tau_{i,j}^k(t) = \frac{Q_1}{cost(p(s,d))} \quad (5)$$

其中, $0 < \rho < 1$, 表示局部信息素挥发系数, Q_1 是常量, 用于调整

信息素强度。

当所有的蚂蚁完成了一次循环后, 选择出目标函数数值最大的路径, 用来完成全局信息素更新。以使得较优解保留下来, 并对后继蚂蚁的行为产生影响, 以达到较快收敛于最优解的目的。使目标函数值最大的路径为最优路径, 其值为 F_{best} 。

$$\tau_{i,j}(t+n) = (1-\alpha)\tau_{i,j}(t) + \alpha\Delta\tau_{i,j}(t) \quad (6)$$

其中, α 为全局信息素挥发系数, $\Delta\tau_{i,j}(t)$ 表示信息素的修改, 其计算公式如下:

$$\Delta\tau_{i,j}(t) = \begin{cases} \frac{Q_2}{L_{best}} & t+n \text{ 过程中的最优路径} \\ 0 & \text{否则} \end{cases}$$

其中, Q_2 是常量。

规则 4 拥塞回避规则。

通过节点的平均队列长度来衡量网络拥塞状态。节点监视自己当前的 MAC 层接口队列, 以便于向网络层实时提供传输信道的拥塞信息。平均缓存队列长度设定阈值 Q_{thresh} , 当 v_i 的队列长度 Q_i 超过这个阈值则认为节点处于拥塞状态, 就立即向自己的上游节点发送拥塞通告信息, 使上游节点尽量避免使后续分组的传输路径经过该拥塞节点, 以缓解节点的拥塞程度。

3.3 算法的实现过程

GAAC 算法的实现步骤如下:

步骤 1 精简网络拓扑, 删除不满足 QoS 约束条件的节点以及与之相连的链路, 再删除不满足带宽要求的链路, 得到一个新的网络拓扑, 基于此拓扑图进行搜索;

步骤 2 随机产生一组实数编码;

步骤 3 根据遗传算法进行若干次迭代, 依据目标函数, 生成网络初始信息素分布;

步骤 4 根据公式(3)初始信息素分布, 在源节点放置 m 只蚂蚁, 每个蚂蚁依据状态转移概率选择下一跳节点;

步骤 5 蚂蚁完成一步后, 根据公式(4)进行局部信息素更新, 同时按照拥塞回避规则回避拥塞节点;

步骤 6 所有的蚂蚁完成一次循环后, 根据公式(6)进行全局信息素更新;

步骤 7 若满足结束条件, 即循环次数 $N_c \geq N_{max}$ 则循环结束并输出最优解; 否则跳转到步骤 4。

4 仿真实验与结果分析

为了更好地测试算法性能, 做了以下仿真实验: 模拟一个具有 100 个主机随机分布在 1 000 m x 1 000 m 区域的网络, 源节点和目标节点从网络中随机选取, 采用 Random Waypoint 运动模型, 节点的移动速度在 0~20 m/s 范围内变化。MAC 层采用 IEEE802.11DCF, 节点覆盖范围为 250 m, 信道容量为 2 Mb/s, 包长为 512 Byte, 节点的业务流服从泊松分布, 节点缓存队列阈值 Q_{thresh} 为 40, 仿真时间为 500 s, 结果是 10 次仿真的平均值。参数取值为: $M=30$; 各个路径信息素初值 τ_c 设为 60, 遗传算法求解结果转换的信息素是经过路径信息素加 2, $\alpha=1, \beta=2, Q_1=100, Q_2=10, \rho_1=\rho_2=0.5, \tau_{i,j}(0)=10, N_{max}=200$ 。

为全面比较算法的性能, 仿真使用端到端时延和包投递率这两个性能评价参数, 与 Ant-AODV 算法性能进行比较。

第一个场景是节点的停留时间在 0~900 s 变化, 停留时间的长短反映了网络拓扑的变化速率。结果图 3、图 4 所示:

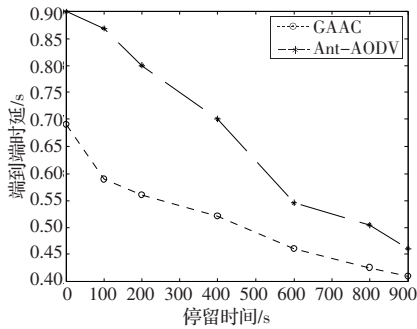


图3 不同停留时间下的端到端时延

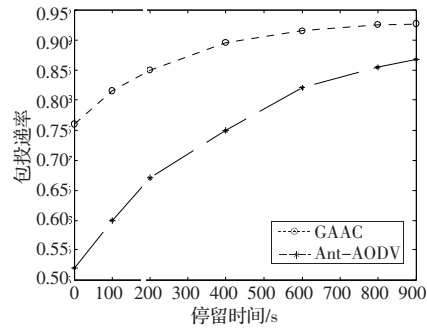


图4 不同停留时间下的包投速率

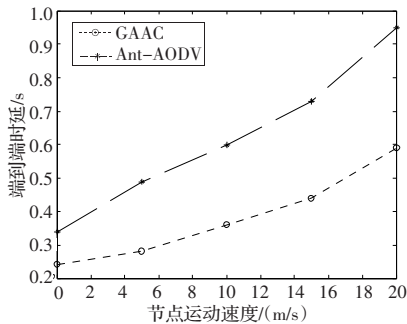


图5 不同移动速度下的端到端时延

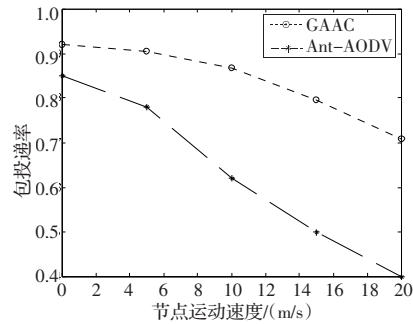


图6 不同移动速度下的包投速率

从图3可以看出,停留时间在0~900 s内,GAAC表现出比Ant-AODV更好的整体性能。在暂停时间0 s到100 s的变化区间上,GAAC维持了较低的传输时延,显示出了比Ant-AODV更好的时延性能,但是在停留时间较大的区间,因为随着停留时间的增大,网络拓扑结构稳定,GAAC在时延方面的优势不复存在。这说明GAAC更适合于动态的网络环境。

这和我们预想的结果一致,是由它们的实现机制决定的。在寻找路由过程中,首先利用遗传算法的快速性、随机性、全局收敛性等特点来生成初始信息素分布,然后利用蚁群算法的并行性、正反馈性、求解效率高等特点求精解,从而可在较短的时间内找到满足多个QoS约束的路由,减少了端到端的时延。而随着停留时间的增大,网络拓扑结构变得越稳定,平均端到端时延越来越小。

图4表明了,随着停留时间的增大GAAC包投递率在增大,因为停留时间增大以后,网络拓扑结构比较稳定。而且GAAC包投递率一直比Ant-AODV要高,实现了较高的包投递率。同时较高的包投递率,也表明GAAC算法的可靠程度更高。

第二个场景是不同移动速度下三种算法端到端时延和包投递率的比较,反应了算法对动态网络环境的适应情况,如图5、图6所示。

图5表明,在静止环境下两个算法的端到端时延差别不大,随着节点移动速度的增加,Ant-AODV端到端时延越来越大,而GAAC在增加到一定程度后,趋势是渐缓的。显示出GAAC对网络结构变化良好的适应性。

图6表明,随着网络变化速率和节点移动速率的增加,两种算法的包投递率都是下降的。这是符合移动网络的特点的,运动场景越激烈,路由的维护难度就越大,必然导致包投递率的下降。而对于GAAC,本身是为了更好适应高速动态变化的网络而设计的。可以发现节点运动很激烈的时候,它依然能够

保持一定的数据发送率。GAAC包投递率明显高于Ant-AODV,节点在任何移动状态都达到70%以上。

5 结论和今后的工作

本文针对Ad Hoc网络QoS多播路由问题,给出了一个具有多个约束条件的QoS多播路由模型,并提出了一种基于遗传蚁群算法的QoS多播路由算法。并对于遗传蚁群算法应用于Ad Hoc网络QoS多播路由时产生的拥塞问题,采取拥塞回避策略,从而实现了业务流的均衡分配。仿真结果表明,比其它单一采用蚁群算法进行路由选择更适合于动态Ad Hoc网络环境。但本文算法没有考虑Ad Hoc网络节点能量受限的特点,所以下一步将对遗传蚁群算法考虑节能的QoS路由做进一步的研究。

参考文献:

- [1] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.
- [2] Fei Xiang, Luo Junzhou, Wu Jieyi, et al. QoS routing based on genetic algorithm[J]. Computer Communications, 1999, 22(15): 1394-1399.
- [3] Hussein O, Saadawi T. Ant routing algorithm for mobile ad-hoc networks (ARAMA)[C]// Proc of the 2003 IEEE International Conference on Performance.
- [4] Liu Z, Kwiatkowska M Z, Constantinou C. A biologically inspired QoS routing algorithm for mobile ad hoc networks[C]// AINA 2005: the 19th International Conference on Advanced Information Networking and Applications, 28-30 March 2005, 1: 426-431.
- [5] Holland J H. Adaptation in natural and artificial systems[M]. [S.l.]: The University of Michigan Press, 1975.