

一种求解有状态服务选取问题的遗传算法

赵秀涛, 张 斌, 孙若男, 葛 亮

(东北大学 信息科学与工程学院, 辽宁 沈阳 110819)

摘 要: 基于 QoS 的 Web 服务选取问题, 通常认为应用 workflow 中的任务是相互独立的, 而在很多实际应用中, 工作流的某些任务之间往往需要共享状态信息, 由此增加了任务绑定约束, 使得求解复杂度提高, 影响了选取效率. 针对现有方法的不足, 提出了一种面向有状态服务选取的遗传算法, 其中重新定义了交叉操作和变异操作, 使得所有个体均满足任务状态关联绑定约束, 同时在个体评价策略中引入罚函数, 并进行个体相似性判断以防止过早收敛. 实验表明, 提出的算法在有状态服务选取问题中, 可求得质量良好的解, 且收敛速度快, 选取效率亦优于现有算法.

关 键 词: 服务选取; 服务质量; 有状态服务; 状态关联绑定约束; 遗传算法

中图分类号: TP 311.5 **文献标志码:** A **文章编号:** 1005-3026(2015)01-0001-05

A Genetic Algorithm for Stateful Service Selection

ZHAO Xiu-tao, ZHANG Bin, SUN Ruo-nan, GE Liang

(School of Information Science & Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: ZHANG Bin, E-mail: zhangbin@mail.neu.edu.cn)

Abstract: Tasks in the workflow of an application are generally considered to be independent of each other in current web service selection based on QoS. In practice, however, state information is often shared among some tasks in the workflow, which adds binding constraints between tasks and web services, resulting in higher time complexity and lower selection efficiency. Aiming at drawbacks of the existing methods, a genetic algorithm for stateful service selection was proposed. In the proposed algorithm, genetic operations including crossover and mutation were redefined in order to make all individuals meet state-correlate binding constraints among tasks. In addition, to prevent premature convergence, penalty function was introduced into individual evaluation strategy; moreover, similarity judgment between individuals was also included in the algorithm. The experiments results showed that with regards to stateful service selection, good solution and fast convergence rate can be obtained using the proposed algorithm; furthermore, the proposed algorithm is more efficient than the existing algorithms.

Key words: service selection; QoS (quality of service); stateful service; state-correlate binding constraints; genetic algorithm

在面向服务的应用中, 面对众多具有相同功能且服务质量 (QoS) 不同的候选服务, 如何为应用工作流的每个任务选择并绑定一个服务, 以达到在满足用户对组合服务 QoS 约束的基础上最优组合服务的全局 QoS, 依然是基于 QoS 的 Web 服务选取问题的研究热点^[1].

当前研究多假定任务间相互独立, 而忽略了

任务间的状态关联关系^[2]. 在 Web 服务中, 状态是一些不包含在请求消息中的信息片段^[3]. 实际工作流中的某些任务之间往往是状态关联的, 它们彼此共享某些状态信息. 为实现任务间的状态共享, 要求以服务包含的操作为基本选取粒度, 且具有状态关联关系的任务绑定的操作必须包含于同一服务, 这种允许不同操作间共享状态信息的

服务称作有状态服务 (stateful service)^[4]. 目前对面向有状态服务的 Web 服务选取问题研究较少. 文献[4]研究了多个用户并发访问组合服务时的服务选取, 通过在运行中限制共享状态任务集中后续任务的操作绑定关系保证有状态约束; 文献[5]将服务选取问题构建为一个混合整数规划问题, 并将共享状态的任务绑定到相同服务作为最优化问题的一个约束. 这些研究均采用线性规划进行求解, 存在计算代价大、可扩展性差等不足. 为了降低计算代价, 可采用启发式算法求解, 其中遗传算法已被证明是一种有效的 Web 服务选取方法^[6], 然而面对选取问题中的有状态服务, 现有遗传算法无法直接应用.

针对上述问题, 本文提出了一种求解面向有状态服务的 Web 服务选取问题的遗传算法, 其中定义了新的交叉操作和变异操作, 使得每一代个体均满足任务之间的绑定约束, 同时在适应度函数中引入了罚函数, 以适应多个约束条件产生的复杂搜索超平面的情形. 为了防止过早收敛, 算法在选择操作之前增加了相似性判断过程, 从而增加群体多样性. 与线性规划方法相比, 该算法具有全局搜索最优解的能力和良好的可扩展性. 实验表明, 本文的服务选取算法与现有遗传算法相比可以获得质量更好的解, 同时在保证解的质量的基础上, 其执行效率优于当前求解有状态服务选取的线性规划算法.

1 面向有状态服务的 Web 服务选取问题及优化模型

1.1 面向有状态服务的 Web 服务选取问题

如图 1 所示, 任务 T_1, T_2, \dots, T_7 构成一个应用的工作流, 任务的功能通过绑定某个操作 (如 op11, op31 等) 来实现, 这些操作包含于不同服务, 如 S_1, S_3 等. 某些任务之间存在状态共享关系, 如 T_3, T_4 与 T_2, T_5, T_7 分别共享不同状态. 由于状态不包含在服务的请求消息中, 为了实现状态共享, 具有状态关联的任务绑定的操作必须属于同一服务. 服务 S_3, S_6 分别包含了实现其

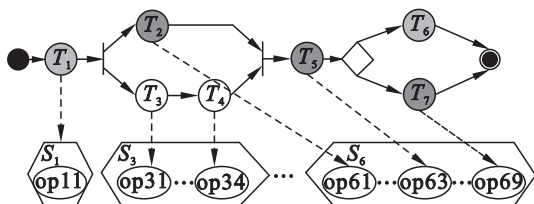


图 1 面向有状态服务的 Web 服务选取问题示例

Fig. 1 An example of stateful service oriented selection

任务功能且共享状态的操作, 称为有状态服务.

为便于描述, 给出以下定义.

定义 1 抽象组合服务 (abstract composite service, 简称 ACS). ACS 表示为一个二元组 $\langle T, SR \rangle$, 其中, $T = \{T_i | 1 \leq i \leq m, i \in \mathbf{N}\}$ 为任务集合, m 是任务数量; $SR = \{seq, loop, and, or\}$, 表示 ACS 包含的结构关系, 即序列结构、循环结构、并行结构和选择结构^[4].

定义 2 状态关联绑定约束 (state-correlate binding constraints, 简称 SBC). 对于 ACS 中的任意两个任务 T_i, T_j , 如果在实例化 ACS 时, 要求二者绑定的操作必须属于同一服务且共享相同的状态, 则称任务 T_i, T_j 之间存在状态关联绑定约束.

因此, 面向有状态服务的 Web 服务选取问题的目标是, 在满足给定的抽象组合服务端到端 QoS 约束和任务间状态关联绑定约束的基础上, 找到所有任务与操作之间的最优绑定关系.

定义 3 有状态任务 (stateful task). 对于 ACS 中的任意任务 T_i , 如果 ACS 中存在与具有状态关联绑定约束的任务, 则称为有状态任务.

相应地, 如果 ACS 中不存在与 T_i 具有状态关联绑定约束的任务, 则称 T_i 为无状态任务.

根据状态共享关系的不同, 可将抽象组合服务 ACS 中的全部有状态任务 $T_i (1 \leq i \leq m)$ 划分为多个互不相交的集合 O_1, O_2, \dots, O_h , 每个集合 O_e 中的元素共享相同的状态 rp_e , 称 O_e 为状态 rp_e 的有状态任务组. 可知, 任意 $O_i, O_j (i \neq j)$ 共享的状态都不同, 即 $rp_i \neq rp_j$. 在实际应用中, ACS 的任务间状态关联关系通常是由工作流的设计人员确定的, 因此与已有研究相同, 本文假定状态关联绑定约束是已知的.

1.2 面向有状态服务的 Web 服务选取优化模型

为了判断不同任务与操作绑定关系的优劣, 需要确定用户关注的 Web 服务的 QoS 属性^[7], 为简单起见且不失一般性, 仅考虑 3 个常用 QoS 属性: 响应时间 (response time)、可靠性 (reliability) 以及服务费用 (cost). 其中, 响应时间 t_R 为服务接受请求后, 处理并返回结果所消耗的时间; 可靠性 R 为一段时间内服务成功执行的概率; 服务费用 C 为每次调用服务所产生的费用.

根据上述 QoS 属性, 可以采用不同方法计算组合服务的 QoS 属性值^[1,4,8]. 对于任意服务, 如果其中某一操作提供不同 QoS 等级, 则根据 QoS 等级将其分解为多个功能相同的操作. 下面给出面向有状态服务的 Web 服务选取问题的优化模型.

给定一个包含 m 个任务的 ACS, 有状态任务组的集合为 $O = \{O_e \mid 1 \leq e \leq h\}$, 任务 $T_i (1 \leq i \leq m)$ 的候选操作集合 $OP_i = \{op_{i,1}, op_{i,2}, \dots, op_{i,n_i}\}$, 其中 n_i 表示候选操作数量, 所有任务的候选操作所属的服务构成集合 $CS = \{s_1, s_2, \dots, s_M\}$, 用户 u 对组合服务的全局 QoS 约束为: 最大响应时间不超过 $t_{R,\max}^u$, 最小可靠性高于 R_{\min}^u , 最大服务费用低于 C_{\max}^u . 服务选取的目标是同时最优响应时间、可靠性和服务费用. 以 $\mathbf{x} = (op_{1,j_1}, op_{1,j_2}, \dots, op_{m,j_m})$ 表示可行解, 则优化模型为

$$\max F(\mathbf{x}) = \sum_{\tau \in \{1,2,3\}} w_{\tau} f_{\tau}(\mathbf{x}). \quad (1)$$

$$\text{s. t. } t_R(\mathbf{x}) \leq t_{R,\max}^u, \quad (2)$$

$$R(\mathbf{x}) \geq R_{\min}^u, \quad (3)$$

$$C(\mathbf{x}) \geq C_{\max}^u, \quad (4)$$

$$\omega(op_{i,j_i}) = \omega(op_{k,j_k}),$$

$$op_{i,j_i}, op_{k,j_k} \in \bigcup_{l=1}^m OP_l, \quad (5)$$

$$\forall T_i, T_k \in O_e, 1 \leq e \leq h.$$

式(1)中, w_1, w_2, w_3 分别表示用户给出的 t_R, R, C 的权重, 且 $w_1 + w_2 + w_3 = 1$; f_1, f_2, f_3 分别为 $t_R(\mathbf{x}), R(\mathbf{x})$ 和 $C(\mathbf{x})$ 标准化后的值. 式(2) ~ 式(4)分别表示可行解 \mathbf{x} 满足响应时间、可靠性和服务费用的全局约束. 式(5)表示任务状态关联绑定约束, 其中 $\omega(op_{i,j_i}), \omega(op_{k,j_k})$ 分别是指操作 op_{i,j_i}, op_{k,j_k} 所属的服务, 其意义是, 对于属于同一有状态任务组 O_e 的任意两个任务 T_i, T_k , 它们绑定的操作 op_{i,j_i} 和 op_{k,j_k} 所属的服务是相同的, 且二者共享相同的状态 rp_e .

2 基于面向有状态服务选取的遗传算法求解最优解

提出了一种求解面向有状态服务选取问题的遗传算法 (stateful service oriented selection genetic algorithm, 简称 SSOSGA).

针对任务状态关联绑定约束, SSOSGA 在个体编码中要求有状态任务组中任务对应的基因位取值所属的服务相同. 同时, 需要在交叉操作中确定可用的交叉点, 并在变异操作中要对有状态任务组的基因位同时变异, 使得交叉和变异产生的新个体均属于可行解空间, 进而提高求解效率.

为了降低在解空间中无对应可行解的个体的适应度, 使其遗传到下一代群体中的概率减小^[9], 在 SSOSGA 中采用包含罚函数的个体评价策略. 与此同时, 采用相似性判断^[10] 增加群体多样性和高适应度值个体的主导地位以防止 SSOSGA

过早收敛.

1) 个体编码. SSOSGA 中的个体采用符号编码方式, 个体长度为 ACS 的任务数 m . 将基因位按照共享状态的类型分组, 每个基因位的序号 G_i 与 ACS 中的某个任务 T_j 具有对应关系, $T_j = \Omega(G_i)$, 且第 i 个基因位的取值范围是 $[1, m_i]$, 其中 m_i 为第 i 个基因位对应任务的候选操作个数, 而且每个有状态任务组对应的基因位取值所属的服务相同. 图 2 给出了个体编码示意图.

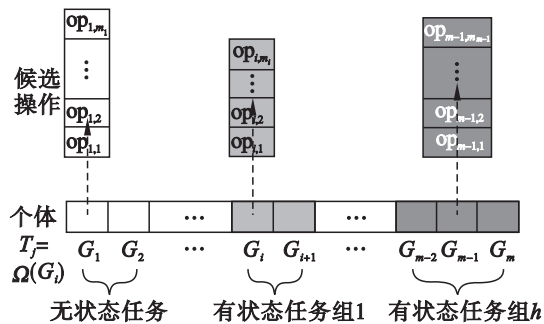


图2 个体编码

Fig. 2 Individual encoding

2) 交叉操作. 交叉操作采用两点交叉, 但限制了交叉点的取值范围以满足任务状态关联绑定约束. 首先, 根据交叉概率随机选择个体对. 然后, 确定可用交叉点, 包括: ①个体中不同任务组的分割点; ②无状态任务基因位的分割点; ③绑定操作属于同一服务的有状态任务组基因位的分割点. 最后, 在所有可用交叉点中随机选择两个交叉点进行交叉操作.

3) 变异操作. 变异操作中所有基因位都是可变异的. 对于无状态任务, 基因位变异后成为新的个体; 对于任意有状态任务 $T_i \in O_e$, 如果第 i 个基因位变异后的值 $op_{i,k}$ 所属的服务 s_l 与变异前相同, 则将其作为新的个体; 否则, 将 $\forall T_j \in O_e (j \neq i)$ 绑定的服务替换成 s_l , 并将 T_j 的基因位的值替换成 s_l 中的某个功能等价的操作, 当 s_l 中存在多个功能等价的操作能够实现 T_j 时, 随机从这些操作中选择一个作为其基因位的值.

4) 个体评价策略. 为了评价解的质量, 定义适应度函数如下:

$$\text{Fit}(\mathbf{x}) = F(\mathbf{x}) - P(\mathbf{x}). \quad (6)$$

式中, $P(\mathbf{x})$ 为罚函数, 以 $h_1(\mathbf{x}), h_2(\mathbf{x}), h_3(\mathbf{x})$ 分别表示 $t_R(\mathbf{x}) - t_{R,\max}^u, R_{\min}^u - R(\mathbf{x})$, 和 $C(\mathbf{x}) - C_{\max}^u$, 则

$$P(\mathbf{x}) = \sum_{\tau \in \{1,2,3\}} w_{\tau} \gamma_{\tau} h_{\tau}(\mathbf{x}). \quad (7)$$

且式(7)中的系数

$$\gamma_{\tau} = \begin{cases} 1, & h_{\tau}(\mathbf{x}) > 0; \\ 0, & h_{\tau}(\mathbf{x}) \leq 0. \end{cases}$$

SSOSGA 采用随机遍历抽样选择个体,该方法是轮盘赌选择法的一种改进,其特点是只需一次轮盘旋转,可避免早期的高适应度个体迅速占据种群.另外,以离线性能评估准则作为算法终止条件^[11].设第 T 次迭代的离线性能为 $X(T)$,则对于给定阈值 σ ,当 $X(T) - X(T-1) \leq \sigma$ 时,终止搜索.同时,为了避免搜索时间过长,设定一个最大遗传代数.

5) 算法描述.

算法 1 面向有状态服务选取的遗传算法 (SOSGA)

- 输入:ACS,有状态服务组集合及候选服务集.
输出:所有任务与候选操作的最优绑定关系.
- 1) 根据任务关联状态绑定约束过滤所有有状态任务的候选服务集合;
 - 2) 随机产生包含 M 个个体的初始种群 $P(0)$;
 - 3) while $X(T) - X(T-1) \leq \sigma$ or $T \leq T_{\max}$ do;
 - 4) 计算适应度函数 $\text{Fit}(x)$;
 - 5) 采用随机遍历抽样选择个体;
 - 6) 通过修改的两点交叉操作生成新的个体;
 - 7) 通过修改的变异操作生成新的个体;
 - 8) 评价个体间相似度;
 - 9) 计算离线性能 $X(T+1)$;
 - 10) $T++$;
 - 11) end while
 - 12) 将最优个体解码为最优解 EP^* ;
 - 13) return.

3 实验与分析

3.1 实验设置

硬件环境:CPU i5 M560 2.67 GHz,内存 6.0 GB;软件环境:Windows 7 Ultimate 操作系统,JDK 1.6.7.

实验采用一定规则随机生成 workflow,任务数量 m 取值范围为 10 ~ 50, T_i 的候选操作数量 n_i 取值范围为 20 ~ 100,且所有有状态任务组的大小均为 5.实验假定操作的响应时间、可靠性和费用等 QoS 属性权重相等,三者的值分别在一定的区间内以随机方式生成,响应时间区间为 0 ~ 10 s,可靠性区间为 0.7 ~ 1.0,费用区间为 1 ~ 10,并且根据文献[4]中的标记树递归规则计算组合服务 QoS.

根据遗传算法参数的常用取值范围^[9],SSOSGA 主要参数:种群大小 $P=50$,交叉概率 $P_c=0.75$,变异概率 $P_m=0.06$,阈值 $\sigma=1.0 \times 10^{-5}$,最大迭代数 $T_{\max}=100$.

3.2 实验结果及分析

3.2.1 SSOSGA 的收敛速度

以图 1 中的问题示例为实验对象,每个任务的候选操作数量为 50,与未采用罚函数的适应度函数相比较得到结果如图 3 所示.由图 3 可知,当未在适应度函数中引入罚函数时,种群的平均适应度明显要低一些,更容易收敛到局部最优.这是因为部分不满足多个约束条件的无可行解的个体被遗传到下一代,进而影响了收敛速度.

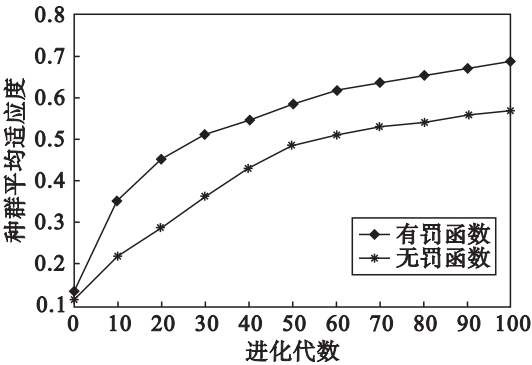


图 3 SSOSGA 的收敛性
Fig. 3 Convergence of SSOSGA

3.2.2 与现有算法比较

为了验证 SSOSGA 的有效性和性能开销,将其与不考虑有状态服务的遗传算法和线性规划算法(MOSES)^[4]进行组合服务 QoS 及算法消耗时间方面的对比.这里,不考虑有状态服务的遗传算法是指在 SSOSGA 中采用基本遗传算法的交叉和变异操作,并且取消个体编码规则中对有状态任务组基因位取值的限制,将其称为改进的服务选取遗传算法(improved service selection genetic algorithm,简称 ISSGA).

1) 组合服务的 QoS.对于不同的 m, n_i 组合,有状态任务组的个数固定在 $\lfloor m/2 \rfloor$,得到组合服务的 QoS 如表 1 所示.由表 1 可知,对于 m, n_i 的

表 1 组合服务的 QoS Table 1 QoS of composite services				
m	n_i	SSOSGA	ISSGA	MOSES
10	50	0.754	0.631	0.797
20	50	0.834	0.710	0.853
30	50	0.793	0.668	0.822
40	50	0.801	0.670	0.849
50	50	0.734	0.655	0.761
25	20	0.772	0.678	0.800
25	40	0.753	0.640	0.794
25	60	0.784	0.665	0.805
25	80	0.730	0.578	0.748
25	100	0.693	0.537	0.724

不同组合,SSOSGA 得到的组合服务 QoS 略低于 MOSES,这是因为线性规划算法通常能够更容易获得最优解.但与 ISSGA 相比,SSOSGA 使得每一代个体均满足任务状态关联绑定约束,因此能够更快地得到质量更好的解.

2) 服务选取消耗的时间.与 1) 相同,改变 m 和 n_i 时,有状态任务组的个数固定 $\lfloor m/2 \rfloor$,其结果如图 4,图 5 所示.

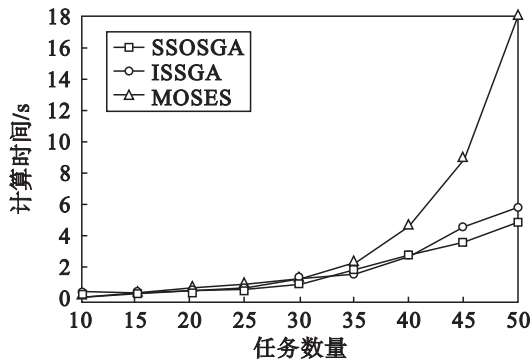


图 4 任务数量对计算时间的影响

Fig. 4 Impact of the number of tasks on solving time

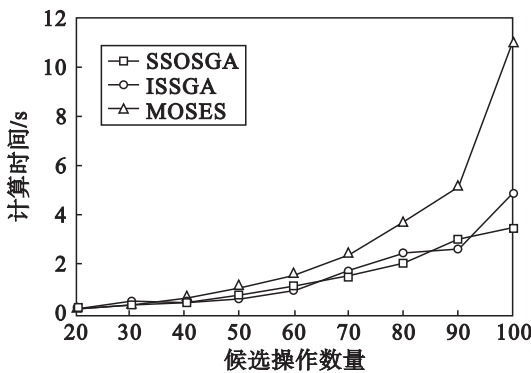


图 5 候选操作数量对计算时间的影响

Fig. 5 Impact of the number of candidate operations on solving time

由图可知,3 种算法的计算时间均随着任务数量和候选操作数量的增加而增加,当任务或候选操作较少时,SSOSGA 的计算时间略高于 MOSES,但当任务或候选操作较多时,MOSES 的计算时间迅速增加,并超过了前者.

3.3 实验结论

1) 在适应度函数中引入罚函数能够有效提高 SSOSGA 的收敛速度,并且更可能获得最优解;

2) SSOSGA 求得的解能够在满足全局 QoS 约束的前提下近似线性规划得到最优解;

3) 与采用线性规划的方法相比,SSOSGA 在面对较大的问题规模时,能够在更短的时间内得到最优解;同时,解的质量明显优于现有遗传算法.

4 结 论

本文针对工作流中多个任务之间存在状态关联的组合服务选取问题,提出了一种考虑状态关联绑定约束的 Web 服务选取方法的遗传算法.对比实验表明,本文的服务选取算法能够在令人满意的时间内收敛到最优解,并且在保证解的质量的基础上,其执行效率在一定程度上优于已有算法.

参考文献:

- [1] Hirochi W, Junichi S, Yuji Y, et al. A multiobjective optimization framework for SLA-aware service composition [J]. *IEEE Transactions on Services Computing*, 2012, 5(3): 358 – 372.
- [2] Yu T, Zhang Y, Lin K J. Efficient algorithms for web service selection with end-to-end QoS constraints [J]. *ACM Transactions on the Web*, 2007, 1(1): 1 – 26.
- [3] Papazoglou M P. Web services: principles and technology [M]. London: Prentice Hall, 2008: 143 – 166.
- [4] Valeria C, Emiliano C, Vincenzo G, et al. MOSES: a framework for QoS driven runtime adaptation of service-oriented systems [J]. *IEEE Transactions on Software Engineering*, 2012, 38(5): 1138 – 1159.
- [5] Ardaga D, Pernici B. Adaptive service composition in flexible processes [J]. *IEEE Transactions on Software Engineering*, 2007, 33(6): 369 – 384.
- [6] Pejman E, Rastegari Y, Majlesi E P, et al. Web service composition methods; a survey [C]//Proceedings of International Multi Conference of Engineers and Computer Scientists. Hong Kong, 2012: 603 – 607.
- [7] Menascé D A. QoS issues in web services [J]. *IEEE Internet Computing*, 2002, 6(6): 72 – 75.
- [8] Zeng L, Benatallah B, Ngu A H, et al. QoS-aware middleware for web services composition [J]. *IEEE Transactions on Software Engineering*, 2004, 30(5): 311 – 327.
- [9] 陈国良,王熙法,庄镇泉,等. 遗传算法及其应用[M]. 北京:人民邮电出版社,2001:67 – 85.
(Chen Guo-liang, Wang Xi-fa, Zhuang Zhen-quan, et al. Genetic algorithm and its applications [M]. Beijing: Posts and Telecommunications Press, 2001: 67 – 85.)
- [10] Baker J E. Adaptive selection methods for genetic algorithms [C]//Proceedings of the 1st International Conference on Genetic Algorithms. Hillsdale, 1985: 101 – 111.
- [11] De Jong K A. An analysis of behavior of a class of genetic adaptive systems [D]. Ann Arbor: University of Michigan, 1975.