

基于元胞遗传算法的 Web 服务选择

陈彦萍^{1,2}, 樊梦茹¹

(1 西安邮电大学 计算机学院, 陕西 西安 710121; 2 西安邮电大学 大数据处理研究中心, 陕西 西安 710121)

摘 要: 为了提高遗传算法在服务选择中的收敛速度与稳定性, 将元胞遗传算法引入服务选择中, 用元胞遗传算法典型的“生命游戏”规则及改进的演化规则取代传统遗传算法中的单点交叉操作. 实验结果表明, 采取规则 4 作为演化规则的元胞遗传算法可以很好地解决传统遗传算法运行结果不稳定和收敛性差的现象.

关键词: Web 服务选择; 元胞遗传算法; 生命游戏

中图分类号: TP393.09

文献标识码: A

文章编号: 1000-7180(2015)11-0116-04

Web Service Selection Based on the Cellular Genetic Algorithm

CHEN Yan-ping^{1,2}, FAN Meng-ru¹

(1 School of Computer Science, Xi'an University of Post and Telecommunications, Xi'an 710121, China;

2 Big Data Process Research Center, Xi'an University of Post and Telecommunications, Xi'an 710121, China)

Abstract: In order to improve the genetic algorithm convergence speed and stability, we bring the cellular genetic algorithm into the field of service selection. We use a typical cellular genetic algorithm rule Game of Life and improved rules to replace the single-point crossover operation in the traditional genetic algorithm. The experimental results show that the Rule Four can solve the instability of result and poor convergence phenomenon in the traditional genetic algorithm.

Key words: Web service selection; cellular genetic algorithm; Game of Life

1 引言

在 Web 服务组合研究中, 基于服务质量(QoS)的服务选择研究是一个重点^[1]. 目前比较流行的服务选择算法是采用启发式算法设计的选择算法, 例如遗传算法、蚁群算法、粒子群优化算法等^[2]. 文献[3]中提出了一种基于遗传算法的多 QoS 约束服务选择算法, 可以实时提供满足用户 QoS 需求的组合服务. 文献[4]中在遗传算法中采用关系矩阵编码方式, 克服了传统编码方式表示的局限性. 文献[5]中提出了一种二次编码模型, 缩短了染色体的编码长度, 提高编码可读性. 文献[6]中提出了二叉树编码方式, 降低了遗传算法的计算量. 文献[7]中提出了一种基于树型二重结构编码方式的遗传算法. 文献

[8]中将混沌理论引入到遗传算法的关系矩阵编码方案中, 在交叉过程中, 采用混沌时间序列代替随机选择, 实验结果表明该方法可以增强遗传算法的收敛能力. 采用传统遗传算法所设计的 Web 服务选择算法, 其通用性强、容易实现, 而且具有并行性和全局搜索能力; 但容易出现早熟收敛现象, 运行结果不稳定且收敛性差. 而这些缺点严重影响算法在服务组合问题上的应用^[2]. 元胞遗传算法^[9] (Cellular Genetic Algorithm, CGA) 是遗传算法的一个分支, 它将元胞自动机和遗传算法有机地结合在一起, 而引入元胞自动机的遗传算法能够克服原算法易陷入局部最优解的问题. 本文针对传统遗传算法在服务选择中存在的不足, 将元胞遗传算法引入 Web 服务选择中, 用元胞自动机中的演化规则代替传统遗传

收稿日期: 2015-01-16; 修回日期: 2015-03-05

基金项目: 陕西省自然科学基金项目(2012JQ8029); 中国博士后基金项目(2014M552453); 西安市科技计划项目(CXY1437

(8)); 西安邮电大学“西邮新星”团队支持计划

算法的交叉操作. 通过实验验证基于元胞遗传算法的服务选择可以快速稳定地收敛到最优值.

2 Web 服务选择

本文采用文献[10]中提出的组合服务选择模型. 将组合服务的执行流程划分为 L 个服务类 $\{C_1, C_2, \dots, C_L\}$, 第 i 个服务类 $C_i (1 \leq i \leq L)$ 由原子服务集合 $C_i = \{S_i^1, S_i^2, \dots, S_i^{N_i}\}$ 组成, N_i 表示原子服务集合 C_i 中候选服务个数. 基于 QoS 的服务选择问题就是根据这些 QoS 属性从每个服务类中选择一个服务. 每个候选服务都有 m 个 QoS 属性, 我们将其定义为 $q = (q_1, q_2, \dots, q_m)$. 本文我们定义的 QoS 属性由 5 个分量组成, 分别是响应时间、价格、有效期、可靠性和安全等级 5 个 QoS 属性.

针对基于多维 QoS 服务选择问题建立如下模型:

$$\begin{aligned} \max \quad & f_k(X) = f_k \left[\sum_{j=1}^{N_1} q_k \times x_{1j}, \sum_{j=1}^{N_2} q_k \times x_{2j}, \right. \\ & \left. \sum_{j=1}^{N_3} q_k \times x_{3j}, \dots, \sum_{j=1}^{N_L} q_k \times x_{Lj} \right], k = 1, 2, \dots, m \\ \text{s. t.} \quad & \begin{cases} x_{ij} \times (1 - x_{ij}) = 0 \\ \forall i = 1, 2, \dots, L, \sum_{j=1}^{N_i} x_{ij} = 1 \\ x_{ij} \geq 0, i = 1, 2, \dots, L, j = 1, 2, \dots, N_i \end{cases} \end{aligned} \quad (1)$$

求 $X = [x_{11}, x_{12}, \dots, x_{1N_1}, \dots, x_{L1}, x_{L2}, \dots, x_{LN_L}]^T$.

在模型(1)中可以得知以下几点:

(1) $f_k(X)$ 表示组合方案 X 在目标 q_k 下的属性值, 即通过原子服务的属性来计算出组合服务的 QoS 属性;

(2) $x_{ij} \times (1 - x_{ij}) = 0$ 表明了候选服务的状态, $x_{ij} = 1$ 表示选中该服务, $x_{ij} = 0$ 表示没有选中该服务, 这样就保证了在每一个服务类中只有一个服务被选中;

(3) $\forall i = 1, 2, \dots, L, \sum_{j=1}^{N_i} x_{ij} = 1$ 表示在每一个服务类中每次能且只能选择一个候选服务.

对于 QoS 属性而言, 包括成本型(越小越好)和效益型(越大越好), 为了能够更好地处理极大极小(极小极大)问题, 引入极大熵函数式(式(2))作为评价多目标优化性能的目标函数^[10]. 该极大熵函数作为元胞遗传算法的适应值函数进行评价, 进而对问题进行求解.

$$F(X) = \frac{1}{p} \ln \left\{ \sum_{k=1}^m \exp[-p \times f_k(x)] \right\} \quad (2)$$

3 元胞遗传算法

目前, 对元胞自动机应用于遗传算法的研究可大致分为两种情况: 一种是利用元胞自动机丰富的演化规则来代替传统遗传算法的某些操作如交叉; 另一种情况是将单个的元胞个体看作种群的个体, 借助元胞自动机模型的邻居结构实现遗传操作. 在传统遗传算法中交叉算子对整个优化过程中起着至关重要的作用, 因此本文主要研究第一种情况, 利用元胞自动机的演化规则替代传统遗传算法的交叉操作.

3.1 元胞自动机典型规则—生命游戏

元胞构成:

- (1) 元胞分布在规则划分的网格上;
- (2) 元胞状态的取值为 0 或 1, 状态“1”代表“生”, 状态“0”代表“死”;
- (3) 元胞邻居的形式为 Moore 型, 即元胞将其周围相邻的 8 个元胞作为其邻居;
- (4) 一个元胞的状态由该时刻的生死状态以及周围的 8 个邻居的状态之和决定.

演化规则:

- (1) 在当前时刻, 如果一个元胞的状态为“生”, 且周围 8 个邻居元胞的状态中有两个或三个为“生”, 则该元胞在下一时刻的状态保持为“生”, 否则为“死”. 用数学表达式为

if $S_i^t = 1$ then

$$S_i^{t+1} = \begin{cases} 1 & \text{if } \sum S_N^t = 2 \text{ or } 3 \\ 0 & \text{if } \sum S_N^t < 2 \text{ or } \sum S_N^t > 3 \end{cases} \quad (3)$$

- (2) 在当前时刻, 如果一个元胞的状态为“死”, 且周围 8 个相邻元胞的状态中恰好有三个为“生”, 则该元胞在下一时刻“复活”, 否则保持为“死”. 用数学表达式为

$$\text{if } S_i^t = 0 \text{ then } S_i^{t+1} = \begin{cases} 1 & \text{if } \sum S_N^t = 3 \\ 0 & \text{if } \sum S_N^t \neq 3 \end{cases} \quad (4)$$

式(3)和式(4)中, S_N^t 表示 t 时刻中心元胞的邻居的状态.

3.2 生命游戏交叉方式

在传统的遗传算法中, 我们采用选择、交叉、变异算子对个体进行操作, 在保证种群多样性中交叉算子起着重要作用. 图 1 中给出了传统遗传算法单点交叉情况.

由图 1 可知, 交叉后的基因种类并没有发生很

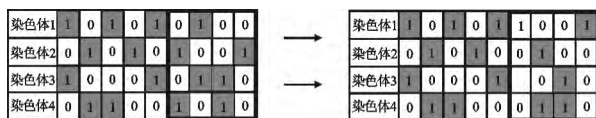


图1 传统遗传算法单点交叉方式

大的变化. 如果变异率很小的话, 解的多样性将变得很低, 这可能导致算法在运行时出现过早收敛的情况.

我们采用“生命游戏”规则来替代 GA 中的交叉算子, 这样对种群而言, 不再是配对的两条染色体之间进行交叉操作, 而是对所有非边界元胞进行检查, 计算该元胞邻居存活数, 进而判断该元胞在下一时刻的状态. 图 2 所示为生命游戏交叉方式.

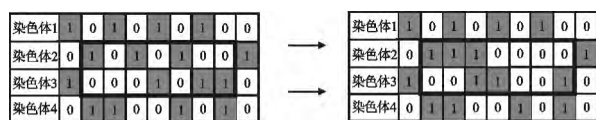


图2 生命游戏交叉方式

从图 2 中可以看出 CGA 采用元胞自动机的演化规则, 改变了传统遗传算法固定的交叉方式, 这样使个体之间的交互全方位, 保证了种群的多样性.

3.3 演化规则拓展

“生命游戏”规则要求比较苛刻, 多次运算后元胞演化逐渐稳定, 大量元胞死亡. 这样就会导致生命密度过小, 而由于物种缺乏繁殖机会, 最终会出现生命危机, 从而破坏优秀个体的继承^[9]. 为此设计了一些具有代表性的演化规则, 如下所示.

$$\begin{aligned}
 \text{规则 1: } & \begin{cases} \text{if } S_i^t = 1 \text{ then } S_i^{t+1} = \begin{cases} 1 & \text{if } \sum S_N = 2 \\ 0 & \text{if } \sum S_N \neq 2 \end{cases} \\ \text{if } S_i^t = 0 \text{ then } S_i^{t+1} = \begin{cases} 1 & \text{if } \sum S_N = 3 \\ 0 & \text{if } \sum S_N \neq 3 \end{cases} \end{cases} \\
 \text{规则 2: } & \begin{cases} \text{if } S_i^t = 1 \text{ then } S_i^{t+1} = \begin{cases} 1 & \text{if } \sum S_N = 1, 2 \\ 0 & \text{if } \sum S_N \neq 1, 2 \end{cases} \\ \text{if } S_i^t = 0 \text{ then } S_i^{t+1} = \begin{cases} 1 & \text{if } \sum S_N = 3 \\ 0 & \text{if } \sum S_N \neq 3 \end{cases} \end{cases} \\
 \text{规则 3: } & \begin{cases} \text{if } S_i^t = 1 \text{ then } S_i^{t+1} = \begin{cases} 1 & \text{if } \sum S_N = 1, 2, 3, 4 \\ 0 & \text{if } \sum S_N \neq 1, 2, 3, 4 \end{cases} \\ \text{if } S_i^t = 0 \text{ then } S_i^{t+1} = \begin{cases} 1 & \text{if } \sum S_N = 5, 6, 7, 8 \\ 0 & \text{if } \sum S_N \neq 5, 6, 7, 8 \end{cases} \end{cases} \\
 \text{规则 4: } & \begin{cases} \text{if } S_i^t = 1 \text{ then } S_i^{t+1} = \begin{cases} 1 & \text{if } \sum S_N = 2, 4, 6, 8 \\ 0 & \text{if } \sum S_N \neq 2, 4, 6, 8 \end{cases} \\ \text{if } S_i^t = 0 \text{ then } S_i^{t+1} = \begin{cases} 1 & \text{if } \sum S_N = 1, 3, 5, 7 \\ 0 & \text{if } \sum S_N \neq 1, 3, 5, 7 \end{cases} \end{cases}
 \end{aligned}$$

上述规则 3 模拟了一种周期性变化, 可以使得存活的元胞数目稳定在 51% 左右; 规则 4 采取一种奇偶性规则, 可以使存活的元胞数目稳定在 80% 左右. 采取这样的规则作为元胞遗传算法交叉算子, 就可以有效地保证优秀个体的继承.

3.4 基于 CGA 的服务选择过程

整个选择过程如下:

```

begin
    t ← 0
    初始化 P(t)
    评价 P(t)
    While(终止条件不满足) do
        Begin
            编码 encoding()
            选择 roulette()
            交叉 crossover()
            变异 mutation()
            解码 decoding()
            评价新个体即后代 C(t)
            从 P(t) 和 C(t) 中选择 P(t+1)
            t ← t + 1
        end
    end

```

4 实验与分析

为了验证基于元胞遗传算法的 Web 服务选择问题的可行性, 对比元胞遗传算法和传统遗传算法在解决 Web 服务选择问题时的性能, 本文设计了一个顺序执行的旅游测试场景(订票、订酒店、租车、找导游), 在本场景中总共定义了 4 个服务类, 共 100 个候选原子服务, 每个服务类中的候选服务个数分别为 10、20、30、40, 因此总共有 240 000 (10 × 20 × 30 × 40) 种组合方案, 通过穷举算法我们得到组合方案最优值是 15, 服务选择的非劣集: 服务类 1 中选择第 4、8 号服务, 服务类 2 中选择第 11、14 号服务, 服务类 3 中选择第 13、20、27 号服务, 服务类 4 中选择第 1、15、25、28、33、35 号服务.

在本文的实验中, 由于 CGA 算法中的“交叉”操作是针对个体元胞进行的, 相对传统遗传算法的交叉操作而言, 其交叉率已经很大了, 所以本实验中令传统遗传算法的交叉概率 $P_c = 0.7$, CGA 算法的交叉概率 $P_c = 0.1$, 其他算法参数相同, 令种群规模 $P = 80$, 变异概率 $P_m = 0.01$, 进化代数 $T = 1\ 000$, 表 1 所示为两种算法各自独立运行 10 次后的结果对比.

表 1 传统 GA 与 CGA 算法结果对比

传统 GA						“生命游戏”规则					
次数	类 1	类 2	类 3	类 4	进化结果	次数	类 1	类 2	类 3	类 4	进化结果
1	4	11	13	1	15.0	1	4	14	13	1	15.0
2	4	11	13	1	15.0	2	4	11	13	25	15.0
3	8	11	13	25	15.0	3	4	11	13	15	15.0
4	8	14	13	1	15.0	4	8	11	13	1	15.0
5	4	14	13	1	15.0	5	8	14	13	1	15.0
6	8	14	13	25	15.0	6	4	11	13	1	15.0
7	4	11	27	1	15.0	7	4	11	27	1	15.0
8	4	14	27	15	15.0	8	8	14	13	25	15.0
9	4	14	20	1	15.0	9	4	11	27	1	15.0
10	4	11	13	1	15.0	10	4	14	13	1	15.0

从表 1 可以看出当进化代数 $T=1\ 000$ 时,基于“生命游戏”规则的 CGA 对解决服务选择是有效的,两种算法运行 10 次后每次的进化结果值均为 15.0,每次选择出的服务都包含在非劣集中,其正确率是 100%。

为了验证基于元胞遗传算法的服务选择可以快速收敛到最优值,在实验中令两种算法的进化代数从 $T=500$ 增至 $T=10\ 000$,每次迭代次数增加 500 次,其他参数同上述实验参数一致,独立运行 10 次,计算达到最优解值次数以平均解值,实验结果如图 3、图 4 所示。

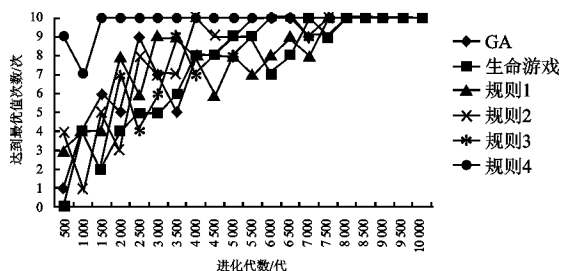


图 3 GA 与 CGA 达最优值次数对比

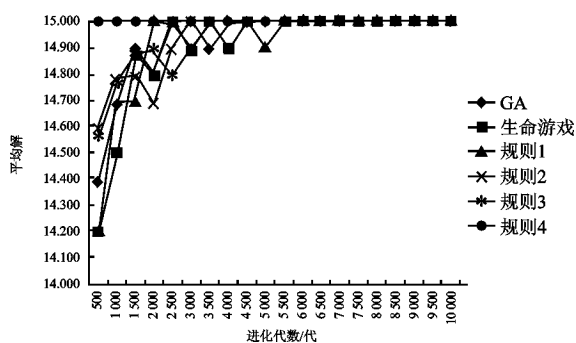


图 4 GA 与 CGA 平均解对比

从图 3 和图 4 中可以看出基于规则 4 的 CGA 算法当进化代数达到 1 500 代后逐渐稳定收敛到最优解 15.0,约为其他算法的 4~5 倍,基于规则 4 的 CGA 算法能快速稳定收敛到最优值。分析其原因:“生命游戏”规则、规则 1、规则 2 对生命的要求较为苛刻,影响优秀个体的继承,而规则 3、4 却能很好地保持活元胞数量,进而进行优秀个体的继承。采用规则 4 的 CGA 可以很有效的解决“生命游戏”交叉规则导致的种群损失问题。所以在基于元胞遗传算法的服务选择问题中,我们采取规则 4 作为交叉规则的元胞遗传算法可以很好地解决传统遗传算法运行结果不稳定和收敛性差的现象。

5 结束语

本文将元胞遗传算法引入 Web 服务选择中,用元胞的演化规则代替传统遗传算法的交叉操作,在实验中我们采用了“生命游戏”规则及其他改进规则代替传统的交叉操作。实验结果表明:基于规则 4 的 CGA 算法能够很好地解决基于传统遗传算法的服务选择问题存在的不足,提高算法的收敛速度和稳定性。下一步,我们将对算法的性能进行更深入的分析和改进,从而更好地提高算法各项性能。

参考文献:

- [1] 何炎祥,吴钊. 动态 Web 服务组合关键技术与性能分析[M]. 北京:清华大学出版社,2011.
- [2] 李金忠,夏洁武,唐卫东,等. 基于 QoS 的 Web 服务选择算法综述[J]. 计算机应用研究,2010,27(10): 3622-3628.
- [3] 莫振华,蔡鸿明,姜丽红. 基于遗传算法的多 QoS 约束服务选择[J]. 计算机应用与软件,2009,26(3):4-6.
- [4] 张成文,苏森,陈俊亮. 基于遗传算法的 QoS 感知的 Web 服务选择[J]. 计算机学报,2006,29(7): 1029-1037.
- [5] RongxiWang, LiMa, YanpingChen. The web service selection based on the genetic algorithm[C]// 2011 International Conference on Computer Science and Service System. Nanjing: IEEE,2011:1861-1864.
- [6] 杨艳梅,赵逢禹,韩文冬. 基于二叉树编码遗传算法的 SOA 服务选择[J]. 计算机应用,2009,29(8): 2276-2280.
- [7] 李淑芝,杨书心,王继升. 基于树型二重结构编码遗传算法的服务选择[J]. 计算机应用,2008,28(12): 3017-3020.

(下转第 124 页)

计算两个 9×9 的随机矩阵需花费 $30 \mu\text{s}$ 的时间, 加速比将近 5. 随着矩阵维数的增加能更好地发挥脉动阵列的处理优势, 将得到更好的加速比.

4 结束语

在卡尔曼滤波的实时计算中存在着大量的矩阵乘法运算. 本文在资源消耗相对节约的情形下, 基于“赛灵思”XCV5VLX110T-FPGA 及脉动阵列的设计思想, 构建了 3×3 的矩阵阵列乘法器, 为设计其余运算模预留了相当的资源. 同时乘法器具有良好的扩展性, 可结合矩阵的分块运算, 实现维数更高的矩阵乘法. 在后续的研究中, 可从运算效率、工作频率等方面入手进一步提升乘法器的性能.

参考文献:

- [1] Brown R G. Introduction to random signal analysis and Kalman filtering[M]. New York: Wiley, 1983.
- [2] Golub G H, Van Loan C F. 矩阵计算[M]. 袁亚湘, 译. 北京: 科学出版社, 2011.
- [3] Cappello J D, Strenski D. A practical measure of FPGA floating point acceleration for High Performance Computing[C]// Proceedings of the 2003 IEEE 24th International Conference on Application-specific Systems. Washington, DC: IEEE, 2013: 160-167.
- [4] 田翔, 周凡, 陈耀武, 等. 基于 FPGA 的实时双精度浮点矩阵乘法器设计[J]. 浙江大学学报: 工学版, 2008, 42(9): 1611-1615.
- [5] 刘沛华, 鲁华祥, 龚国良, 等. 基于 FPGA 的全流水双精度浮点矩阵乘法器设计[J]. 智能系统学报, 2012, 7(4): 302-306.
- [6] Bensaali F, Amira A, Sotudeh R. Floating-point matrix product on FPGA[C]// proceedings of the IEEE/ACS International Conference on Computer Systems and Applications, 2007 Computer Systems and Application. Amman: IEEE, 2007: 466-473.
- [7] Kestur S, Davis J D, Chung E S. Towards a universal FPGA matrix-vector multiplication architecture[C]// proceeding of the 2012 IEEE 20th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). Toronto, ON: IEEE, 2012: 9-16.
- [8] 吴淑泉, 王前, 谢运祥. 适于消谐模型求解的矩阵乘法器设计与实现[J]. 华南理工大学学报: 自然科学版, 2003, 31(8): 1-5.
- [9] Fleming S T, Thomas D B. Hardware acceleration of matrix multiplication over small prime finite fields[C]// proceedings of 9th international Conference on Reconfigurable Computing: architectures tools, and applications. Berlin, Heidelberg: ACM, 2013: 103-114.

作者简介:

王 阳 男, (1991-), 硕士研究生. 研究方向为精确制导自动目标识别. E-mail: sunville2013@nudt.edu.cn
陶华敏 女, (1972-), 副教授, 硕士研究生导师. 研究方向为精确制导自动目标识别.

(上接第 119 页)

- [8] Chengwen Zhang, Xiuqin Lin. A genetic algorithm with improved convergence capability for QoS-Aware web service selection[C]// Proceedings of the international Conference on Management and Service Science, wuhan: IEEE, 2009, : 1-4.
- [9] 李学伟, 吴金培, 李雪堰. 实用元胞自动机导论[M]. 北京: 北京交通大学出版社, 2013.
- [10] 陈彦萍, 张建科, 孙家泽, 等. 一种基于混合智能优化的

服务选择模型[J]. 计算机学报, 2010, 33(11): 2116-2125.

作者简介:

陈彦萍 女, (1979-), 博士, 教授. 研究方向为服务计算.
樊梦茹(通讯作者) 女, (1989-), 硕士研究生. 研究方向为服务计算. E-mail: fanmengru_1017@163.com