

文章编号: 1001-0920(2010)01-0074-05

交互式遗传算法基于用户认知不确定性的定向变异

巩敦卫¹, 郝国生^{1,2}, 严玉若²

(1. 中国矿业大学 信息与电气工程学院, 江苏 徐州 221008;

2. 徐州师范大学 计算机科学与技术学院, 江苏 徐州 221116)

摘要: 利用用户认知的不确定性设计定向变异算子. 首先, 采用主成分分析法辨识用户认知的不确定性; 然后, 给出用户认知不确定性的区间表示与更新策略; 最后, 将用户认知的不确定性指导定向变异算子, 包括: 选择待变异的进化个体, 确定变异位置, 以及变异方法等. 将所提方法应用于人眼图形优化, 实验结果验证了该方法的优越性.

关键词: 遗传算法; 交互; 认知不确定性; 定向变异

中图分类号: TP183

文献标识码: A

Directional mutation based on user's uncertain cognitive in interactive genetic algorithms

GONG Dun-wei¹, HAO Guo-sheng^{1,2}, YAN Yu-ruo²

(1. School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221008, China; 2. School of Computer Science and Technology, Xuzhou Normal University, Xuzhou 221116, China.

Correspondent: HAO Guo-sheng, E-mail: guoshenghaoxz@tom.com)

Abstract: A directional mutation operator is designed by using the user's uncertain cognitive. The main ingredient analysis is used to identify the user's uncertain cognitive. An interval-based expression and an update strategy of the user's uncertain cognitive are given. Finally, the user's uncertain cognitive is to guide a directional mutation operator, including the choice of candidate individuals to be mutated, the determination of positions to mutate, as well as the strategy to mutate. The proposed algorithm is applied to eyes optimization. The experimental results show the advantageous of the algorithm.

Key words: Genetic algorithm; Interaction; Uncertain cognitive; Directional mutation

1 引言

遗传算法不要求被优化问题目标函数可微或连续^[1], 但需要被优化问题目标函数可显式表示. 然而, 许多复杂优化问题的目标函数难以显式给出, 如与人的主观因素密切相关的服装设计等. 20 世纪 80 年代后期提出的交互式遗传算法是解决上述问题的有效方法.

交互式遗传算法由用户赋予个体适应值, 是由用户与进化机制相互协作解决优化问题的技术^[2]. 目前, 交互式遗传算法已成功应用于知识获取、工程设计、语言处理与韵律控制等许多领域.

提高交互式遗传算法的性能, 一直是交互式遗传算法研究的核心问题. 文献[3]根据用户偏好, 把

搜索空间划分为不同的子空间. 在解的质量较好空间中, 分配较多进化代数. [4]提出了在成对个体比较评价基础上的适应值推断方法. [5]提出了异地多用户分布式交互式遗传算法. 这些方法都有效地改进了算法性能, 但没有考虑用户认知特点, 也没有很好地利用用户认知特点.

人们往往对最满意或最不满意的事物判别最有把握, 而对介于二者之间的事物判别把握不大. 这意味着, 用户对适应值最高或最低个体的认知有较小的不确定性, 而对适应值居中的个体的认知有较大的不确定性. 如果根据个体的适应值, 确定用户对个体认知的不确定性, 并据此采用合适的遗传操作, 将会提高交互式遗传算法的性能.

收稿日期: 2009-02-23; 修回日期: 2009-04-04.

基金项目: 国家自然科学基金项目(60775044); 教育部“新世纪优秀人才支持计划”项目(NCET-07-0802).

作者简介: 巩敦卫(1970—), 男, 江苏铜山人, 教授, 博士, 从事智能优化与控制等研究; 郝国生(1972—), 男, 河北万全人, 副教授, 博士, 从事进化计算的研究.

变异算子是遗传算法的重要算子之一,对交互式遗传算法而言,也同样如此.采用合适的变异算子可以明显改善算法的性能,如不同的进化个体采用不同的变异概率,采用不同的基因位变异进化个体,针对不同的优化问题,采用域知识设计专门的变异算子,以及变化变异算子在算法流程中的顺序等^[6-8].这些方法均利用了进化个体的适应值.

交互式遗传算法中,个体的适应值是由用户赋予的,因此设计高性能的变异算子应该充分考虑用户对进化个体认知的不确定性.本文给出了基于用户认知不确定性的定向变异算子设计方法.

2 基于主成分分析的用户认知不确定性辨识

前已阐述,用户对适应值最高或最低个体的认知有较小的不确定性,而对适应值居中个体的认知有较大的不确定性.因此用户对适应值越接近种群适应值的进化个体的认知不确定性越大,而对适应值越远离种群适应值个体的认知不确定性越小.鉴于此,基于个体适应值与种群适应值的距离可描述用户认知的不确定性.

记 $x(t)$ 为第 t 代进化种群,规模为 n , $x_i(t)$ 为其第 i 个进化个体, $f(x_i(t))$ 为 $x_i(t)$ 的适应值.记种群 $x(t)$ 的适应值(即进化个体适应值的均值)为 $\mu(x(t))$, $x(t)$ 中进化个体适应值的方差为 $\sigma^2(x(t))$, 则有

$$\mu(x(t)) = \frac{1}{n} \sum_{i=1}^n f(x_i(t)),$$

$$\sigma^2(x(t)) = \frac{1}{n-1} \sum_{i=1}^n (f(x_i(t)) - \mu(x(t)))^2.$$

记 $f(x_i(t))$ 的规范化适应值为 $f^*(x_i(t))$, 则有

$$f^*(x_i(t)) = \frac{f(x_i(t)) - \mu(x(t))}{\sigma(x(t))}. \quad (1)$$

容易看出, $|f^*(x_i(t))|$ 越小,用户对 $x_i(t)$ 认知的不确定性越大;反之亦然.因此,式(1)可用来定量地描述用户认知的不确定性.

记

$$(x_i(t)) = \frac{1}{\sqrt{2}} \frac{1}{\sigma(x(t))} e^{-\frac{(f(x_i(t)) - \mu(x(t)))^2}{2\sigma^2(x(t))}},$$

则 $(x_i(t))$ 值越大, $f(x_i(t))$ 越接近种群适应值,从而用户对 $x_i(t)$ 认知的不确定性越大.根据 $(x_i(t))$ 的大小,依降序重排 $x_i(t)$, $i = 1, 2, \dots, n$.在不引起混淆的情况下,记重排后的进化个体仍为 $x_i(t)$, $i = 1, 2, \dots, n$.

记 $c_u = \sum_{i=1}^m (x_i(t)) / \sum_{i=1}^n (x_i(t))$ 为前 m 个进化个体对用户认知不确定性的累积贡献率^[9].给定

阈值 θ , 记满足 $c_u > \theta$ 的最小的 m 对应的进化个体集合为 $x(t)$, 则 $x(t)$ 可表示为

$$x(t) = \{x_i(t), i = 1, 2, \dots, m\}, \quad (2)$$

其中 m 是优化问题 $\min_m \{m | c_u > \theta\}$ 的解.

类似地,可得到用户认知不确定性较小的进化个体集合,记为 $x_{1-}(t)$, 则有

$$x_{1-}(t) = \{x_i(t), i = m + 1, m + 2, \dots, n\}. \quad (3)$$

通过式(2)和(3),种群 $x(t)$ 分割成两个互不相交的个体集合 $x(t)$ 和 $x_{1-}(t)$.用户对 $x_{1-}(t)$ 中个体的偏好比较明确,因此可利用 $x_{1-}(t)$ 提取用户的偏好知识,以指导种群的后续进化.

3 用户认知不确定性的区间表示与更新

对个体而言,如果其基因型某些部分对应的表现型是独立的,则称这些部分为基因意义单元^[10].本节考虑进化个体采用实数编码时,基因意义单元的表现与用户满意的、不满意的,以及未知的等位基因意义单元的求取与更新.

3.1 基因意义单元的区间表示

设个体编码长度为 d , 包含 l 个基因意义单元,第 k 个基因意义单元为 $x_{ik}(t)$, $k = 1, 2, \dots, l$.对 $x_{ik}(t)$ 的不同取值,称之为等位基因意义单元.由于用户对微小差别的鉴别能力有限,因此当用户很难分辨出差异时,则认为是相同类型的等位基因意义单元.

记用户对 $x_{ik}(t)$ 的鉴别阈值为 d_k .考虑两进化个体 $x_i(t)$ 和 $x_j(t)$ 的第 k 个基因意义单元,其等位基因意义单元分别为 $x_{ik}(t)$ 和 $x_{jk}(t)$.当 $|x_{ik}(t) - x_{jk}(t)| < d_k$ 时,用户无法区别由 $x_{ik}(t)$ 和 $x_{jk}(t)$ 确定的个体表现型,因此可认为区间 $[x_{ik}(t), x_{jk}(t)]$ 或该区间的点具有相同的表现型.当除了 $x_{ik}(t)$ 和 $x_{jk}(t)$ 不同外,其他基因意义单元的取值完全相同时,如果用户对 $x_i(t)$ 不满意,却对 $x_j(t)$ 满意,则有理由认为用户对 $x_i(t)$ 不满意的根源在于对 $x_{ik}(t)$ 对应的表现型不满意.

求同算子的目的是确定具有相同表现型的等位基因意义单元.求异算子的目的是确定具有不同表现型的等位基因意义单元.下面通过例子,说明进化个体实数编码的求同求异算子.取 $d = l = 4$, $k = 0.01$, $k = 1, 2, 3, 4$.用 null 表示结果为空,用 \oplus 和 \ominus 分别表示求同和求异算子,则当 $x_i(t) = (0.5238, 1.4368, 3.8566, 2.6314)$, $x_j(t) = (0.5234, 0.1765, 3.8470, 2.6314)$ 时,有

$$x_i(t) \oplus x_j(t) =$$

$$([0.5234, 0.5238], \text{null},$$

$$[3.8470, 3.8566], [2.6314, 2.6314]),$$

$$x_i(t) \ominus x_j(t) =$$

(null, [1.4368, 1.4368], null, null).

通过求同和求异算子,可以确定哪些等位基因意义单元分别是用户满意的、不满意的,以及未知的,且这些等位基因意义单元采用区间表示.

下面以个体的第 k 个基因意义单元的满意等位基因意义单元的构成及其变化为例进行说明.采用链表方式表示用户满意的等位基因意义单元,由于采用区间表示,链表的结点就是一个区间,这些结点按区间从小到大的顺序连接,如图 1 所示.

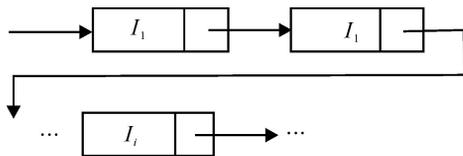


图 1 用户满意的等位基因意义单元链表

3.2 用户满意的等位基因意义单元的更新

由于用户满意的等位基因意义单元链表的结点是区间,当有新的用户满意的等位基因意义单元插入时,可能需要更新某个(些)结点区间的上下端点.下面讨论链表的建立及更新的元操作.

最初,用户满意的等位基因意义单元链表为空.假设 $[a, \bar{a}]$ 是第 1 个用户满意的等位基因意义单元,其中 a 和 \bar{a} 分别是区间的下端点和上端点,这时可将结点 $[a, \bar{a}]$ 直接插入到链表中.

当第 2 个用户满意的等位基因意义单元 $[b, \bar{b}]$ 需要插入到链表时, $[b, \bar{b}]$ 与 $[a, \bar{a}]$ 的关系共有 6 种可能,如图 2 所示.图 2 中,从左到右分别表示两区间上下端点的大小关系(箭头从小数指向大数)、区间图形示意、区间比较与合并后得到的新区间,以及类型编号.主要分为两类,第 1 类为两区间没有重叠,对应类型 0 和 3;第 2 类为两区间有重叠,对应类型为 1, 2, 4, 5, 需要合并区间后更新节点.

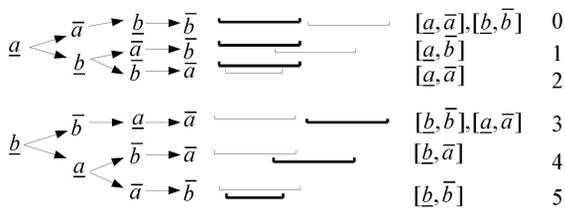


图 2 两区间的位置关系及链表更新

下面考虑第 3 个用户满意的等位基因意义单元 $[c, \bar{c}]$ 在链表中的插入方式.首先考虑 $[a, \bar{a}]$ 与 $[c, \bar{c}]$ 的位置关系,如图 3 所示.对于类型 0,需要再考虑 $[c, \bar{c}]$ 和 $[b, \bar{b}]$ 的位置关系.假设链表中的结点满足 $[a, \bar{a}] \cap [b, \bar{b}]$,则需要继续进行类似于图 3 的比较过程.对于类型 2, 3, 4, 同图 2 中插入第 2 个结点时类型 2, 3, 4 的情况.对于类型 1 和 5,则需要进一步判断 \bar{c} 与 $[b, \bar{b}]$ 的关系.这里不再赘述.

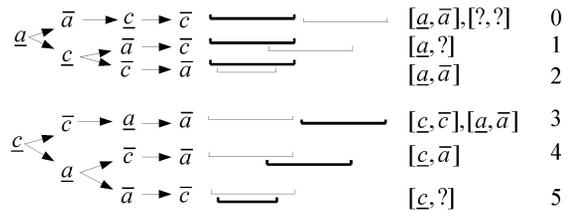


图 3 第 3 个用户满意的等位基因意义单元的插入

此外,若两结点满足 $[a, \bar{a}] \cap [b, \bar{b}]$,当 $b - \bar{a} < \epsilon$ 时,则合并结点为 $[a, \bar{b}]$.

这样便建立了表示用户满意的等位基因意义单元链表.表示用户不满意的以及未知的等位基因意义单元链表的过程类似,不再赘述.

本文需要充分利用上述用户认知信息,改进传统的遗传算子,从而提高算法的性能.

4 基于用户认知不确定性的定向变异

本节根据用户认知不确定性设计定向变异算子,其思想是:选择用户不确定性较小且适应值较低的进化个体,对用户不满意的等位基因意义单元进行变异,使得变异后的等位基因意义单元来自用户满意或未知的等位基因意义单元.

4.1 待变异进化个体的选择

首先考虑 $x(t)$ 中用户认知不确定性较小且适应值较低的进化个体.一般来讲,如果进化个体的适应值小于种群适应值 $\mu(x(t))$,那么该个体的适应值较低.记这类进化个体的集合为 $x_i^l(t)$,则 $x_i^l(t)$ 可表示为

$$x_i^l(t) = \{x_i(t) \mid f(x_i(t)) < \mu(x(t)), x_i(t) \in x_1(t)\} \quad (4)$$

类似的,可得到用户认知不确定性较低且进化个体适应值较高的进化个体集合,记为 $x_i^u(t)$,则 $x_i^u(t)$ 可表示为

$$x_i^u(t) = \{x_i(t) \mid f(x_i(t)) > \mu(x(t)), x_i(t) \in x_1(t)\} \quad (5)$$

由于用户对 $x_i^l(t)$ 中的进化个体评价较低,有必要对它们实施概率为 1 的变异操作.

对于 $x(t)$ 中除 $x_i^l(t)$ 之外的进化个体,即 $x(t)$ 或 $x_i^u(t)$ 中的进化个体,要么用户对这些进化个体的认知不明确,要么认知已经明确但适应值比较大,对它们依较低的概率实施变异操作.因此对 $x(t)$ 中的进化个体可分成两类:第 1 类是 $x_i^l(t)$ 的个体,实施概率为 1 的变异操作;第 2 类是 $x(t)$ 或 $x_i^u(t)$ 中的进化个体,依较低的概率实施变异操作.

4.2 变异位置的确定

由于用户对 $x_i^l(t)$ 和 $x_i^u(t)$ 中的进化个体的

认知不确定性较小,可通过对这些进化个体的基因型求同或求异确定变异位置.

首先,利用求同算子确定变异位置.用户认为 $x_i^l(t)$ 中的进化个体比较低劣,如果这些个体具有相同(近)的某一(些)等位基因意义单元,则可认为该等位基因意义单元是用户不满意的,因此应该对其进行变异.鉴于此, $\oplus\{x_i(t) \mid x_i(t) = x_i^l(t)\}$ 中非空的等位基因意义单元所处的位置就是变异位置.

其次,利用求异算子确定变异位置.在种群进化的后期,个体之间的差异较小.如果 $x_i^l(t)$ 中的进化个体与 $x_i^u(t)$ 中进化个体有相异的等位基因意义单元,则用户对 $x_i^l(t)$ 中的进化个体赋予较低适应值的原因,可能是由于这些相异的等位基因意义单元,因此应当对其进行变异. $\forall x_i(t) = x_i^l(t), x_j(t) = x_i^u(t), x_i(t) = x_j(t)$ 中非空的等位基因意义单元位置就是变异位置.

4.3 变异方法

本文仍通过对 $x_i^l(t)$ 和 $x_i^u(t)$ 中的个体的基因型求同或求异,确定变异后的等位基因意义单元.标记

$$x_i^u =_{t=0}^{T_1} x_i^u(t), x_i^l =_{t=0}^{T_1} x_i^l(t),$$

其中 T_1 为某进化代数.在种群进化初期,通过对 x_i^l 中的进化个体求同,即 $\oplus\{x_i(t) \mid x_i(t) = x_i^l, t \in \{0, 1, \dots, T_1\}\}$,确定用户不满意的等位基因意义单元.在种群进化后期,通过对 x_i^l 和 x_i^u 中的进化个体求异,即 $\forall x_i(t_1) = x_i^l, x_j(t_2) = x_i^u, x_i(t_1) = x_j(t_2)$,确定用户不满意的等位基因意义单元.某基因意义单元变异后的等位基因意义单元从用户不满意的等位基因意义单元之外选取.

本文提出的待变异的进化个体、变异的基因意义单元,以及变异后的等位基因意义单元的选择并非随机的,而是基于用户认知的不确定性进行合理选择,都具有方向性,因此称这样的变异算子为定向变异.

5 在三维人眼图形优化中的应用

5.1 背景

三维人脸自动生成在动漫人物造型中有着重要的应用价值,其中双眼是人脸的重要组成部分,因此采用交互式遗传算法进化用户满意的人眼图形.

本文中,人眼图形是由基准人眼坐标值经仿射变换后得到的.仿射变换包括:平移、缩放、剪切和旋转等.其中:平移变换由沿坐标轴平移的 3 个参数 p_1, p_2 和 p_3 确定;缩放变换由沿坐标轴缩放的 3 个参数 p_4, p_5 和 p_6 确定;剪切变换由沿 x 轴剪切的参

数 p_7, p_8 ,沿 y 轴剪切的参数 p_9, p_{10} 和沿 z 轴剪切的参数 p_{11}, p_{12} 确定;旋转变换由绕轴旋转的参数,绕 y 轴旋转的参数和 z 绕轴旋转的参数确定.

不同的人眼图形具有不同的参数,因此对上述参数进行优化,可得到满足用户需求的人眼图形.本文以基准人眼图形作为优化目标,因此反映目标的参数取值如下:

$$p_1 = p_2 = p_3 = 0, p_4 = p_5 = p_6 = 1, \\ p_7 = p_8 = p_9 = p_{10} = p_{11} = \\ p_{12} = = = = 0.$$

本文的目的是,通过采用交互式遗传算法进化优化上述参数,使得优化后的人眼是基准人眼.采用实数对它们编码,形成进化个体的基因型.

5.2 参数设置

算法中的参数设置如下:种群规模为 8;采用正弦轮盘赌选择;单点交叉,位置在某两个实数之间,交叉概率为 0.8;对 $x_i^l(t)$ 采用概率为 1 的定向变异方法,对 $x(t) \setminus x_i^l(t)$ 采用单点变异,变异概率为 0.5;最优保留策略.取 $\alpha = 0.6, \delta_k = 0.01, k = 1, 2, \dots, 15$.仿射变换的参数取值范围分别是

$$p_1, p_2 \in [-0.005, 0.05], \\ p_3 \in [-0.008, 0.03], \\ p_4 \sim p_6 \in [0.4, 1.1], \\ p_7 \sim p_{12} \in [-0.2, 0.2], \\ , , \in [-/10, /10].$$

5.3 实验设计

为了验证本文方法的性能,将其与传统交互式遗传算法和文献[3]的方法进行了比较.由 10 个实验人员每人进行 2 组实验,每组独立运行 10 次,终止条件是种群进化 20 代.

比较的性能指标包括:用户对优化人眼图形的满意程度以及优化人眼图形与基准人眼图形的距离.其中,对于第 1 个指标,要求用户从“非常不满意”、“不满意”、“满意”和“非常满意”4 个选项中选择的一个作为评价结论,并分别赋予 -2, -1, 1 和 2 等分值;第 2 个指标采用下式:

$$f_d = \frac{3}{i=1} |p_i| + \frac{12}{i=7} |p_i| + \frac{6}{i=4} |1 - p_i| + \dots \quad (6)$$

进行衡量,越小越好,最小值为 0.

5.4 结果分析

表 1 是实验人员给出的主观评价.从表 1 可以看出,与传统算法相比,本文算法得到的优化结果用户比较满意.

表 2 是最终的最优人眼图形与目标人眼图形的参数距离 f_d 的平均值.可以看出,本文算法得到的



表1 用户对优化结果的满意程度比较

	本文算法	文献[3]	传统算法
非常不满意(-2)	5	5	7
不满意(-1)	11	14	18
满意(1)	76	75	70
非常满意(2)	8	6	5
总分	71	63	48

表2 优化结果与目标人眼图形的距离 f_d

	本文算法	文献[3]	传统算法
均值	1.07	1.21	1.50

优化结果与目标人眼图形更加接近,与目标人眼图形之所以有差别,是因为用户对细微图形的辨别能力有限。

6 结论

本文目的是提高交互式遗传算法性能,采用的方法是利用用户认知不确定性设计定向变异算子。

本文基于主成分分析,识别了种群中用户认知不确定性较小的个体,根据这些个体得到了用户满意的、不满意的,以及未知的等位基因意义单元。此外,再结合进化个体的适应值,选择了待变异的进化个体、需要变异的基因意义单元,以及变异后的等位基因意义单元。

将本文方法应用于人眼进化,与传统交互式遗传算法的对比结果验证了所提方法的优越性。

采用合适的代理模型代替用户评价进化个体,是提高交互式遗传算法性能的另一个重要途径。如何基于用户认知的不确定性,选择合适的样本训练代理模型,是需要进一步研究的问题。

参考文献(References)

- [1] Holland J H. Adaptation in natural and artificial systems [M]. MIT Press, 1975.
- [2] Takagi H. Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human

evolution [J]. Proc of the IEEE, 2001, 89(9): 1275-1296.

- [3] Szeto K Y, Zhao S Y. Adaptive spatial allocation of resource for parallel genetic algorithm [J]. Studies in Computational Intelligence, 2008, 129(5): 389-398.
- [4] Watanabe Y, Yoshikawa T, Furuhashi T. A study on application of fitness inference method to PC-IGA [C]. Proc of IEEE Congress on Evolutionary Computation. Swissote, 2007: 1450-1455.
- [5] Miki M, Yamamoto Y, Wake S, et al. Global asynchronous distributed interactive genetic algorithm [C]. Proc of IEEE Int Conf on Systems, Man and Cybernetics. Taipei, 2006: 3481-3485.
- [6] 苏小红, 杨博, 王亚东. 基于进化稳定策略的遗传算法[J]. 软件学报, 2003, 14(11): 1863-1868.
(Su X H, Yang B, Wang Y D. A genetic algorithm based on evolutionarily stable strategy [J]. J of Software, 2003, 14(11): 1863-1868.)
- [7] Tinos R, Yang S. Evolutionary programming with q gaussian mutation for dynamic optimization problems [C]. Proc of IEEE Congress on Evolutionary Computation. Hong Kong, 2008: 1823-1830.
- [8] 薛文涛, 吴晓蓓, 徐志良. 基于双变异算子的免疫规划[J]. 控制与决策, 2007, 22(12): 1412-1416.
(Xue W T, Wu X B, Xu Z L. Immune programming based on double mutation operators [J]. Control and Design, 2007, 22(12): 1412-1416.)
- [9] 向东进. 实用多元统计分析[M]. 北京: 中国地质大学出版社, 2005.
(Xiang D J. Practical multivariate statistical analysis [M]. Beijing: China University of Geosciences Press, 2005.)
- [10] 巩敦卫, 郝国生, 周勇, 等. 交互式遗传算法原理及其应用[M]. 北京: 国防工业出版社, 2007.
(Gong D W, Hao G S, Zhou Y, et al. The principle and application of interactive genetic algorithm [M]. Beijing: National Defense Industry Press, 2007.)

(上接第73页)

- [12] Khalil H K. Nonlinear system [M]. Upper Saddle River NJ: Prentice Hall, 2002.
- [13] Misra V, Cong W B, Towsley D. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED [C]. Proc of the ACM SIGCOMM 2000. Stockholm: ACM, 2000: 151-160.
- [14] Chiu D, Jain R. Analysis of the increase and decrease

algorithms for congestion avoidance in computer networks[J]. Computer Networks and ISDN Systems, 1989, 17(1): 1-14.

- [15] Steven W. TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms [OE/BL]. [1997-07-15]. <http://www.faqs.org/rfcs/rfc2001.html>.