

基于多目标约束遗传算法的SDN路径增强算法

周睿¹, 何利文¹, 唐澄澄¹, 侯小宇¹, 陆钱春²

(1. 南京邮电大学, 江苏 南京 210003;

2. 中兴通讯股份有限公司, 江苏 南京 210012)

摘要: SDN控制器的出现有效地解决了传统网络设施僵化等问题, 它可以通过自身的网络拓扑模块掌握全局拓扑, 并且计算出符合全局最优的网络路径。但是, 现如今的网络环境日益复杂, 想要计算出可以有效满足带宽、时延、丢包率等约束的路由变得尤为困难, 传统的精确算法已经很难满足各种网络业务的需求。实际上计算满足多目标多约束的路径问题本质上属于多目标优化问题(MOP), 很难用精确的数学方法解决。所以借助启发式算法的思想, 针对该问题, 提出了一种改进的多目标约束遗传算法来找到全局最优的路径, 在传统遗传算法的基础上根据网络的特性, 对初始种群生成的算法进行了改进, 使得初始路径满足了大部分的约束条件, 实现对传统的计算路径算法的增强及优化的目标。实验结果表明, 该算法可以在满足各种业务约束条件的前提下达到多个路径优化目标的效果。

关键词: SDN控制器; 路由算法; 多约束遗传算法; 多目标优化

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2019)07-0017-06

doi: 10.3969/j.issn.1673-629X.2019.07.004

SDN Path Enhancement Algorithm Based on Multi-objective Constraint Genetic Algorithm

ZHOU Rui¹, HE Li-wen¹, TANG Cheng-cheng¹, HOU Xiao-yu¹, LU Qian-chun²

(1. Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

2. ZTE Corporation, Nanjing 210012, China)

Abstract: The emergence of SDN controller effectively solves the problems such as the rigidity of traditional network facilities. It can master the global topology through its own network topology module and calculate the network path that conforms to the global optimal. However, nowadays, with the increasingly complex network environment, it is particularly difficult to calculate the routing that can effectively meet the constraints of bandwidth, delay, packet loss rate and so on, and the traditional precision algorithm has been difficult to meet the requirements of various network services. In fact, the computation of a path problem that satisfies multiple objectives and multiple constraints is essentially a multi-objective optimization problem (MOP), which is difficult to solve with precise mathematical methods. Therefore, with the aid of the heuristic algorithm, an improved multi-objective constrained genetic algorithm to find the global optimal path is put forward. Based on the characteristics of the network, the algorithm of initial population generation is improved based on the traditional genetic algorithm, so that the initial path satisfies most of the constraints, achieving the goal of enhancing and optimizing the traditional computation path algorithm. The experiment shows that the proposed algorithm can achieve the effect of multiple path optimization objectives under the premise of meeting various business constraints.

Key words: SDN controller; routing algorithm; multi-objective genetic algorithm; multi-objective optimization

1 概述

网络的稳定性是网络正常运行的重要前提, 减少网络变更对生产生活的影响就变得至关重要, 这同时还会减少相关设备和协议的巨大安装成本。想在传统

网络上有所创新变得困难, 几乎没有什么办法来尝试新的网络协议, 因此人们普遍认为网络设施已经变得僵化了^[1]。但是随着移动设备和内容的爆炸式增长、服务器虚拟化, 以及云计算的出现推动了网络的进一

收稿日期: 2018-05-06

修回日期: 2018-09-12

网络出版时间: 2019-03-21

基金项目: 2012年国家发改委信息安全专项(GJ215001); 南京邮电大学引进人才科研启动基金资助项目(NY212012)

作者简介: 周睿(1994-), 男, 硕士研究生, 研究方向为SDN网络路径增强和优化; 何利文, 博士, 教授, 研究方向为网络与信息安全、图像处理、云计算和大数据的技术与应用等。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190321.0904.002.html>

步发展,行业开始重新审视传统的网络架构。传统网络结构是分层的,由分布式排列的交换机和路由器构成,这种设计在客户机-服务器计算时是占主导地位的,但是这样的静态架构不适合今天的企业数据中心、校园网、运营商环境的计算和存储需求^[2]。近年来,SDN(software defined networking,软件定义网络)的提出,为目前出现的这些网络问题找到了新的方向。不同于传统的网络,SDN 控制器将网络设备的数据平面和控制平面分离,使用一个集中的控制器来决定网络中所有转发组件的行为^[3]。并且为相关的网络应用提供可编程的接口,利用软件的方式对整个网络的资源进行控制,革命性地改变了现有的网络架构^[4]。

在传统的网络体系架构中,路由协议改进非常困难,这主要是由于设备更新换代以及网络自身的复杂性比较高,同时也会带来非常巨大的开销等^[5]。传统的 QoS 体系结构(基于分布式控制平面)被证明是过于静态的,无法获得更广泛的应用^[6]。SDN 的出现直接降低了新型网络开发的成本,其倡导控制转发分离、网络能力接口的开放、软硬件解耦和网络功能的虚拟化,这将会推动网络架构向软件化、集约化、智能化和开放化的目标网络架构演进^[7]。

它的关键技术包括控制和转发解耦、实现控制集中化和网络能力开放化这几个方面,一个 SDN 控制器可以对转发平面多台设备进行通信,同时提供了一系列的开放编程接口,这样对网络业务的加载和控制更加灵活^[8]。以往的路由算法通常是由分布式实现,即交换机需要生成并维护转发表,实现对网络的集中配置与管理,而在 SDN 网络中,数据转发由控制器集中管理^[9],因此,只需要通过控制器来设计与数据转发有关的程序。在具有全局的网络状态和应用需求的情况下,可以在一个集中的系统中执行得更高效^[10],SDN 控制器的出现大大简化了路由创新的门槛。

网络资源的中心化让以前的分布式算法控制路由转发成了历史,这也是造成网络低效的原因之一。现在通过中心控制器可以高效地管理数据转发,让网络流量更加可控,可以直接使用 Dijkstra 算法^[11]来计算最短路,达到最佳路由。然而,SDN 控制器支持的最短路径算法无法较好地部署三层路由技术,还有就是 Dijkstra 算法的时间复杂度并不低,随着网络结点的增加,CPU 的消耗越来越高,网络寻址的效率也会降低。SDN 现在最常见的应用是在 IDC 网络中,这种网络通常会有成千上万个网络节点,因此,提出一种效率更高的算法是极具意义的。与此同时,网络服务提供商需要根据客户的 SLA(服务等级协议)为其定制特定级别的高质量网络服务,即需要满足如带宽、时延、严格下一跳等等条件^[12]。Qos 需求路由问题可以理解为

寻找满足多个约束条件的可行路径问题,这是属于 NP 难问题,通常无法在多项式时间内求解^[13]。所以文中采用基于启发式算法的遗传算法来解决这个问题。

遗传算法(genetic algorithm, GA)是由 J. H. Holland 等于 20 世纪 70 年代提出并发展起来的^[14],也是启发式算法的一种,还是一种具有全局搜索能力的直接、并行、随机的优化方法。它通过模仿生物的进化,遵循进化的主要原则,本质上是先通过随机方式选出初始种群,根据特定的评定标准,通过选择、交叉、变异的遗传算子的优化作用,优选出性能最佳的一系列个体^[15]。因为遗传算法具有很强的全局寻优能力和适应性,被广泛地应用于路径规划问题。所以这里采用多目标约束遗传算法解决多约束路由问题,既能一次计算产生多条可选路径路由,也可以根据业务的需求动态调整路由计算的参数,来得到所需要的路径。

文中提出的遗传算法是在已有物理拓扑状态下根据具体的业务标准需求的多目标约束遗传算法,可以在并行的情况下自适应地计算出符合路径约束条件的若干条可用路径,保证每个业务都有充足的备用路径可供调整。对该算法的主要思想和流程进行了介绍,然后对算法展开了详细描述,最后对算法在给定的网络环境下的性能进行了分析,验证该算法对 SDN 路径增强问题的有效性。

2 SDN 网络约束

2.1 网络模型及问题描述

对整个网络环境建立数学模型,网络拓扑关系用有向图 $G = (V, E)$ 来描述。其中 $V = \{v_i\} (i = 1, 2, \dots, n)$ 是网络中网元节点的集合, $|V| = n$ 包括交换机、路由器或者主机节点, E 为链路集合, $E = \{e_1, e_2, \dots, e_m\}$ 表示网络拓扑中有向图 G 的所有 m 对有向弧的集合, $|E| = m$, 每两个节点之间是用弧链接的,其中 $0 < m < \frac{n(n-1)}{2}$, $e^k = (v_i, v_j)$, 对应的费用和时延分别为 $c_k =$

$c(e_k) (c_k \geq 0)$ 和 $d_k = d(e_k) (d_k \geq 0)$, 带宽为 b_k 。存储图 G 的数据结构用邻接矩阵 L 表示,其中 L 是一个 $n \times n$ 的 0-1 矩阵,即 $L = (l_{i,j}) n \times n \in \{0, 1\} n \times n$ 。

$$L_{i,j} = \begin{cases} 0, & (i,j) \quad v_i, v_j \in V, e_k = (v_i, v_j) \notin E \\ 1, & (i,j) \quad v_i, v_j \in V, e_k = (v_i, v_j) \in E \end{cases}$$

即如果两节点之间有链路连通,则矩阵对应的元素为 1, 否则为 0。同理,链路带宽容量矩阵用 $B = \{b_{i,j}\} , v_i, v_j \in V, e_k = (v_i, v_j) \in E$ 表示,其中 $b_{i,j}$ 表示节点 i 和节点 j 之间链路的带宽容量大小;时延矩阵 $D = \{d_{i,j}\} , v_i, v_j \in V, e_k = (v_i, v_j) \in E, d_{i,j}$ 表示节点 i 和节点 j 之间的链路时延的值, c_{ij} 表示节点 i 和节点 j 之间的链路开销的值。

2.2 约束优化模型

假定两个节点 v_i, v_j 之间有一条不带环的通路路径,记为: $\text{path} = (v_i, v_j)$, 如果路径 path 经过链路 e_k , 那么 $e_k \in \text{path}$, 否则 $e_k \notin \text{path}$ 。文中路径是由染色体个体的形式表示的,即每个染色体代表一条路径,有 x_1, x_2, \dots, x_n , 这里使用变长的实数编码的方式,以 s 和 d 为其起始节点,每个个体所表示的路径经过的节点数量是不定的。多约束路由问题中的多约束条件可以表述为:

$$\sum_{i=s}^t \sum_{j \neq i} c_{i,j} \cdot x_{i,j} = \text{Cost} \quad (1)$$

$$i = \sum_{i=s}^t \sum_{j \neq i} d_{i,j} \cdot x_{i,j} = \text{Delay} \quad (2)$$

$$\sum_{i=s}^t \sum_{j \neq i} x_{i,j} \leq \text{Hop} \quad (3)$$

$$g_q(\vec{x}) = h(\vec{x}) - \text{Hop} \leq 0 \quad (4)$$

$$\forall (x_{ij} = 1) b_{i,j} \cdot x_{i,j} > \text{BandWidth}$$

$$x_{i,j} = \begin{cases} 1 & x \text{ 个体的路径经过 } L_{ij} \\ 0 & x \text{ 个体的路径不经过 } L_{ij} \end{cases}$$

$$h_1(\vec{x}) = c(\vec{x}) - \text{Cost} = 0 \quad (5)$$

$$h_2(\vec{x}) = d(\vec{x}) - \text{Delay} = 0 \quad (6)$$

其中, x_{ij} 表示路径 x 中对应的链路有效因子,当 (v_i, v_j) 属于 x 时, $x_{ij} = 1$, 否则 $x_{ij} = 0$ 。Cost、Delay、Hop、Bandwidth 分别是 SDN 路由中要求的额定开销、时延、跳数和带宽的可变参数,在这里一般表示为常数。

3 改进的多目标约束的遗传算法

3.1 路径编码

根据网络拓扑路径的特点,采用链式变长实数编码的方式,如:源节点是 0.0,目的节点是 120.3,路径依次经过节点 0.0、3.5、50.9、70.3、62.8、109.5、120.4,那么路径编码就是 0.0→50.9→70.3→62.8→109.5→120.4。这样的编码和解码的过程相对来讲比较简单,代表每一个节点的实数根据网络中节点的数量会做相应的调整,随着网络规模的变化相应地改变小数位的位数,如图 1 所示。

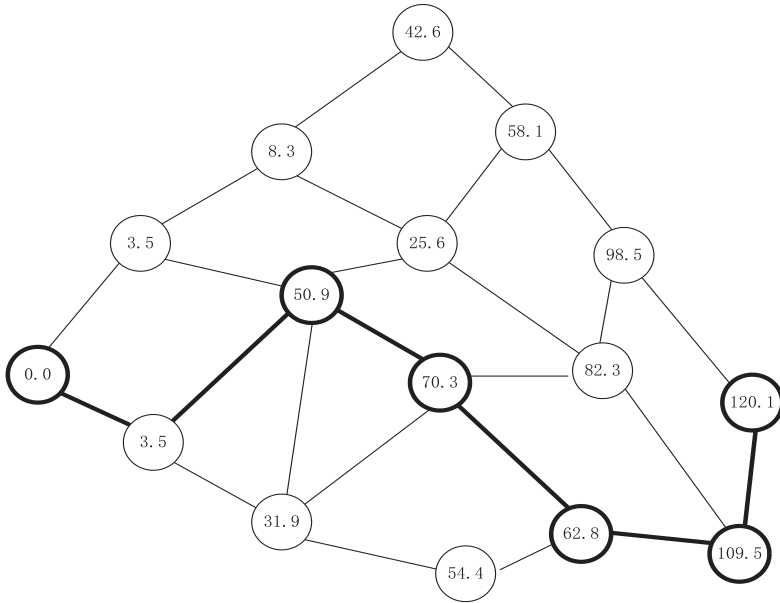


图 1 网络拓扑

3.2 种群初始化

遗传算法需要通过初始种群的生成的筛选满足路径规划类约束条件。这里引入了 He L 在用混合遗传算法解决电信网络备用路由的研究中提出的 SPA 算法(最短路径优先)^[16], 并且根据约束条件进行改进, 在原有随机路径选择生成的过程中满足相应的约束条件。算法流程如下:

(1) 从源节点开始;

(2) 寻找所有与源节点相连接的节点, 记录这些节点作为当前节点;

(3) 检查这些当前节点中是否有目的节点;

(4) 如果有, 则记录源节点和目的节点之间的链路, 计算完成, 否则, 继续计算;

(5) 继续查找与当前节点相连的节点, 作为当前节点;

(6) 查看这些节点是否有源节点, 如果有则将其从列表中删除;

(7) 重复步骤 3, 直到找到最短路径为止。

约束条件为: 严格下一跳和松散下一跳; 排除某个节点; 亲和力属性的处理。

改进的 SPA 算法:

- (1) 从源节点开始;
- (2) 寻找所有与源节点相连接的节点, 记录这些节点为当前节点集合;
- (3) 检查当前节点是否有目的节点;
- (4) 如果有, 则记录源节点和目的节点之间的链路, 计算完成, 否则继续计算;
- (5) 检查当前节点中是否需要排除的节点, 如果有则从当前节点中删除, 继续计算;
- (6) 检查当前节点集合中是否有严格下一跳或者松散下一跳的节点, 如果集合中有下一跳节点就走去, 否则继续计算;
- (7) 检查与当前节点集合中节点连接的所有链路上的亲和力属性有无满足当前业务的亲和力属性, 如果有就走该链路, 否则继续计算;
- (8) 继续查找与当前节点相连的节点, 作为当前节点;
- (9) 查看这些节点是否有源节点, 如果有则将其从列表中删除;
- (10) 重复步骤 3, 直到找到可行的路径为止。

3.3 个体选择

个体选择概率取决于种群中个体的适应度及其分布, 为了简单有效地控制选择压力并且使得个体有更好的鲁棒性, 选择使用轮盘赌的方法。解决路径规划的优化问题必须先解决问题模型中的约束条件问题, 包括一些等式和不等式的约束条件, 在遗传算法中对约束条件的处理会带来一些额外的参数。首先要将所有的等式约束条件转化成不等式约束条件, 这样就只处理不等式约束条件即可。文中采用静态惩罚函数法来处理不等式约束条件, 适应度函数定义为:

$$\text{Fitness}(\vec{x}) = f(\vec{x}) + \sum_{j=1}^s P_{k,j} C_j^2(\vec{x}) \quad (7)$$

其中, $P_{k,j}$ ($k = 1, 2, \dots, q, j = 1, 2, \dots, s$) 为特定的惩罚系数, q 是定义的对各个约束条件的违反水平。利用 SPA 算法计算出若干条路径作为个体, 随机选取 N 个这样的个体, $X_0^1, X_1^1, \dots, X_N^1, k = 0$; 然后从第 K 代

种群中以概率 $P = \frac{F(X_k^j)}{\sum_{l=1}^N F(X_k^l)}$ 选择子代种群

$X_k^1, X_k^2, \dots, X_k^N$ 。

3.4 交叉运算

交叉运算产生子代, 子代继承父代的基本特征, 主要包括两个内容: 第一是对种群中的个体进行随机配对并按照设定的交叉概率 P_m 来确定需要交叉的个体对; 第二是设定个体的交叉点, 然后对个体的部分片段相互交换。交叉算子的设计与编码方式有关, 在最优路径规划问题中有几种代表性的交叉算子, 如: 顺序交叉、类 OX 交叉等。这些交叉算子在产生新个体的过程中没有目的性, 对于实数编码的最优规划问题, 这些交叉可能破坏亲代的较优基因, 从而使交叉算子的搜索能力大大降低。基于此, 文中拟用在重复节点位置交叉且只进行一点交叉的操作方式, 具体实现步骤如下:

- 随机选取两个个体作为带交叉个体;
- 找到两个带交叉个体的共同点(源节点和目的节点除外)的集合;
- 从共同节点的集合中随机选择一个节点作为交叉节点;
- 检查两个带交叉个体在交叉节点之前和之后的内容是否相同。如果相同, 则取消此次的交叉操作。

具体的操作过程如下:

- (1) 设选取的两个待交叉样本为 0.0, 3.5, 50.9, 70.3, 62.8, 109.5, 120.4 和 0.0, 23.1, 50.9, 25.6, 62.8, 98.5, 120.4, 如图 2 所示;
- (2) 两者重复节点为 {50.9, 62.8};
- (3) 随机选择节点 50.9 作为交叉节点;
- (4) 检查发现两者带交叉样本在节点 50.9 之前和之后的内容均不相同, 因此可以进行此次操作, 交叉之后会产生新的个体:

0.0, 3.5, 50.9, 25.6, 62.8, 98.5, 120.4

0.0, 23.1, 50.9, 70.3, 62.8, 109.5, 120.4

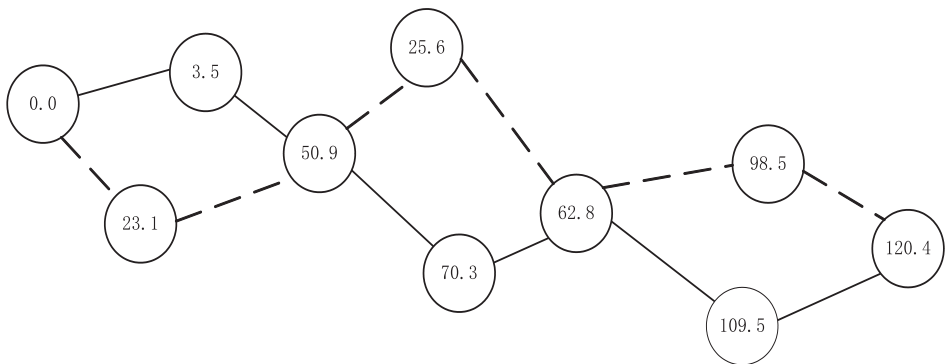


图 2 交叉操作

3.5 变异操作

变异操作是对个体的某些基因值做变动。目的有两个,第一使遗传算法具有局部的随机搜索能力,当经过交叉操作群体已接近最优解时,利用变异算子可以加速向最优解收敛;第二是使遗传算法可维持群体的多样性,以防止早熟现象。变异算子的设计与编码方法有关,对于实数编码的TSP问题,可采用逆转变异、对换变异和插入变异等。逆转变异,也称倒位变异,是指在个体编码中随机选择两点,再将这两点内的字串按反序插入到原位置中。倒位变异考虑了与原边的邻接关系,能将巡回路线上的优良基因性能较好地遗传到下一代,提高寻优速度。基于此,该项目具体实现步骤如下:

- (1) 随机选取一个个体作为待变异个体;
- (2) 在待变异个体中随机选择一个节点(起点和终点除外)作为待变异节点;
- (3) 找到与当前待变异节点直接相连的节点集合(该集合中不包括起点、终点以及待变异个体中的节点);

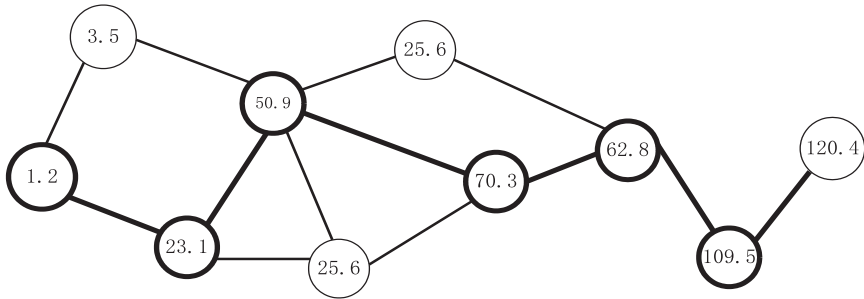


图3 变异操作

3.6 终止条件

初始种群规模相对较大或者网络拓扑规模相对较小时,算法收敛速度则相对较快。基于这一原因,算法终止条件选定为“迭代达到最大世代数”或者“种群中半数以上位置被生成的最优个体占据”。

3.7 算法完整流程

第一步:通过改进的约束型SPA算法计算出若干符合要求的路径并对其进行编码成为该算法中的个体,个体的表现型为变长的实数串,然后在其中随机选择出 N 条符合条件的个体作为初始化种群, N 即为初始化种群的规模,将其表示为: $X_0^1, X_1^1, \dots, X_N^1, k=0$;其中 k 代表种群的代数。

第二步:计算适应度值: $(X_k^i), i=1, 2, \dots, N$ 。

第三步:从第 k 代种群中以概率 $P = \frac{F(X_k^i)}{\sum_{i=1}^N F(X_k^i)}$ 选

择子代种群 $X_k^1, X_k^2, \dots, X_k^N$ 。

第四步:在子代种群 $X_k^1, X_k^2, \dots, X_k^N$ 中以相同的概

(4)从节点集合中随机选取一个节点作为变异后节点;

(5)检查待变异节点之前和之后的节点是否与变异后节点直接相连。若直接相连,则用变异后节点替代待变异节点完成变异过程;否则,放弃此次操作,回到第4步,直至将节点集合中的所有节点全部选遍。

操作的实现过程如下:

- (1) 选择待变异的个体为0.0, 23.1, 50.9, 70.3, 62.8, 109.5, 120.4, 如图3所示;
- (2) 随机选择节点50.9为变异节点;
- (3) 与节点50.9直接相连的节点集合为{3.5, 31.9, 25.6};
- (4) 随机选择节点3.5作为变异后节点;
- (5) 经过检查发现节点23.1与节点3.5并不直接相连,所以取消此次变异操作,接着选取节点31.9作为变异后的节点,检查后发现节点23.1和节点31.9、节点31.9和节点70.3直接相连,所以此次变异操作是可以的,变异后的新个体为:0.0, 23.1, 31.9, 70.3, 62.8, 109.5, 120.4。

率选出两个个体并以重组概率 P_r 重组,重复至遍历全部子代个体,产生新的种群 $X_k^1, X_k^2, \dots, X_k^N$ 。

第五步:以变异概率 P_m 执行变异操作,形成下一代种群 $X_{k+1}^1, X_{k+1}^2, \dots, X_{k+1}^N$ 。

第六步:检查代数是否已经达到设定的值,如果满足则停止遗传算法,否则重复第二步。

4 实验仿真与分析

实验的网络路由模型采用满足多约束的最小费用模型,网络规模为500节点24k链路,费用和时延皆在[1,10]中随机选取。带宽以1M为单位,链路的带宽分别在10G、20G、30G、40G中随机选取,随机网络生成后,随机选取源点和宿点并用最短路算法求解最小时延路的时延 D_1 和最小费用路的时延 D_2 ,取 $d = (D_2 - D_1)/4$ 。时延约束取不大于 $D_2 + d$;请求的业务带宽要求在1G、800M、500M、50M中随机选取,跳数约束为不大于80跳。

图4的实验结果是在初始种群规模为50,交叉变

异概率分别为 0.5, 0.3 的情况下生成的, 由此说明文中算法对这种大规模的约束网络模型具有比较快的收敛速度。

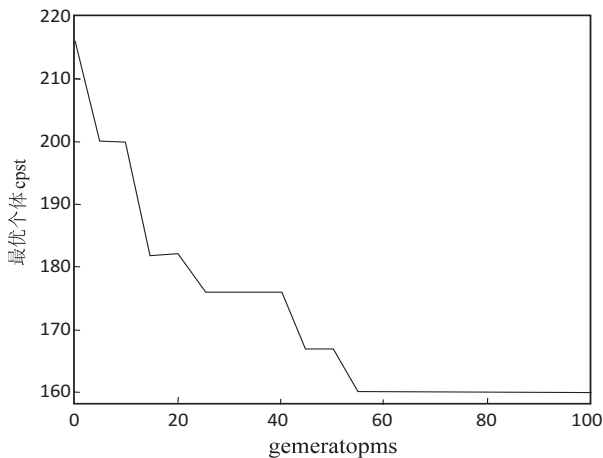


图 4 遗传代数与最优个体适应度值的变化

Dijkstra 算法在求解最短路径的过程中, 无论起始节点距离终点多远都需要遍历整网的拓扑, 在节点数为 n 的网络中, Dijkstra 算法的复杂度为 $O(n)$ 。对于节点数和边数比较大的大规模图, 当 n 比较大时, 该算法所需计算机的时间资源与空间资源将急剧增加; 对 GA 算法, 路径长度主要影响算法的收敛时间, GA 算法的搜索空间大小为路径的长度, 所以根据算法的特点决定文中的遗传算法更适合大规模网络路径的计算。图 5 是 KSP 算法和改进遗传算法同时计算 10 条无约束路径的耗时, 实验结果也验证了这个结论。

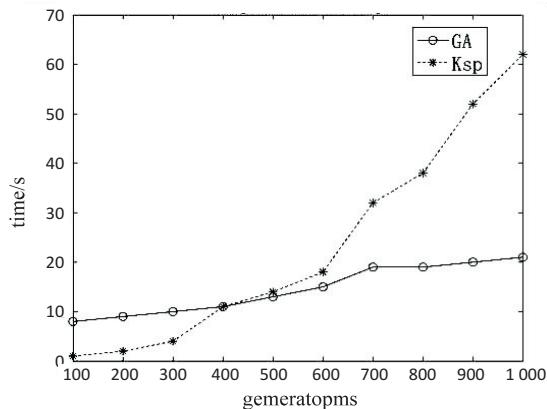


图 5 KSP 算法与文中算法的耗时比较

5 结束语

文中提出的基于多目标多约束的改进遗传算法可以在保证一定性能的前提下, 在大规模网络中计算出符合所有约束的最优路径。并且, 该算法主要具有以下特点: 改进了初始路径的生成方式, 显著提高了算法的搜索效率, 可以解决严格下一跳和松散下一跳的约束; 实数编码, 简化了编码和解码的操作, 省略了复杂

的解码过程; 启发式交叉策略, 加快了算法收敛的速度; 该算法较传统的 K 最短路径算法更适合大规模网络路径的计算。

参考文献:

- [1] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.
- [2] 赖香武, 彭大芹, 黄德玲, 等. 基于 SDN 的数据中心网络路由算法研究[J]. 无线互联科技, 2016(24): 38-40.
- [3] 吕小兵. 基于 OpenFlow 的 SDN 网络路由算法的设计与实现[J]. 自动化与仪器仪表, 2017(9): 128-130.
- [4] 邓书华, 卢泽斌, 罗成程, 等. SDN 研究简述[J]. 计算机应用研究, 2014, 31(11): 3208-3213.
- [5] 邓 谦. 基于盒覆盖的 SDN 路由算法研究[D]. 长沙: 湖南师范大学, 2015.
- [6] DAS S. A unified control architecture for packet and circuit network convergence[D]. USA: Stanford Univeristy, 2012.
- [7] 马军锋. SDN/NFV 关键技术问题分析和标准化进展[J]. 中兴通讯技术, 2016, 22(6): 12-16.
- [8] 李 光, 王延松, 范成法. M-ICT 时代 SDN 技术实践与创新[J]. 中兴通讯技术, 2016, 22(5): 61-66.
- [9] FEAMSTER N, REXFORD J, ZEGURA E. The road to SDN: an intellectual history of programmable networks[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(2): 87-98.
- [10] TOMOVIC S, RADUSINOVIC I. Fast and efficient bandwidth-delay constrained routing algorithm for SDN networks [C]//IEEE netsoft conference and workshops. Seoul, South Korea: IEEE, 2016: 303-311.
- [11] 汪胡青, 孙知信. 基于 Dijkstra 算法的多约束多播路由算法的研究[J]. 计算机技术与发展, 2011, 21(12): 5-8.
- [12] UPADHYAYA S, DEVI G. Characterization of QoS based routing algorithms[J]. International Journal of Computer Science & Emerging Technologies, 2010(20): 50-52.
- [13] 陈乐瑞, 孔金生. 基于改进遗传算法的网络路由优化研究[J]. 计算机应用与软件, 2013, 30(4): 135-137.
- [14] SAMPSON J R. Adaptation in natural and artificial systems (John H. Holland)[J]. SIAM Review, 2006, 18(3): 529-530.
- [15] HAMIDA S B, SCHOENAUER M. ASCHEA: new results using adaptive segregational constraint handling [C]//Proceedings of the 2002 congress on evolutionary computation. Honolulu, HI, USA: IEEE, 2002: 884-889.
- [16] HE L, MORT N. Hybrid genetic algorithms for telecommunications network back-up routing[J]. BT Technology Journal, 2000, 18(4): 42-50.