

基于线性规划的大规模网络故障恢复机制

刘 珂

(黄淮学院 信息工程学院,河南 驻马店 463000)

摘 要: 从大规模故障中进行网络恢复时可用的修复资源有限,需要经过多个修复阶段才能实现网络恢复。在过渡修复期间,网络运营商需确保重要流量的可达性。基于此,研究多个过渡修复阶段流量恢复率最大化和被切换路径数量最小化的问题,将其建模为线性规划问题,针对数百个节点构成的网络,基于分而刺穿思想提出启发式算法,获得流量恢复率和被切换路径数量的帕类托最优解。仿真实验结果表明,该算法生成的次优解与线性规划生成的最优解只相差 4%,与现有算法相比,可将受影响的路由元素数量下降 60% 左右,同时保证流量比相当。

关键词: 大规模故障;网络恢复;流量修复率;被切换路径数量;线性规划

中文引用格式: 刘 珂. 基于线性规划的大规模网络故障恢复机制[J]. 计算机工程,2016,42(7):104-108,116.

英文引用格式: Liu Ke. Massive Network Failure Recovery Mechanism Based on Linear Programming[J]. Computer Engineering,2016,42(7):104-108,116.

Massive Network Failure Recovery Mechanism Based on Linear Programming

LIU Ke

(School of Information Engineering, Huanghuai University, Henan, Zhumadian, Henan 463000, China)

[Abstract] In a scenario of recovery from massive failures, a network is repaired through multiple restoration stages because availability of repair resources is limited. During repair transition, it is crucial to ensure the accessibility of important flow for network operators. Therefore, this paper discusses the problem of maximizing the flow recovery ratio involving transient repairing stages and minimizing the number of switched paths. It formulates the problem as Linear Programming(LP), proposes a heuristic algorithm based on the dividing and impaling idea for networks consisting of a few hundred nodes, and obtains pareto-optimal solutions of the flow recovery ratio and the number of switched paths. Simulation results show that the algorithm can produce sub-optimal solutions within a 4% difference from optimal solutions, produced by LP, and it reduces the number of affected routing entries by about 60% compared with the current method while ensuring equivalent flow ratio.

[Key words] massive failure; network recovery; flow repair ratio; number of switched path; Linear Programming(LP)

DOI: 10.3969/j.issn.1000-3428.2016.07.018

1 概述

随着通信网络规模的不断增大与复杂性越来越高,发生大规模故障的概率不断上升。因为这些网络具有重要的作用,所以在发生故障时保证网络的可用性成为载波主干网络运营商们的重要课题^[1-2]。研究表明,解决上述故障可通过底层的保护机制^[3]或上层的恢复机制^[4]来处理(比如单链路或单节点故障、双故障和共享风险链路群组(SRLG)故障)。文献[5]提出一种基于贝叶斯征兆解释度的链路故障定位算法。该算法以概率加

权的二分图作为故障传播模型,定义一种新的参数贝叶斯征兆解释度,并基于该参数对潜在链路故障进行判断,得出最优故障假设集合,实现链路故障定位。文献[6]为提高网络单链路(节点)故障的快速恢复能力以增强网络可生存性,提出一种基于模拟退火粒子群优化及二分图最小点集覆盖算法的无环替代重路由技术优化方法。文献[7]提出一种基于颜色树的多播树生成方法及单节点/链路故障的多播通信恢复方案。然而,当网络故障大规模发生时会使网络组件陷入严重障碍。如果物理容量不足,则该方法无法有效处理大规模故障。因

基金项目: 河南省科技攻关计划基金资助项目(122102210430)。

作者简介: 刘 珂(1980-),男,讲师、硕士,主研方向为信息检索、网络安全。

收稿日期: 2015-02-06 **修回日期:** 2015-03-18 **E-mail:** 1913938431@qq.com

此,需要故障组件迅速恢复并重建通信路径。针对这些要求,文献[8-9]提出一种渐进式恢复方案。该方案认为修复资源(如物理和人力资源)的可用性有限且受到多个时间限制,因此假设故障组件可逐渐修复。该方案的目标是在多个过渡修复期间恢复尽量多的流量。有2种类型的恢复:一种是同时优化物理链路修复次序和传输路径^[8];另一种是只优化物理链路修复次序^[9]。然而,这2种方案均无法抑制多阶段过渡带来的风险和运营开销。例如,最优光(light)链路的切换不仅会影响交换机的运营负载,还会影响分布式IP或集中式SDN网络的上层路由。逻辑拓扑结构的逻辑链路由光路径构成,逻辑拓扑结构发生变化后需要对路由表的寄存元素进行重新配置,瞬时中断也会导致服务中断。为了弥补以上方案的不足,本文对多个过渡修复阶段流量恢复率最大化和被切换路径数量最小化这一问题进行研究,将该问题阐述为线性规划(LP)问题,并针对数百个节点构成的网络提出一种启发式算法,获得流量恢复率和被切换路径数量的帕类托最优解。

2 大规模故障场景假设与要求

本文对大规模故障场景做出如下假设:(1)至少有10%的物理网络资源发生故障^[10]。(2)修复故障组件需要时间,且物理和人力资源有限。因此,需要通过多个阶段逐步修复故障资源,如图1所示。

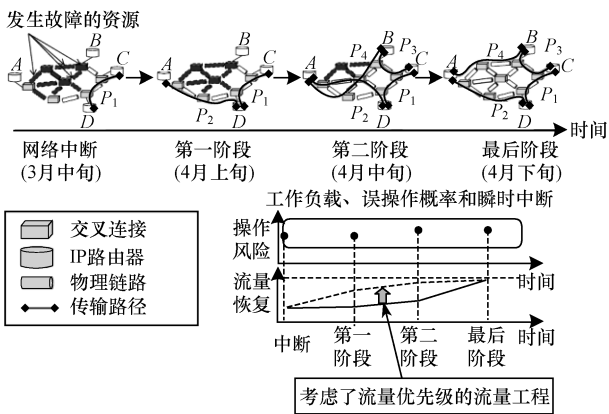


图1 大规模故障后基于多个阶段的网络恢复示意图

以上述假设为基础,本文相关定义要求如下:

(1)主干网在全国分布,由400个~500个节点构成。

(2)多个过渡期间的流量恢复率应该实现最大化。然而,大规模故障可能会导致过渡期间资源不足。因此,根据流量的优先级来进行网络恢复。对每组节点分配不同的优先级,高优先级的流量优先使用可用资源。

(3)为了降低运营负载、减少流量的瞬时中断,

被切换的路径数量应该最小。例如,在图1中,路径 P_2 的路由在从第1阶段至最后一个阶段的过渡期间发生变化。如果只考虑流量恢复优化,则这一情况将会出现。如果保证这一情景下有足够的流量恢复率,则建议不要变更第2阶段路径 P_2 的路由。

3 问题阐述

本节首先对网络模型进行定义,然后将多个过渡修复阶段下的网络流量恢复率最大视为线性规划问题 LP_e ,最后将使网络流量恢复率最大化和被切换的路径数量最小化问题建模为线性规划问题 LP_p 。

3.1 网络模型

图2为网络模型示意图。网络由多个报文层节点(如IP/MPLS节点)及传导层节点组成(如电力或光学交叉连接(EXC/OXC)),如图2(a)所示。通过交叉连接对报文层节点间的电学或光学路径进行配置。可以将这些路径看成是上层的逻辑链路。用户流量即发生于报文层节点间,根据上层(IP)路由策略及逻辑链路来传输流量。

虽然网络由多层节点构成,但是可以将这些节点看成是单层节点,如图2(b)所示:报文层节点被看成是入向和出向流量点,由上层协议(如分布式IP路由协议或集中式SDN网络)来确定报文层路由。然后,将网络看成是一个平面图 $G=(V,E)$,其中, V 表示一组物理节点(交叉链路); E 表示一组物理链路。每条链路 $(i,j) \in E$ 的容量表示为 C_{ij} 。

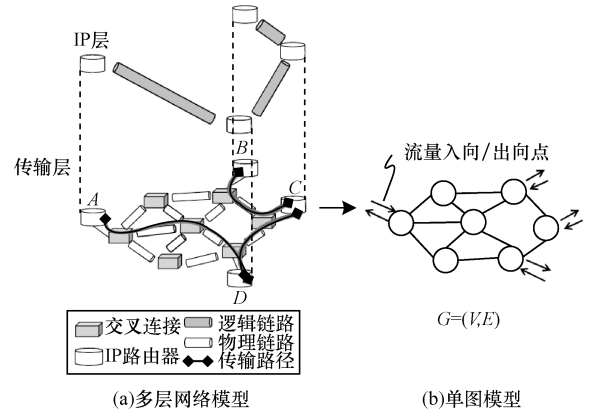


图2 网络模型示意图

根据上述假设,发生故障的组件(链路或节点)通过 K 重修复阶段即可实现缓慢修复。在本文标记法中,第1阶段从网络中断和急救阶段开始($k=0$),最后一个阶段为完全修复阶段($k=K$)。为了更好地表示物理组件的修复,引入二元常量 r_{ijk} 。如果链路 (i,j) 在阶段 $k(k>0)$ 被修复,则 r_{ijk} 为1。如果链路 (i,j) 在阶段0时未发生故障,则 r_{ij0} 也为1。否则, r_{ijk} 为0,即 r_{ijk} 满足如下条件:

$$\sum_{k=0}^K \sum_{(i,j) \in E} r_{ijk} = |E| \quad (1)$$

本文假设当与节点相连的所有链路可用时,该物理节点即被修复。整个网络的容量,即报文节点传输路径的可用性之和,表示为 M 。每个传输路径表示为 $p \in P$, 路径 p 的修复优先级 w_p 表示为:

$$w_{p-1} = \beta w_p, p = 1, 2, \dots, P \quad (2)$$

$$w_p = \beta M \quad (3)$$

参数 β 的数值足够大,路径的修复优先级 w_p 与其下标大小成反比。

3.2 线性规划描述 1 (LP_c): 流量优化

设 x_{ijk}^p 表示阶段 k 链路 (i, j) 的路径 p 的流量, F_k^p 表示阶段 k 路径 p 的端到端流量, 多个过渡修复阶段下的网络流量恢复率最大问题被认为是线性规划问题 LP_c , 具体描述如下:

$$\max \sum_{k=0}^K \sum_{p \in P} w_p F_k^p \quad (4)$$

$$\text{s. t. } \sum_{j: (i, j) \in E} x_{ijk}^p - \sum_{j: (j, i) \in E} x_{ijk}^p = \begin{cases} F_k^p & \text{if } i = s(p) \\ -F_k^p & \text{if } i = t(p) \\ 0 & \text{otherwise} \end{cases}$$

$$p \in P, i \in N, k = 0, 1, \dots, K \quad (5)$$

$$\sum_{p \in P} x_{ijk}^p \leq C_{ij} \sum_{l=0}^k r_{ijl} \quad i, j \in E, k = 0, 1, \dots, K \quad (6)$$

$$x_{ijk}^p \geq 0, F_k^p \geq 0 \quad (7)$$

式(4)表示以多阶段累积加权流量最大化为目标的目标函数。利用参数 w_p , 从优先级较高的路径开始依次确定路径 $p \in P$ 的路由和数量。式(5)是因为每个阶段的流量守恒约束。项 $s(p)$ 和 $t(p)$ 分别表示 p 的源点和终点。式(6)表示容量约束。如果链路 (i, j) 直到当前阶段 k 才实现修复, 则容量被充分利用。否则, r_{ijk} 之和为 0, 于是容量也被认为是 0。鉴于式(6), 该问题可看成是线性规划问题。

3.3 线性规划描述 2 (LP_p): 流量和路径切换优化

下面讨论使网络流量恢复率最大化, 同时使被切换的路径数量最小化的线性规划问题 LP_p 。其中, 上文目标函数式(4)由式(8)替代, 且新增式(9)。

$$LP_p \quad \min \sum_{k=1}^K \sum_{p \in P(i, j) \in E} t_{ij}^{pk} + \lambda (KM - \sum_{k=0}^K \sum_{p \in P} w_p F_k^p) \quad (8)$$

$$\text{s. t. } -t_{ij}^{pk} \leq x_{ijk}^p - x_{ij(k-1)}^p \leq t_{ij}^{pk} \quad i, j \in E, k = 1, 2, \dots, K, p \in P \quad (9)$$

在约束式(9)中, 变量 t_{ij}^{pk} 表示变量 x_{ijk}^p 在阶段 k 和 $k-1$ 间的绝对值之差。因此, 目标函数的第 1 项式(8)表示变量 x_{ijk}^p 在多阶段过渡期间差值最小化。路径路由的直接控制为弓形路径问题^[11], 所以路径路由的直接控制要求列出可能的当前路径。列出所有的当前路径在实践中并不可行, 所以该方法在实践中不可靠。通过使变量 x_{ijk}^p 之差最小化, 即可间接地使路径之差最小。式(8)第 2 项最小化等价于式(4)等累积流量最大化: 在该式中, 当累积流量增

加时, 第 2 项的值下降。通过改变 λ , 可改变第 2 项的强度, 即根据 λ 可计算不同的帕类托最优解。

4 启发式算法

4.1 大规模问题的求解方法

LP_c 问题可以看成经过优先级排序后的多商品流量最大问题^[8]。虽然基于多商品流的线性规划表述可提供一个严格最优解, 但是当节点数量线性增长时, 约束数量呈指数增长, 其计算过程在实践中不可行^[12-14]。因此, 本文提出一种称为“分而刺穿”的启发式算法, 可作用于由数百个节点构成的网络上。

本文降低计算量的直接策略就是将多阶段并行优化问题分割为多个单独阶段优化的子问题。为了求解分割后经过优先级排列的多商品最大流问题, 可用经典的 Edmonds-karp 算法^[15], 该算法可计算一对源节点和目标节点间的最大流, 计算复杂度为 $O(|V||E|^2)$ 。然而, 对问题进行简单的分割并不能保证多阶段过渡时路径的一致性。因此, 引入“刺穿”(impale)机制, 在该机制下, 过渡阶段的部分解由最终阶段(前向锁)和初始阶段(后向锁)的解确定, 以便抑制路径切换。

4.2 算法内容

为了便于算法描述, 表 1 为本文算法需要用到的符号含义。

表 1 相关符号标记法

符号	含义
$G_i = (V_i, E_i)$	阶段 i 包括节点 V_i 和无向链路 E_i 的图
P_i	阶段 i 输入路径组成的集合
P_i^{out}	阶段 i 输出路径组成的集合
$F(p)$	返回路径 p 的流量
Edmondskarp(p, G)	返回图 G 路径 p 的最大流量
Available(G)	返回图 G 的可用资源
Requested(F)	返回 F 请求的资源
Register(p, P^{out})	记录路径 p , 以设置 P^{out}

算法 1 描述了单个阶段时经过优先级排列后多商品流问题的计算方法。首先, 按照其权重 w_p 的次序, 利用 Edmonds-karp 算法计算路径 p 的最大流。然后, 对可用资源进行更新(第 4 行), 将 p 登记为输出信息(第 5 行)。在本文部署中, 当没有最大流时, Edmondskarp() 子程序为路径 p 返回 0 值。因此, 该算法只为 P 中的所有路径重复执行上述步骤。

算法 1 经过优先级排列的多商品最大流问题单阶段计算过程

```

1: Maxflow( $P, G$ )
2: for ( $p = 0, 1, \dots, P, w_p > w_{p+1}$ )
3:      $F(p) = \text{Edmondskarp}(p, G)$ 

```

4: Available(G) - = Requested($F(p)$)
 5: Register(p, P_k^{out})
 6: 返回 P_k^{out}

算法 2 描述了利用“刺穿”(impale)机制抑制路径切换时的多阶段计算过程。利用算法 1(第 1 行和第 2 行)来计算阶段 K 和 0 的最大流。然后,通过前向锁过程(第 4 行~第 7 行)和后向过程(第 8 行~第 11 行)来计算阶段 1~阶段 $K-1$ 期间的最大流。在前向锁进行过程中,开始时将阶段 K 尽可能多的路径作为阶段 k 的解。具体地说,如果在阶段 K 有足够多的可用资源来满足路径 p 的资源需求,则对可用资源进行更新(第 6 行),将 p 记录为 P_k^{out} (第 7 行)。在前向锁过程之后,后向过程类似:阶段 0 尽可能多的路径作为阶段 k 的解。在前向锁和后向锁过程之后,计算剩余资源条件下阶段 k 的最大流,并将路径添加给 P_k^{out} (第 12 行)。最后,算法输出所有阶段的路径集合,然后终止(第 13 行)。

在时间复杂性方面,计算每条路径的最大流过程是算法 1 的主要内容,时间复杂度为 $O(|P||V||E^2|)$ 。前向和后向过程的复杂度为 $O(|P||V^2|)$ 。因为这些阶段的计算互相独立,所以从阶段 1~ $K-1$ 的多阶段路径计算可采用并行计算策略。

算法 2 采用 impale 机制的多阶段最大流计算过程

1: $P_K^{\text{out}} = \text{Maxflow}(P_K, G_K)$
 2: $P_0^{\text{out}} = \text{Maxflow}(P_0, G_0)$
 3: for ($k = 1, 2, \dots, K-1$)
 4: for ($p \in P_K^{\text{out}}$)
 5: if (Available(G_K) > Requested($F(p)$))
 6: Available(G_K) - = Requested($F(p)$)
 7: Register(p, P_k^{out})
 8: for ($p \in P_0^{\text{out}}$)
 9: if (Available(G_K) > Requested($F(p)$))
 10: Available(G_K) - = Requested($F(p)$)
 11: Register(p, P_k^{out})
 12: $P_k^{\text{out}} = P_k^{\text{out}} \cup \text{Maxflow}(P_k, G_k)$
 13: 返回 $P_k^{\text{out}}, k = 0, 1, \dots, K$

5 性能评估

5.1 目标条件

本文从 2 个角度评估本文算法的性能:(1)传统方法^[8-9]和本文方法在线性规划方面的性能比较。(2)从求解精度和可扩展性角度评估启发式算法相对线性规划方法的有效性。表 2 对各评估方法进行了总结。基于传统方法^[8-9]的线性规划和启发式算法分别表示为 LP_c 和 HS_c 。注意,利用算法 2(删除了第 4 行~第 11 行“刺穿”(impale)部分)来计算传统方法的启发解,可对流量恢复率和切换路径数量进行优化,线性规划和启发式算法分别表示为 LP_p 和 HS_p 。

表 2 评估方法列表

目标函数	线性规划	启发式算法
流量恢复	LP_c	HS_c
流量恢复 + 切换路径数量	LP_p	HS_p

评估指标 1:过渡期间累积流量恢复(CTR)率和被切换的路径总量。CTR 率表示传输路径可用性之和,且进行正规化,以便对 LP_c 算法有 $\sum_{l=0}^K \sum_{p \in P} w_p F_p^l = 1$ 。假设路径切换频率取决于拓扑密度。本文讨论 2 种类型的拓扑结构,这 2 种结构的节点度(每个节点的链路数量)不同:高密度 Cost239 模型(11 个节点和 25 条链路),稀疏型 Abilene 模型(11 个节点和 14 条链路)。每条链路的容量为 100 Gb/s,定义 7 条入向/出向节点与 IP 路由器相连。为了评估路径切换对 IP 层的影响,本文还定义了 IP 路由器中路由元素的数量。每个路由器有 100 个~1 200 个路由元素。最大值的确定基于如下假设:大约有 300 000 条互联网完全路由基于“/24”掩蔽路由聚合进行登记。根据第 2 节描述的假设,Cost239 和 Abilene 模型分别有 14(56%)条和 5(36%)条链路故障。这些故障路径数量会影响修复阶段的数量,且按照随机次序通过 5 个阶段($K=5$)进行修复。在线性规划的阐述中,参数 λ 从 0.1 变为 10, β 设为 10。

评估指标 2:启发式算法的最优性和计算时间。CTR 方面的最优性表示启发式解的总接收流量与线性规划解的总接收流量之比。切换路径数量方面的最优性则刚好相反,表示为线性规划解切换路径总量与启发式解的切换路径总量之比。因此,当启发式解接近于最优解时,最优性达到 100%。为了评估最优性,本文衡量了入向/出向特征各不相同的 10 个样本。考虑到可读性,按照升序次序比较样本中的切换路径数量。为了评估计算量,根据平均节点度为 2 的 Waxman 拓扑模型^[16],将节点数量和路径数量设置为 400 和 100。从链路总量中随机选择 50% 左右的链路作为故障链路,并通过 5 个阶段加以修复。本文使用 CPLEX 作为 LP 求解程序,基于四核 Xeon 3.4 GHz,16 GB 内存环境来测量计算时间。

5.2 流量恢复率与被切换路径数量(LP_c vs LP_p)

图 3 给出了 CTR 率和被切换路径数量间的关系。通过改变参数 λ , LP_p 可获得多个帕类托最优解。帕类托最优性表示一种求解分配状态,在该状态下,无法使其中一个解变优的情况下不让至少一个其他解变差。显然, LP_c 的解不是帕类托最优。例如,为了保证充分的流量恢复(CTR 率为 1.0),与本文帕类托最优解相比, LP_c 要求 Abilene 多进行一次路径切换, Cost239 模型多进行 14 次路径切换。

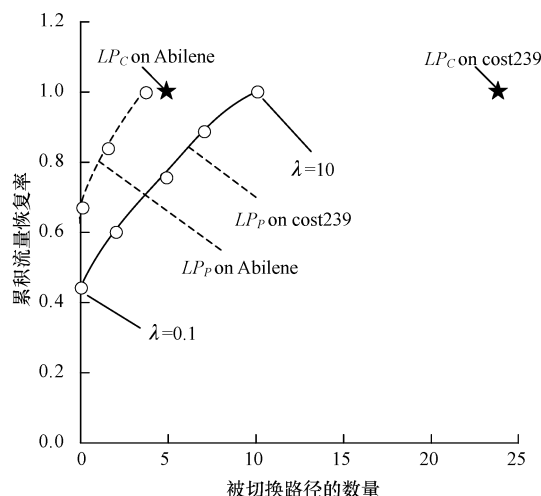


图3 CTR和被切换路径总量间的关系

通过更改 λ 参数, LP_p 算法可生成多个帕类托解。从图3可以看出, 结果依赖于拓扑密度, 高密度 Cost239 模型路径切换数量下降幅度显著。当节点度上升时, 网络总容量增加至给定流量水平, 即对 CTR 率进行优化的解空间得以拓展。该结果表明, 作为对 CTR 率进行优化的前提, 对被切换路径数量进行优化的解空间也得到了拓展。

为说明底层传输路径切换对 IP 层的影响, 图4给出了 Cost239 模型 CTR 率为 1.0 时 IP 路由器中发生变化的路由元素的数量。

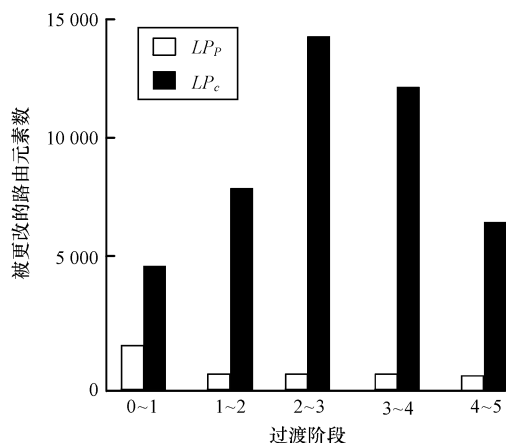


图4 IP路由器在过渡期间发生变更的路由元素数量

一个路由器有 100 个 ~ 1 200 个元素, LP_c 算法十几个切换路径就会导致路由元素进行数千次更新。很显然, 随着更新数量的不断上升, 服务中断的风险相应也会上升。利用本文 LP_p 算法, 这些更新的次数可压缩到 1 000 次以下 (不包括首次过渡), 更新次数下降至 60% 左右。过渡期间的更新次数取决于受到影响的路由器中记录的元素数量。如果在某些恢复阶段 (比如第 2 阶段 ~ 第 3 阶段, 第 3 阶段 ~ 第 4 阶段) 切换路径数量上升, 更新次数也会上升。

5.3 计算时间 (LP vs 启发式算法)

图5给出了 LP_c , LP_p , HS_c 和 HS_p 的计算时间。在数值示例中, LP_p 对 100 个节点的计算时间约为 5 h。观察发现, LP_p 算法中的约束数量会呈指数增长: 当有 50 个节点时约束数量为 5 000 个左右, 当有 100 个节点时约束数量为 90 000 个左右。另一方面, 与 HS_c 相比, HS_p 的计算量略有上升, 但是它它对 400 个节点仍可在 140 s 内完成计算任务。根据第 4 节描述的时间复杂度, 当路径数量上升时, 计算时间呈线性上升。如果允许数小时内的计算时间, 则 HS_p 可用于 10 000 条路径场景。

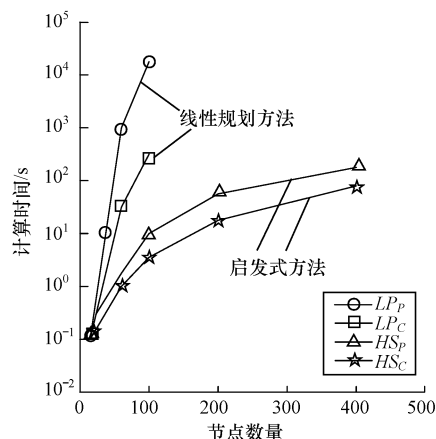


图5 四核 Xeon 3.4 GHz, 16 GB 内存环境下的计算时间

6 结束语

本文对流量恢复率优化和路径切换时过渡风险抑制进行研究, 将其阐述为线性规划问题, 并计算出帕类托最优解。对高密度网络模型, 本文算法的性能更高, 与当前算法相比, 可将受影响的路由元素数量下降 60% 左右, 同时保证流量比相当。为了提升可扩展性, 本文提出一种“分而刺穿”算法。该算法可在数分钟内完成计算任务, 生成的次优解与最优解只差 4% 左右。下一步工作将回顾启发式算法的精度对节点数量和阶段的依赖性, 并将物理资源的修复次序应用到多目标优化问题中。

参考文献

- [1] Woo S, Kim H. An Empirical Interference Modeling for Link Reliability Assessment in Wireless Networks [J]. IEEE/ACM Transactions on Networking, 2013, 21(1): 272-285.
- [2] Silva I, Guedes L A, Portugal P, et al. Reliability and Availability Evaluation of Wireless Sensor Networks for Industrial Applications [J]. Sensors, 2012, 12(1): 806-838.
- [3] Dhadwal D, Arora A, Singh V R. Optimization of Shared Path Protection for SONET/SDH Network [J]. International Journal of Scientific and Engineering Research, 2014, 5(7): 1258-1264.

(下转第 116 页)

种适用于中小型信息系统风险评估的轻量级方法。

在真实信息系统的长期运维中,通常需要定期对系统实施风险评估,在多次评估中,存在大量的重复工作环节。在后续的研究中,将针对如何应用以往评估过程及结果减少后续评估的工作量,以及如何利用每次评估的结果指导系统安全措施的选择进行研究。

参考文献

- [1] 戚湧,王艳,李千目,等.基于情景感知的网络安全风险评估模型与方法[J].计算机工程,2013,39(4):158-164.
- [2] 王兴明,陈裕颈,王芙蓉,等.IT运维操作风险评估方法初探[C]//银行和第三方支付行业信息安全等级保护技术大会论文集.北京:北京市公安局与中国人民银行营业管理部,2013:72-75.
- [3] Singh A, Picot A, Kranz J. Information Security Management(ISM) Practices;Lessons from Select Cases from India and Germany[J]. Global Journal of Flexible Systems Management,2013,14(4):225-239.
- [4] Calkins M. Risk Management: Capturing the Right Situation with Prudent Case Use in Scenario-based Modeling[J]. Developing a Virtue-Imbued Casuistry for Business Ethics,2014,42:225-241.
- [5] 姚远,肖应霖,谈向东,等.信息安全风险管理在企业访问控制管理中的应用研究[J].信息网络安全,2012(12):77-79.
- [6] 何丽.实时风险评估综述[J].电脑知识与技术,2014(6):1176-1177,1196.
- [7] 林云威,陈冬青,彭勇,等.基于D-S证据理论的电厂工业控制系统信息安全风险评估[J].华东理工大学学报:自然科学版,2014,40(4):500-505.
- [8] 罗森林,张蕾,郭亮,等.一种高效的攻击树串行建模方法[J].北京理工大学学报,2013,33(5):500-504.
- [9] Lazzarini B, Mkrtchyan L. Analyzing Risk Impact Factors Using Extended Fuzzy Cognitive Maps[J]. IEEE Systems Journal,2011,5(2):288-297.
- [10] Papageorgiou E I. Learning Algorithms for Fuzzy Cognitive Maps——A Review Study[J]. IEEE Transactions on Systems,2011,42(2):1-14.
- [11] 彭珍,田立勤,吴静,等.基于大型模糊认知图的复杂系统建模与推理研究[J].计算机科学,2013,40(6):203-205,210.
- [12] 康兵义.证据认知图及其在社会经济系统建模中的应用[D].重庆:西南大学,2013.
- [13] Jetter A, Schweinfurt W. Building Scenarios with Fuzzy Cognitive Maps:An Exploratory Study of Solar Energy[J]. Futures,2012,43(1):52-66.
- [14] 王旭,陈涛,张建业,等.信息安全风险评估技术在电力系统的应用[J].网络安全技术与应用,2013(4):71-74.
- 编辑 刘冰
- (上接第108页)
- [4] Kamamura S, Miyamura T, Pelsser C, et al. Minimum Backup Configuration-creation Method for IP Fast Reroute[C]//Proceedings of GLOBECOM'09. Honolulu, USA:IEEE Press,2009:1-6.
- [5] 王汝言,吴晴,熊余,等.基于贝叶斯征兆解释度的链路故障定位算法[J].计算机应用研究,2013,30(3):712-714.
- [6] 王明鸣,孟相如,庄绪春,等.基于无环替代重路由优化的网络可生存性增强方法[J].微电子学与计算机,2014,10(2):110-113.
- [7] 孙伟,罗俊海,肖志辉.基于SimCT的多播路由及故障恢复研究[J].电信科学,2012,27(12):90-96.
- [8] Wang J, Qiao C, Yu H. On Progressive Network Recovery After a Major Disruption[C]//Proceedings of the 30th IEEE International Conference on Computer Communications. Washington D. C., USA:IEEE Press,2011:1925-1933.
- [9] Yu H, Yang C. Partial Network Recovery to Maximize Traffic Demand[J]. IEEE Communications Letters,2011,15(12):1388-1390.
- [10] Horie T, Hasegawa G, Kamei S, et al. A New Method of Proactive Recovery Mechanism for Large-scale Network Failures[C]//Proceedings of the 23rd IEEE International Conference on Advanced Information Networking and Applications. Bradford, UK:IEEE Press,2009:951-958.
- [11] Maurras J F, Vanier S. Network Synthesis Under Survivability Constraints[J]. Quarterly Journal of the Belgian, French and Italian Operations Research Societies,2004,2(1):53-67.
- [12] Madry A. Faster Approximation Schemes for Fractional Multicommodity Flow Problems via Dynamic Graph Algorithms[C]//Proceedings of the 42nd ACM Symposium on Theory of Computing. New York, USA:ACM Press,2010:121-130.
- [13] Xu K, Liu H, Liu J, et al. LBMP: A Logarithm-barrier-based Multipath Protocol for Internet Traffic Management[J]. IEEE Transactions on Parallel and Distributed Systems,2011,22(3):476-488.
- [14] Labovitz C, Iekel-Johnson S, McPherson D, et al. InternetInter-domain Traffic[J]. ACM SIGCOMM Computer Communication Review,2011,41(4):75-86.
- [15] Edmonds J, Karp R M. TheoreticalImprovements in Algorithmic Efficiency for Network Flow Problems[J]. Journal of the ACM,2002,19(2):248-264.
- [16] Medina A, Lakhina A, Matta I, et al. BRITE: An Approach to Universal Topology Generation[C]//Proceedings of the 9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. Cincinnati, USA:IEEE Press,2001:346-353.
- 编辑 索书志