Evolutionary Computation at work for the optimization of link state routing protocols

Vitor Pereira
Centro Algoritmi / Dept. Informatics,
University of Minho
Portugal
vitor.pereira@algoritmi.uminho.pt

Pedro Sousa
Centro Algoritmi / Dept. Informatics,
University of Minho
Portugal
pns@di.uminho.pt

Miguel Rocha
Centre of Biological Engineering /
Dept. Informatics,
University of Minho
Portugal
mrocha@di.uminho.pt

ABSTRACT

This work explores the optimization of a real-valued parameter, assigned to each network node running the Distributed Exponentially-weighted Flow SpliTting (DEFT) routing protocol, in order to address changes on traffic conditions. This new proposal avoids the need to alter link weights and forwarding paths, by adjusting traffic splitting. In this context, we explore the use of Evolutionary Algorithms both in single and multi-objective optimization problems, to obtain solutions that minimize network's congestion.

KEYWORDS

Evolutionary Computation, Traffic Engineering

1 INTRODUCTION

The Distributed Exponentially-weighted Flow SpliTting (DEFT) [1, 2] is a link-state routing protocol built around the well-known shortest path algorithm from graph theory. Using topological information broadcast by all nodes on the network, each router calculates the shortest path to all others, which minimizes the sum of weights assigned to links. While most implementations use Equal-Cost Multi-Path (ECMP) [4] to equally distribute traffic along shortest paths, DEFT enables routers to forward portions of the traffic over non shortest paths, and thus enables a better use of paths which may otherwise be underutilized.

This work explores the utilization of distinct values for the parameter p in DEFT, assigned to each node, and their optimization by means of Evolutionary Algorithms (EAs). We show that by optimizing such values it is possible to accommodate traffic variations without the need to reconfigure the installed link weights, and therefore without changing the network routing paths. The EAs are used both to optimize link weights, for single and multi-objective problems, as well as p values.

2 MATHEMATICAL MODEL AND PROBLEM DEFINITION

Network topologies are modeled as directed graphs $G(N, A)$, with capacity constrains c_a, for each arc $a \in A$. Considering a given weight configuration w and a traffic matrix D, the distribution of traffic among the available resources is achieved using the DEFT routing protocol. DEFT assigns flows to a next-hop with a probability that exponentially decreases with the extra length of the path when compared with the shortest path. The distance from a node u to a node t, when traffic is routed thought a node v, can be expressed as $d_v^t + w_{u,v}$, where d_v^t is the shortest distance from the next-hop v to t, and $w_{u,v}$ is the weight of the link (u,v). The extra length of the path from u to t through v, when compared to the shortest path from u to t, is $h_{u,v}^t = d_v^t + w_{u,v} - d_u^t$. The flow proportion on the outgoing link (u,v) destined to t, at u, is computed by Equation 1:

$$P\left(h_{u,v}^t\right) = \frac{\Gamma\left(h_{u,v}^t\right)}{\sum_{(u,i) \in A} \Gamma\left(h_{u,i}^t\right)}, \tag{1}$$

where the exponential function Γ, Equation 2, maps the extra length $h_{u,v}^t$ of a path into the $[0;1]$ range.

$$\Gamma\left(h_{u,v}^t\right) = \begin{cases} e^{-\frac{h_{u,v}^t}{p}}, & \text{if } d_v^t < d_u^t \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

As the extra length $h_{u,v}^t$ is an integer value, the parameter p was initially thought as a constant to scale the penalizing function Γ to a range $[w_{\min}, w_{\max}]$ of possible link weights. This work considers the values of p to be distinct variables assigned to each node u, that can be optimized to improve traffic distribution. To evaluate the network resources utilization and congestion level we resort to the piecewise linear cost function Φ, proposed by Fortz and Thorup [3], that also serves as main objective function.

Given a network topology and one or more traffic matrices, for which an optimal link weights configuration is already installed, we aim to find a configuration of p values which minimizes the network congestion Φ when new traffic requirements are imposed, without any change to the installed link weights. The problem requires a pre-optimization of link weights, performed by a single-objective EA (SOEA), for a single traffic matrix (Scenario A), or with NSGAII [5], when two traffic matrices are provided (Scenario B).

Table 1: P-values Optimization - Scenario A: Single previously known traffic matrix

Measure	$Rand30_2$							$Rand50_2$						
	D_0	D_1			D_2			D_0	D_1			D_2		
	Opt.	Before	After	Opt	Before	After	Opt.	Opt.	Before	After	Opt	Before	After	Opt.
Congestion	1.39	8.61	3.28	1.49	16.45	5.93	1.76	1.59	8.22	1.75	1.83	31.29	13.46	11.18
Avg Link Utilization	35%	37%	37%	37%	43%	44%	43%	37%	38%	38%	37%	50%	51%.	49%
Max. Link Utilization	75%	121%	115%	76%	144%	117%	90%	94%	130%	100%	101%	155%	132%	121%

Table 2: P-values Optimization - - Scenario B: Two previously known traffic matrices

Measure	$Rand30_2$								$Rand50_2$							
	$D_{0,1}$	$D_{0,2}$	D_1			D_2			$D_{0,1}$	$D_{0,2}$	D_1			D_2		
	Opt.		Before	After	Opt	Before	After	Opt.	Opt.		Before	After	Opt	Before	After	Opt.
Congestion	1.46	1.39	18.91	2.43	1.73	34.12	16.29	4.07	1.23	1.28	17.76	1.99	1.59	28.30	8.06	5.32
Avg Link Utilization	34%	33%	43%	44%	42%	54%	55%	55%	25%	28%	37%	38%	37%	45%	46%	45%
Max. Link Utilization	77%	77%	155%	101%	90%	140%	135%	109%	89%	82%	132%	104%	94%	136%	118%	116%

3 CONFIGURATION SETTINGS

The SOEA and NSGAII have a similar configuration. The individuals that populate the initial populations are randomly generated. While for link weights optimization the solutions are integer vectors that take values in the range [1; 20] (each gene encodes a link weight), solutions for parameters p optimization are vectors of real values, one for each node, in the range [0.1 ; 10]. When $p = 0.1$, no traffic is forwarded on non shortest path links, and when $p = 10$ the non shortest path links are used nearly as if they were on shortest paths.

In the integer representation, the reproduction operators (all with the same probability) were: Random mutation, Incremental/decremental mutation, Uniform crossover and Two point crossover, while for real representations, the reproduction operators were: Gaussian Perturbation mutation, Linear Genome Random mutation, Uniform crossover and Real Value Arithmetical crossover. For the optimization of DEFT weights, a p value of 1 was applied to all nodes. The experiments were run for two synthetic networks $Rand30_2$ (30 nodes, 110 edges) and $Rand50_2$ (50 nodes, 196 edges).

4 RESULTS

In the experiments, we consider that a DEFT link weight configuration is already installed and optimized for a previously known traffic matrix D_0 (Scenario A), or for two traffic matrices $D_{0,1}$ and $D_{0,2}$ (Scenario B). These experiments analyze scenarios where heavy traffic needs to be forwarded, where demands are in the order of 30, 40 and 50% of the network capacity. Due to traffic changes, where the new traffic conditions are represented as two traffic demand matrices D_1 and D_2, which need to be accommodated, the network starts to experience congestion problems. Tables 1 and 2 include the congestion values, average link utilization and maximal link utilization for the distinct moments in time: the initial optimized configuration (Opt.); when the new demands are imposed while preserving the installed weights configuration and default p values (Before); and after p value optimization (After). For comparison, we also include the results for the link weights optimization of the new traffic demands. The results are the average of 10 runs with a termination criteria of 1000 iterations for link weights optimization and 200 iterations for the p values optimization.

The results show that, by optimizing p values to address new traffic requirements, the congestion levels on the network can be reduced, while preserving the installed weights. In some cases, the network continues to perform efficiently, with a congestion measure under 10.6, and therefore it was possible to accommodate the changes on traffic without any disruption on the forwarding paths. Although there are some links that continue to be over-utilized after the p values optimization, this became almost negligible. It is important to note that this solution aims to keep the forwarding paths configuration, while temporary traffic changes are imposed.

5 CONCLUSIONS

This work presented two applications of EAs to link state routing protocols optimization. Single objective and multi-objective EAs were used to obtain near optimal solutions to link weight setting problems, for DEFT routing protocols. Taking advantage of a parameter used by the routing algorithm, we experimentally analyzed a possible approach to respond to traffic changes responsible for congestion in networks. By optimizing the values of p assigned to each node on the network it was possible, in most cases, to reduce the overall network congestion for DEFT enabled networks, without resorting to a re-optimization of link weights.

REFERENCES

[1] D. Xu, M. Chiang, and J. Rexford. DEFT: Distributed exponentially-weighted flow splitting. *Proc. IEEE Conf. Comput. Commun.*, pp. 71–79, 2007.
[2] V. Pereira, M. Rocha and P. Sousa. Optimizing Load Balancing Routing Mechanisms with Evolutionary Computation. In Proc. of Intelligent Environments 2016 (FI&SN), pp. 298-307, 2016.
[3] B. Fortz. Internet traffic engineering by optimizing ospf weights. In *Proceedings of IEEE INFOCOM*, pp. 519–528, 2000.
[4] C. Hopps, Analysis of an Equal-Cost Multi-Path Algorithm, Internet Engineering Task Force , IETF , RFC 2992 (Informational).
[5] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evolutionary Computation, 6(2): pp. 182–197, 2002.