Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 52 (2016) 77–84

www.elsevier.com/locate/endm

A Game-theoretic Algorithm for Non-linear Single-Path Routing Problems

Josselin Vallet [1], Olivier Brun [2], Balakrishna Prabhu [3]

CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
Univ. de Toulouse, LAAS, F-31400 Toulouse, France

Abstract

This paper is devoted to non-linear single path routing problems, which are known to be NP-hard even in the simplest cases. We propose a Best Response algorithm, based on Game Theory, providing single-path routings with modest relative errors with respect to optimal solutions, while being several orders of magnitude faster than existing techniques.

Keywords: best response, single-path routing, game theory, non-linear programming

1 Introduction

We consider single-path routing problems with an additive and non-linear objective function. These routing problems belong to the class of non-linear mathematical programs involving integer variables (actually, binary variables in our case). These mathematical programs are known to be extremely hard

[1] Email: jvallet@laas.fr
[2] Email: brun@laas.fr
[3] Email: balakrishna.prabhu@laas.fr

http://dx.doi.org/10.1016/j.endm.2016.03.011

to solve, both from a theoretical and from a practical point of view. Even in the simplest case with binary variables, quadratic function and equality constraints, they are known to be NP-hard ([3]).

Several approaches have been proposed to solve non-linear integer programming problems. Some transform the discrete problem into a continuous one (see for example [8]). Despite their qualities, those techniques do not scale very well with the size of the problems. A recent alternative is the so-called *Global Smoothing Algorithm* [7], which seems to scale better while providing fairly good approximations (see Section 3.1 for details).

Heuristics and meta-heuristics have also been used to solved non-linear integer programming problems. Among others, ant-inspired optimization techniques are known to be efficient for solving various routing problems ([10], [6]). In this paper, we use the heuristic method proposed in [6] for comparison purposes (see Section 3.2 for details).

We propose an approximation algorithm, inspired from Game Theory [5], for solving non-linear single-path routing problems. Indeed, we assume that individual flows are allowed to independently select their path to minimize their own cost function. We note that a similar algorithm was proposed in [1] for scheduling of strictly periodic tasks. As will be shown numerically, the main merit of this algorithm is that it is several orders of magnitude quicker than the method discussed above while providing good optimization results.

1.1 Problem statement

Consider a network represented by a directed graph $G = (V, E)$. To each edge $e \in E$ is associated a capacity c_e, and a non-decreasing, continuously differentiable and convex latency function $\ell_e : \mathbb{R}_+ \to \mathbb{R}_+$. For any set $\pi \subset E$, we define the constant δ_π^e as 1 if $e \in \pi$, and 0 otherwise.

We are given a set $\mathcal{K}$ of flows, each one associated with a traffic demand λ_k. We let s_k and t_k be the origin and destination of flow k. Each traffic flow has to be routed in the network over a single path. We let Π_k be the set of all paths available for routing traffic between s_k and t_k. A routing strategy for flow k is defined as a vector $\mathbf{x}_k \in \mathcal{S}_k = \left\{\mathbf{x}_k \in \{0,1\}^{|\Pi_k|} \ : \ \sum_{\pi\in\Pi_k} x_{k,\pi} = 1\right\}$, where $\mathcal{S}_k$ is the set of all feasible routing strategies for flow k, and the binary variable $x_{k,\pi}$ is 1 if flow k is routed over path π, and 0 otherwise. The vector $\mathbf{x} = (\mathbf{x}_k)_{k\in\mathcal{K}}$ will be called a routing strategy. It belongs to the product strategy space $\mathcal{S} = \bigotimes_{k\in\mathcal{K}} \mathcal{S}_k$. For each edge $e \in E$, we define the function $y_e : \mathcal{S} \to \mathbb{R}_+$ as the amount of traffic flowing through link e under strategy $\mathbf{x}$. We define the cost of a routing strategy $\mathbf{x} \in \mathcal{S}$ as $\phi(\mathbf{x}) = \sum_{e\in E} \ell_e\left(y_e(\mathbf{x})\right)$. We

seek for a routing strategy with minimum cost, that is for the optimal solution of the following optimization problem:

$$\text{minimize } \phi(\mathbf{x}) = \sum_{e\in E} \ell_e\left(y_e\right) \qquad \text{(OPT-R)}$$

$$\text{subject to } \; y_e = \sum_{k\in\mathcal{K}} \sum_{\pi\in\Pi_k} \delta_\pi^e x_{k,\pi}\lambda_k \qquad e\in E, \qquad (1)$$

$$\sum_{\pi\in\Pi_k} x_{k,\pi} = 1 \qquad k\in\mathcal{K}, \qquad (2)$$

$$y_e \le c_e \qquad e\in E, \qquad (3)$$

$$x_{k,\pi}\in\{0,1\} \qquad \pi\in\Pi_k, k\in\mathcal{K}. \qquad (4)$$

2 Best-response algorithm

The best-response algorithm we propose takes its inspiration from an algorithm of the same name in Game theory [5]. In a game, the best-response of a player is defined as its optimal strategy conditioned on the strategies of the other players. It is the best response that the player can give for a given strategy of the others. The algorithm consists of players taking turns to adapt their strategy based on the most recent known strategy of the others. This algorithm is similar to the Gauss-Seidel iterative scheme.

The proposed Best-response algorithm converts the optimization problem (OPT-R) into the following game. Let us think of individual flows as players. The number of players is therefore $N = |\mathcal{K}|$. In the following, we denote by $\mathcal{P} = \{1, 2, \ldots, N\}$ the set of players. The game is assumed to be played sequentially with players taking turns in some fixed order until the strategies of the players converge. Let $\pi_j(n)$ denote the strategy (equivalently, the path) of player j at the beginning of the nth turn. If $k(j)$ is the flow corresponding to player j, we thus have $\pi_j(n)\in\Pi_{k(j)}$. The vector $\boldsymbol{\pi}(n)$ describes the strategy of the N players at iteration n. Using standard Game Theory notation, $\boldsymbol{\pi}_{-j}(n) = [\pi_1(n), \pi_2(n), \ldots, \pi_{j-1}(n), \pi_{j+1}(n), \ldots, \pi_N(n)]$ is the vector of paths of individual flows other than flow j. We define $y_e(\boldsymbol{\pi}(n))$ as the traffic routed through link e at turn n:$y_e(\boldsymbol{\pi}(n)) = \sum_{j\in\mathcal{P}} \delta^e_{\pi_j(n)}\lambda_{k(j)}$

In its turn, player j computes its path (using any shortest path algorithm) so as to minimize its own cost, that is, player j solves the following problem :

$$\text{minimize}_{\pi\in\Pi_{k(j)}} u_j\left(\pi, \boldsymbol{\pi}_{-j}(n)\right) = \sum_{e\in\pi} \ell_e\left(\lambda_{k(j)} + y_e(\boldsymbol{\pi}_{-j}(n))\right), \qquad \text{(OPT-}j\text{)}$$

and sets $\pi_j(n+1)$ to that value of π that gives the solution. Note that player j solves the same problem as (OPT-R) except that it takes into account only the terms that are affected by its own traffic. Note also that player j deviates from strategy $\pi_j(n)$ to a new strategy $\pi'_j(n)$ if and only if $u_j\left(\pi'_j(n), \boldsymbol{\pi}_{-j}(n)\right) < u_j\left(\boldsymbol{\pi}(n)\right)$. The pseudocode for the heuristic is given in Algorithm 1.

Algorithm 1 Best-response

Require: $\boldsymbol{\pi}(0)$
1: $n \leftarrow 0$
2: **repeat**
3: **for** $j = 1 \dots N$ **do**
4: $\pi_j(n+1) \leftarrow \text{argmin}\,(\text{OPT-}j)$
5: **end for**
6: $n \leftarrow n+1$
7: **until** $\boldsymbol{\pi}(n) \neq \boldsymbol{\pi}(n-1)$.
8: **return** $\boldsymbol{\pi}(n)$

The general convergence of the algorithm is not addressed in this paper. However, when all flows have the same traffic demands ($\lambda_k = \lambda$, $\forall k$), the convergence of the algorithm directly follows from the fact that the game is a congestion game [9]. For heterogeneous traffic demands, see [4] for a proof of convergence in the case of linear latency functions. Also, in [2], the authors found that the price of anarchy is bounded for affine and polynomial cost functions with non negative coefficients.

3 Global Smoothing Algorithm and Ant Colony Optimization

In this section, we present two existing approximation techniques that can be used to solve (OPT-R). We will use those algorithms for comparison purposes.

3.1 Global Smoothing Algorithm (GSA)

This algorithm was introduced in [7]. Designed to handle non linear optimization with binary variables, it may provide good integer solutions. Its original approach is to use a sequence of non-linear optimization without integer constraints. It optimizes the cost with two penalty functions : a logarithmic barrier penalty (designed to smooth the function and keep the current point away from the integer grid) and an "exact" penalty (describing how far the current point is from the integer grid). Both are weighted by one parameter,

which is updated at each iteration to gradually bring the current point on the integer grid. The sequence of penalized problem should form a trajectory leading to a good approximation of the solution. We refer to [7] for further details.

3.2 Ant Colony Optimization (ACO)

This meta-heuristic algorithm belongs to the ant colony algorithms family, which are efficient on problem that requires to find good paths in a graph.

The algorithm we consider was presented in [6]. It is designed to optimize an MPLS network - a direct application of our problem - and provides good approximated results within reasonable execution times. Basically, L ants are defined, each one exploring the space of solution depending on probability distributions. At each iteration, each ant construct a solution by routing each flow, leaving a trace of pheromones on its way, influencing the next ants. The algorithms stops after a fixed maximum number of iterations. This algorithm gives a solution after the first iteration, and each subsequent iteration can improve the solution.

4 Numerical Results

We analyse here the efficiency of the Best Response (BR) algorithm on two non linear cost functions, compared to GSA and ACO. The optimal multipath routing is considered as the lower bound to which we will compute a relative "error". Simulations are performed on 8 real network topologies (see Table 1), found in IEEE literature and in [11]. We consider at most 2 shortest paths (in terms of number of hops) for each source/destination pair. For each network, we consider a set of 100 full random traffic matrices. For all matrices, there exists a single-path routing solution such that no link capacity is excedeed. The number of variables associated to each scenario is shown in Table 1.

For ACO, we consider at most 50 iterations and 5 ants : experiments on our problems revealed that increasing those parameters did not improve the results, while increasing execution times. For GSA, two different multirouted starting points were tested : a random point, and a point close to the monorouted solution obtained with the BR algorithm. Results presented in the next sections are composed of the best solutions obtained from these two starting points, and were obtained using Matlab 2013b, on a Intel Core i5-2430M processor at 2.4 GHz, running under Linux with 4 GB of memory.

Topology	N	M	Binary variables
ABOVENET	19	68	664
ARPANET	24	100	1104
BHVAC	19	46	684
EON	19	74	684
METRO	11	84	220
NSF	8	20	112
PACBELL	15	42	420
VNSL	9	22	140

Table 1
Topologies : number of nodes (**N**), links (**M**) and number of binary variables

4.1 Quadratic cost function

We consider here the following quadratic cost function : $\ell_e(y_e) = \left(\frac{y_e}{c_e}\right)^2$

The results are summarized in Table 2. First, we can observe that each algorithm provide good optimization results on this cost functions, as the average relative error remains below 4.37% for all configurations. Secondly, GSA seems to provides the best results, but when looking at the execution times shown on table 4, we observe that BR outperforms clearly GSA and ACO, as its worst execution time remains below 1.08 seconds, while ACO reaches 50 seconds and GSA 300 seconds.

Topology	BR			GSA			ACO		
	min	max	avg	min	max	avg	min	max	avg
ABOVENET	1,42	5,81	4,16	$\simeq 0$	13,32	7,19	0,17	4,87	2,98
ARPANET	1,20	3,01	1,95	1,09	5,40	2,57	4,42	8,49	5,99
BHVAC	0,15	1,46	0,69	0,09	4,35	0,94	3,93	8,30	5,78
EON	$\simeq 0$	3,85	2,47	$\simeq 0$	3,63	0,89	2,24	6,63	4,53
METRO	2,23	22,67	9,43	1,64	8,90	4,47	2,63	13,32	7,41
NSF	0,10	16,92	4,07	0,04	4,43	1,76	0,15	7,46	2,21
PACBELL	2,04	5,46	3,72	0,32	3,00	1,44	2,30	8,00	4,76
VNSL	0,21	7,27	2,31	$\simeq 0$	8,57	3,07	0,09	4,92	1,34
All	0,00	22,67	3,60	$\simeq 0$	13,32	2,79	0,09	13,32	4,37

Table 2
Quadratic function relative error to OLS (%)

4.2 M/M/1 cost function

In this section, we study the results obtained with the cost function associated to an M/M/1 queueing model (Kleinrock's delay function) : $\ell_e(y_e) = \dfrac{y_e}{c_e - y_e}$

Results are presented in Table 3 : ACO and BR are clearly better than GSA in terms of obtained cost. GSA encounters difficulties on some instances, providing some bad maximum errors. BR provides the best average optimization results, while ACO is best for minimizing the maximum error. But as shown in Table 4, BR seems to offer the best trade-off between optimization efficiency and execution times, as the worst time remains below 0.6 seconds.

Topology	**BR**			**GSA**			**ACO**		
	min	max	avg	min	max	avg	min	max	avg
ABOVENET	$\simeq 0$	1,10	0,46	$\simeq 0$	87,54	5,95	1,94	6,23	3,59
ARPANET	$\simeq 0$	2,99	0,31	$\simeq 0$	77,01	3,44	2,58	7,64	4,82
BHVAC	$\simeq 0$	1,54	0,38	$\simeq 0$	64,58	8,57	3,30	8,54	5,17
EON	$\simeq 0$	4,03	0,52	$\simeq 0$	58,09	3,15	1,71	8,92	3,63
METRO	$\simeq 0$	12,90	1,46	$\simeq 0$	30,89	3,62	0,20	14,99	2,77
NSF	$\simeq 0$	20,85	1,03	0,01	60,14	3,75	0,01	8,77	1,47
PACBELL	$\simeq 0$	2,07	0,45	$\simeq 0$	45,13	5,30	1,44	7,14	4,00
VNSL	$\simeq 0$	2,05	0,66	$\simeq 0$	24,07	3,51	0,34	6,82	2,46
All	$\simeq 0$	20,85	0,67	$\simeq 0$	87,54	4,53	0,01	14,99	3,26

Table 3
MM1 function relative error to OLS (%)

Scenario	**Quadratic function**			**MM1 function**		
	BR	GSA	ACO	BR	GSA	ACO
Min	0,02	1,02	4,73	0,01	0,75	4,98
Max	1,08	297,73	48,83	0,58	400,82	50,95
Average	0,24	30,85	22,16	0,14	27,45	22,40

Table 4
Execution time (s) for all tested scenarios

5 Conclusion

The numerical results obtained with the proposed Best Response algorithm are promising : the relative error to optimal solution remains very modest (equivalent to other techniques), while the execution times are substantially

lower than the others. Those results make us think that this algorithm should clearly be considered for large scale problems, where other techniques show their limits. Future work will try to consider the theoretical results associated with the algorithm.

References

[1] Al Sheikh, A., O. Brun, P. Hladik and B. Prabhu, *A best-response algorithm for multiprocessor periodic scheduling*, in: *Real-Time Systems (ECRTS), 2011 23rd Euromicro Conference on*, 2011, pp. 228–237.

[2] Awerbuch, B. and et al., *The price of routing unsplittable flow* (2005).

[3] Cela, E., "The Quadratic Assignment Problem: Theory and Algorithms." Kluwer Academic Publishers, 1998.

[4] Fotakis, D., S. Kontogiannis and P. Sprirakis, *Selfish unsplittable flows*, Theoretical Computer Science (2005), pp. 226–239.

[5] Fudenberg, D. and J. Tirole, "Game Theory," MIT Press, Cambridge, MA, 1991.

[6] Garcia, J., A. Rachdi and O. Brun, *Optimal lsp placement with qos constraints in diffserv/mpls networks*, in: *Proceedings of the 18th International Teletraffic Congress - ITC-18*, Teletraffic Science and Engineering **5**, 2003 pp. 11 – 20.

[7] Murray, W. and K.-M. Ng, *An algorithm for nonlinear optimization problems with binary variables*, Computational Optimization and Applications **47** (2010), pp. 257–288.

[8] Pardalos, P., *Continuous approaches to discrete optimization problems*, in: G. Di Pillo and F. Giannessi, editors, *Nonlinear Optimization and Applications*, Springer US, 1996 pp. 313–325.

[9] Rosenthal, R. W., *A class of games possessing pure-strategy nash equilibria*, International Journal on Game Theory **2** (1973), pp. 65–67.

[10] Schoonderwoerd, R., O. Holl, J. Bruten and L. Rothkrantz, *Ant-based load balancing in telecommunications networks*, Adaptive Behavior **5** (1996), pp. 169–207.

[11] Spring, N., R. Mahajan, D. Wetherall and T. Anderson, *Measuring isp topologies with rocketfuel*, Networking, IEEE/ACM Transactions on **12** (2004), pp. 2 – 16.