

The edge-disjoint path problem on random graphs by message-passing

Fabrizio Altarelli,^{1,2,*} Alfredo Braunstein,^{1,3,2} Luca Dall'Asta,^{1,2} Caterina De Bacco,^{4,†} and Silvio Franz⁴

¹*DISAT and Center for Computational Sciences, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy*

²*Collegio Carlo Alberto, Via Real Collegio 30, 10024 Moncalieri, Italy*

³*Human Genetics Foundation, Via Nizza 52, 10126 Torino, Italy*

⁴*LPTMS, Centre National de la Recherche Scientifique et Université Paris-Sud 11, 91405 Orsay Cedex, France.*

We present a message-passing algorithm to solve the edge disjoint path problem (EDP) on graphs incorporating under a unique framework both traffic optimization and path length minimization. The *min-sum* equations for this problem present an exponential computational cost in the number of paths. To overcome this obstacle we propose an efficient implementation by mapping the equations onto a weighted combinatorial matching problem over an auxiliary graph. We perform extensive numerical simulations on random graphs of various types to test the performance both in terms of path length minimization and maximization of the number of accommodated paths. In addition, we test the performance on benchmark instances on various graphs by comparison with state-of-the-art algorithms and results found in the literature. Our message-passing algorithm always outperforms the others in terms of the number of accommodated paths when considering non trivial instances (otherwise it gives the same trivial results). Remarkably, the largest improvement in performance with respect to the other methods employed is found in the case of benchmarks with meshes, where the validity hypothesis behind message-passing is expected to worsen. In these cases, even though the exact message-passing equations do not converge, by introducing a reinforcement parameter to force convergence towards a sub optimal solution, we were able to always outperform the other algorithms with a peak of 27% performance improvement in terms of accommodated paths. On random graphs, we numerically observe two separated regimes: one in which all paths can be accommodated and one in which this is not possible. We also investigate the behaviour of both the number of paths to be accommodated and their minimum total length.

arXiv:1503.00540v1 [cond-mat.dis-nn] 2 Mar 2015

* Presently at Capital Fund Management, 23 Rue de l'Université, 75116 Paris, France

† caterina.de-bacco@lptms.u-psud.fr

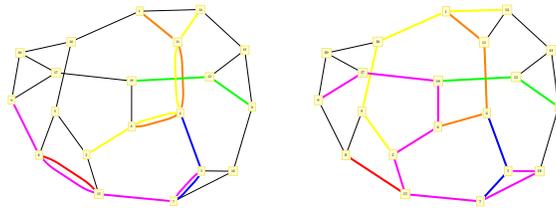


FIG. 1: An instance of the EDP problem over a 3-regular random graph of $V = 20$ and $M = 6$: examples of solutions of the unconstrained (left) and optimal (right) EDP problem are displayed. In the latter, the purple communication is redirected along a longer path to avoid edge-overlap. The yellow one has two shortest paths of equal length (degeneracy) in the unconstrained case, but once EDP is enforced this degeneracy is broken and only one of the two is optimal (right).

I. INTRODUCTION

The optimization of routing and connection requests is one of the main problems faced in traffic engineering and communication networks [1]. The need to deliver Quality of service (QoS) [2, 3] performances, when transmitting data over a network subject to overload and failures, requires both efficient traffic management and resource optimization.

Some aspects of these problems can be formalized using the edge-disjoint path (EDP) problem. This is a constrained optimization problem that is defined as follows. For a given network and a set of communication requests among pairs of users, the EDP consists in finding the maximum number of communications that can be accommodated at the same time, under the constraint that different paths cannot overlap on edges. Moreover, the additional requirement of minimization of the total path length can be considered. Apart from a purely theoretical interest [4], the EDP finds a wide range of applications: in very-large-scale-integration (VLSI) design, in admission control and virtual circuit routing and in all-optical networks. In VLSI design it is required to route wires on a circuit avoiding overlaps, along with minimizing the length of the wires [5, 6]. In admission control and virtual circuit routing [7–9] one needs to reserve in advance a given path for each communication request so that once the communication is established no interruption will occur. This has applications in real-time database servers, large-scale video servers [10–12], streaming data and bandwidth reservation in communication networks [13–16] and in parallel supercomputers. All these applications require high quality data transmission and full bandwidth exploitation. Routing via edge disjoint paths allows for an efficient bandwidth allocation among users because overlap avoidance means full bandwidth exploitation by each single user. An area that has attracted particular attention in the last decade is communication transmission in all-optical networks. Along an optical fiber different communications cannot be assigned the same wavelength to transmit data. Moreover, a unique wavelength must be assigned on all the edges contributing to the path assigned for a given communication. Routing communications under the above two requirements define the problem of routing and wavelength assignments (RWA) in this type of networks [17]. These two constraints suggest that a strategy that iteratively builds edge disjoint paths solutions could allow for a more efficient bandwidth management, namely by using an overall smaller number of wavelengths. This leaves available the remaining ones (according to the edge capacity) to be used either by new users entering the network or by allowing current users to exploit higher bandwidth. This strategy has indeed been applied using greedy [18] and genetic algorithms [19] with performances comparable to other methods based on integer linear programming, graph coloring or bin packing.

The EDP is classified among Karp’s NP-hard combinatorial problems [20, 21]. Defining the approximation ratio of a given algorithm as the ratio between the result obtained in term of cost/profit by the algorithm and the optimal one (or viceversa depending on what order gives the maximum ratio), the EDP problem is hard to approximate in the worst case; it has been proved that even an approximation with ratio $O(m^{\frac{1}{2}-\epsilon})$ is NP-Hard. The best known approximation ratio for the number of accommodated paths is $O(\min\{n^{2/3}, \sqrt{m}\})$ [22, 23] where n and m are the number of nodes and edges in the graph, respectively. Negative results on worst-case inapproximability did not stop progress on heuristic approaches. The problem has been studied intensely with a variety of classical techniques: heuristic greedy algorithms [13, 14, 18, 24], elaborated strategies using bin packing [25], integer/linear programming relaxations [26–29], post-optimization [30], Montecarlo local search [31], genetic algorithms [32–34], particle swarm optimization [35] and ant colony optimization [36], among them.

In this paper we propose a distributed algorithm to solve the EDP problem based on message-passing (MP) techniques (or cavity method) [37]. This method has been extensively employed to address problems in spin glass theory [38–40], combinatorial optimization [41] and more recently in routing problems on networks [42–45]. The evaluation of the equations at the core of the MP technique requires, for each vertex i in the underlying graph, to solve a local combinatorial optimization problem, performing a minimum over a set which is exponentially large in the number of neighbors of i . We propose an efficient method to perform this calculation, by mapping it into a minimum-weight matching problem on a complete auxiliary graph with vertices in the set ∂i of neighbors of i , that can be solved by classical algorithms [46]. With this construction, each iteration of the MP equations can be computed in a time which is polynomial in the number of graph edges (and linear in average for sparse random graphs).

The MP algorithm is tested on computer-generated instances of different classes of random graphs to study the scaling properties with the system size and to compare the performances against a greedy algorithm. We also considered the EDP problem on some benchmark instances found in the literature, for which we could compare the message-passing results with those obtained using other types of algorithms: greedy, ant colony optimization [36] and Montecarlo local search [31].

The paper is organized as follows. In Section II we define the EDP optimization problem, for which we present the message-passing equations in Section III, together with the mapping on a matching problem that simplifies their actual implementation. Section IV reports the results of simulations on random graphs and the scaling of the relevant quantities with the system size, while the comparison between the performances of the message-passing algorithm and other methods is discussed in Section V. Conclusions are given in Section VI.

II. THE EDGE-DISJOINT PATHS PROBLEM

Given a network and a set of M communications requests between pairs of senders and receivers, the standard EDP consists in finding the maximum number of accommodated paths which are mutually edge disjoint. In the applications described in the Introduction, the length of the communication paths is a quantity that, directly or indirectly, affects the overall transmission performances, in terms of transmission delays, infrastructure cost and network robustness. We take into account this aspect by considering the Minimum Weight Edge Disjoint Paths (MWEDP) problem, a generalization of the EDP problem that combines in a unique framework both path length optimization and edge disjointness. An instance of the MWEDP problem is defined by a graph $G(\mathcal{V}, \mathcal{E})$, where \mathcal{V} denotes the set of nodes and \mathcal{E} is the set of edges, by an assignment of edge weights w , that we assume to be non-negative real numbers and by a set of M communication requests $\{(S^\mu, R^\mu)\}_{\mu=1, \dots, M}$ between ordered pairs of nodes. We denote by π_μ , a path, i.e. a set of consecutive edges $e \in \mathcal{E}$, that connects a sender S^μ with the corresponding receiver R^μ . The optimization problem consists in finding M pairwise edge-disjoint paths π_μ while minimizing the total edge weight $\sum_\mu w(\pi_\mu)$, where $w(\pi_\mu) = \sum_{e \in \pi_\mu} w(e)$.

The classical EDP problem could be trivially recovered by assigning zero weight to all edges in $G(\mathcal{V}, \mathcal{E})$ and a positive cost to each communication that is not accommodated. Alternatively, any solution of the MWEDP problem can be reinterpreted as a solution of the classical EDP problem by slightly modifying the original instance of the graph $G(\mathcal{V}, \mathcal{E})$ by introducing an extra edge between each pair (S^μ, R^μ) with sufficiently large cost, such that the algorithm could still always find a solution possibly using these expensive extra edges. By construction, the cost of each of these M extra edges should be larger than the maximum possible weight a single path can take. Then the solution of the classical EDP problem is obtained from any solution of the MWEDP problem by discarding the paths passing through the extra edges. In the present paper, we keep information about path length minimization by assigning unit weights (i.e. $w_{ij} = 1, \forall (ij) \in \mathcal{E}$) to the original edges of the graph $G(\mathcal{V}, \mathcal{E})$ and a fixed cost $|\mathcal{E}| + 1$ to the extra edge added between each pair (S^μ, R^μ) .

We introduce M -dimensional variables $\bar{I}_{ij} = (I_{ij}^1, \dots, I_{ij}^M)$ with entries $I_{ij}^\mu \in \{\pm 1, 0\}$ representing the communication passing along an edge:

$$I_{ij}^\mu = \begin{cases} 1, & \text{if communication } \mu \text{ passes from } i \text{ to } j, \\ -1, & \text{if communication } \mu \text{ passes from } j \text{ to } i, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

We call these vectors *currents* as they must satisfy current conservation at each node i (Kirchhoff law):

$$\sum_{j \in \partial i} I_{ij}^\mu + \Lambda_i^\mu = 0, \quad \forall \mu = 1, \dots, M, \quad (2)$$

where we defined for each node i and each communication μ a variable Λ_i^μ such that

$$\Lambda_i^\mu = \begin{cases} 1 & \text{if } i = S^\mu, \\ -1 & \text{if } i = R^\mu, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The constraint of edge-disjointness specifies that for each edge (ij) , at most one of I_{ij}^μ is non-zero, therefore each vector \bar{I}_{ij} can be parametrized by a variable taking $2M + 1$ different values. Notice that the set of variables $\{\bar{I}_{ij}\}_{(ij) \in \mathcal{E}}$ completely specifies the state of the network. In this multi-flow formalism, the MWEDP problem is a combinatorial optimization problem in which the global cost function $C(\{\bar{I}_{ij}\}) = \sum_{(ij) \in \mathcal{E}} w_{ij} f(\|\bar{I}_{ij}\|)$ depends additively on the total net current $\|\bar{I}_{ij}\| = \sum_\mu |I_{ij}^\mu|$ along the edges, and the edge-disjointness is ensured by defining

$$f(\|\bar{I}_{ij}\|) = \begin{cases} 0, & \text{if } \|\bar{I}_{ij}\| = 0, \\ 1, & \text{if } \|\bar{I}_{ij}\| = 1, \\ +\infty, & \text{if } \|\bar{I}_{ij}\| > 1, \end{cases} \quad (4)$$

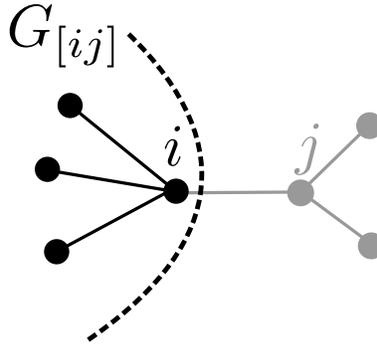


FIG. 2: The modified cavity graph $G_{[ij]}$.

where $|I_{ij}^\mu| = 0, 1$ denotes the absolute value of I_{ij}^μ . Thus configurations with more than one communication passing along an edge have infinite cost and, in the case of unit weights, the total cost $C(\{\bar{I}_{ij}\})$, if finite, represents exactly the total path length, i.e. the number of edges traversed by paths.

III. THE MESSAGE-PASSING ALGORITHM

On a tree, the optimization problem defined in Sec II can be solved exactly by iteration using the following message-passing algorithm. Let us assume that G is a tree and consider the subtree $G_{[ij]}$ defined by the connected component of i in $G \setminus (ij)$ (see Figure 2). We define $E_{ij}(\bar{I}_{ij})$ to be the minimum cost $C(\{\bar{I}_{ij}\})$ among current configurations that satisfy Kirchoff's laws on all vertices of $G_{[ij]}$ given that we fix an input (or output) extra current \bar{I}_{ij} entering (or exiting) node i . Because of the absence of cycles, it is possible to write a recursive equation for E_{ij} as a sum of cost contributions coming from neighbors of i in the subtree, plus the single cost contribution due to the current \bar{I}_{ij} passing along edge (ij) . We call these quantities $E_{ij}(\bar{I}_{ij})$ messages and they verify the *min-sum* recursion relation [39]:

$$E_{ij}(\bar{I}_{ij}) = \min_{\{\bar{I}_{ki}\}_{k \in \partial i \setminus j} \text{ constraint}} \left\{ \sum_{k \in \partial i \setminus j} E_{ki}(\bar{I}_{ki}) \right\} + f(\|\bar{I}_{ij}\|) \quad (5)$$

where *constraint* is the Kirchoff law at node i and ∂i denotes the neighborhood of i . This relation is exact for trees and can be considered as approximately correct for locally tree-like graphs, such as sparse random ones [37, 39], where correlations between neighbors of a given node decay exponentially. One can develop further this recursion to obtain a set of three types of message-passing equations, one for each type of node, i.e. for each value of Λ_i^μ . A fixed point of these equations can be found by iteration from arbitrary initial values for the messages until convergence. Then, one can collect at each edge the incoming and the outgoing converged messages to find the optimal configuration $\{\bar{I}_{ij}^*\}_{(ij) \in E}$ such that:

$$\bar{I}_{ij}^* = \arg \min_{\bar{I}} \left\{ E_{ij}(\bar{I}) + E_{ji}(-\bar{I}) - f(\|\bar{I}\|) \right\} \quad (6)$$

where the last term is subtracted to avoid double counting of the cost of the single edge (ij) .

A. The mapping into a weighted matching problem.

The min-sum algorithm as in (5) presents a computational bottleneck coming from the fact that for each output current \bar{I}_{ij} there is a large number of possible neighborhood's configurations $\{\bar{I}_{ki}\}_{k \in \partial i \setminus j}$ that are consistent both with the edge-disjoint constraints and with Kirchoff's law. In the calculation of the minimum in (5) one needs in fact to consider all possible combinations of paths entering and exiting node i ; the number of such combinations grows exponentially with the degree of node i . Nevertheless, the calculation can be performed efficiently by reducing it to a maximum weight matching problem [46] on an auxiliary weighted complete graph G'_i . The nodes of G'_i are the neighbors $k \in \partial i$ and the (symmetric) weights matrix Q will be defined as

$$Q_{kl} = - \min_{1 \leq |v| \leq M} \{ E_{ki}(v) + E_{li}(-v) \} + E_{ki}(0) + E_{li}(0) \quad (7)$$

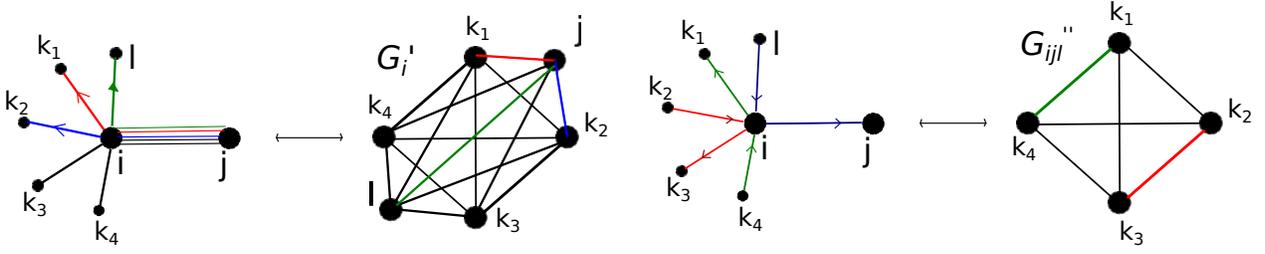


FIG. 3: Mapping into a weighted matching problem. Left: intermediate step where G'_i is built. On the leftmost part we show an example of several communications passing along (ij) and exiting along the remaining neighbors $k \in \partial i \setminus j$. Right: the final step where G''_{ijl} is built; the best configuration around node i when the blue current passes through (ij) is given by the minimum weighted matching on the complete auxiliary graph G''_{ijl} . Edges red and green represent the best matching, i.e. the configuration where two other communications enter/exit neighbors of $i \setminus j$.

where $E_{kl}(\nu) = E_{kl}(\bar{I}_{kl})$ with $I_{kl}^\mu \equiv \delta_{\mu,\nu}$ for $\nu > 0$, $I_{kl}^\mu \equiv -\delta_{\mu,\nu}$ for $\nu < 0$ and $I_{kl}^\mu \equiv 0$ for $\nu = 0$. Notice that this notation maps the M -dimensional vectors \bar{I}_{ij} to the $2M + 1$ possible current configurations ν allowed by the edge-disjointness constraint along a given edge. The computation of matrix Q , that requires $O(Mk^2)$ operations, should be performed only once at the beginning of the update routine for node $i \in G$.

Consider now a neighbor $j \in \partial i$ and a given μ passing through edge (ij) , we want to update $E_{ij}(\bar{I}_{ij})$. Assuming to know the other vertex $l \in \partial i \setminus j$ where the current μ entering (resp. exiting) node i can exit (resp. enter), then the least costly configuration in the remaining neighborhood is given by

$$q_{jl}^{\min} = -M_{jl} + \sum_{k \in \partial i \setminus \{j,l\}} E_{ki}(0) \quad (8)$$

where M_{jl} is the maximum weight of a matching on a complete graph G''_{ijl} with $k - 2$ nodes, built from G'_i by removing nodes j and l (and all their incident edges). Recall that a matching is a subset of edges of G''_{ijl} that do not share any vertex[46]. This is indeed equivalent to assigning to some of the remaining pairs of neighboring nodes currents $\nu \in [-M, \dots, M]$ that enters through one of them and exits through the other, such that the overall cost of the configuration is minimum. The key point is that the matching condition, i.e. the fact that edges in the solution set cannot have a vertex in common, in our problem translates in the condition of forbidding edge overlaps. Hence, thanks to this auxiliary mapping, we are able to reduce the computation of the update rule for the MP equations of the edge disjoint path problem to the solution of a standard (polynomial) combinatorial optimization problem, i.e. maximum weight matching. In Figure 3 we give a diagrammatic representation of the mapping. Note that in the maximum matching problem, edges in the input graph with negative weight can be simply removed. Notice that the neighboring current ν can also be a priori equal to μ in this algorithm, because the configurations where μ appears in more than one pair of edges will be eliminated in the minimization calculation as they have higher cost in our formulation. The minimum weight is thus independent of μ , i.e. of which message we are updating, a fact that allows reducing the complexity of the algorithm by a factor M .

Finally one needs to minimize over l given the matrix q^{\min} :

$$E_{ij}^{t+1}(\mu) = \min_{l \in \partial i \setminus j} \left\{ E_{il}^t(\mu) + q_{jl}^{\min} \right\} + c_{ij}(\mu) \quad (9)$$

where $c_{ij}(\mu)$ is the cost of edge (ij) , that in our case is 0 if $\mu = 0$ and 1 otherwise. We can notice that in order to evaluate each term inside the brackets we need to perform a matching optimization on each of the $(k - 2)$ -node complete graphs G''_{ijl} built $\forall l \in \partial i \setminus j$. Each of these matching routine has complexity $O(k^3 \log k)$ [47] and there are $O(k^2)$ possible combinations of j and l . Reminding that we first need to evaluate the weight matrix Q , the overall complexity of this algorithm will be:

$$O(k^5 \log k + Mk^2)$$

which is polynomial in the variables k and M . Once we have performed this whole procedure, we get all the information we need to calculate the $2M + 1$ update messages $E_{ij}^{t+1}(\mu)$, for each $j \in \partial i$, adding a term $O(kM)$ to the final complexity (which is nonetheless negligible compared to the previous two).

The case of $\mu = 0$, in which no current passes through edge (ij) regardless of what happens on the other edges, is addressed by calculating a matching on the $(k - 1)$ -node complete graph composed of all nodes $l \in \partial i \setminus j$. If i is either a sender or a receiver, i.e. $\Lambda_i^\mu \in \{\pm 1\}$ for a given $\mu \in [1, \dots, M]$, the same computation can be performed provided that an auxiliary node, indexed by the communication label μ is added to the original graph G and connected to node i , such that its exiting messages will be fixed once at the beginning in the following way and never updated: $E_{\mu i}(\nu) = -\infty$ if $0 < \nu = \mu$ (sender) or $0 < -\nu = \mu$ (receiver), and $E_{\mu i}(\nu) = +\infty$ otherwise.

B. The role of reinforcement.

In order to aid and speed-up convergence of the MP equations, we used a reinforcement technique [48, 49], in which a set of external local fields $h_{ij}^t(\mu) = E_{ij}^t(\mu) + E_{ji}^t(-\mu) - c_{ij}(\mu)$ act on the messages gradually biasing them to align with themselves. The reinforcement is introduced by promoting edge costs to become communication-dependent quantities defined as linear combinations of the cost at the previous time-step and the reinforcement local fields:

$$c_{ij}^{t+1}(\mu) = c_{ij}^t(\mu) + \gamma_t h_{ij}^t(\mu) \quad (10)$$

with $c_{ij}^0(\mu) = c_{ij}$. This cost will then be inserted into equation (9) to replace the term $c_{ij}(\mu)$. This has the effect to lead the messages to converge faster, gradually bootstrapping the system into a simpler one with large external fields. In practice we choose $\gamma_t = t\rho$ and one has to choose the growth rate of γ by tuning the reinforcement parameter ρ , that controls the trade-off between having a faster convergence and reaching a better solution. We tested ρ on instances on three types of graphs to finally choose to fix it to $\rho = 0.002$ in the rest of the simulations. In Figure 4 we could notice that this value achieves comparable results (inset) in terms of M_{acc}/M to lower ρ in less time.

In Figure 5(left) we report the number of converged instances (over 100 realizations) for standard MP (without reinforcement) on four types of random graphs (as described in the next section) and fixed size $V = 1000$ and average degree $\langle k \rangle = 3$. The convergence failure of the standard MP increases considerably with M/V until it reaches a peak value, then it decreases. On the contrary, when reinforcement is used, convergence is always achieved in less than 100 steps (right panel).

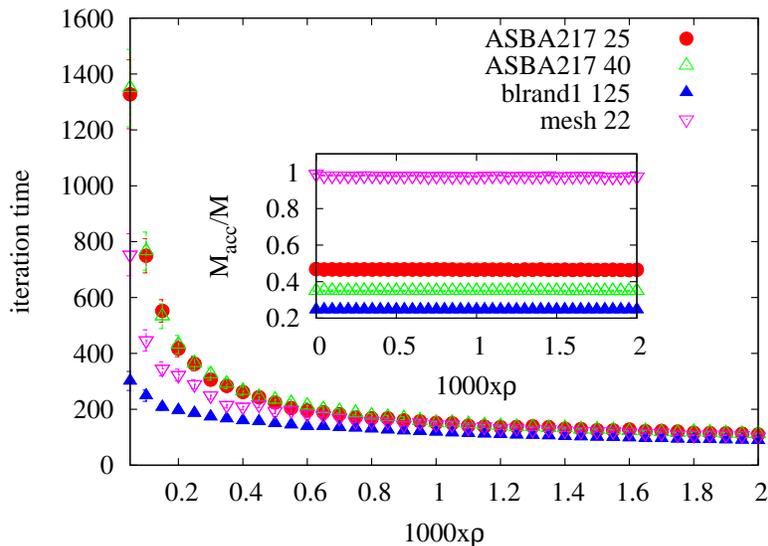


FIG. 4: Reinforcement performance. Number of iterations to reach convergence as a function of reinforcement parameter ρ on BRITE graphs AS-BA217 ($V=100$) with $M = 25, 40$, brand1 ($V=500$) with $M = 125$ and mesh 15×15 with $M = 22$. Inset: the number of accommodated paths M_{acc} is substantially unchanged in the range of parameter values under study.

IV. RESULTS ON RANDOM GRAPHS

First, we tested the MP algorithm on various types of random graphs, with fixed size $V = |\mathcal{V}| = 1000$ and average degree $\langle k \rangle = 3, 5, 7$: regular random graphs (Reg), Erdős-Rényi random graphs (ER) [50], random graphs with power-law distribution (SF) [51] and a set of graphs (RER) obtained adding edges independently with probability p starting from a k_0 -regular random graph (for large V , the final average degree of such graphs is $\langle k \rangle = k_0 + d$, with $d = pV$). We compared the performance with a multi-start greedy algorithm (MSG) [36]. This heuristic algorithm calculates paths by iteratively choosing a (random) communication μ , finding the corresponding shortest path and removing the edges belonging to the path from the graph. The process is repeated until either there are no paths left to be routed or no communications can be accommodated anymore in the graph. The multi-start version repeats the same procedure a given number of times and keeps the best solution in terms of M_{acc} , the number of accommodated paths. A bounded-length version [52] of MSG has been used to develop an iterative algorithm to solve the RWA using EDP in [18]: its performance was comparable to the one obtained using a linear programming

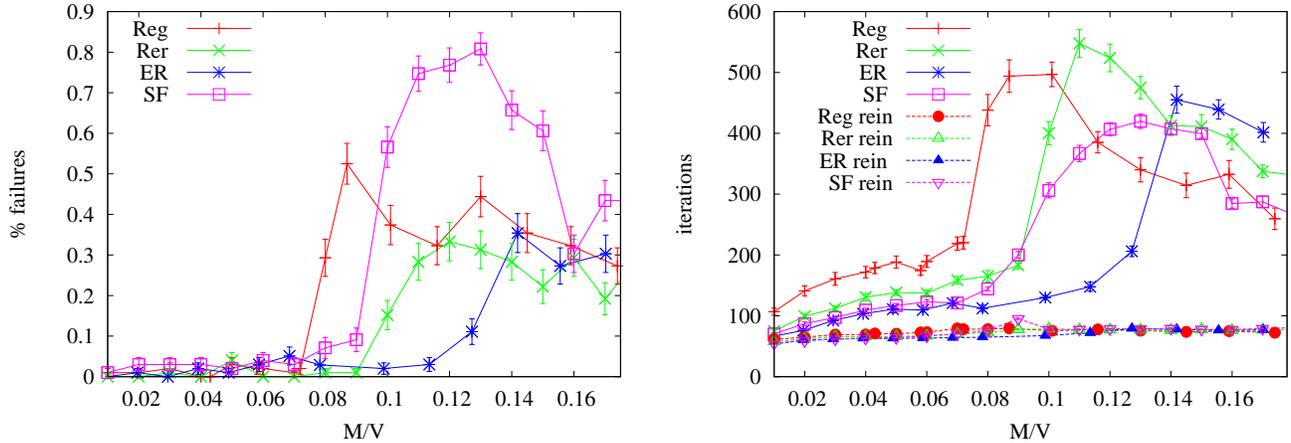


FIG. 5: Left: Fraction of instances in which convergence standard MP fails (reinforced MP always converged in our experiments). Right: number of iterations for convergence for standard MP and reinforced MP ($\rho = 0.002$) in case of random graphs of $V = 1000$ and $\langle k \rangle = 3$ as a function of M/V . Notice how the reinforcement term, besides ensuring convergence, greatly improves the convergence time.

solver on graphs of small sizes ($V \leq 40$) but with faster execution times. This makes it suited to be tested on larger graphs. A disadvantage of the greedy method is that it relies heavily on the order in which communications are accommodated (it disregards the information about sender-receiver pairs other than the ones already accommodated). The difference in the performances of the message-passing and greedy algorithm could then be used to assess the relevance of local information usage in such optimization problem. We tested both the standard multi-start and the bounded-length version but we found equal results with the first being slightly faster, in our tests, in terms of execution times. Thus we decided to use the standard MSG in our simulations. First we compared the results in terms of number of accommodated paths M_{acc} by calculating the ratio M_{acc}/M . In Figure 6 we show the behavior of M_{acc}/M for each type of random graph and $V = 10^3$, $\langle k \rangle = 3$ using MP, reinforced MP and MSG. Both MP versions perform better than MSG, with the standard MP giving better results. The corresponding results for $\langle k \rangle = 5$ are similar (not reported) but the value $M_{acc}/M < 1$ is reached at higher values of M/V and standard MP and MP with reinforcement give almost always the same solutions. The case $\langle k \rangle = 7$ is not reported because, given the high number of edges, the solutions are often trivial (i.e. $M_{acc}/M = 1$), a part from the case of SF graphs where we have instead $M_{acc}/M < 1$ due to the presence of many small degree nodes. We also studied the total path length as a function of M/V for the solutions, obtained with the different algorithms. We consider the ratio between the total path lengths obtained with greedy and MP for solutions in which the number M_{acc} of accommodated path is the same. In Figure 7 we can see that MP always outperforms the MSG algorithm for all types of graph under study. The results for the SF graph with $\langle k \rangle = 7$ are quite different from the other graphs: both for MP and MSG the ratio departs from 1 at rather small values of M/V , possibly because the maximum number of accommodated paths is limited by the existence of many small degree nodes that act as bottlenecks, preventing the use of many alternative edge-disjoint routes. The scaling behavior of the fraction $1 - M_{acc}/M$ of unaccommodated communications and the average total path length L/V of accommodated paths with the system size in the solutions obtained using the MP algorithm is shown in Figure 8 for regular random graphs and ER random graphs. These quantities are plot as functions of the scaling variable $x = \frac{M \log V}{V}$. Note that when paths do not interact, x is a measure the total path length per site, as the average path length is proportional to $\log V$. In the top panels, two regimes are visible: for small x , all communications can be accommodated, whereas at some value x^* the curves for different values of V depart from zero. This behavior can be interpreted as a SAT/UNSAT transition, in analogy with the terminology of constraint-satisfaction problems [41]. The collapse of the curves L/V for different values of V is very good in the region in which all paths can be accommodated. On the contrary, in the UNSAT region, the curves for different sizes do not collapse anymore, though the relative difference between them seems to decrease by increasing the system size, and the curves for the largest graphs analyzed ($V = 8000, 10000$) are almost superimposed. We argue that x is the correct scaling variable in the limit of infinitely large graphs, and the observed mismatch could be due to finite-size effects. The change of slope in the roughly linear behavior of the average total length L/V is motivated by the fact that in the SAT region, all communications can be accommodated at the cost of taking longer paths with respect to those actually accommodated in the UNSAT region.

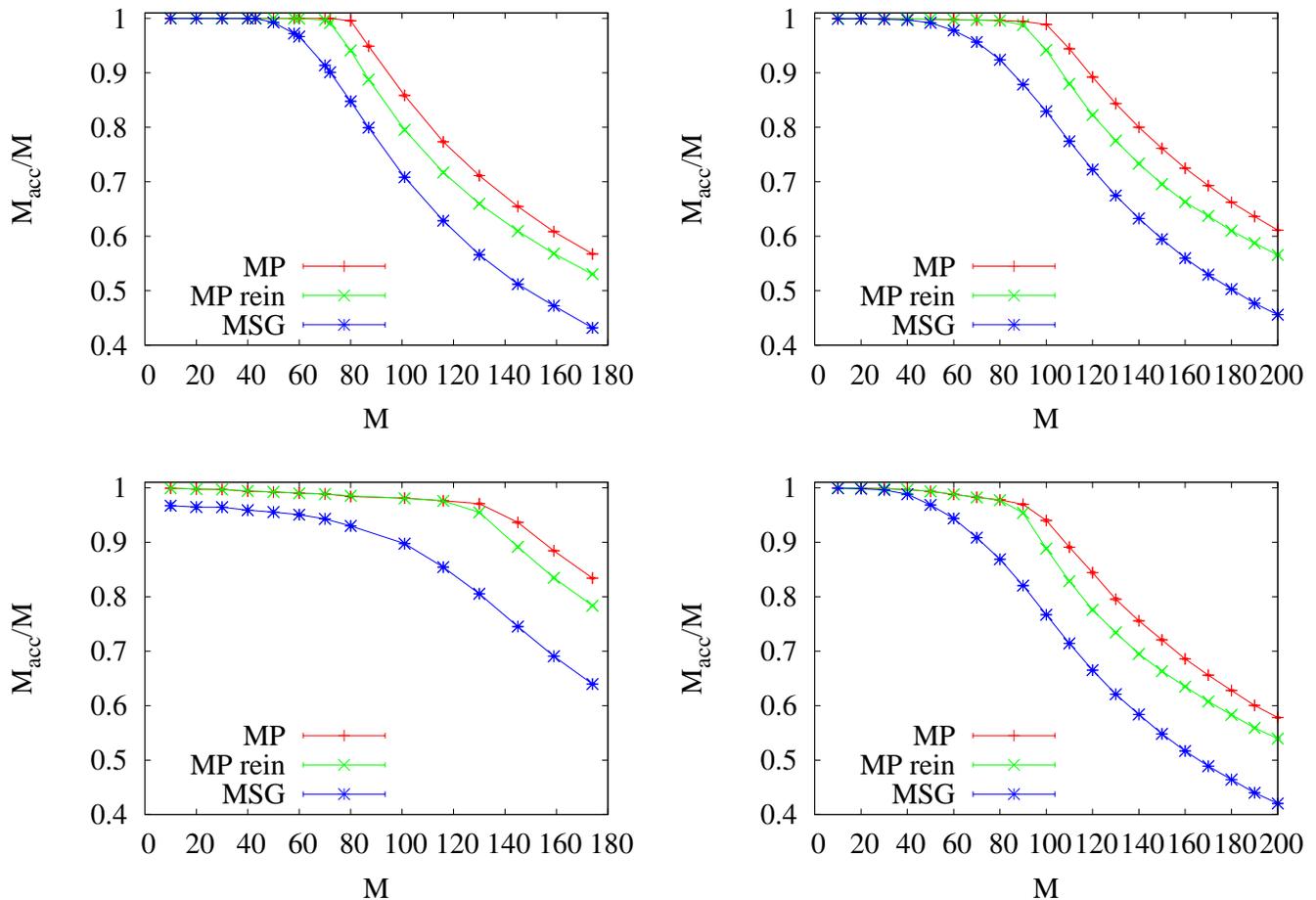


FIG. 6: MP vs greedy performance. We plot the performance in terms of M_{acc}/M for (from top to bottom) regular, RER, ER and SF graphs of fixed size $V = 10^3$ and average degree $\langle k \rangle = 3$. Error bars are smaller than the size of the symbols.

V. COMPARISON WITH OTHER METHODS

A comparison between the performances of the MP algorithm and those of alternative algorithms proposed in the literature [31, 36] is reported in Table I. As benchmark instances we used: two internet-like topologies generated using the BRITe graph generator [53] with parameters set as in [36]; mesh graphs of sizes 15×15 and 25×25 , Steiner and planar graphs as reported in [31]. For each of these graphs we used the same set of sender-receiver pairs of size $M = 0.10V, 0.25V, 0.40V$ used in [31]. For each of these instances we ran the MP, MP with reinforcement and MSG algorithms 20 times and collected the average, minimum and maximum number of accommodated paths M_{acc} along with the average computational time in seconds. All results are reported in Table I.

A. Other optimization methods.

A part from the multi-start greedy, we used as comparison two more structured algorithms. The first one is an Ant Colony Optimization metaheuristic [36]. This method builds an EDP solution incrementally from partial solutions provided by a set of M ants. Each ant generates a path for a given communication making probabilistic decisions during the construction steps. These are made by processing local information modeled as *pheromone* information provided by other ants. The advantage of this method is to divide the EDP in subproblems and to use local information. The drawback is that it relies on several parameters that need to be carefully tuned in order to have a sensitive solution. Moreover the computational time increases considerably with the system size. The second algorithm is a Montecarlo-based Local Search [31], that uses as main Montecarlo step a path rewiring based on rooted spanning trees. Unfortunately the running time grows rapidly with the system size, making

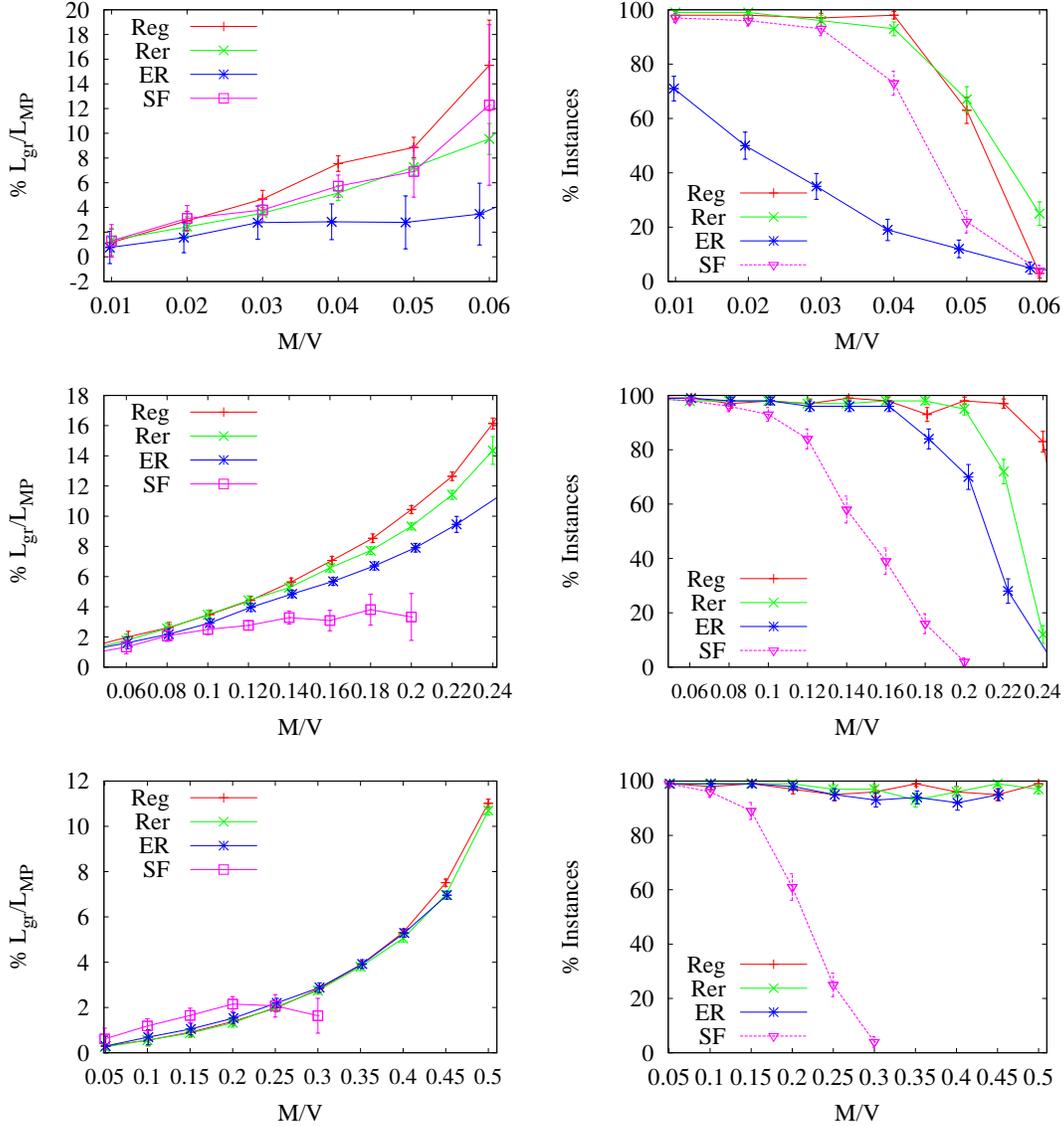


FIG. 7: Length performance. We plot (left) the relative performance of MSG over MP in terms of total length of the solution paths: $y = 100(L_g/L_{MP} - 1)$. Here L_g and L_{MP} denote the total path lengths calculated with MSG and MP respectively. We use Reg, RER, ER and SF graphs of fixed size $V = 10^3$ and average degree $\langle k \rangle = 3, 5, 7$ (from top to bottom). On the right we report the number of instances where the two algorithms find the same solution in term of M_{acc}/M over 100 realizations.

it computationally expensive when used on large graphs. Results are reported in Table I. Finally, we performed simulation using the multi-start greedy heuristic described above.

B. Results.

In Table I we report the performance comparison in terms of M_{acc} between the two versions of MP (with and without reinforcement) and the other 3 types of algorithms. The message-passing always performs equal or better than the other methods. Surprisingly the best performances are given for meshes and planar graphs, where we would expect the failure of MP due to the existence of short loops. What we find instead is that, even though the standard MP converges in few of these instances on meshes, the version with reinforcement always finds a solution that is always better than the other algorithms. The larger performance gap is seen on larger set of commodities and bigger graphs. Performance improvement reaches 27% with respect to LS, the best one between the other algorithms tested. The same considerations can be made in the case of planar graphs. We

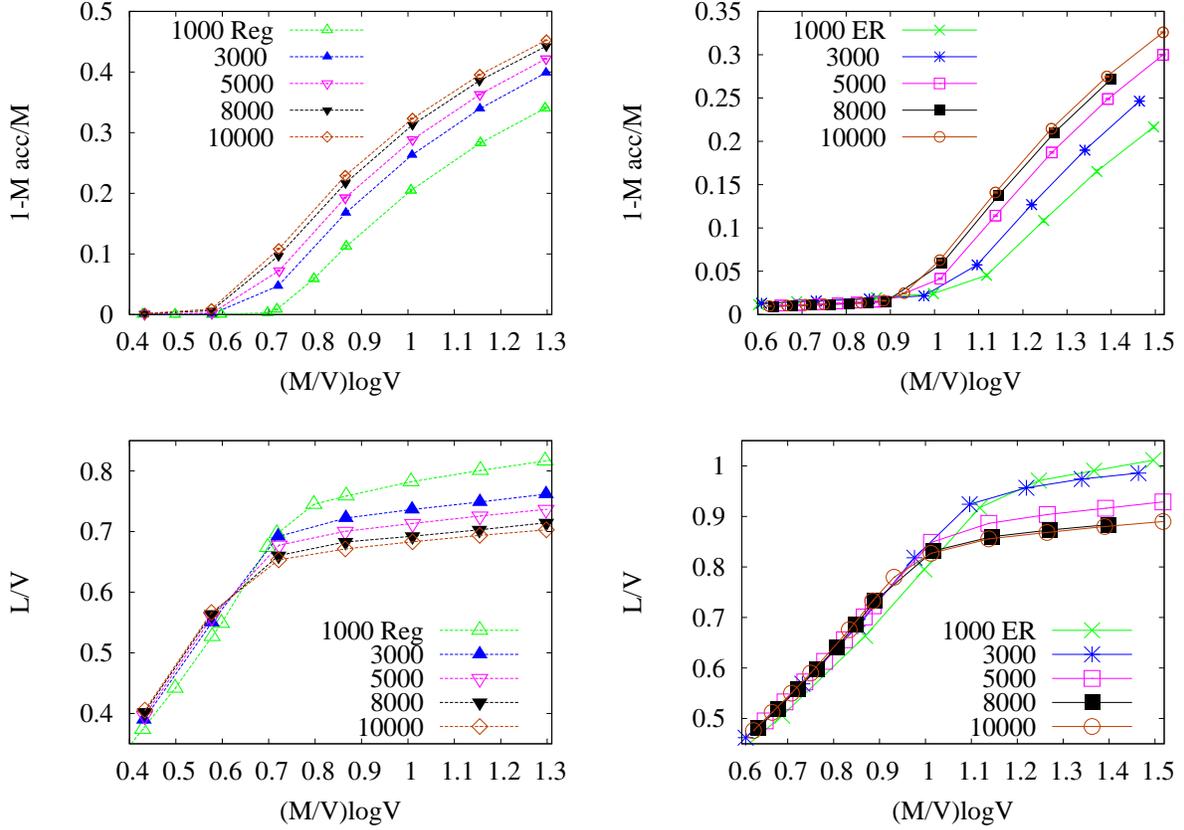


FIG. 8: Finite-size effects. We plot $1 - M_{acc}/M$ (top) and the total length per node L/V (bottom) for Reg (left) and ER (right) graphs as a function of the scaling variable $\frac{M \log V}{V}$. We can notice the finite-size effects decreasing with system size leading to the curves corresponding to the biggest graphs $V = 8000, 10000$ to almost superimpose. Note that in the SAT phase the total length grows linearly in $\log V$ for all system sizes as expected but in the UNSAT phase the graphs split. Error bars are smaller than point size.

claim that this gap would increase with system size, but unfortunately the size of benchmark graphs remains limited to $V \leq 500$. Moreover these alternative algorithms do not consider path length optimization, thus we cannot compare the performance with respect to this variable. The ACO has been recently tested on several types of graphs (still with $V \leq 500$) against a Genetic Algorithm (GA) in [54]. It performed better than GA in the case of BRITE graphs 1-6 and 14% worse in the case of 10×10 and 15×15 mesh graphs. The MP algorithm always outperforms ACO and in the case of 15×15 mesh the gap reaches 23.5%. Unfortunately neither the GA has been tested on larger graphs nor gives results in terms of path length of the solutions.

VI. CONCLUSIONS

The EDP problem is a combinatorial optimization problem that finds applications in several traffic engineering problems, from VLSI design to routing and access control management in communication networks. In this work we proposed a min-sum message-passing algorithm to find the maximum number of communications M_{acc} that can be accommodated in a network subject to edge-disjoint constraints and minimizing total path length at the same time. We devised an efficient method to implement these equations by exploiting a mapping into a minimum weight matching problem on an auxiliary graph. The standard MP algorithm and the version with reinforcement consistently outperform alternative algorithms found in the literature on different types of benchmark graphs in terms of the fraction M_{acc}/M of accommodated communications. We found two different behaviors: on some “easy” instances, all algorithms accommodate all requests, providing the same results and suggesting that these could be the optimal ones; there are non-trivial instances in which $M_{acc}/M < 1$, but the message-passing algorithm always outperforms the other algorithms in terms of the number of accommodated paths. In particular we obtained better results in the case of meshes and planar graphs, even though these topologies are not locally tree-like as required by the cavity method. In these cases, we could always ensure convergence of the MP equations by exploiting a reinforcement technique. The quality

of solutions improves with decreasing the reinforcement parameter, such that we could always find better solutions than those obtained using the other algorithms under study. Unfortunately, for the heuristic algorithms employed on the benchmarks we could not access other relevant metrics such as the average total path length, as it was not considered before in the literature [31, 36]. Nonetheless we could directly compute such quantity for a multi-start greedy heuristic in several graphs, finding that MP always gives a lower average path length for solutions with the same fraction of accommodated communications.

In conclusion, combining the good performance results, in terms of traffic and path length, with the polynomial time implementation, the use of the MP algorithm opens new perspectives in the solution of relevant routing problems over communication networks such as the RWA in optical networks. In particular, it would be interesting to apply the MP algorithm in the iterative construction of RWA solutions over communication networks with finite link capacity, as it has been done for other types of EDP algorithms.

ACKNOWLEDGEMENT

We thank P. Quang Dung for kindly providing us the detailed benchmark instances. This work was supported by the Marie Curie Training Network NETADIS (FP7, grant 290038). L.D. also acknowledges the Italian FIRB Project No. RBFR10QUW4.

REFERENCES

-
- [1] Tanenbaum A S. *Computer Networks 4th Edition*. Prentice Hall, 2003.
 - [2] Gerald R Ash. *Traffic Engineering and QoS Optimization of Integrated Voice & Data Networks*. Morgan Kaufmann, 2006.
 - [3] X et al Masip-Bruin. Research challenges in qos routing. *Computer communications*, 29(5):563–581, 2006.
 - [4] Neil Robertson and Paul D Seymour. Graph minors. xiii. the disjoint paths problem. *Journal of combinatorial theory, Series B*, 63(1):65–110, 1995.
 - [5] Sabih H Gerez. *Algorithms for VLSI design automation*, volume 8. Wiley Chichester, England, 1999.
 - [6] Wun-Tat Chan, Francis YL Chin, and Hing-Fung Ting. Escaping a grid by edge-disjoint paths. *Algorithmica*, 36(4):343–359, 2003.
 - [7] Baruch Awerbuch, Rainer Gawlick, Tom Leighton, and Yuval Rabani. On-line admission control and circuit routing for high performance computing and communication. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 412–423. IEEE, 1994.
 - [8] Mario Gerla and Jack Tzu-Chieh Tsai. Multicenter, mobile, multimedia radio network. *Wireless networks*, 1(3):255–265, 1995.
 - [9] Akkaya K and Younis M. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349, 2005.
 - [10] Kang G Shin and Parameswaran Ramanathan. Real-time computing: A new discipline of computer science and engineering. *Proceedings of the IEEE*, 82(1):6–24, 1994.
 - [11] Abinash Mahapatra, Kumar Anand, and Dharma P Agrawal. Qos and energy aware routing for real-time traffic in wireless sensor networks. *Computer Communications*, 29(4):437–445, 2006.
 - [12] Hussein F Salama, Douglas S Reeves, and Yannis Viniotis. Evaluation of multicast routing algorithms for real-time communication on high-speed networks. *Selected Areas in Communications, IEEE Journal on*, 15(3):332–345, 1997.
 - [13] Sumpter Z, Burson L, Tang B, and Chen X. Maximizing number of satisfiable routing requests in static ad hoc networks. *IEEE GLOBECOM 2013 Conference Proceedings*, 2013.
 - [14] Srinivas A and Modiano E. Minimum energy disjoint path routing in wireless ad-hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 122–133. ACM, 2003.
 - [15] Jain S and Das S R. Exploiting path diversity in the link layer in wireless ad hoc networks. *Ad Hoc Networks*, 6(5):805–825, 2008.
 - [16] X Li and L Cuthbert. Node-disjointness-based multipath routing for mobile ad hoc networks. In *Proceedings of the 1st ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, PE-WASUN '04, pages 23–29. ACM, 2004.
 - [17] Chatterjee B C, Sarma N, Sahu P P, et al. Review and performance analysis on routing and wavelength assignment approaches for optical networks. *IETE Technical Review*, 30(1):12, 2013.
 - [18] Manohar P, Manjunath D, and Shevgaonkar RK. Routing and wavelength assignment in optical networks from edge disjoint path algorithms. *Comm. Lett., IEEE*, 6(5):211–213, 2002.
 - [19] Chia-Chun Hsu, Hsun-Jung Cho, and Shu-Cherng Fang. Routing and wavelength assignment in optical networks from maximum edge-disjoint paths. In *Genetic and Evolutionary Computing*, pages 95–103. Springer, 2014.
 - [20] R M Karp. *Reducibility among combinatorial problems*. Springer, 1972.
 - [21] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Series of books in the mathematical sciences. W. H. Freeman, 1979.
 - [22] Chandra Chekuri and Sanjeev Khanna. Edge disjoint paths revisited. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 628–637. Society for Industrial and Applied Mathematics, 2003.

- [23] Thomas Erlebach. Approximation algorithms for edge-disjoint paths and unsplittable flow. In *Efficient Approximation and Online Algorithms*, pages 97–134. Springer, 2006.
- [24] Chen C and Banerjee S. A new model for optimal routing and wavelength assignment in wavelength division multiplexed optical networks. In *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proc. IEEE*, volume 1, pages 164–171. IEEE, 1996.
- [25] Skorin-Kapov N. Routing and wavelength assignment in optical networks using bin packing based algorithms. *Eur. J. Op. Res.*, 177(2):1167–1179, 2007.
- [26] Banerjee D and Mukherjee B. A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. *Sel. Ar. in Comm., IEEE*, 14(5):903–908, 1996.
- [27] Stavros G Kolliopoulos and Clifford Stein. Approximating disjoint-path problems using packing integer programs. *Mathematical Programming*, 99(1):63–87, 2004.
- [28] Asuman E Ozdaglar and Dimitri P Bertsekas. Routing and wavelength assignment in optical networks. *IEEE/ACM Transactions on Networking (TON)*, 11(2):259–272, 2003.
- [29] Alok Baveja and Aravind Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *Mathematics of Operations Research*, 25(2):255–280, 2000.
- [30] Belgacem L, Charon I, and Hudry O. A post-optimization method for the routing and wavelength assignment problem applied to scheduled lightpath demands. *Eur. J. Op. Res.*, 232(2):298–306, 2014.
- [31] Quang Dung Pham, Yves Deville, and Pascal Van Hentenryck. Ls (graph): a constraint-based local search for constraint optimization on trees and paths. *Constraints*, 17(4):357–408, 2012.
- [32] Thiago F Noronha, Mauricio GC Resende, and Celso C Ribeiro. A biased random-key genetic algorithm for routing and wavelength assignment. *Journal of Global Optimization*, 50(3):503–518, 2011.
- [33] Mitsuo Gen, Runwei Cheng, and Lin Lin. *Network models and optimization: Multiobjective genetic algorithm approach*. Springer, 2008.
- [34] Storn R and Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. glob. opt.*, 11(4):341–359, 1997.
- [35] Ali Hassan, Chris Phillips, and Jonathan Pitts. Dynamic routing and wavelength assignment using hybrid particle swarm optimization for wdm networks. In *EPSRC PostGraduate Network Symposium (PGNet)*, 2007.
- [36] Blesa M and Blum C. Ant colony optimization for the maximum edge-disjoint paths problem. In *App. Ev. Comp.*, pages 160–169. Springer, 2004.
- [37] Mézard M and Parisi G. The cavity method at zero temperature. *J. Stat. Phys.*, 111(1-2):1–34, 2003.
- [38] Mézard M, Parisi G, and Virasoro M A. *Spin glass theory and beyond*, volume 9. World scientific Singapore, 1987.
- [39] Mézard M and Montanari A. *Information, physics, and computation*. Oxford University Press, 2009.
- [40] Mézard M and Parisi G. The bethe lattice spin glass revisited. *Eur. J. Phys B*, 20(2):217–233, 2001.
- [41] Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.
- [42] Yeung C H and Saad D. Competition for shortest paths on sparse graphs. *Phys. Rev. Lett.*, 108(20):208701, 2012.
- [43] Yeung C H, Saad D, and Wong KY M. From the physics of interacting polymers to optimizing routes on the london underground. *Proc. Nat. Ac. Sci.*, 110(34):13717–13722, 2013.
- [44] M Bayati, C Borgs, Alfredo Braunstein, J Chayes, Abolfazl Ramezanzpour, and Riccardo Zecchina. Statistical mechanics of steiner trees. *Physical review letters*, 101(3):037208, 2008.
- [45] C. De Bacco, S. Franz, D. Saad, and C. H. Yeung. Shortest node-disjoint paths on random graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(7):P07009, July 2014.
- [46] L. Lovász and D. Plummer. *Matching Theory*. AMS Chelsea Publishing Series. American Mathematical Soc., 2009.
- [47] Zvi Galil, Silvio Micali, and Harold Gabow. An $o(ev \log v)$ algorithm for finding a maximal weighted matching in general graphs. *SIAM Journal on Computing*, 15(1):120–130, 1986.
- [48] Alfredo Braunstein and Riccardo Zecchina. Learning by message passing in networks of discrete synapses. *Physical review letters*, 96(3):030201, 2006.
- [49] Fabrizio Altarelli, Alfredo Braunstein, John Realpe-Gomez, and Riccardo Zecchina. Statistical mechanics of budget-constrained auctions. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(07):P07002, 2009.
- [50] Erdős P and Rényi A. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–ñ61, 1960.
- [51] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, 2001.
- [52] Jon Michael Kleinberg. *Approximation algorithms for disjoint paths problems*. PhD thesis, Citeseer, 1996.
- [53] Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers. Brite: An approach to universal topology generation. In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*, pages 346–353. IEEE, 2001. Available at <http://www.cs.bu.edu/brite/>.
- [54] Chia-Chun Hsu and Hsun-Jung Cho. A genetic algorithm for the maximum edge-disjoint paths problem. *Neurocomputing*, 2014.

Appendix A: Convergence criterion

Given a decision variable d^t to be calculated at each iteration update step t , an integer variable n and a time step T_{max} we have convergence if, for n consecutive iteration steps, d^t does not change, and we fix a maximum iteration time T_{max} to update MP

equations. Formally this writes:

$$\exists t_0 \in [1, T_{max} - n] \quad s.t. \quad d^{t_0+i} = d^{t_0} \quad \forall i = 1, \dots, n \quad (A1)$$

In our simulations we defined the decision variable as the total difference of the optimal currents (calculated edge by edge) between two consecutive iteration steps :

$$d^t = \sum_{(ij) \in E} [1 - \delta_{\mu_{ij}^t, \mu_{ij}^{t-1}}] \quad (A2)$$

where $\mu_{ij}^t = |\min_{\mu=-M, \dots, M} \{E_{ij}(\mu) + E_{ji}(-\mu) - c_{ij}(\mu)\}|$ and convergence is reached when $d^t = 0$ for n consecutive time steps.

Appendix B: Benchmark results

Instance Name	V	E	(k)	MP			MP rein = 0.002			MSG (greedy)			ACO			LS			MP gain vs.		
				$\langle M \rangle$	M_{min}	M_{max}	MSG	ACO	LS												
blrand10_1	500	1020	4.08	16.00	16	16	16.00	16	16	13.65	13	15	14.80	14	16	16.00	16	16	6.67	0.00	0.00
blrand25_1	500	1020	4.08	32.00	32	32	32.00	32	32	27.75	26	30	31.85	31	32	32.00	32	32	6.67	0.00	0.00
blrand40_1	500	1020	4.08	38.00	38	38	38.00	38	38	33.10	32	35	37.85	37	38	37.90	37	38	8.57	0.00	0.00
blrand10_2	500	1020	4.08	26.00	26	26	25.65	25	26	23.85	23	25	25.25	25	26	26.00	26	26	4.00	0.00	0.00
blrand25_2	500	1020	4.08	35.00	35	35	35.00	35	35	30.75	29	33	34.75	34	35	34.95	34	35	6.06	0.00	0.00
blrand40_2	500	1020	4.08	37.00	37	37	37.00	37	37	32.45	31	34	36.95	36	37	36.95	36	37	8.82	0.00	0.00
blsdeg10_1	500	1020	4.08	17.00	17	17	16.89	15	17	14.65	14	16	15.95	15	16	17.00	17	17	6.25	6.25	0.00
blsdeg25_1	500	1020	4.08	36.00	36	36	36.00	36	36	31.55	30	33	35.80	35	36	36.00	36	36	9.09	0.00	0.00
blsdeg40_1	500	1020	4.08	34.00	34	34	34.00	34	34	29.00	28	31	33.65	33	34	34.00	34	34	9.68	0.00	0.00
blsdeg10_2	500	1020	4.08	20.00	20	20	19.85	19	20	16.90	16	18	19.20	19	20	20.00	20	20	11.11	0.00	0.00
blsdeg25_2	500	1020	4.08	34.00	34	34	34.00	34	34	28.45	27	30	32.95	32	34	33.90	33	34	13.33	0.00	0.00
blsdeg40_2	500	1020	4.08	37.00	37	37	37.00	37	37	31.75	30	33	36.50	35	37	37.00	37	37	12.12	0.00	0.00
mesh15_10_1	225	420	3.73	22.00	22	22	22.00	22	22	20.60	20	22	19.65	19	21	21.55	21	22	0.00	4.76	0.00
mesh15_25_1	225	420	3.73	36.00	36	36	35.10	35	36	28.30	27	30	27.70	26	29	32.00	31	33	20.00	24.14	9.09
mesh15_40_1	225	420	3.73	43.00	43	43	42.50	42	43	30.10	28	32	35.30	32	38	38.80	37	40	34.38	13.16	7.50
mesh15_10_2	225	420	3.73	-	-	-	19.89	19	20	19.75	19	20	17.50	17	19	19.45	19	20	0.00	5.26	0.00
mesh15_25_2	225	420	3.73	35.00	35	35	34.70	33	35	29.25	29	30	29.20	28	31	33.05	32	34	16.67	12.90	2.94
mesh15_40_2	225	420	3.73	42.00	42	42	41.35	41	42	29.80	29	32	34.00	33	36	37.60	36	39	31.25	16.67	7.69
mesh25_10_1	625	1200	3.84	-	-	-	47.25	46	48	40.70	40	42	32.85	29	36	41.00	39	43	14.29	33.33	11.63
mesh25_25_1	625	1200	3.84	-	-	-	68.30	67	69	48.40	47	51	45.00	42	49	55.55	54	59	35.29	40.82	16.95
mesh25_40_1	625	1200	3.84	-	-	-	88.74	88	90	54.35	53	58	57.70	53	61	69.30	67	72	55.17	47.54	25.00
mesh25_10_2	625	1200	3.84	-	-	-	44.33	43	46	40.05	38	42	30.10	28	33	37.90	36	40	9.52	39.39	15.00
mesh25_25_2	625	1200	3.84	-	-	-	67.22	65	70	48.90	47	52	45.60	44	48	54.70	52	59	34.62	45.83	18.64
mesh25_40_2	625	1200	3.84	-	-	-	88.55	87	90	54.05	51	57	57.75	54	61	68.85	66	71	57.89	47.54	26.76
steinb4_10	50	100	4.00	5.00	5	5	0.00	0.00	0.00												
steinb4_25	50	100	4.00	12.00	12	12	0.00	0.00	0.00												
steinb4_40	50	100	4.00	20.00	20	20	19.90	19	20	0.00	0.00	0.00									
steinb10_10	75	150	4.00	7.00	7	7	0.00	0.00	0.00												
steinb10_25	75	150	4.00	18.00	18	18	18.00	18	18	18.00	18	18	17.85	17	18	18.00	18	18	0.00	0.00	0.00
steinb10_40	75	150	4.00	28.00	28	28	27.65	27	29	25.10	24	27	24.35	23	26	27.30	27	28	7.41	11.54	3.57
steinb16_10	100	200	4.00	10.00	10	10	0.00	0.00	0.00												
steinb16_25	100	200	4.00	25.00	25	25	25.00	25	25	25.00	25	25	24.35	24	25	25.00	25	25	0.00	0.00	0.00
steinb16_40	100	200	4.00	36.12	36	37	36.00	36	36	33.20	32	34	32.45	32	34	35.95	35	37	8.82	8.82	0.00
steinc6_10	500	1000	4.00	50.00	50	50	50.00	50	50	50.00	50	50	49.10	47	50	50.00	50	50	0.00	0.00	0.00
steinc6_25	500	1000	4.00	125.00	125	125	122.55	121	124	107.50	106	110	89.90	85	94	104.95	102	108	13.64	32.98	15.74
stienc6_40	500	1000	4.00	145.84	144	147	140.40	139	142	114.10	112	117	109.80	106	117	121.40	119	125	25.64	25.64	17.60
steinc11_10	500	2500	10.00	50.00	50	50	0.00	0.00	0.00												
steinc11_25	500	2500	10.00	125.00	125	125	125.00	125	125	125.00	125	125	123.30	122	125	125.00	125	125	0.00	0.00	0.00
steinc11_40	500	2500	10.00	200.00	200	200	200.00	200	200	200.00	200	200	194.25	190	198	200.00	200	200	0.00	1.01	0.00
steinc16_10	500	12500	50.00	50.00	50	50	0.00	0.00	0.00												
steinc16_25	500	12500	50.00	-	-	-	125	125	125	125.00	125	125	125.00	125	125	125.00	125	125	0.00	0.00	0.00
steinc16_40	500	12500	50.00	-	-	-	200	200	200	200.00	200	200	200.00	200	200	200.00	200	200	0.00	0.00	0.00
plan50_10	50	135	5.40	5.00	5	5	0.00	0.00	0.00												
plan50_25	50	135	5.40	12.00	12	12	0.00	0.00	0.00												
plan50_40	50	135	5.40	20.00	20	20	19.90	19	20	0.00	0.00	0.00									
plan100_10	100	285	5.70	10.00	10	10	0.00	0.00	0.00												
plan100_25	100	285	5.70	25.00	25	25	0.00	0.00	0.00												
plan100_40	100	285	5.70	37.00	37	37	37.05	37	38	35.80	35	37	34.00	33	36	36.00	35	37	2.70	5.56	2.70
plan200_10	200	583	5.83	20.00	20	20	0.00	0.00	0.00												
pan200_25	200	583	5.83	-	-	-	48.95	48	50	46.50	46	48	41.80	39	43	45.95	45	48	4.17	16.28	4.17
plan200_40	200	583	5.83	-	-	-	60.65	58	62	52.95	52	56	49.35	47	51	55.70	54	58	10.71	21.57	6.90
plan500_10	500	1477	5.91	50.00	50	50	50.00	50	50	50.00	50	50	44.95	42	47	50.00	50	50	0.00	6.38	0.00
plan500_25	500	1477	5.91	-	-	-	92.29	90	94	78.15	76	80	60.95	57	65	78.20	77	80	17.50	44.62	17.50
plan500_40	500	1477	5.91	-	-	-	122.31	119	124	92.60	90	95	82.85	78	86	100.15	97	102	30.53	44.19	21.57

TABLE I: Message-passing and multi-start greedy performances. Columns 1-4 give the characteristics of the benchmark. For each algorithm, columns 1-3 represent the average, the minimum and the max number of accommodated paths over 20 runs of a given set of commodity instance respectively. ACO and LS performances are reported in [31, 36]. Performance comparison between MP and the other algorithms is given in the three last columns, representing the performance ratio $100 \cdot (M_{acc}^{BP}/M_{acc}^{alg} - 1)$ where alg indicates the algorithm used (MSG, ACO and LS respectively). We use as M_{acc}^{MP} the best one between MP with and without reinforcement.