Efficient delay routing

Miriam Di Ianni*

Istituto di Elettronica, Università di Perugia, via G. Duranti 1/A, I-06123 Perugia, Italy

Abstract

In this paper the computational complexity of finding packet routing schemes provably efficient with respect to the end-to-end delay is studied. The attention is focused on polynomial-time algorithms able to optimize the end-to-end delay when the number of packets in the network increases, and the number of packets that can be accepted in the network in order to keep the end-to-end delay within a constant value. In particular, the hardness of approximating in polynomial time both the minimum end-to-end delay and the maximum number of accepted packets within any sublinear error in the number of packets is proved. © 1998—Elsevier Science B.V. All rights reserved

1. Introduction

Efficient routing of messages is a fundamental task in parallel and distributed systems. In fact, a large amount of routing algorithms aiming to minimize the completion time of delivering packets have been proposed in the past [22, 24, 29, 30]. However, they were mainly devised to deal with particular topologies. The first step towards the design of topology independent routing algorithms was the randomized technique proposed by Valiant and Brebner [42, 43], even if the authors first used it to route on hypercubes. Successively, a series of fundamental papers [1, 41, 36, 38, 23, 26, 32] showed the effective advantage of randomization in the design of efficient routing strategies.

Universal deterministic packet routing has been significantly approached by Leighton et al. [25, 26]. Their solution to routing consists of two steps: during the first one the paths to be followed by the packets are selected, while the second step, usually called scheduling, is used for timing packet movements in order to minimize the total delivery time without violating network constraints (limited bandwidth and limited channel queue size). In [25], the authors proved the existence of a schedule bringing all packets to their respective destinations in \(O(C + L)\) steps for any set \(\mathcal{P}\) of paths used to route the packets, where \(C\) is the congestion (maximal number of paths in \(\mathcal{P}\) using the same channel) and \(L\) the dilation (length of the longest path in \(\mathcal{P}\)) of

* E-mail: diianni@istel.ing.unipg.it.
Such a proof was based on the Lovasz's local lemma and, thus, inherently nonconstructive. By using an algorithmic form of Lovasz's local lemma recently discovered by Beck [7], in [28] a polynomial-time algorithm able to find such a schedule has been presented. Finally, by exploiting the results and the techniques used in [25,26,28], an $\varepsilon$-approximating algorithm for the minimum routing time (that is, an algorithm able to devise an asymptotically optimum routing scheme) [40] and an $\log^{1-\varepsilon}|V|$-approximating algorithm for the minimum on-line routing time [34] have been recently proposed.

All previously cited papers refer to store-and-forward networks. However, the interest in routing algorithms has recently turned also to wormhole routed and optical networks [6,11,15,39,2,3,37].

Usually, the delivery time of a packet is expressed as the sum of network latency and end-to-end delay. While the first quantity is strictly dependent on the architectural choices for the network (topology, switching technique) and is not affected by network congestion, the end-to-end delay is a byproduct of congestion and measures the number of times a packet must wait for traversing the links because of their limited bandwidth. In order to study the effects of network congestion on its performance, in this paper the attention is focused on the end-to-end delay.

The end-to-end delay is assuming increasing relevance due to the recent introduction of switching techniques yielding network latencies which can be considered almost independent from the lengths of the routes. In the traditional store-and-forward, when a packet is received by a node the entire packet is stored in one of its free buffers. The packet is forwarded to a neighboring node only if the corresponding channel is available and that node has a free buffer. The major drawback of store-and-forward is its distance-dependent latency: if $b$ is the bandwidth, $p$ the size of a packet and $r$ the length of its route, store-and-forward latency is roughly $l = r p/b$. Virtual cut-through [33] stores packets in buffers only when the channels they require are busy. In this case, the latency can be expressed as $l = r p_h/b + p/b$, where $p_h$ is the length of the header field. Notice that, if $p_h \ll p$, the distance $r$ produces a negligible effect. Wormhole routing [33] divides each packet into a number of flits (flow control digits) for transmission, with the header flit governing the route. As the header advances, the remaining flits follow in a pipeline fashion. If the header flit encounters a channel already in use, it is blocked until the channel becomes available. Once a channel has been acquired by a packet it is released only when the last flit has been transmitted on the channel. In this case, the latency is given by $l = r p_f/b + p/b$, where $p_f$ is the length of a flit. Again, if $p_f \ll p$, the path length $r$ does not significantly affect network latency. Notice that, in networks using the last two switching techniques, the end-to-end delay is the main factor which strongly affects the delivery time of packets.

In this paper, two different problems are considered, both related to the attempt of devising routing algorithms provably efficient with respect to the end-to-end delay: route a set of packets in order to either minimize the maximum end-to-end delay or maximize the number of packets that can be delivered within a fixed end-to-end delay. The problem of routing within a fixed delay (also known as call admission problem) has several motivations. First, it is a useful tool in real-time situations, when
packets must be delivered within a fixed deadline, after which they become useless. Secondly, in optical networks it is more convenient to destroy collided packets and to ask for their retransmission than buffering the ones that cannot be immediately transmitted, since electronic-optical conversions (necessary whenever a packet must be stored in a buffer) are very time-expensive. In this case, a good routing strategy is partitioning the packets in sets of pairwise nonconflicting packets, each set allowed to be transmitted with delay 0. Similar call admission problems have already been considered in [4, 5, 16, 14]. From the previous discussion about the incidence of network latency on the delivery time, the relevance of the minimum delay routing problem should be clear. Furthermore, there is also a more theoretical reason leading to consider it which is related to the results presented in [25]. Indeed, even if the bound $O(C + L)$ on the delivery time of a schedule is asymptotically optimal, it was still unknown if it is possible to find in polynomial time a schedule that completes the delivery of packets within $L + O(C)$ steps. Notice that, while $L$ is a "physical" constraint on the delivery time of any schedule, the number of steps required after the first $L$ ones is a measure of the goodness of a schedule with respect to the congestion $C$.

1.1. Results and paper organization

The minimum end-to-end delay problem has already been considered in [9, 10], where the authors proved the impossibility of optimally routing in polynomial time and the hardness of approximating the minimum end-to-end delay in a store-and-forward network model in which the main resource to be shared among packets is storage inside nodes. This means that each node can contain at each time step a number of packets at most equal to the number of packets that can be transmitted (in a single time step) along any of its outgoing edges. Such a model will be called buffer model. In this paper, a different network model is considered, corresponding to the case in which there is no bound on the number of packets a node is allowed to contain at each time step while the bandwidth is kept bounded. Such a model will be called channel model: in a channel network, the main resource to be shared among packets is edge bandwidth.

Of course, the channel model is able to represent both the store-and-forward and the virtual cut-through switching modes. Concerning store-and-forward, the channel model is more relevant than the buffer model, since limited bandwidth is a much more crucial bottleneck for the throughput of a network than limited buffer availability. In fact, all the routing algorithms in the literature cited in the introduction refer to the channel model. Notice that, wormhole routing can be seen as a generalization of store-and-forward in the channel model. Thus, all the negative results presented in this paper for the channel model can be immediately extended via generalization to wormhole routing. The model will be better described in the next subsection.

In Section 2 the minimum delay routing problem is considered in the channel model. In Section 2.1 the power of centralized strategies is studied. In particular, it is proved that, if the paths to be used by packets are chosen in advance (according to the routing
paradigm proposed in [25]) the minimum end-to-end delay cannot be approximated within a relative error in $O(k^{1-\delta})$ for any $\delta > 0$, where $k$ is the number of packets, unless $P=NP$. It follows that it is impossible to find in polynomial time schedules whose delivery time is $L+O(C)$. As an easy corollary, it is then shown that, in the same hypothesis, the minimum end-to-end delay cannot be approximated within a relative error in $O(L^{(1/2) - \delta})$ for any $\delta > 0$. In other words, this means that it is impossible to find in polynomial time schedules whose delivery time is $L+O(L^{(1/2) - \delta})$ for any $\delta > 0$.

Notice that, routing algorithms separating the phase in which the choice of the paths occur from the scheduling of channel requirements are realistic only in centralized environments, since in this case paths can be easily precomputed and whatever can be done without precomputed paths can also be done (with roughly the same performance) with precomputed paths. However, in large distributed systems centralized strategies are somehow impractical, since they require a large amount of information to be exchanged between nodes in the network. More frequently, the strategy to solve collisions and to decide the outgoing edge onto which to forward a packet at a given time is chosen by each node according to the knowledge of the state of its neighborhood only. A relevant issue is thus to investigate the performance of such local strategies. A first contribution to this aim has already been given in [10] by introducing a "local optimum" criterion: if two or more packets are simultaneously requiring a buffer in a node and one of them can alternatively choose another node belonging to a different shortest path towards its destination and which is not requested by any other packet then it will use such node. Due to its similarity with deflection routing, this kind of strategies will be called deflection strategies. Deflection strategies are practically relevant. Indeed, if each node has the knowledge of the occupancy state of its neighbors only, it has not sufficient information for delaying a packet which can advance along another shortest path. The behavior of similar greedy strategies has been investigated also in [8] and in [31] with respect to the minimum delivery time and, as remarked in the second paper, such greedy policies are used in many cases.

According to the standard definition of approximability [17, 19], the performance of a local strategy should be compared to the performance of an optimal one, no matter if this optimum can be actually achieved by a local strategy. This clearly implies that all negative results obtained for global strategies can be immediately extended to the performance achieved by any local strategy. However, this criterion is often too pessimistic, that is, if global strategies are unrealistic, the performance of a local strategy should be compared to the optimum achievable by local strategies (see [21] for more discussions). In Section 2.2, the performance of deflection strategies is considered in the channel model and it is proved that approximating the minimum end-to-end delay with respect to deflection strategies within a relative error in $O(k^{1-\delta})$ or in $O(L^{1-\delta})$ is NP-hard for any $\delta > 0$.

In Section 3 the call admission problem is considered: whenever a set of nodes in the network wants to send messages to other nodes, the call admission algorithm selects a subset of the requests which can be satisfied within a fixed end-to-end delay.
In Section 3.1 it is proved (again by means of approximation preserving reductions) that, once the paths to be used by packets are fixed, it is impossible to approximate the maximum number of packets which can be accepted in the network in order to route them with no end-to-end delay within a relative error in $O(r^{1-\delta})$, for any $\delta > 0$, where $r$ is the number of communication requests, unless $P=NP$. Next, in Section 3.2 it is also considered a call admission control that does not work on precomputed paths, that is, it is required to find a set of paths in order to maximize the number of packets that can be delivered within a fixed delay. If the choice of the paths and the scheduling of channel assignments happen according to the rules of deflection strategies, it is proved that the maximum number of packets cannot be approximated with a relative error in $O(r^{1-\delta})$, for any $\delta > 0$. Again, the relative error is computed with respect to an optimum obtained by a deflection strategy.

Finally, in the last section some conclusive remarks and open questions are briefly discussed.

1.2. Preliminary definitions

A (channel) network is represented as a graph, where nodes stand for sites containing the processing elements and edges for communication links which must be shared by the messages. A bandwidth is associated to each edge, representing the maximum number of packets which can be simultaneously transmitted on it. In this paper it is always assumed that the bandwidth of each edge is 1 and that transmission along every edge takes exactly one time unit.

Each packet in the network follows a route starting at its source node (which generates the packet) and ending at its destination (which eliminates the packet from the network). Each packet transmission along one edge requires one unit of its bandwidth. If a packet is requiring an edge already assigned to some other packet, it cannot be transmitted along that link and it is stored in the (potentially unbounded) output queue of that link.

A network is in the initial configuration if all packets are into their respective source nodes; it is in the final configuration if each packet has reached its own destination and, thus, has been removed by consumption. The routing ends when the network reaches the final configuration. The number of steps (i.e., of configurations met) necessary to a routing algorithm to pass from the initial configuration to the final one is called delivery time or length of the routing algorithm.

If a routing algorithm $S$ is such that at a given time $i$ a packet $p$ has neither reached its destination nor is transmitted along any edge, then $p$ is said to be delayed at instant $i$ of $S$. The delay $d(p,i)$ of packet $p$ at instant $i$ denotes how many times $p$ has been delayed from instant 0 to instant $i$. The end-to-end delay of routing algorithm $S$, in symbols $d(S)$, is the maximum of the $d(p,l)$'s where $l$ denotes the length of $S$.

Recall that a proof of intractability for a problem under some restrictive hypothesis implies (by generalization) its intractability even in the general case. In view of this, since all results presented in this paper are hardness proofs, the assumption on the
bandwidth and on the transmission time onto every edge are not restrictive. Furthermore, following the same line of reasoning, layered networks (i.e., having a layered support graph) are always considered. Layered graphs are such that nodes are partitioned into \( L + 1 \geq 2 \) sets or levels \( V^i \), with \( 0 \leq i \leq L \) and edges exist only between consecutive levels. As a further restriction, only instances such that all the source nodes are included in level 0, and all the destinations are included in level \( L \) are taken into account.

2. Approximating the minimum end-to-end delay

The attention in this section is focused on the possibility of devising polynomial-time algorithms able to find routing strategies yielding approximate delays, that is, delays having bounded relative error with respect to the optimal ones. Here, the relative error of an algorithm \( \mathcal{A} \) for a minimization problem \( \Pi \) is defined as follows:

\[
\frac{m(S_\mathcal{A}(x))}{m(S^*(x))},
\]

where \( S^*(x) \) is an optimum solution relative to an instance \( x \) of \( \Pi \), \( S_\mathcal{A}(x) \) is the approximate solution found by \( \mathcal{A} \), and \( m(S^*(x)) \) and \( m(S_\mathcal{A}(x)) \) are, respectively, their sizes. A problem is said to be \( \varepsilon \)-approximable if a polynomial-time algorithm \( \mathcal{A} \) exists such that the relative error is never greater than \( \varepsilon \).

In [9, 10] the minimum delivery time routing and the minimum end-to-end delay routing problems have been studied in the buffer model. Although the buffer and the channel models look quite similar, transforming a buffer network \( N \) into a channel network \( N' \) in order to preserve the delivery order of the packets is not trivial. As an example, consider the intuitive transformation of \( N \) into \( N' \), mapping each node \( u \) of \( N \) into an edge \((u, u')\) of \( N' \) and assigning to it a bandwidth equal to the number of buffers included in \( u \). Consider now, the buffer network \( N \) in Fig. 1(a) in which every node contains one buffer: if packet \( p_2 \) occupies the buffer contained in \( u \) at the first step and packet \( p_3 \) occupies the buffer contained in \( v \) at the second step, a schedule of length \( L + 4 = 7 \) is obtained in which no pair of packets reach their destinations at the same time. In Fig. 1(b), the network \( N' \) corresponding to \( N \) (according to the previously described transformation) is shown: in this case edges \((u, u')\) and \((v, v')\) have both bandwidth 1. By using the same priorities as before in order to assign edges to packets (i.e., edge \((u, u')\) is assigned first to \( p_2 \) and edge \((v, v')\) is assigned first to \( p_3 \)), we get a schedule of length \( L + 1 \) in which packets \( p_1 \) and \( p_2 \) reach their destinations at the same time.

The problems encountered in transforming a buffer network into a channel one are mainly due to the following consideration: in a buffer network, a packet holds a buffer in a node until it occupies a new buffer in the next node on its path, thus forbidding to the first node to accept any other packets even if they will use different outgoing channels; conversely, in a channel network, a packet holds a channel only while using
it, and the channel is released even if, after having traversed it, the packet is blocked. In
spite of the previously remarked differences, it is possible to "simulate" a buffer network
by a channel network so that for any routing algorithm in the first network there exists
a routing algorithm in the second one such that the packets are delivered in the same
sequence by both of them. However, the simulation is very complicated and holds
only under very restrictive hypothesis. If such hypothesis are sufficient to extend the
results for the approximation of the minimum delay in [10], they make uninteresting
the simulation. Furthermore, it enlarges the number of levels of the channel network
of an over-linear factor. For all of these reasons, and since the networks in [10] are
too "long" to prove a bound on the relative error as a function of the dilation, in this
paper the nonapproximability of the minimum delay is proved by original reductions.

2.1. Centralized strategies

It is first proved that the minimum end-to-end delay problem cannot be approximated
in polynomial time even by strategies such that the decisions about packet transmissions
are taken according to the knowledge of the state of the entire network.

Theorem 1. Finding a schedule to route a set of k packets onto precomputed paths
such that the achieved end-to-end delay is at most $O(k^{1-\delta})$ times the optimum one
is NP-hard for any $\delta > 0$.

Proof. The proof is a linear reduction from MIN-COLORABILITY [20]: given a graph $G = (V, E)$, find the minimum-size partition $V_1, \ldots, V_{h_{\min}}$ of $V$ such that no pair of nodes
contained in a same $V_i$ are adjacent in $G$. In particular, the reduction transforms a graph $G = (V, E)$ into a layered network containing in level 0 as many packets as the nodes
in $V$. The paths to be used by the packets are built so that two paths are edge-disjoint
if and only if the two nodes corresponding to the packets that will use them are not
adjacent in $G$. Furthermore, the set of paths has the following property: if a subset of
the packets reaches the last level of the network (containing their destinations) with
the same delay, the corresponding subset of $V$ does not contain any pair of adjacent
nodes. This implies that the original graph $G$ can be colored with $h$ colors if and only if the corresponding network admits a schedule having end-to-end delay $h - 1$. Such a reduction preserves approximability properties and thus, since $\text{MIN-COLORABILITY}$ has been recently proved to be not approximable with an error in $O(1\sqrt{\ln n})$ for any $\delta > 0$ [13], this will allow to prove the assertion.

The idea of the proof is similar to the one presented in [10]. However, since the dilation of the network built in the cited paper is very large ($O(k^3)$), it is now presented a more compact way of representing the edges of the graph in the instance of the reduced problem.

Let $(G = (V,E))$ be an instance of $\text{MIN-COLORABILITY}$ with $V = \{a_1, \ldots, a_n\}$. The reduction maps $(G = (V,E))$ into a network $N^G$ containing $n$ source nodes $s_1, \ldots, s_n$ corresponding, respectively, to $a_1, \ldots, a_n$. The packet contained in $s_i$ will be denoted as $x_i$. An edge $(v_i, v_j)$ in $G$ is represented in the pair of levels $l, l+1$ of $N^G$ if there exists an edge $e$ in $N^G$ from level $l$ to level $l+1$ such that both paths of the two packets $x_i$ and $x_j$ contain $e$.

The network is built as a sequence of $n$ identical filters such that the $n$ outputs of one filter are connected to the $n$ inputs of the next one (see Fig. 2(c)). The goal of a filter is to create conflicts between pairs of packets representing adjacent nodes in the input graph $G$. Each filter is a layered network with $n$ nodes in the first and in the last levels and $L_F = 4\left\lfloor\frac{n-1}{2}\right\rfloor + 2\left\lfloor\frac{n-1}{2}\right\rfloor + 2$ levels. Each pair of inner levels represents a subset of non consecutive (that is, having no end in common) edges, as described in the following:

1. levels $4i - 3 \geq 1$ and $4i - 2 \leq \left\lfloor\frac{n-1}{2}\right\rfloor$ 2 represent the set of edges

$$E(4i - 3) = \{(v_1, v_{1+i}), (v_{i_1}, v_{i_1+i}), \ldots, (v_{i_k}, v_{i_k+i})\},$$

with $i_j = \min\{i \notin \{1, 1+i, i_1 + i, \ldots, i_{j-1} + i\}\}$, eventually included in $G$;

2. levels $4i - 1 \geq 3$ and $4i \leq \left\lfloor\frac{n-1}{2}\right\rfloor$ represent the set of edges

$$E(4i - 1) = \{(v_{1+i}, v_{1+2i}), (v_{i_1}, v_{i_1+i}), \ldots, (v_{i_k}, v_{i_k+i})\},$$
with $i_j = \min\{l \notin \{1 + i, 1 + 2i, i_1 + i, \ldots, i_{j-1} + i_1 + i\}\},$ eventually included in $G$ and if they are not represented in the two levels above. Notice that this second set is nonempty whenever $G$ includes paths of length 3; for instance, in Fig. 2, edge $(v_2, v_3)$ needs to be represented in levels 3 and 4 since levels 1 and 2 are used to represent edge $(v_1, v_2)$.

(3) levels $2i - 1 \geq [(n - 1)/2]$ and $2i \leq L_F - 1$ represent the set of edges

$$E(2i - 1) = \{(v_1, v_1 + i), (v_1, v_1 + i), \ldots, (v_i, v_i + i)\},$$

with $i_j = \min\{l \notin \{1 + i, 1 + 2i, i_1 + i, \ldots, i_{j-1} + i_1 + i\}\},$ eventually included in $G$.

Let us denote by $E'(2i - 1)$ the set of edges in $G$ included in $E(2i - 1)$, $1 \leq 2i - 1 \leq L_F - 2$. By construction, each $E'(2i - 1)$ contains pairwise non-consecutive edges and $E = \bigcup_{i \geq 1} E'(2i - 1)$. Each of the two levels that have to represent $E'(2i - 1)$ contains $n - |E'(2i - 1)|$ nodes with exactly $|E'(2i - 1)|$ nodes in level $2i - 1$ having indegree two and $|E'(2i - 1)|$ nodes in level $2i$ having outdegree two. The $j$th node with indegree 2 is connected to the $j$th node with outdegree 2 and such edge must be used by both of the two packets corresponding to the two ends of the $j$th edge in $E'(2i - 1)$. All the remaining nodes have indegree and outdegree one: the $j$th edge connecting such nodes is used by the $j$th packet corresponding to a node which is the end of no edge in $E'(2i - 1)$. As an example of the transformation see Fig. 2.

Clearly, such a network can be constructed in polynomial time and includes $L + 1 = nL_F$ levels.

It must be proved now that $G$ can be colored with $h \leq n$ colors if and only if a schedule for the network exists with end-to-end delay $h - 1$.

Suppose first that a partition of $V$ into $h$ subsets $V_1, \ldots, V_h$ of pairwise non-adjacent nodes exists; then, packets are partitioned into $h$ sets $P_1, \ldots, P_h$ where each $P_i$ contains packets associated to the nodes in $V_i$. Consider the schedule $S$ in which packets in $P_i$ leave level 0 with a delay equal to $i - 1$, for $i = 1, \ldots, h$. Since nodes in $V_i$ are pairwise not adjacent, all packets included in a same set $P_i$, $i = 1, \ldots, h$, use distinct edges within the network and, thus, they are not furtherly delayed. Hence, all packets in $P_i$ reach their destinations with delay $i - 1$, that is, the end-to-end delay of such a schedule is $h - 1$.

Conversely, suppose the network admits a schedule with end-to-end delay $h - 1$ ($h \leq n$). Hence, the packets arrive at the end of the last filter partitioned into $h$ sets $P_1, \ldots, P_h$. Thus, none of the $P_i$ contains any pair of packets which are in conflict for the use of some edge. Indeed, this property is true when the set of packets having delay 0 (i.e., the set $P_1$) leaves the first filter. However, some pair of packets leaving the first filter with the same delay may have a conflict for the use of some edge in the next levels: for instance, this happens if packet $x$ and packet $y$ have a conflict in level $i$ of the first filter and both of them are delayed by some packet in $P_1$, respectively, in levels $j_x < i$ and $j_y > i$. But, at the end of the second filter, packets having delay 1 (i.e., the set $P_2$) are pairwise nonconflicting. In general, at the end of the $i$th filter this property is true for the set $P_i$. Thus, after at most $h$ filters none of the sets $P_i$ can contain conflicting packets. It follows that nodes corresponding to packets included in...
a same set \( P_1 \) are pairwise not adjacent, that is, nodes of \( G \) can be partitioned into \( h \) pairwise disjoint subsets \( V_1, V_2, \ldots, V_h \), where each \( V_i \) contains the nodes corresponding to packets included in \( P_i \).

Suppose that a polynomial-time \( \varepsilon(k) \)-approximation algorithm \( \mathcal{A} \) for the minimum end-to-end delay schedule problem exists (\( k \) being the number of packets to be transmitted), that is, there exists a \( \varepsilon(k) \in O(k^{1-\delta}) \), for some \( \delta > 0 \), such that for any networks \( N \), \( \mathcal{A} \) yields a scheduling \( S_{\mathcal{A}}(N) \) whose end-to-end delay \( d(S_{\mathcal{A}}(N)) \) satisfies the following relation:

\[
\frac{d(S_{\mathcal{A}}(N))}{d^*(N)} < \varepsilon(k),
\]

where \( d^*(N) \) denotes the optimum end-to-end delay for \( N \). Consider a graph \( G \) with \( k \) nodes and transform it according to the reduction above. Then apply \( \mathcal{A} \) to the corresponding \( N^G \): let \( V_1, \ldots, V_h(\mathcal{A}) \) be the partition of the nodes in \( G \) induced by the schedule \( S_{\mathcal{A}}(N^G) \). Since the size of an optimum coloring for \( G \) is \( h_{\text{min}} = d^*(N^G) + 1 \) then

\[
\frac{h(\mathcal{A})}{h_{\text{min}}} = \frac{d(S_{\mathcal{A}}(N)) + 1}{d^*(N) + 1} < \varepsilon(k) + 1.
\]

This implies that any approximation algorithm for the minimum end-to-end delay problem can be used also for \textsc{min-colorability} with the same asymptotical performance. Since \textsc{min-colorability} cannot be approximated with an error in \( O(k^{1-\delta}) \) for any \( \delta > 0 \) [13], the assertion is proved.

As a consequence of the previous theorem, the minimum end-to-end delay problem cannot be optimally solved in polynomial time. Since the length \( l \) of a schedule \( S \) for an \( L+1 \)-levels layered network \( N \) satisfies the relation \( l = L + d(S) \), this implies that it is impossible to optimally solve in polynomial time the minimum delivery time problem, unless \( P = NP \). However, the same cannot be said concerning its approximation properties.

The delay, besides function of the congestion of the network (i.e., of the number of packets in the network), also depends on the dilation. Informally, a packet having to travel a long distance is more likely to be delayed a larger number of times than a packet having to cover a few hops. Thus, a noticeable consequence of Theorem 1 is expressed in the following corollary.

\textbf{Corollary 2.} Finding a schedule to route a set of \( k \) packets onto precomputed paths such that the achieved end-to-end delay is at most \( O(L^{(1/2) - \delta}) \) times the optimum one is \( \text{NP-hard} \) for any \( \delta > 0 \), where \( L \) is the dilation.

\textbf{Proof.} The assertion directly follows from Theorem 1 by noticing that the network built in the proof is such that \( L = O(k^2) \). \( \square \)
2.2. Local strategies

The approximability of the minimum delay routing problem is now studied with respect to deflection strategies, that is, local strategies that force packets to advance along free shortest paths whenever possible. Notice that, deflection strategies can be meaningfully defined only for the arbitrary paths case.

Theorem 3. Finding a deflection strategy which achieves an end-to-end delay at most $O(k^{1-\delta})$ times the optimum one is NP-hard for any $\delta > 0$, where $k$ is the number of packets.

Proof. The proof is based on a reduction from the DISJOINT CONNECTING PATHS problem (in short DCP) that produces a large gap between the minimum end-to-end delay in a network corresponding to a yes instance of DCP and the minimum end-to-end delay in a network corresponding to a no instance. DCP is defined as follows: given a graph $G$ and a set $\{(s_1, t_1), (s_2, t_2), \ldots, (s_h, t_h)\}$ of pairs of nodes of $G$, decide if $G$ contains $h$ pairwise disjoint paths, each connecting a pair $(s_i, t_i)$, $i = 1, \ldots, h$. DCP is a well-known NP-complete problem [17] and it has been proved to remain NP-complete also for instances restricted to layered graphs [9].

Given the layered graph $G$ with $L_G + 1$ levels and the $h$ pairs of nodes, the network $N^G$ of the corresponding instance of the minimum end-to-end delay deflection strategy problem is composed by $h$ identical subnetworks $N^1, N^2, \ldots, N^h$ plus a final "funnel" $F$. In turn, each $N^i$ contains $2L_G + 3$ levels and is partitioned into two further subnetworks, $N^i_1$ and $N^i_2$:

1. $N^i_1$ contains $h$ pairs of source nodes at level 0. The two packets belonging to a pair, $x_j^i$ and $y_j^i$, are forced to use the same edge from a node in level 1 to a node in level 2 (see Fig. 3). This device is used in order to avoid schedules with delay 0: they would generate inconsistencies because of the ratio in the definition of relative error.

2. The pair of levels 2 and 3 of $N^i_1$ exactly corresponds to the pair of levels 0 and 1 of $G$. In general, the pair of levels $2j$ and $2j + 1$ of $N^i_1$ exactly corresponds to the pair of levels $j$ and $j + 1$ of $G$. The remaining pairs of levels ($2j - 1$ and $2j$) only contain "vertical" edges, that is, edges connecting the $l$th node in level $2j - 1$ to the $l$th node in level $2j$.

3. $N^i_2$ does not contain any source node, starts at level 4 and consists of $2L_G - 1$ levels.
(4) Each level of $N^2_i$ contains $h$ nodes and $N^2_i$ consists of a set of $h$ disjoint chains;
(5) Level $2j + 1$ of $N^1_i$ and level $2j + 2$ of $N^2_i$, $j \geq 1$, are a complete bipartite graph.

$F$ contains 3 levels, with the first and the second levels containing $h^2 + 1$ nodes:
all nodes of the last level of $N^2_i$, $i = 1, \ldots, h$, are connected with the last node of
the first level of $F$, while node $j$ of the last level of $N^1_i$ is connected with node $h(i - 1) + j$ of the first level of $F$. Node $h(i - 1)j$ of the first level of $F$ is connected
with node $h(i - 1)j$ of its second level which, in turn, is connected with the common
destination of $x^j$ and $y^j$, $j = 1, 2, \ldots, h^2$. Finally, the last node of the first level of $F$ is
connected with the last node of second level which, in turn, is connected with every
destination.

$N^G$ contains $2L_G + 6$ levels and can be constructed in polynomial time. In Fig. 4
it is shown an example of the reduction. For the sake of simplicity, levels 0 and 1
of $N^1_i$ and $N^3_i$ (shown in Fig. 3) have not been drawn and only packets $x$ have been
depicted; finally, the last node of the first two levels of $F$ has been drawn in the
center.

Notice that, if a packet $x_i^j$ or $y_j^i$ enters $N^2_i$ then there is a unique (shortest) path it
can follow to reach its destination.

Notice also that $N^1_i$ contains $h$ disjoint paths between the source and the destination
of each packet $x_i^j$ (or $y_j^i$) if and only if the input graph $G$ contains $h$ disjoint paths
between the pairs $(s_i, t_i)$ of nodes. Thus, if $G$ contains $h$ disjoint paths between
the pairs $(s_j, t_j)$ an optimum deflection strategy with end-to-end delay $d^* = 1$ is easily
achieved when all the $x_i^j$'s follow the disjoint paths in $N^1_i$, followed by the $y_j^i$'s in a
pipeline fashion, for any $i = 1, \ldots, h$. Indeed, in this case no packet is forced to wait
when it is possible for it to advance.

Conversely, if $G$ does not contain the $h$ disjoint paths, then all deflection strategies
have end-to-end delay $d \geq 2h - 1$. In fact, whenever two packets $x_i^j$, $x_k^j$ (or $y_i^j$, $y_k^j$)
have a conflict for using some edge of $N^1_i$, one of them is forced to advance in the
first node of the next level of $N^2_i$, according to the definition of deflection strategy.
This implies that, in order to reach their own destinations, all of them have to pass through the "funnel" in the last node of the first level of $F$.

Suppose now an $f(h)$-approximation algorithm $\mathcal{A}$ for the minimum end-to-end delay routing problem exists, where $f(h) = o(h)$. It will be shown that the reduction and the $f(h)$-approximation algorithm for the minimum end-to-end delay deflection strategy problem correspond to a polynomial-time algorithm which decides $\text{DCP}$, a contradiction with the NP-completeness of this last problem. Indeed, consider an instance of $\text{DCP} (G, (s_1, t_1), \ldots, (s_h, t_h))$. Without loss of generality, since $f(h) = o(h)$ and $\text{DCP}$ is still NP-complete when $h$ is bigger than any positive constant [17], it is sufficient to consider an instance in which $h > (f(h) + 1)/2$. Then transform it into a network $N^G$ according to the reduction and apply $\mathcal{A}$ to $N^G$. If the $h$ disjoint paths exist in $G$, $\mathcal{A}$ finds a deflection strategy having end-to-end delay $d < f(h)$. Similarly, if the $h$ disjoint paths do not exist, the algorithm finds a deflection strategy having end-to-end delay $d \geq 2h - 1 > f(h)$.

Since $k = 2h^2$, till now it has been proved that the minimum end-to-end delay deflection strategy problem cannot be approximated with an error in $O(k^{(1/2) - \delta})$ for any $\delta > 0$. Notice now that the same reasoning can be repeated for a network $N^G$ composed by $h'$ identical subnetworks $N^1, N^2, \ldots, N^{h'}$, for any $r > 0$: in this case, it can be shown that the minimum end-to-end delay deflection strategy problem cannot be approximated with an error in $O(k^{r/(r+1) - \delta})$ for any $\delta > 0$. Since this claim holds for any $r > 0$, the assertion is completely proved. □

It is interesting to observe that the network $N^G$ in the proof of the previous theorem can be easily modified in order to place the structure that induces the large gap arbitrarily "far" from the position in which the decisions that generate a "big" delay are taken. In other words, this means that the theorem holds even for deflection strategies in which nodes have the knowledge of all the nodes which are at a distance of at most $i$ from them, for any $i > 0$.

Also in this case it is possible to express the bound on the error as a function of the dilation (well-defined in a layered network even in the arbitrary paths case) as stated in the next corollary.

**Corollary 4.** Finding a deflection strategy which achieves an end-to-end delay at most $O(L^{1-\delta})$ times the optimum one is NP-hard for any $\delta > 0$, where $L$ is the dilation.

**Proof.** The NP-completeness proof of $\text{DCP}$ restricted to layered graphs in [9] is a polynomial-time reduction from $\text{VERTEX COVER}$: given a graph $G - (V, E)$ and an integer $k$, decide if $V' \subseteq V$ of cardinality at most $k$ exists such that for every edge $(u, v) \in E \ u \in V'$ or $v \in V'$. The layered graph instance of $\text{DCP}$ built in the reduction contains $L + 1 \in O(|E|)$ levels and $k \in O(|E|)$ source–destination pairs.

Since the number of levels (and, thus, the dilation) of the network built in the proof of Theorem 3 is within a constant factor of $L$, the assertion follows as a consequence of that theorem. □
3. Call admission

In this section the attention is focused on call admission problems: whenever a set of nodes in the network wants to send messages to other nodes, the call admission algorithm selects a subset of the requests which can be satisfied within a fixed end-to-end delay. The problem is to select the largest set of requests to be satisfied. Here, the relative error of an algorithm $\mathcal{A}$ for a maximization problem $\Pi$ is defined as follows:

$$\frac{m(S^*(x))}{m(S_{\mathcal{A}}(x))},$$

where $S^*(x)$, $S_{\mathcal{A}}(x)$, $m(S^*(x))$ and $m(S_{\mathcal{A}}(x))$ are defined similar to Section 2.

As already remarked in the introduction, the main motivation for this section is related to applications in optical networks. Thanks to WDM, optical networks usually allow for high bandwidth and the results of this paper could seem at first glance quite restrictive, since limited to bandwidth 1. However, they can be trivially extended to each bound on the bandwidth by generalization.

3.1. Centralized strategies

It is now proved that the maximum call admission problem cannot be approximated in polynomial time even by strategies allowed to use the knowledge of the state of the entire network. In this case, the following theorem holds:

**Theorem 5.** Approximating the maximum number of communication requests which can be accepted by a channel network in order to be satisfied with no end-to-end delay with an error in $O(r^{-\delta})$ is $\mathsf{NP}$-hard for any $\delta > 0$, where $r$ is the number of communication requests.

**Proof.** The theorem is proved for the paths being fixed before the requests are submitted to the system. As noticed in the introduction, such assumption is realistic when dealing with centralized strategies.

The proof is an approximation preserving reduction from the well-known $\mathsf{NP}$-hard $\mathsf{MAX-Clique}$ problem [17]: given a graph $G = (V, E)$, find the maximum-size clique included in $G$. The reduction from $\mathsf{MAX-Clique}$ is such that the original graph $G$ contains a clique of $h$ nodes if and only if the corresponding network can accept $h$ communication requests that can be scheduled with end-to-end delay 0. Such a reduction preserves the approximation property and thus, since $\mathsf{MAX-Clique}$ has been proved to be not approximable with an error of $O(|V|^{1-\delta})$ for any $\delta > 0$ [18], a similar result will hold for the call admission problem.

Let $(G = (V, E))$ be an instance of $\mathsf{MAX-Clique}$ with $r = |V|$ and $m = |E|$. The reduction maps $(G = (V, E))$ into a layered network $N^G$ containing $r$ nodes $s_1, \ldots, s_r$ in level 0, each of which corresponds to a node of $G$, and $r$ nodes $t_1, \ldots, t_r$ in level $L$. Every pair $(s_i, t_i)$ is a communication request submitted to the flow control procedures of the system.
Fig. 5. Network corresponding to graph $G$.

$N^G$ contains $L - 1 = r(r - 1) - 2m$ inner levels; it is described in terms of the paths chosen for the communication requests. Each pair of inner levels $(2i - 1, 2i)$ corresponds to a pair of nonadjacent nodes $v, w$ of the input graph $G$: they contain $r - 1$ nodes with exactly one node in level $2i - 1$ having indegree two and one node in level $2i$ having outdegree two. Such nodes are connected by an edge that must be used by both of the two packets representing $v$ and $w$. All the other nodes have indegree and outdegree one (see Fig. 5).

Clearly, such a network can be constructed in polynomial time.

By the above construction, a set of packets reaches level $L$ with no end-to-end delay if and only if they correspond to pairwise adjacent nodes in $G$. Thus, $G$ contains a clique of $h$ nodes if and only if $h$ packets exist which can be scheduled with end-to-end delay 0. This implies that the reduction satisfies the property according to which any set of $h$ admissible packets (i.e., packets admitting a 0 end-to-end delay schedule) corresponds to a $h$-clique for graph $G$. Suppose that a polynomial-time $\varepsilon(r)$-approximation algorithm $\mathcal{A}$ for the maximum call admission problem exists ($r$ being the number of communication requests), that is, there exists a $\varepsilon(r) \in O(r^{1-\delta})$ such that for any network $N$, $\mathcal{A}$ accepts $R(\mathcal{A}, N)$ requests that can be scheduled with end-to-end delay 0, with

$$\frac{R^*(N)}{R(\mathcal{A}, N)} < \varepsilon(r),$$

where $R^*(N)$ denotes the maximum number of requests that can be scheduled with end-to-end delay 0. Then, $\mathcal{A}$ can be applied to the network $N^G$ which corresponds to some graph $G$ with $r$ nodes according to the reduction described above: since the size $h_{\text{max}}$ of a maximum clique in $G$ is the same as the maximum number of admissible requests, then the above relation bounds also the ratio between the approximate and the maximum clique in $G$. In other words, $\mathcal{A}$ can be easily transformed into an $\varepsilon(r)$-approximation algorithm for the maximum clique problem. Since this last problem cannot be approximated with an error in $O(|V|^{1-\delta})$ for any $\delta > 0$ [18] and $r = |V|$, the assertion is proved. $\Box$
3.2. Local strategies

Finally, the approximability of the maximum call admission problem is studied with respect to deflection strategies, that is, local strategies that force packets to advance along free shortest paths whenever possible.

A similar result to the one proved in Theorem 5 can be proved in this case. Notice that, similarly to Theorem 3, in the next theorem the error is computed with respect to an optimum reachable by a deflection strategy.

**Theorem 6.** Approximating the maximum number of communication requests which can be accepted by a channel network in order to be satisfied with no end-to-end delay by a deflection strategy within an error in $O(r^{-\delta})$ is NP-hard for any $\delta > 0$, where $r$ is the number of communication requests.

**Proof.** The proof is based on a reduction from the DISJOINT CONNECTING PATHS problem (in short, DCP) that produces a large gap between the maximum number of accepted requests in a network corresponding to a yes instance of DCP and the maximum number of accepted requests in a network corresponding to a no instance. The DCP problem has already been defined in the proof of Theorem 3.

Given an instance of DCP, that is, a layered graph $G$ with $L_G + 1$ levels and $h$ pairs of nodes, the network $N^G$ of the corresponding instance of the deflection call admission problem is partitioned into four layered subnetworks, $N^1$, $N^2$ and $N^3$ which are “in parallel”, and $N^4$ which is in sequence with the others: $N^1$ directly corresponds to graph $G$, while $N^2$, $N^3$ and $N^4$ are used to force a large number of refused requests when $G$ does not contain the $h$ disjoint paths. In particular, only $N^1$ and $N^2$ contain the communication requests: when the $h$ disjoint paths exist in $G$, packets whose source is contained in $N^1$ may reach their destinations by using only edges in $N^1$ and packets whose source is contained in $N^2$ may reach their destinations by using only edges in $N^2$. Conversely, when the $h$ disjoint paths do not exist in $G$, packets whose source is contained in $N^1$ need to use some edge in $N^2$ thus “disturbing” packets whose source is contained in $N^2$ and forcing them to pass through a “funnel” induced by $N^3$ and $N^4$.

Level 0 of $N^1$ contains $h$ packets, $x_1, \ldots, x_h$, corresponding to the pairs $(s_1, t_1), \ldots, (s_h, t_h)$. $N^1$ is further divided into a sequence of subnetworks $N^1_0, \ldots, N^1_{L_G}$, each corresponding to a pair of consecutive levels in $G$. In particular, $N^1_0$ corresponds to the pair of levels $(0, 1)$ of $G$, $1 \leq i \leq L_G$, and includes $4h^2 + 4$ levels. Denote as $m_j$ the number of nodes in level $j$ of $G$: in level 0 of $N^1_0$ there are $m_{i-1}$ nodes, in all its remaining levels there are $m_i$ nodes. Edges in $N^1_i$ are described in the following:

1. nodes $1, \ldots, m_{i-1}$ in level 0 of $N^1_i$ and nodes $1, \ldots, m_i$ in its level 1 are connected as those in $G$, more formally: nodes $j_1$ and $j_2$, $1 \leq j_2 \leq m_{i-1}$ and $1 \leq j_1 \leq m_i$,” are adjacent in $N^1_i$ if and only if the corresponding nodes are adjacent in $G$;
2. node $j \in \{1, \ldots, m_i\}$ of level $l+1$ is connected to node $j \in \{1, \ldots, m_i\}$ of level $l+1$.
Level 0 of $N^2$ contains $2h^2$ packets $y_1, \ldots, y_{2h^2}$. Also $N^2$ is divided into $L_G$ layered subnetworks $N_{1}^2, \ldots, N_{L_G}^2$, each including $4h^2 + 4$ levels. Each level $l \neq 2, 3$ in $N_{l}^2$ contains $2h^2$ nodes, one more node being added to levels 2 and 3. For simplicity of description, the extra nodes in levels 2 and 3 are denoted, respectively, as $u^l$ and $v^l$ and they are not counted as nodes of such levels. Edges in $N_{l}^2$ are described in the following:

1. Node $j$ of level $l$ is connected to node $j$ of level $l + 1$, $l = 0, \ldots, 4h^2 + 3$ and $j = i, \ldots, 2h^2$;
2. $u^l$ is connected to $v^l$ and $v^l$ is connected to the first node in level 4 of $N_{l}^2$;
3. Node $j$ of level $2j + 3$ is connected to node $j + 1$ of level $2j + 4$, $j = 1, \ldots, 2h^2 - 1$.

$N^1$ and $N^2$ are connected by $m_l$ edges, from any node in level 1 of $N_{l}^1$ to node $u^l$.

$N^3$ does not contain any source node and is still divided into $L_G$ layered subnetworks: each $N_{l}^3$ includes $4h^2 + 4$ levels, with levels 0, 1, 2, 3 and 4 empty. Each (nonempty) level of every $N_{l}^3$ contains $2h^2$ nodes, node $j$ of level $l$ being connected to node $j$ of level $l + 1$. $N^2$ and $N^3$ are connected as follows: node $j$ in level $2j + 2$ of $N_{l}^2$ is connected to node $j$ in level $2j + 3$ of $N_{l}^3$, $j = 1, \ldots, 2h^2$.

$N^4$ contains 4 levels: level 0 includes $h + 4h^2$ nodes, levels 1 and 2 $h + 2h^2 + 1$ nodes, and the last level the $h + 2h^2$ destinations. Node $j$ in level $l$ of $N^4$ is connected to node $j$ in level $l + 1$, $l = 0, 1$ and $j = 1, \ldots, h + 2h^2$. The last $2h^2$ nodes in level 0 are connected to node $h + 2h^2 + 1$ of level 1 which, in turn, is connected to node $h + 2h^2 + 1$ of level 2. Finally, this last node is connected with nodes $h + j$ of level 3, $j = 1, \ldots, 2h^2$ and node $h + 2h^2$ in level 2 is connected to first $h$ nodes in level 3.

$N^1 \cup N^2 \cup N^3$ and $N^4$ are connected by "vertical" edges, i.e., node $j$ in the last level in $N^1 \cup N^2 \cup N^3$ is connected to node $j$ in the first level of $N^4$. In Fig. 6 an example
of the reduction is shown even if, in order to limit its size, only the first $h^2$ packets
$y_j$ and the corresponding portion of $N^3$ are shown.

Observe that $N^1$ (and consequently $N^G$) contains $h$ disjoint paths between the source
and the destinations of each packet $x_i$ if and only if the input graph $G$ contains $h$
disjoint paths between the pairs $(s_i, t_i)$ of nodes. Thus, if $G$ contains $h$ disjoint paths
between the pairs $(s_i, t_i)$ then an optimum deflection call admission algorithm accepting
$r^* = h + 2h^2$ requests is easily achieved when all the $x_i$ follow the disjoint paths in $N^1$
and each $y_j$ follows the $j$th chain in $N^2$. Notice that no packet is forced to wait when
it is possible for it to advance.

Conversely, if $G$ does not contain the $h$ disjoint paths all deflection call admission
algorithms accept $r \leq h + 1$ requests. Indeed, whenever two packets $x_i$ have a con-
lict in some node of $N^1$, one of them is forced to advance in some node $u^j$ of the
next level of $N^2$. According to the restrictions imposed on legal routings, since such
packet must reach the last “column” in $N^2$, this implies that all the $y_j$ are pushed
into $N^3$ and, thus, they have to pass through the “funnel” in the rightmost nodes
in levels 1 and 2 of $N^4$. This implies that exactly one of them can be routed with
delay 0.

Suppose an $f(h)$-approximation algorithm $A$ for the maximum deflection call admission
problem exists, where $f(h) = o(h)$, and consider an instance $(G, (s_1, t_1), \ldots, (s_h, t_h))$
of DCP. Without loss of generality, since $f(h) = o(h)$ and the DCP problem is still NP-
complete when $h$ is bigger than any positive constant [17], the instance can be chosen
such that $h > f(h)$. Then, $(G, (s_1, t_1), \ldots, (s_h, t_h))$ is transformed into a network $N^G$
according to the reduction and $A$ is applied to $N^G$. If the $h$ disjoint paths exist,
$A$ accepts $\alpha \geq (h + 2h^2)/f(h) > h + 1$ requests. Conversely, if the $h$ disjoint paths do
not exist, the algorithm accepts $\beta \leq h + 1$ requests.

This implies that the reduction and the $f(h)$-approximation algorithm for the max-
imum deflection call admission problem correspond to a polynomial-time algorithm
deciding the DCP problem, a contradiction.

Similar to Theorem 3, since $r = h + 2h^2$, till now it has been proved that the deflec-
tion call admission problem cannot be approximated with an error in $O(r^{(1/2)-\delta})$
for any $\delta > 0$. Notice now that the same reasoning can be repeated for a network
$N^G$ containing $h + 2h^2$ communication requests, for any $q > 0$: in this case, since
$A$ accepts $\alpha \geq (h + 2h^q)/f(h) > h^{q-1} + 1$ requests if the $h$ disjoint paths exist in
$G$ and $\beta \leq h + 1$ requests if the $h$ disjoint paths do not exist, it is easy to verify
that the deflection call admission problem cannot be approximated with an error in
$O(r^{\delta/(q+1)-\delta})$ for any $\delta > 0$. Since this claim holds for any $q > 0$, the assertion is comple-
tely proved.

It is interesting to observe that, similar to the network in the proof of Theorem
3, also the network $N^G$ in the proof of the previous theorem can be easily modi-
fied in order to place the structure that induces the large gap arbitrarily “far” from
the position in which the decisions that generate a large number of collisions are
taken.
4. Conclusions

In this paper some computational complexity results have been shown related to the problem of finding efficient end-to-end delay packet routing schemes, trying to optimize both the end-to-end delay, when the number of packets (and, thus, the congestion) increases, and the number of packets which can be accepted in the network in order to keep the end-to-end delay low.

Unfortunately, all the results are negative, even with respect to approximate solutions. This means that it is impossible to design an algorithm that is efficient (with respect to running time) and performs well (with respect to the quality of solutions) in the worst case. It is thus worth, at this point, to carry out an average case analysis, that is, to study if the minimum end-to-end delay routing problem is NP-complete on the average with respect to some reasonable probability distribution in the set of instances, or even to search for heuristics that are on the average approximating.

Finally, since little is known about the actual performances of the various proposed heuristics, another interesting issue would be their effective implementation and simulation in distributed environments.

Different open problems are related to the unlikeliness of the existence of a polynomial-time algorithm able to find a schedule with delivery time \( L + O(C) \) or \( L + O(L^{1/2}) \). They concern the minimum constants \( \alpha \) and \( \beta \) such that a schedule with delivery time \( \alpha L + O(C) \) or \( L + O(L^\beta) \) can be found in polynomial time.

References


[34] R. Ostrovsky, Y. Rabani, Universal \( O(\text{congestion} + \text{dilation} + \log^{1+\epsilon} N) \) local control packet switching algorithms, Proc. 29th Ann. ACM Symp. on Theory of Computing (STOC), 1997, pp. 644–653.


