Neural Approximation of Team Optimal Dynamic Routing Strategies in IP Networks

Marco Baglietto, Raffaele Bolla, Roberto Bruschi, Franco Davoli, and Riccardo Zoppoli

Departement of Communications, Computer and Systems Science
DIST–University of Genova, Via Opera Pia 13, 16145 Genova, Italy.
email: {mbaglietto, lelus, roberto.bruschi, franco, rzop}@dist.unige.it

Abstract— **The dynamic-routing problem in a packet-switching telecommunication network is addressed by a receding-horizon approach. The nodes of the network must accomplish the following tasks: i) generating routing decisions to minimize the expected total delay, spent by messages in the queues at the nodes and on the network links, on the basis of local information and possibly of some data received from other nodes, typically the neighboring ones; ii) computing their routing strategies by measuring local variables and exchanging a small amount of data with other nodes. The first task leads to regard the nodes as the cooperating decision makers of a team organization. The second task calls for a computationally distributed algorithm. Such requirements and the well-known impossibility of solving team optimal control problems under general conditions suggest two main approximating assumptions: 1) the team optimal control problem is stated in a receding-horizon framework, and 2) each decision maker acting at a node is assigned a given structure, in which a finite number of parameters have to be determined, in order to minimize the cost function (this makes it possible to approximate the original functional optimization problem by a nonlinear programming one). Among the various possible fixed-structure functions, feed-forward neural networks have been chosen for their powerful approximation capabilities. The neural approximation of such team-optimal routing strategies is computed in a numerical example, and used in network routing simulations performed by means of ns-2, in order to show the feasibility and effectiveness of the methodology.**

I. Introduction

Dynamic routing in telecommunication networks may constitute a formidable task, if the word "dynamic" is to be interpreted in the real control theory sense, i.e., a control system based on the knowledge of the instantaneous systems state or some collection of observations on the systems state. In this respect, the transfer mode adopted in the network and the level of abstraction at which the route selection is operated can make a big difference in the practical implementation of the control scheme. Dynamic routing strategies have been widely investigated and are indeed applied in circuit-switched telephone networks (see, e.g., [1], [2], [3], [4], [5], [6], [7], [8] and [9]), where the state to be considered is related to the available capacity, the dynamic evolution of the system is considered at the call level, and relatively simplified searches make sense (e.g., alternate routing on double-hop paths if the direct route is congested). Correspondingly, similar techniques carry out to the ATM networking context [1], [10], [11], [12], [13] and [14], where routing is effected, possibly with bandwidth allocation and Call Admission Control, at the call level, under the constraint that cell-level Quality of Service (QoS) guarantees are respected. In another respect, [15], [16], [17], [18], [19] and [20] consider optimization problems in routing based on flow-level information, possibly with QoS constraints, even in the framework of game theory, but without a dynamic control formulation.

Some basic difficulties of optimal dynamic routing in a packet-switching environment, based on the status of the network queues, in the presence of centralized or even decentralized information, are evidenced in [21], [22], [23], [24], [25], [26], [27] and [28]. Actually, two facts become crucial in a large-scale telecommunication network: 1) due to communication delays throughout the network, it may be impossible for a single decision center to gather and process all the information characterizing the state of the network (i.e., the lengths of the queues at the nodes and the amount of information traveling along the links) and to send the nodes new routing variables. An informationally decentralized decision structure must then be sought, in which the nodes act as "the cooperating decision makers of a team". This means that each node makes its decisions on the basis of a "personal" information set (i.e., the lengths of the local queues and, possibly, some data coming from other nodes, typically the neighboring ones), and that it aims at minimizing a cost function that is common to all the decision makers of the team (see [29] for the fundamentals of a team organization). 2) The place where the routing or control strategies are computed represents a critical issue. If the characteristics of the communication network (topology, statistics of input-data flows, transmission costs, etc.) were constant over time, the algorithm to design the routing strategies could be implemented at a single computing center. Instead, if such characteristics may undergo unpredictable changes, the availability of a distributed algorithm, enabling the nodes to "adapt locally" the routing strategies on the basis of local information, constitutes an attractive property.

These points will be addressed in the present paper, with respect to the problem of optimal dynamic routing in a store-and-forward packet switching network. The communication network is modeled as a graph, in which a set of nodes are connected through a set of links. The links cannot be overloaded by traffic beyond their finite capacities. The routing problem consists of directing packets from the nodes, where they originate, to their destinations, in such a way as to minimize a given cost function. Whenever the flows of packets entering the communication network vary over time, the nodes may be requested to modify the amount of information to be sent to their neighbors in real time. In this case, a dynamic routing problem arises, which is here addressed by using a receding horizon (RH) control scheme: a discrete-time model is dealt with and, at any time instant, a finite horizon (FH) cost is minimized. Only the control actions relative to the current time instant are applied. The solution of a sequence of FH optimal control problems ensures the routing of the traffic flows on the links over an infinite horizon (IH).

As to the solution of the FH optimal control problem, we shall consider the results reported in [30] and [31], where the problem of clearing the queues at the nodes over a finite horizon was addressed. We define the dynamic routing problem as a team optimization one, by introducing a stochastic discrete-time version of the model presented by Segall in [24] (based on a dynamic system, in which the queue lengths at the nodes are the state variables) and reconsidered in [25], by adding various types of delay (e.g., processing and propagation delays) to the queueing ones. Segall's model was addressed in [26], where a centralized routing algorithm was studied to minimize the total amount of messages waiting in the queues of the network nodes. This algorithm, however, becomes too complex for a practical application when the number of nodes increases. Another centralized algorithm, based on Segall's model, was proposed in [27]. The method exploits a geometrical interpretation of the optimal state trajectory, but it does not seem easily applicable to the case of multiple destinations and of stochastic inputs to the network.

Computationally distributed routing algorithms were considered by Sarachik and Özgüner [28] (their algorithm, however, is valid only for a single destination in the network) and by Iftar and Davison [25], who presented a routing controller that guarantees the clearing of the queues at the nodes in the absence of external inputs, and keeps the lengths of the queues limited as the external message arrival rates are bounded by certain quantities. Our method shares some points with Gallager's distributed algorithm [32]. This algorithm addresses a quasi-static routing context, in which each node constructs its routing strategy on the basis of periodic updating information received from the neighboring nodes. Our team theory approach provides a suitable framework for the correct statement of an informationally decentralized FH optimization problem, but it enables one to determine the optimal control strategies analytically only in few cases, typically under LQG (linear system, quadratic cost, Gaussian disturbances) assumptions and for teams characterized by partially nested information structures (see [33] for general issues related to this point). This drawback and the need for a computationally distributed algorithm lead us to approximate the original team optimal control problem, which is stated in terms of functional optimization, to a nonlinear programming one. This is accomplished by assigning fixed-structure control strategies to each decision maker acting at a network node, in which a certain number of parameters have to be optimized (throughout the paper, the terms "routing" and "control" will be considered as synonyms). According to the time invariance of the adopted model, the approximate routing functions relative to the first FH control stage are used as stationary control functions to manage the routing flows on the infinite horizon.

Various fixed-structure control strategies can be used (i.e., linear combinations of algebraic or polynomial basis functions, nonlinear approximators like feed-forward neural networks, radial basis functions, linear combinations of sinusoidal functions with variable frequencies and phases, etc.). How to choose a nonlinear approximator (which benefits in general from better approximation capabilities than those of traditional linear ones) for solving a given functional optimization problem is a most important but still open issue. We have chosen feed-forward neural networks and optimized their parameters by a stochastic approximation algorithm. Such a choice has been greatly motivated by successful results obtained in solving highly nonlinear optimal control problems [34], [35], [36] and [37]. The technique presented in the paper is the same already employed in [38]; however, the model has been modified to better reflect the case of packet network routing, and examples related to this case are explicitly reported.

MAIN NOTATIONS

$\mathcal{C} = (\mathcal{N}, \mathcal{L})$	directed graph with a set $\mathcal{N}$ of N nodes and a set $\mathcal{L}$ of oriented links describing the communication network.
DM_i	decision maker acting at node $i \in \mathcal{N}$.
$\mathcal{P}(i)$	set of nodes that are upstream neighbors of node i.
$\mathcal{S}(i)$	set of nodes that are downstream neighbors of node i.
$\mathcal{N}^i = \mathcal{N} \setminus \{i\}$	
$\tilde{\mathcal{S}}(i) = \mathcal{S}(i) \bigcup \{i\}$	
C_{ij}	capacity of link $(i,j) \in \mathcal{L}$.
p^d_{ki}	delay of messages with

	destination d routed from node k to node i.
$b_{ij}^{d}\left(t\right)$	discrete-time continuous variable specifying the length of the queue of messages at time t, at node i routed to link (i,j), whose destination is node d.
$f_{ij}^{d}\left(t\right)$	discrete-time continuous variable representing the traffic flow on link (i,j) starting from node i in the time interval $[t,t+1]$ with destination d.
$f_{ij}\left(t\right)$	overall traffic flow on link (i,j) starting from node i in the time interval $[t,t{+}1]$.
$r_{i}^{d}\left(t\right)$	stochastic external input flow entering node i in the time interval $[t,t+1]$ with destination d.
$u_{ij}^{d}\left(t\right)$	control variable representing the portion of the traffic with destination d arriving at node i in the time interval $[t,t+1]$ that is routed from node i to node $j\in\mathcal{S}\left(i\right)$.
$\underline{x}(t)$	state vector of the communication network.
$\underline{I}_i(t)$	personal information vector of DM_i at time t.
$\underline{u}_i(t)=\underline{\gamma}_{it}[\underline{I}_i(t)]$	control function of DM_i at time t.
$\underline{u}_i^{*}=\underline{\gamma}_{it}^{*}[\underline{I}_i(t)]$	optimal control function of DM_i at time t.
$\underline{\bar{u}}_i(t)=\underline{\bar{\gamma}}_i[\underline{I}_i(t),\underline{w}_i(t)]$	neural approximator (i.e., neural network) used by DM_i at time t.
$\underline{\hat{u}}_i(t)=\underline{\hat{\gamma}}_i[\underline{I}_i(t),\underline{w}_i(t)]$	neural control function of DM_i at time t.
$\underline{\hat{u}}_i^{*}(t)=\underline{\hat{\gamma}}_i[\underline{I}_i(t),\underline{w}_i^{*}(t)]$	optimal neural control function of DM_i at time t.

II. A DISCRETE–TIME MODEL FOR THE COMMUNICATION NETWORK

Let us consider a communication network, $\mathcal{C}=(\mathcal{N},\mathcal{L})$, consisting of a connected directed graph with a set $\mathcal{N}$ of N nodes and a set $\mathcal{L}$ of oriented links. At each node $i\in\mathcal{N}$, an input flow may enter the network. Each message has a destination node $d\in\mathcal{N}$. Messages are absorbed as soon as they arrive at their destination nodes.

Let us denote by $\mathcal{S}(i)$ the set of nodes (whose cardinality is $|\mathcal{S}(i)|$) that are downstream neighbors of node i, i.e., the set of nodes j for which a directed link (i,j) exists. At each node $i\in\mathcal{N}$ there are $|\mathcal{S}(i)|$ buffers (one for each link (i,j), $j\in\mathcal{S}(i)$) in which messages are stored once they are routed to a node $j\in\mathcal{S}(i)$. Therefore, the network has to be "connected", i.e., each node of the network must be reachable from each other node. (This simplifies the models presented later on; some minor changes allow the number of destination nodes to be smaller than N.) We also assume that the routing tables on the nodes are updated (synchronously throughout the network), at discrete periodic instants $0,1,\ldots,T-1$ (the sampling period is taken to be unity) and let them be the control variables for our dynamic system.

Up to this point, our model may be considered as the discrete–time version of the continuous–time one proposed by Segall in [24] and used in subsequent works. Following [25], we also take into account the various possible delays (besides the queueing ones) that may occur in a real communication network. More specifically, we denote by p_{ij}^{d} the total delay in transmitting a message at node i with destination d (i.e., the time between starting and ending the transmission of a message), in propagating it on the link joining node i to node j, and in processing the message at node j (i.e., in identifying its destination, inserting it in the queue of messages with destination d, and performing the routing computations). We assume that p_{ij}^{d} can be rounded off to an integer, that is, to a multiple of the sample period. Then, we have the following network model:

Dynamics of Model $\mathcal{M}$

$$b_{ij}^{d}(t+1)=b_{ij}^{d}(t)+\left[r_i^{d}(t)+\sum_{k\in\mathcal{P}(i)}f_{ki}^{d}(t-p_{ki}^{d}-1)\right]\cdot u_{ij}^{d}(t)-f_{ij}^{d}(t)\,,\qquad (i,j)\in\mathcal{L},\ d\in\mathcal{N}^i,\ t=0,1,2,\ldots \quad (1)$$

where

$$f_{ij}(t)=g\left[\sum_{d\in\mathcal{N}^i}b_{ij}^{d}(t),C_{ij}\right]\triangleq\min\left[\sum_{d\in\mathcal{N}^i}b_{ij}^{d}(t),C_{ij}\right],\qquad (i,j)\in\mathcal{L},\ t=0,1,2,\ldots \quad (2)$$

$$f_{ij}^{d}(t)=f_{ij}(t)\cdot\frac{b_{ij}^{d}(t)}{\sum\limits_{d'\in\mathcal{N}^i}b_{ij}^{d'}(t)}\,,\qquad (i,j)\in\mathcal{L},\ d\in\mathcal{N}^i,\ t=0,1,2,\ldots \quad (3)$$

subject to the following constraints:

$$u_{ij}^{d}(t)\geq 0\,,\qquad (i,j)\in\mathcal{L},\ d\in\mathcal{N}^i,\ t=0,1,2,\ldots \quad (4)$$

$$\sum_{j\in\mathcal{S}(i)}u_{ij}^{d}(t)=1\,,\ (i,j)\in\mathcal{L},\ d\in\mathcal{N}^i,\ t=0,1,2,\ldots \quad (5)$$

Remark 1. Eqs. (2) indicate that the buffers are served at the maximum allowable rate. Actually, in the present formulation we have chosen to forward the buffer contents at time t in the interval [t, t+1], i.e., the amount of traffic arriving in the interval [t, t+1] will be served in the next interval. Eqs. (3) state that, in the case of saturation, the amount of traffic for each destination is chosen proportionally to the buffer contents relative to the destination itself.

Remark 2. Note that $\underline{x}(t) \stackrel{\triangle}{=} \mathrm{col}\,[b_{ij}^d(t),\,(i,j)\in\mathcal{L},\,d\in\mathcal{N}^i;\,f_{ki}^d(t-\tau),\,i\in\mathcal{N},\,k\in\mathcal{P}(i),\,d\neq i,\,k\neq d,\,\tau=1,\ldots,p_{ki}^d]$ plays the role of a state vector for model $\mathcal{M}$. Note also that in (1) we have assumed that the external inputs $r_i^d(t)$ do not incur any processing delay. This enables us to simplify the notation. Possible delays in $r_i^d(t)$ would not be influenced by the control variables, hence they would not have to appear in the cost function defined below.

The model stated above corresponds to a datagram network with multicommodity flows and bifurcated routing (as happens in minimum average delay routing problems [39], where packets belonging to the same traffic flow may be spread over multiple paths toward the destination. This possibility, which would be desirable to achieve minimum delay and maximum throughput (at least in an open queueing network, disregarding flow control), is most often avoided, in order to preserve the order of packets in the flow. As a matter of fact, in the presence of best effort TCP traffic, splitting the flow increases the reordering burden at the destination and, in the presence of large differences in delay jitter over multiple paths, might give rise to retransmissions; at any rate, it should be certainly avoided in the case of QoS routing of real-time flows (RTP/UDP). If we want to enforce the requirement that all packets to the same destination (note that, in a more general setting, the superscript *d* might indicate a given traffic class, i.e., a destination and/or a flow with specified QoS), we can do so in our model by having constraint (5) be satisfied by a single u_{ij}^d equal to 1 for each given destination *d*. Formally, this can be obtained through the enforcement of the additional constraint

$$u_{ij}^d(t)\cdot u_{il}^d(t)=0\,,\qquad i\in\mathcal{N};\,j,\,l\in\mathcal{S}(i);\,j\neq l;\,d\in\mathcal{N}^i;\,t=0,1,2,\ldots \tag{6}$$

Cost function
We want to minimize the IH weighted traffic cost

$$J_{IH}=\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\left[\sum_{(i,j)\in\mathcal{L}}\sum_{d\in\mathcal{N}^i}\alpha_{ij}^d\,b_{ij}^d(t)+\right.$$
$$\left.+\sum_{(i,j)\in\mathcal{L}}\sum_{d\in\mathcal{N}^i}\sum_{\tau=\max(0,t-p_{ij}^d)}^{t-1}\beta_{ij}^d\,f_{ij}^d(t-\tau)\right] \tag{7}$$

where α_{ij}^d and β_{ij}^d, $(i,j)\in\mathcal{L},\,d\in\mathcal{N}^i$, are positive weight constants.

Remark 3. The presence of the weight coefficients α_{ij}^d and β_{ij}^d allows the cost (11) to take into account a wide variety of practical situations. If such coefficients are set equal to one, the cost functions give the total time spent by the messages at the nodes and on the links of the communication network. The superscripts d in α_i^d and β_{ij}^d may set different priorities on the messages sent to the various destinations. Finally, the pairs i,j in the coefficients β_{ij}^d enable one to associate possible costs that have to be paid to convey messages through the link joining node i to node j. This flexibility may be exploited to model explicitly the case of QoS routing, which is currently the subject of further investigation. In this respect, we may note that our approach might be extended also by explicitly considering, on a shorter decision time scale than the one we are using for routing, the scheduling of multiple flows over the link *(i,j)*, in order to satisfy given QoS requirements (e.g., in a DiffServ or MPLS context). This possibility would give rise to a joint optimization problem of QoS routing and scheduling over multiple time scales, which constitutes a challenging research task.

III. Statement of the Receding Horizon Dynamic Routing Problem

In our decentralized framework, we suppose that each DM makes its routing decisions on the basis of a *personal information set* $\underline{I}_i(t)$ which includes the lengths of the local queues and (possibly) some information received from the other DMs, typcally the neighboring ones. As an example, let us consider an information structure in which each node receives the lengths of the queues of the downstream neighbors with one step of delay. In this case, the personal information sets take on the form

$$\underline{I}_i(t)=\mathrm{col}\big[b_{ij}^d,d\in\mathcal{N}^i,\;j\in\mathcal{S}(i);\;b_{jk}^d(t-1),\,j\in\mathcal{S}(i),\,k\in\mathcal{S}(j),\,d\in\mathcal{N}^j,\big]\,,\qquad i\in\mathcal{N},t=0,1,2,\ldots \tag{8}$$

In general, the control strategies take on the form

$$u_{ij}^d(t)=\gamma_{ijt}^d\left[\underline{I}_i(t)\right],\qquad i\in\mathcal{N}\;j\in\tilde{\mathcal{S}}(i),\,d\in\mathcal{N}^i,\,t=0,1,2,\ldots \tag{9}$$

Let $\underline{r}_\infty \stackrel{\triangle}{=} \mathrm{col}\,[\underline{r}_i^d(t),\,i\in\mathcal{N},\,d\in\mathcal{N}^i,\,t=0,1,2,\ldots]$. We can now state the following

Problem $\mathbf{R}_{IH}$. *Find the control strategies (9) that minimize the expected cost* $\underset{\underline{x}(0),\underline{r}_\infty}{\mathrm{E}}(J_{IH})$.

□

To face Problem R_{IH} we adopt a receding horizon technique. To do this, we first have to fix a finite control horizon T. Let $\underline{r}_i(t) \stackrel{\triangle}{=} \text{col}\left[r_i^d(t+s),\ s=0,1,\ldots,T-1,\ d\in\mathcal{N}^i\right]$, and $\underline{r}(t) \stackrel{\triangle}{=} \text{col}\left[\underline{r}_i(t), i\in\mathcal{N}\right]$. All stochastic variables are characterized by a given probability density function, $p\left[\underline{x}(t),\underline{r}(t)\right]$. It is not necessarily required that the stochastic variables at time t to be mutually independent. Let us now consider Model $\mathcal{M}$, and define the following FH cost function

$$J_{FH}(t) = \sum_{s=1}^{T}\left[\sum_{(i,j)\in\mathcal{L}}\sum_{d\in\mathcal{N}^i}\alpha_{ijs}^d\, b_{ij}^d(t+s) + \sum_{(i,j)\in\mathcal{L}}\sum_{d\in\mathcal{N}^i}\sum_{\tau=\max(0,s-p_{ij}^d)}^{s-1}\beta_{ijs}^d\ f_{ij}^d(t+s-\tau)\right], \quad t=0,1,2,\ldots \quad (10)$$

where $\alpha_{ijs}^d = \alpha_{ij}^d$ and $\beta_{ijs}^d = \beta_{ij}^d$ for $s=0,1,\ldots,T-1$; $\alpha_{ijT}^d > \alpha_{ij}^d$ and $\beta_{ijT}^d > \beta_{ij}^d$ give rise to a suitable "final cost". The presence of this term in the formulation of the FH optimal control problem results to be particularly useful as we want to use a RH control scheme. In the centralized case, this fact in pointed out in [36], where the final cost is essential to prove the stability properties of the RH regulator.

The cost $J_{FH}(t)$ may also be written in the following form (which will be handled more easily than expression (10) in deriving the control strategy approximating the optimal one)

$$J_{FH}(t) = \sum_{s=1}^{T}\sum_{(i,j)\in\mathcal{L}}\sum_{d\in\mathcal{N}^i}\alpha_{ijs}^d\, b_{ij}^d(t+s) + \sum_{s=0}^{T-1}\sum_{(i,j)\in\mathcal{L}}\sum_{d\in\mathcal{N}^i}\left(\sum_{\tau=s+1}^{\min(s+p_{ij}^d,T)}\beta_{ij\tau}^d\right) f_{ij}^d(t+s), \quad t=0,1,2,\ldots \quad (11)$$

For every time instant $t=0,1,2,\ldots$ we can now state the following finite horizon optimization problem

Problem R$_{FH}(t)$. *Find the optimal control strategies*

$$u_{ij}^{d*}(t+s) = \gamma_{ijs}^{d*}[\underline{I}_i(t+s),\, t], \quad i\in\mathcal{N},\ j\in\tilde{\mathcal{S}}(i),\ d\in\mathcal{N}^i,\ s=0,1,\ldots,T-1 \quad (12)$$

that minimize the expected cost $\underset{\underline{x}(t),\underline{r}(t)}{\mathrm{E}}[J_{FH}(t)]$. *($J_{FH}(t)$ can be given by the cost (10) or (11)).*

□

Owing to the time invariance of Model $\mathcal{M}$, from here on, we shall look for FH control strategies that do not depend on the time instant t, but only on the stage s. Then we shall drop the index t from equation (12) and look for a (single) sequence of strategies γ_{ijs}^d, $s=0,1,\ldots,T-1$ inside the finite horizon $[0,T]$. Moreover, again stemming from the time–invariance of Model $\mathcal{M}$, we shall consider $t=0$ as a generic time instant to obtain the strategies (12), and remove the index t from $J_{FH}(t)$ and from "Problem R$_{FH}(t)$": we shall simply write J_{FH} and "Problem R$_{FH}$."

In our RH framework, the control strategies corresponding to the first stage of FH optimal control problem, will be used as time–invariant routing startegies for each DM, i.e.

$$u_{ij}^d(t) = \gamma_{ij0}^{d*}[I_i(t)], \quad i\in\mathcal{N},\ j\in\tilde{\mathcal{S}}(i),\ d\in\mathcal{N}^i,\ t=0,1,2,\ldots \quad (13)$$

Remark 4. It is worth noting that the decision makers DM_i generate their routing control variables on the basis of personal information sets $\underline{I}_i(t)$, but they cooperate on the accomplishment of a common goal (i.e., the minimization of the same cost). Then, they can be considered as "the cooperating decision makers of a team", as defined in the work by Marschak and Radner on team theory [29]. Let us consider the FH framework. As we said previously, we consider DM_i as a single decision maker, placed at node i and generating control actions at stage s. Equivalently, following the work by Ho and Chu [33], one may consider T decision makers $DM_i(0),\ldots,DM_i(T-1)$. In this context, there would be a team of $N\times T$ decision makers, each generating a control action at a single temporal stage. In the following, when convenient and without risk of ambiguity, we shall adopt the context of Ho and Chu. As is well known, team optimal control problems can be solved analytically in very few cases, typically when i) the problem is LQG and ii) the information structure is *partially nested*. i.e., when any decision maker can reconstruct the information owned by the decision makers the actions of which influenced its personal information. Problem R$_{FH}$ is neither LQG nor, in general, characterized by partially nested information structure. Then, it is even more difficult than Witsenhausen's famous counterexample [40] and there is no hope to solve it analytically.

IV. Reduction of the Functional Team Optimization Problem to a Nonlinear Programming Problem

Let us consider a generic time instant t. To simplify the notation, let us aggregate the functions (12) in the following vectorial form

$$\underline{u}_i(s) = \underline{\gamma}_{is}\left[\underline{I}_i(s)\right], \quad i\in\mathcal{N},\ s=0,1,\ldots,T-1 \quad (14)$$

where $\underline{\gamma}_{is} \stackrel{\triangle}{=} \text{col}\left[\gamma_{ijs}^d,\ j\in\tilde{\mathcal{S}}(i),\ d\in\mathcal{N}^i\right]$, and similar definitions hold for $\underline{u}_i(t)$ and, in the following, for $\hat{\underline{\gamma}}_i$ and $\hat{\underline{u}}_i(t)$. We search for approximating optimal strategies of the form

$$\hat{\underline{u}}_i(s) = \hat{\underline{\gamma}}_i\left[\underline{I}_i(s),\underline{w}_i(s)\right], \quad i\in\mathcal{N},\ s=0,1,\ldots,T-1 \quad (15)$$

where the mappings $\hat{\underline{\gamma}}_i$ take on fixed structures, and $\underline{w}_i(s)$ are finite-dimension vectors of parameters to be determined so as to minimize the expected value of the cost (11).

As approximate routing functions, we shall use multilayer feedforward neural networks with sigmoidal activation functions. In the next section, we shall motivate the choice of such nonlinear approximators and why they are to be preferred to linear ones.

Let us consider the i-th neural network at stage s. Clearly, the input variables are the components of the personal information vector $\underline{I}_i(s)$. Let us denote by $\bar{u}_{ij}^d(s)$, $j \in \tilde{\mathcal{S}}(i)$, $d \in \mathcal{N}^i$, the components of the output vector corresponding to the control variables $\hat{u}_{ij}^d(s)$, that are obtained by the following normalization blocks

$$\hat{u}_{ij}^d(s) = \bar{u}_{ij}^d(s) \,/\, \sum_{k \in \mathcal{S}(i)} \bar{u}_{ik}^d(s)\,,$$
$$i \in \mathcal{N},\ j \in \mathcal{S}(i),\, d \in \mathcal{N}^i,\ s = 0, 1, \ldots, T-1 \quad (16)$$

Such normalization blocks enable us to remove the constraints (5). The use of the sigmoidal functions ensures the fulfilment of the non-negativity constraints (4). Finally, we remove the constraints (6) by adding to the cost (11) penalty functions of the form

$$\rho_i^d(s) = \frac{1}{\sum_{j \in \mathcal{S}(i)} [\hat{u}_{ij}^d(s)]^K} - 1$$
$$i \in \mathcal{N},\ d \in \mathcal{N}^i,\ s = 0, 1, \ldots, T-1 \quad (17)$$

where $K \in R_0^+$. It follows that a new cost function is obtained:

$$J_{FH}\left[\underline{w}, \underline{x}(0), \underline{r}\right] = \sum_{s=1}^{T} \sum_{(i,j) \in \mathcal{L}} \sum_{d \in \mathcal{N}^i} \alpha_{ijs}^d\, b_{ij}^d(s) + \sum_{s=0}^{T-1} \sum_{(i,j) \in \mathcal{L}} \left[\sum_{d \in \mathcal{N}^i} \left(\sum_{\tau = s+1}^{\min(s + p_{ij}^d, T)} \beta_{ij\tau}^d \right) f_{ij}^d(s) \right] + \sum_{s=0}^{T-1} \sum_{i \in \mathcal{N}} \sum_{d \in \mathcal{N}^i} K_L\, \rho_i^d(s) \quad (18)$$

where $\underline{r} \triangleq \underline{r}(0)$, K_L is a positive constant, $\underline{w} \triangleq \mathrm{col}\left[\underline{w}_i(s),\, i \in \mathcal{N}, s = 0, \ldots, T-1\right]$; $\underline{w}_i(s)$ is the vector whose components are given by all the weight and bias coefficients of the neural network of the decision maker $DM_i(t)$ (see the control strategies (15)). Thus the functional optimization Problem R_{FH} has been reduced to the unconstrained nonlinear programming

Problem $\mathbf{R}'_{FH}$. *Find the vector $\underline{w}^*$ that minimizes the expected cost* $\underset{\underline{x}(0), \underline{r}}{\mathrm{E}} \left\{ J_{FH}\left[\underline{w}, \underline{x}(0), \underline{r}\right] \right\}$.

□

Remark 4. It is worth noting that we assign a given structure to the control strategies not to obtain a simplified suboptimal solution, but just because we are unable to derive the optimal solution in analytical form.

V. Approximating properties of neural control strategies

In this section, we shall consider approximate control strategies, taking the form of linear combinations of sigmoidal functions, and: *i)* we show that such approximate routing functions benefit from the approximation properties stated by the Weierstrass Theorem, with respect to the optimal ones (in doing this, we shall take into account that neural control strategies are followed by the normalization blocks (16)); *ii)* we try to understand how complex neural control strategies have to be (i.e., how many parameters they have to contain) in order to approximate the optimal control functions $\underline{\gamma}_{is}^*[\underline{I}_i(s)]$, which solve Problem R_{FH}, to a given degree of accuracy.

As to point *i)*, arguments of the functions $\underline{\gamma}_{is}^*[\underline{I}_i(s)]$ must take their values from compact sets. In order to demonstrate this, we make the following two assumptions:

(A1) The vectors $\underline{x}(0)$ and $\underline{r}(0)$ take their values from given compact sets.

(A2) The optimal control functions $\underline{\gamma}_{is}^*[\underline{I}_i(s)]$ are continuous.

Let us now denote by $\mathcal{A}_i(s)$ the set from which the argument vectors $\underline{I}_i(s)$ of the neural and of the optimal control strategies $\hat{\underline{\gamma}}_i$ and $\underline{\gamma}_{is}^*$ take their values. Under assumptions (A1) and (A2), it is easy to prove that the sets $\mathcal{A}_i(s)$ are compact, as they are generated iteratively from compact sets by continuous functions. We define the functions $\underline{n}^i(\cdot) \triangleq \mathrm{col}\left[\underline{n}_d^i(\cdot),\, d \in \mathcal{N}\right]$, where $\underline{n}_d^i(\cdot)$ denotes the mapping induced by the normalization blocks (16) for a given destination d at node i. In the remaining part of the present section, for the sake of notational simplicity and without loss of generality, we consider some given values of the subscript i and of the temporal stage s, and we drop such indexes both from the neural and optimal control strategies, and also from the sets $\mathcal{A}_i(s)$ and the vectors $\underline{I}_i(s)$ and $\underline{w}_i(s)$. Let us assume that the approximating neural routing functions $\hat{\underline{\gamma}}$ contain only one hidden layer composed of ν neural units and that the output layer is composed of linear activation units. Denote such functions by $\hat{\underline{\gamma}}^{(\nu)}(\underline{I}, \underline{w})$. We can now state the following proposition (proved in [41]).

Proposition 1. *Assume that Problem 2 has an optimal solution $\underline{\gamma}^*(\underline{I})$ and let Assumptions (A1) and (A2) be verified. Then, for every $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$, there exist an integer ν and a weight vector $\underline{w}$, (i.e., a neural control strategy $\hat{\underline{\gamma}}^{(\nu)}(\underline{I}, \underline{w})$) such that*

$$\left\| \underline{n}\left[\hat{\underline{\gamma}}^{(\nu)}(\underline{I}, \underline{w})\right] - \underline{\gamma}^*(\underline{I}) \right\| < \varepsilon, \quad \forall\, \underline{I} \in \mathcal{A} \quad (19)$$

□

Like the Weierstrass theorem involving algebraic or trigonometric polynomials, Property A2 shows that the errors due to the introduction of the neural functions can be arbitrarily small, provided that a sufficiently large number ν of neural units are used.

In general, results of the type presented in Proposition 1 are very common in approximation theory. More specifically, Proposition 1 states that the functions implemented by means of feedforward neural networks are *dense* in the space of continuous functions (see, for example, the results given in [42]); in a sense, this can be considered as a *necessary condition* that every approximation scheme should satisfy. However, such results by themselves are not very useful from an operational point of view, in that they do not provide any information about the rate of convergence of the approximation scheme, that is, about the rate at which the approximation error decreases as the number of parameters of the approximating structure (i.e., the number of hidden units or, equivalently, of parameters to be optimized in the neural approximators) increases.

To address this very important issue, we now apply Barron's results on neural approximation [43]. To this end, let us introduce approximating networks that differ slightly from the ones defined to state Proposition 1. The new networks are the parallel of μ single-output neural networks of the type previously described (i.e., containing a single hidden layer and linear output activation units), where μ is the dimension of the control vector generated by the neural routing strategy. Then each network generates one of the μ components of the control vector. For every l such that $1 \leq l \leq \mu$, denote by $\hat{\gamma}_l^{(\nu_l)}(\underline{I}, \underline{w}_l)$ the input-output mapping of the l-th of such networks, where ν_l is the number of neural units in the hidden layer and $\underline{w}_l$ is the weight vector. Then, denote by $\hat{\underline{\gamma}}^{(\underline{\nu})}(\underline{I}, \underline{w})$ the input-output mapping of the parallel of the networks $\hat{\gamma}_l^{(\nu_l)}(\underline{I}, \underline{w}_l)$, where $\underline{w} \triangleq \mathrm{col}(\underline{w}_l,\, l = 1, \ldots, \mu)$ and $\underline{\nu} \triangleq \mathrm{col}(\nu_l,\, l = 1, \ldots, \mu)$.

In order to characterize the ability of the functions $\hat{\gamma}_l^{(\nu_l)}(\underline{I}, \underline{w}_l)$ to approximate the functions $\gamma_l^*(\underline{I})$, we introduce the integrated square error $\int_{\mathcal{A}} \left|\gamma_l^* - \hat{\gamma}_l^{(\nu_l)}\right|^2 \sigma(\mathrm{d}\,\underline{I})$, evaluated on the domain $\mathcal{A}$ (σ is a probability measure). We assume such a domain to contain the origin. Now we need to make some smoothness assumptions on the optimal control functions γ_l^* to be approximated. Following [43], we assume that such functions are characterized by a bound to the average of the norm of the frequency vector weighted by their Fourier transform. However, the functions γ_l^* have been considered on the domain $\mathcal{A}$, which may be a subset of the space $\mathbb{R}^n$, where $n \triangleq \dim(\underline{I})$. If this occurs, in order to introduce the Fourier transforms, we need "to extend" the functions γ_l from domain $\mathcal{A}$ to $\mathbb{R}^n$. Toward this end, we define the functions $\overline{\gamma}_l : \mathbb{R}^n \to \mathbb{R}$ that coincide with $\gamma_l(\underline{I})$ on $\mathcal{A}$. Finally, we define the class of functions

$$G_{c_l} \triangleq \left\{ \overline{\gamma}_l \quad \text{such that} \quad \int_{\mathbb{R}^n} |\underline{\omega}|\; |\Gamma_l(\underline{\omega})|\; \mathrm{d}\underline{\omega} \,\leq\, c_l \right\} \tag{20}$$

where $\Gamma_l(\underline{\omega})$ is the Fourier transform of $\overline{\gamma}_l$ and c_l is some finite positive constant. Then, we can state the following (see [41] for the proof)

Proposition 2. *Assume that Problem R_{FH} has an optimal solution $\underline{\gamma}^*(\underline{I})$, and further assume that $\gamma_l^* \in G_{c_l}$, $l = 1, \ldots, \mu$, for some finite positive scalars c_l. Then, there exist positive integers $\bar{\nu}_l$, $l = 1, \ldots, \mu$, such that for every probability measure σ and for every $\nu_l \geq \bar{\nu}_l$, $l = 1, \ldots, \mu$, there exist weight vectors $\underline{w}_l$, $l = 1, \ldots, \mu$ (i.e., μ neural routing functions $\hat{\gamma}_l^{(\nu_l)}(\underline{I}, \underline{w}_l)$) and positive scalars β and c'_l, $l = 1, \ldots, \mu$ such that*

$$\int_{\mathcal{A}} \left\| \underline{n}\left[\hat{\underline{\gamma}}^{(\underline{\nu})}(\underline{I}, \underline{w})\right] - \underline{\gamma}^*(\underline{I}) \right\|^2 \sigma(\mathrm{d}\,\underline{I}) \leq \beta \sum_{l=1}^{\mu} \frac{c'_l}{\nu_l} \tag{21}$$

where $c'_l = (2rc_l)^2$. r is the radius of the smallest closed sphere (centered in the origin) containing $\mathcal{A}$.

□

It is worth noting that, in a sense, Proposition 2 specifies quantitatively the content of Proposition 1. More specifically, with reference to the l–th component $\gamma_l^*(\underline{I})$ of any control strategy $\underline{\gamma}^*(\underline{I})$, the number of parameters required to achieve an integrated square error of order $O(1/\nu_l)$ is $O(\nu_l n)$, which grows linearly with n, i.e., the dimension of the input vector of the neural network. It is now reasonable to wonder whether such a property is peculiar to neural approximators or is shared by traditional linear approximation schemes (like polynomial and trigonometric expansions) as well as by other classes of nonlinear approximators.

As to linear approximators, in [43] it is shown that, in the class G_{c_l}, functions to be approximated exist for which there is no possibility of choosing ν_l fixed basis functions such that linear combinations of them can achieve an integrated square error of lower order than $(1/\nu_l)^{2/n}$. The presence of $2/n$ instead of 1 in the exponent of $1/\nu_l$ may then give rise to the phenomenon of the "curse of dimensionality". However, such a worst–case performance by linear approximators does not occur for functions characterized by a higher degree of smoothness, like functions with square-integrable partial derivatives of order up to z (hence they belong to Sobolev spaces), provided that z is the least integer greater than $1 + \frac{n}{2}$ (see [44]). Denote such spaces by $W_2^{(z)}$. It can be shown [43] that, for these functions, the integral $\int_{\mathbb{R}^n} |\underline{\omega}||F(\underline{\omega})|\mathrm{d}\,\underline{\omega}$ is finite ($F(\underline{\omega})$ are their Fourier transforms). Then, if c_l is such that $G_{c_l} \supset W_2^{(z)}$ (i.e., $W_2^{(z)}$ is a proper subset of G_{c_l}), neural approximators should behave better than linear ones in the difference set $G_{c_l} \setminus W_2^{(z)}$.

As to nonlinear approximators, approximation properties similar to the ones of the neural mappings described in the paper characterize radial basis functions [45] (for which the centers and the weighting matrices of the radial activation functions can be tuned), linear combinations of trigonometric basis functions [46] (for which the frequencies and phases are adaptable parameters), sums of hinge functions [47] with adaptable hinges, and others. It is worth noting that the aforesaid approximation bound of order $O(1/\nu_l)$ is achieved under smoothness assumptions on the functions to be approximated that depend on the particular approximator considered. For each of such assumptions there are specific spaces the functions to be approximated have to belong to.

VI. Distributed computation of the routing strategies via stochastic approximation

It is well known that gradient descent algorithms with a constant stepsize are particularly suited for distributed computation. Actually, generally speaking, we may think to assign an i-th processor the task of updating the i-th component of the parameter vector $\underline{w}$. Instead, a gradient algorithm using an optimized stepsize, determined with some line-search technique, would require a knowledge of the entire gradient vector (and not a component-wise knowledge of this vector), thus preventing the use of a computationally distributed optimization procedure (for an ample treatment the distributed computation, see [48]). Indeed, in our communication network, distributed computation is a very attractive property, as it enables each decision maker DM_i to compute its optimal control strategy "locally" on the basis of its personal information vector $\underline{I}_i(s)$. Let us now address Problem R'_{FH}, and consider the gradient algorithm

$$\underline{w}^{k+1} = \underline{w}^k - \eta \, \nabla_{\underline{w}} \mathop{\mathrm{E}}_{\underline{x}(0),\underline{r}} J_{FH} \left[\underline{w}^k, \underline{x}(0), \underline{r} \right] , \qquad k = 0, 1, \ldots \quad (22)$$

where η is a fixed stepsize. Unfortunately, in our case, to compute explicitly the expected cost and then its gradient as expressed in (22) is a very hard task. This leads us to compute the "realization" $\nabla_{\underline{w}} J_{FH} \left[\underline{w}^k, \underline{x}(0)^k, \underline{r}^k \right]$ instead of the gradient $\nabla_{\underline{w}} \mathop{\mathrm{E}}_{\underline{x}(0),\underline{r}} J_{FH} \left[\underline{w}^k, \underline{x}(0), \underline{r} \right]$. Then, we consider the updating algorithm

$$\underline{w}^{k+1} = \underline{w}^k - \eta_k \, \nabla_{\underline{w}} J_{FH} \left[\underline{w}^k, \underline{x}(0)^k, \underline{r}^k \right] , \qquad k = 0, 1, \ldots \quad (23)$$

where the index k now denotes both the steps of the iterative procedure and the discrete-time instants at which the vectors $\underline{x}(0)^k, \underline{r}^k$ are generated randomly on the basis of their probability density function $p[\underline{x}(0), \underline{r}]$.

It is worth noting that the probabilistic algorithm (23) is strictly related to the concept of "stochastic approximation". See, for instance, [49] for a description of this method as well for its convergence properties. To ensure (hopefully) the convergence, we take $\eta_k = c_1/(c_2 + k)$, $c1, c2 > 0$, In the simulations performed in Section VII, we also added a "momentum" $\beta \left(\underline{w}^k - \underline{w}^{k-1} \right)$ to (23), as is usually done in training neural networks (β is a suitable positive constant).

Let us now derive the components of the gradient $\nabla_{\underline{w}} J_{FH} \left[\underline{w}^k, \underline{x}(0)^k, \underline{r}^k \right]$, i.e., of the partial derivatives $\partial J_{FH} \left[\underline{w}^k, \underline{x}(0)^k, \underline{r}^k \right] / \partial \underline{w}_i(s)$. From here on, to avoid complicating the equations excessively, we shall not consider the terms (17). To further simplify the notation, in the following we shall drop the index k and simply write J_{FH} instead of $J_{FH} \left[\underline{w}^k, \underline{x}(0)^k, \underline{r}^k \right]$. Let us define the following variables:

$$\lambda_{ij}^d(s) \triangleq \frac{\partial J_{FH}}{\partial b_{ij}^d(s)} , \qquad i \in \mathcal{N}, \, d \in \mathcal{N}^i, \, s = 0, 1, \ldots, T-1 \quad (24)$$

We denote by $\bar{y}_{iji}^d(t)$ the input to $\hat{\underline{\gamma}}_{is}$ corresponding to $b_{ij}^d(t), \, j \in \mathcal{S}(i)$. Similarly, if an input to $\hat{\underline{\gamma}}_{is}$ corresponds to a state variable $b_{kl}^{d'}(t - p_{jk}), \, k \in \mathcal{S}(i), \, l \in \mathcal{S}(k)$, we shall redefine it as $\bar{y}_{kli}^{d'}(t)$. Moreover, $\underline{\bar{y}}_i(s) \triangleq \mathrm{col}[\bar{y}_{iji}^d(s), \; j \in S(i); \, \bar{y}_{kli}^d(s), k \in \mathcal{S}(i), \, l \in \mathcal{S}(k); d \in \mathcal{N}^i]$.

The partial derivatives $\partial J_{FH} / \partial \underline{w}_i(s)$ are obtained by a classical backpropagation (BP) procedure, which, at stage s and for node i, is initialized by the partial derivatives

$$\frac{\partial J_{FH}}{\partial \bar{u}_{ij}^d(s)} = \sum_{k \in \tilde{\mathcal{S}}(i)} \frac{\partial J_{FH}}{\partial \hat{u}_{ik}^d(s)} \frac{\partial \hat{u}_{ik}^d(s)}{\partial \bar{u}_{ij}^d(s)}, \qquad j \in \mathcal{S}(i), d \in \mathcal{N}^i$$

and allows the computation of $\partial J_{FH} / \partial \underline{\bar{y}}_i(s)$. We have

$$\frac{\partial \hat{u}_{ik}^d(t)}{\partial \bar{u}_{ij}^d(t)} = \begin{cases} \sum\limits_{\substack{l \in \mathcal{S}(i) \\ l \neq j}} \bar{u}_{il}^d(t) \Big/ \left[\sum\limits_{l \in \mathcal{S}(i)} \bar{u}_{il}^d(t) \right]^2 , \\ \text{if } k = j , \\ -\bar{u}_{ij}^d(t) \Big/ \left[\sum\limits_{l \in \mathcal{S}(i)} \bar{u}_{il}^d(t) \right]^2 , \\ \text{if } k \neq j . \end{cases} \quad (25)$$

Proposition 3.
We have:

$$\frac{\partial J_{FH}}{\partial \hat{u}_{ij}^d(s)} = \lambda_{ij}^d(s+1) \left\{ \left[\sum_{k \in \mathcal{P}(i)} f_{ki}^d(s - p_{ki}^d - 1) \right] + r_i^d(s) \right\},$$

$$i \in \mathcal{N}, \, d \in \mathcal{N}^i, \, j \in \mathcal{S}(i), \, s = 0, 1, \ldots, T-1 . \quad (26)$$

The variables $\lambda_{ij}^d(s)$ can be computed by means of the following equations ($i \in \mathcal{N}$, $d \in \mathcal{N}^i$, $s = 0, 1, \ldots, T-1$)

$$\lambda_{ij}^d(s) = \alpha_{ijs}^d + \lambda_{ij}^d(s+1) + \sum_{d' \in \mathcal{S}(i)} \frac{\partial f_{ij}^{d'}(s)}{\partial b_{ij}^d(s)} \frac{\partial J_{FH}}{\partial f_{ij}^{d'}(s)} +$$

$$+ \frac{\partial J_{FH}}{\partial \overline{y}_{iji}^d(s)} + step(T-s-2) \cdot \sum_{k \in \mathcal{P}(i)} \frac{\partial J_{FH}}{\partial \overline{y}_{ijk}^d(s+1)}$$

$$i \in \mathcal{N},\ d \in \mathcal{N}^i,\ s = 0, 1, \ldots, T-1 \qquad (27)$$

where $\mathrm{step}\,(a) = 1,$ *if* $a \geq 0$ *and* $\mathrm{step}\,(a) = 0,$ *if* $a < 0$ *. The term* $\dfrac{\partial J_{FH}}{f_{ij}^d(s)}$ *in (27) can be obtained as follows:*

$$\frac{\partial J_{FH}}{\partial f_{ij}^d(s)} = \left(\sum_{\tau=s+1}^{min(s+p_{ij}^d, T)} \beta_{ij\tau}^d \right) - \lambda_{ij}^d(s+1) +$$
$$+ step[T - (s + p_{ij}^d + 1)] \cdot \sum_{l \in \mathcal{S}(i)} \lambda_{jl}^d(s + p_{ij}^d + 1) \cdot$$
$$\cdot \hat{u}_{jl}^d(s + p_{ij}^d + 1)\,. \qquad (28)$$

$$\frac{\partial f_{ij}^{d'}(s)}{\partial b_{ij}^d(s)} = \left. \frac{d\, g(b\,, C_{ij})}{d\, b} \right|_{b = \sum_{\tilde{d} \in \mathcal{R}(j)} b_{ij}^{\tilde{d}}(s)} \cdot \frac{b_{ij}^d(s)}{\sum\limits_{\tilde{d} \in \mathcal{N}^i} b_{ij}^{\tilde{d}}(s)} +$$

$$+ g\Big(\frac{b_{ij}^d(s)}{\sum\limits_{\tilde{d} \in \mathcal{N}^i} b_{ij}^{\tilde{d}}(s)}\,, C_{ij} \Big) \cdot$$

$$\cdot \begin{cases} \displaystyle\sum_{\substack{\tilde{d} \in \mathcal{N}^i \\ \tilde{d} \neq d}} b_{ij}^{\tilde{d}}\,(s) \Big/ \left[\sum_{\tilde{d} \in \mathcal{N}^i} b_{ij}^{\tilde{d}}\,(s) \right]^2, & \text{if} \ \ d = d' \\[2ex] \displaystyle - b_{ij}^{d'}\,(s) \Big/ \left[\sum_{\tilde{d} \in \mathcal{R}(j)} b_{ij}^{\tilde{d}}\,(s) \right]^2, & \text{if} \ \ d \neq d' \end{cases}$$

The recursion is initialized by the conditions

$$\lambda_{ij}^d(T) = \alpha_{ijT}^d\,, \quad i \in \mathcal{N},\ d \in \mathcal{R}(j),\ j \in \mathcal{S}(i) \qquad (29)$$

□

Remark 5. As can be deduced from the mechanism of the forward pass, if the above computation is performed by a single processing center, it is not necessary for the components of the vectors $\underline{x}(0), \underline{r}$ to be independent. Actually, the realizations $\underline{x}(0)^k, \underline{r}^k$ can be generated by the center on the basis of the known probability density function $p[\underline{x}(0), \underline{r}]$. On the contrary, if the random vectors $\underline{X}_i \stackrel{\triangle}{=} \mathrm{col}\,[\underline{x}_i(0), \underline{r}_i]$, $i \in \mathcal{N}$, are mutually independent, each routing node i can generate "locally" its realization $\underline{X}_i^k$ at iteration k of the algorithm, thus determining its personal control strategy. Clearly, the foregoing holds true for an off-line computation of the control strategies. For an on-line computation (or adaptation) of such strategies, the random variables are generated by the stochastic environment, hence no knowledge of the probability density functions is required.

Let us now remark once again that the off–line solution of Problem R'_{FH}, allows the determination of the parameter vectors $w_i(s)$, $i \in \mathcal{N}$, $s = 0, 1, \ldots, T-1$, i.e., of the routing strategies $\underline{\gamma}_{is}$, $i \in \mathcal{N}$, $s = 0, 1, \ldots, T-1$, but only $\underline{\gamma}_{is}$, $i \in \mathcal{N}$ will be retained by each DM_i, and used as IH routing functions.

VII. Simulation results

In this section, some preliminary simulation results are presented, to show the effectiveness of the proposed method, on a simple network. The convergence behaviour of the cost, during the optimazation of the neural networks by the gradient descent algorithm (23) has been investigated in [31], [38], also in relation to the choice of the constant K_L. Let us consider the network depicted in Fig. 1, where all links have the same capacity of 1 Mbps and the same propagation delay of 1 ms. The information structure is described in (8).

There are two destinations, corresponding to nodes 5 and 6. Continuous Bit Rate (CBR) traffic inputs enter the routing nodes. This is actually a worst case pattern, as the neural networks have been trained with mutually independent random variables, uniformly distributed between a minimum and a maximum value. The CBR flows are constantly active for all the duration of the simulation, with the following source-destination pairs (s, d): $(1, 5)$, $(1, 6)$, $(2, 5)$, $(2, 6)$, $(4, 5)$, $(4, 6)$, $(6, 5)$. Flows $(1, 6)$, $(2, 6)$ and $(4, 6)$ are kept constant at 300 kbps; flow $(6, 5)$ is kept constant at 800 kbps. Flows $(1, 5)$, $(2, 5)$ and $(4, 5)$ are initially generating at a peak rate of 500 kbps, and are increased by 50 kbps at each successive simulation test. Five different simulation tests have been performed, by using a UDP/IP protocol stack in NS-2. All simulations have a duration of 20 seconds. The dynamic routing strategy is compared with a shortest path adaptive routing algorithm, whose metric is proportional to the average queue length of buffer serving the link, estimated over a time window of 1 second. In all graphics from Fig. 2 to Fig. 10, a higher performance of the dynamic routing strategy is evidenced, in terms of both throughput and delay.

VIII. Conclusions

The "neural" control strategies described in the paper exhibit the following basic features:

- The routing decision makers acting at the nodes generate control actions as the cooperating members of a team: this meets a very realistic requirement for large-scale communication networks.
- The IH routing Problem has been solved by means of a RH technique. The related team functional FH optimization problem has been reduced to a

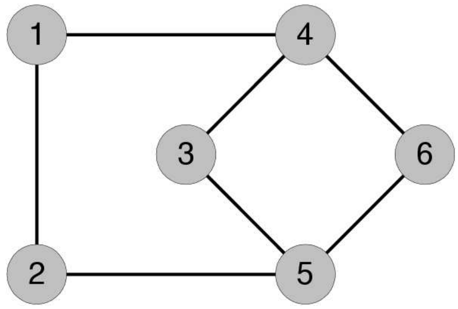

Fig. 1. The network topology used in the test session.

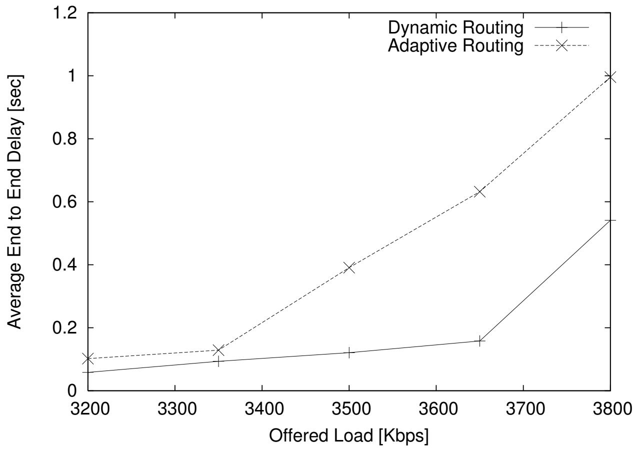

Fig. 2. Average end to end delay of all the packets carried through the network during the simulation tests.

nonlinear programming one that can be solved via a distributed computation scheme. This means that each decision maker can compute (or adapt) its "personal" control strategy "locally" on the basis of a small amount of data, like the lengths of the node queues and possibly the information messages received from some other decision makers, typically the neighboring ones.

- During of the optimization phase, that can be performed off line, t control strategies (one for every FH stage) are computed for each routing node, but only the first one is retained by the DMs.
  During the routing process, a reduced on-line computational effort is requested from each decision maker to compute this stationary control function.

The approximate strategies show a high degree of adaptivity and can face whatever change in the network, including link and node failures. Simulation results allow us to conclude that the proposed method can become a powerful tool for dynamic routing. Moreover, preliminary comparisons with adaptive shortest path routing (with link delay metric), effected under UDP traffic patterns, by means of ns-2 simulations, show encouraging results, with respect to the performance gain obtainable by the dynamic routing strategy. Much work remains to be done to assess the performance, in the case of elastic TCP

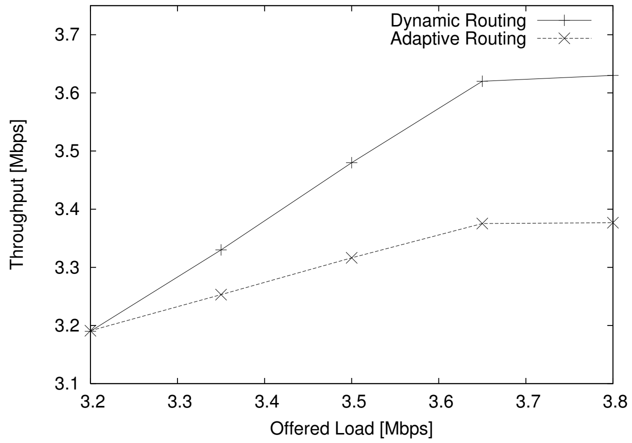

Fig. 3. Total network throughput versus offered load.

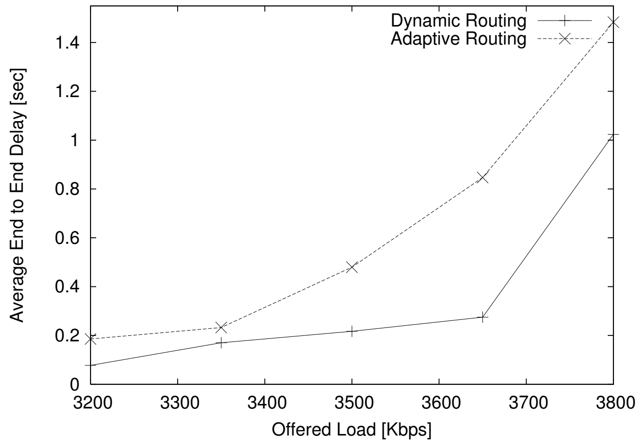

Fig. 4. Average end to end delay of the packets of the flow from node 1 to node 5.

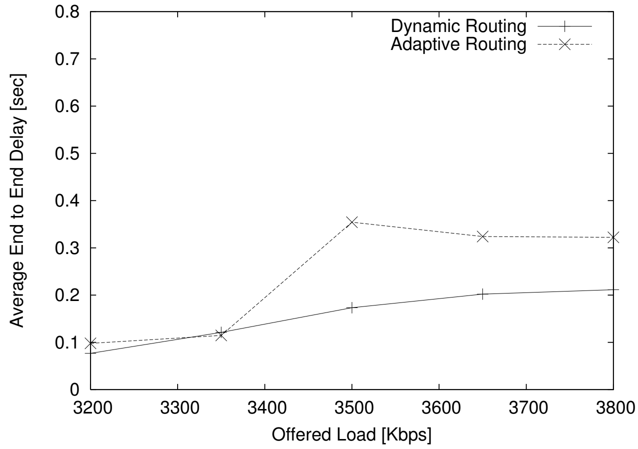

Fig. 5. Average end to end delay of the packets of the flow from node 1 to node 6.

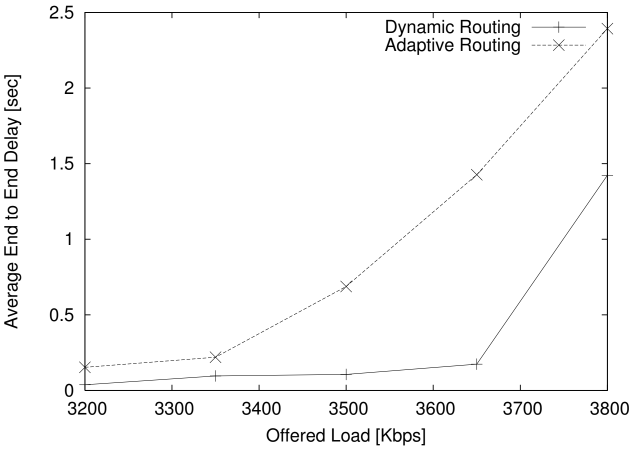

Fig. 6. Average end to end delay of the packets of the flow from node 2 to node 5.

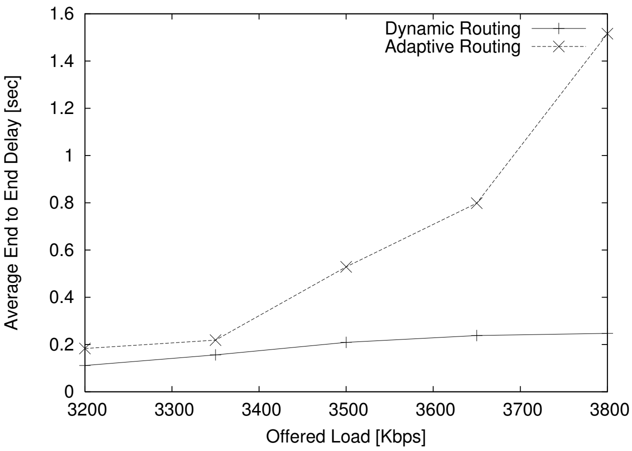

Fig. 7. Average end to end delay of the packets of the flow from node 2 to node 6.

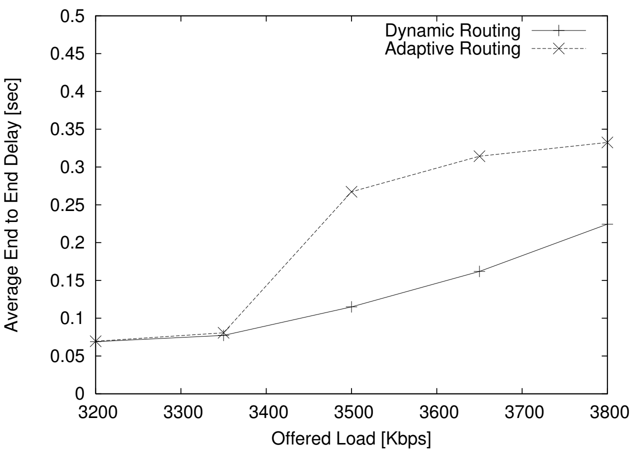

Fig. 8. Average end to end delay of the packets of the flow from node 4 to node 5.

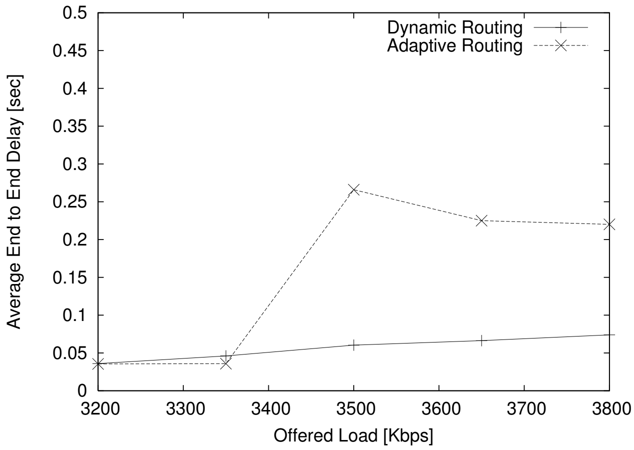

Fig. 9. Average end to end delay of the packets of the flow from node 4 to node 6.

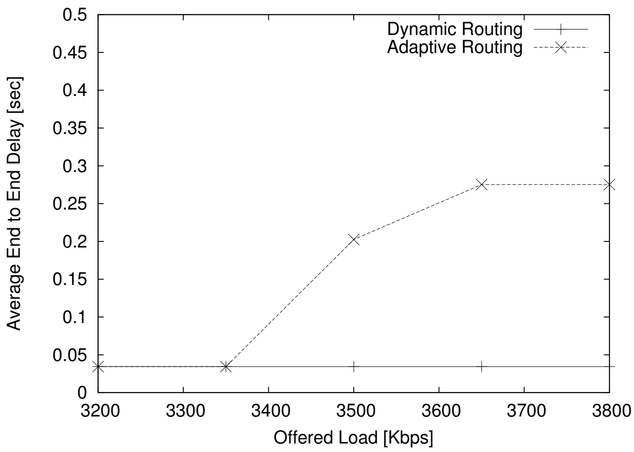

Fig. 10. Average end to end delay of the packets of the flow from node 6 to node 5.

traffic, or under a mix of different (real-time and non-real-time) traffic flows, both in best-effort and in QoS routing conditions. Further refinement is required also on the model, by introducing explicitly the presence of finite buffers and losses, possibly with the inclusion of Active Queue Management (AQM) techniques.

REFERENCES

[1] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*, Springer Verlag, Berlin, 1995.

[2] G. R. Ash, R. H. Cardwell and R. P. Murray, "Design and optimization of networks with dynamic routing,“ *Bell Syst. Tech. J.* , vol. 60, no. 8, pp. 1787-1820, 1981.

[3] G. R. Ash, A. H. Kafker and K. Krishnan, "Servicing and real-time control of networks with dynamic routing,“ *Bell Syst. Tech. J.*, vol. 60, no. 8, pp. 1821-1845, 1981.

[4] D. Mitra, R. J. Gibbens and B. D. Huang, "State-dependent routing on symmetric loss networks with trunk reservations, I,“ *IEEE Trans. Commun.*, vol. 41, p. 10 pages, Feb. 1993.

[5] J. Filipiak, *Modelling and Control of Dynamic Flows in Communication Networks*, Springer-Verlag, Berlin/Heidelberg, 1988.

[6] P. Chemouil, J. Filipiak and P. Gauthier, "Performance issues in the design of dynamically controlled circuit-switched networks,“ *IEEE Commun. Mag.*, vol. 28, no. 10, pp. 90-95, 1990.

[7] A. Girard, *Routing and Dimensioning in Circuit-Switched Networks*, Addison-Wesley, Boston, 1990.
[8] F. P. Kelly, "Routing and capacity allocation in networks with trunk reservation," *Math. Op. Res.*, vol. 15, no. 4, pp. 771-793, 1990.
[9] R. J. Gibbens, F. P. Kelly and S. J. Turner, "Dynamic routing in multiparented networks," *IEEE/ACM Trans. Networking*, vol. 1, no. 2, pp. 261-270, April 1993.
[10] G. Kesidis, J. Walrand and C. J. Chang, "Effective bandwidths for multiclass Markov fluids and other ATM sources," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 424-428, Aug. 1993.
[11] C. A. Courcoubetis, G. Kesidis, A. Ridder, J. Walrand and R. Weber, "Admission control and routing in ATM networks using inferences from measured buffer occupancy," *IEEE Trans. Commun.*, vol. COM-43, no. 2, pp. 1778-1784, 1995.
[12] N. Anerousis and A. A. Lazar, "Virtual Path control of ATM networks with call level quality of service guarantees," *IEEE/ACM Trans. Networking*, vol. 6, no. 2, pp. 222-236, April 1998.
[13] R. Bolla, A. Dalal'ah, F. Davoli, M. Marchese and M. S. Obaidat, "Adaptive distributed admission control and routing in ATM networks," *Internat. J. Commun. Syst.*, vol. 10, pp. 215-225, 1997.
[14] R. Bolla, F. Davoli, M. Marchese and M. Perrando, "QoS-aware routing in ATM and IP-over-ATM," *Computer Commun.*, Special Issue on Performance Evaluation of Telecommunication Systems: Models, Issues and Applications, vol. 24, no. 9, pp. 811-821, May 2001.
[15] Y.A. Korilis, A. A. Lazar and A. Orda, "Achieving network optima using Stackelberg routing strategies," *IEEE/ACM Trans. Networking*, vol. 5, no. 1, pp. 161-173, Feb. 1997.
[16] Y.A. Korilis, A. A. Lazar and A. Orda, "Capacity allocation under noncooperative routing," *IEEE Trans. Automat. Contr.*, vol. 42, no. 3, pp. 309-325, March 1997.
[17] Y.A. Korilis, A. A. Lazar and A. Orda, "Architecting noncooperative networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 7, pp. 1241 1251, Sept. 1995.
[18] R.A. Guérin and A. Orda, "QoS routing in networks with inaccurate information: theory and algorithms," *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 350-364, June 1999.
[19] A. Orda, "Routing with end-to-end QoS guarantees in broadband networks," *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 365-374, June 1999.
[20] G. Fodor, P. Barta, Á. Marquetant, A. Rácz and T. Henk, "Modelling and analysis of routing and resource allocation techniques in multi-service networks," *Internat. J. Commun. Syst.*, vol. 12, no. 2, pp. 103-123, March 1999.
[21] A. Ephremides, P. Varaiya and J. Walrand, "A simple dynamic routing problem," *IEEE Trans. Automat. Contr.*, vol. AC-25, no. 4, pp. 690-693, 1980.
[22] L. Tassiulas and A. Ephremides, "Throughput properties of a queueing network with distributed dynamic routing and flow control," *Adv. Appl. Prob.*, vol. 28, pp. 285-307, 1996.
[23] A. Orda ,R. Rom and M. Sidi , "Minimum delay routing in stochastic networks," *IEEE/ACM Trans. Networking*, vol. 1, no. 2, pp. 187-198, April 1993.
[24] A. Segall, "The modeling of adaptive routing in data–communication networks," *IEEE Trans. Commun.*, vol. 25, pp. 85–95, 1977.
[25] A. Iftar and E. J. Davison, "A decentralized discrete–time controller for dynamic routing," *Internat. J. of Control*, vol. 69, pp. 599–632, 1998.
[26] F. H. Moss and A. Segall, "An optimal control approach to dynamic routing in networks," *IEEE Trans. Automat. Contr.*, vol. 17, pp. 329–339, 1982.
[27] G. J. Stassinopoulos and P. Konstantopoulos, "Optimal congestion control in single destination networks," *IEEE Trans. Commun.*, vol. 33, pp. 792–800, 1985.
[28] P. E. Sarachik and Ü. Özgüner, "On decentralized dynamic routing for congested traffic networks," *IEEE Trans. Automat. Contr.*, vol. 27, pp. 1233–1238, 1982.
[29] J. Marschak and R. Radner, *Economic Theory of Teams*. Yale University Press, 1972.
[30] T. Parisini and R. Zoppoli "Team theory and neural networks for dynamic routing in traffic and communication networks," *Information and Decision Technologies*, vol. 19, pp. 1–18, 1993.
[31] M. Baglietto, T. Parisini, and R. Zoppoli, "Team theory and neural approximators for dynamic routing in communication networks," *American Control Conference*, San Diego, CA, 1999.
[32] R. G. Gallager, "A minimum delay routing algorithm using distributed computation," *IEEE Trans. Commun.*, vol. 25, pp. 73–85, 1977.
[33] Y. C. Ho and K.C. Chu, "Team decision theory and information structures in optimal control, problems - Part 1," *IEEE Trans. Automat. Contr.*, vol. 17, pp. 15–22, 1972.
[34] T. Parisini and R. Zoppoli, "A receding–horizon regulator for nonlinear systems and a neural approximation," *Automatica*, vol. 31, n. 10, pp. 1443–1451, 1995.
[35] T. Parisini and R. Zoppoli, "Neural approximation for multistage optimal control of nonlinear stochastic systems," *IEEE Trans. Autom. Contr.*, vol. 41, pp. 889–895, 1996.
[36] T. Parisini, M. Sanguineti and R. Zoppoli, "Nonlinear stabilization by receding–horizon neural regulators," *Internat. J. of Control*, vol. 70, n. 3, pp. 341–362, 1998.
[37] T. Parisini and R. Zoppoli, "Neural approximations for infinite–horizon optimal control of nonlinear stochastic systems," *IEEE Trans. on Neural Networks*, vol. 9, n. 6, pp. 1388-1408, 1998.
[38] M. Baglietto, T. Parisini, R. Zoppoli, "Distributed-information neural control: the case of dynamic routing in traffic networks," *IEEE Trans. Neural Networks*, vol. 12, no. 3, pp. 485-502, May 2001
[39] D. Bertsekas and R. Gallager, *Data Networks*, 2^{nd} Ed., Prentice-Hall, Englewood Cliffs, NJ, 1992.
[40] H. S. Witsenhausen, "A counterexample in stochastic optimum control," *SIAM J. Control*, vol. 6, no. 1, pp. 131–147, 1968.
[41] M. Baglietto, "Nonlinear approximators for team optimal control problems," *PhD Thesis*, DIST, Univ. of Genoa, 1999.
[42] M. Leshno, V. Ya, A. Pinkus and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, pp. 861–867, 1993.
[43] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Inform. Theory*, vol. 39, pp. 930–945, 1993.
[44] A. Pinkus, *N–Widths in Approximation Theory*. Springer–Verlag, New York, 1986.
[45] F. Girosi, "Regularization theory, Radial Basis Functions and networks," in *From Statistics to Neural Networks. Theory and Pattern Recognition Applications*, V. Cherkassky, J.H. Friedman, and H. Wechsler, Eds., Springer–Verlag, Subseries F, Computer and Systems Sciences, 1994.
[46] L. K. Jones, "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training," *Annals of Statistics*, vol. 20, pp. 608–613, 1992.
[47] L. Breiman, "Hinging hyperplanes for regression, classification, and function approximation," *IEEE Trans. Inform. Theory*, vol. 39, pp. 993–1013, 1993.
[48] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*. Prentice Hall, Englewood Cliffs, NJ, 1989.
[49] B. T. Polyak and Ya. Z. Tsypkin, "Pseudogradient adaptation and training algorithms," *Automation and Remote Control*, vol. 12, pp. 377–397, 1973.