

A TABU SEARCH ALGORITHM FOR DYNAMIC ROUTING IN ATM CELL-SWITCHING NETWORKS

P Cortes^{1†}, J Muñuzuri¹ and J. Fernández¹

¹ Seville University
Grupo Ingeniería Organización.
Escuela Superior Ingenieros, Camino de los Descubrimientos s/n.
Sevilla 41092. SPAIN
†E-mail: pca@esi.us.es
† URL: <http://io.us.es/P.Cortes/>

Abstract

This paper deals with the dynamic routing problem in ATM cell-switching networks. We present a mathematical programming model based on cell loss and a Tabu Search algorithm with short-term memory that is reinforced with a long-term memory procedure. The estimation of the quality of the solutions is fast, due to the specific encoding of the feasible solutions. The Tabu Search algorithm reaches good quality solutions, outperforming other approaches such as Genetic Algorithms and the Minimum Switching Path heuristic, regarding both cell loss and the CPU time consumption. The best results were found for the more complex networks with a high number of switches and links.

Keywords: Tabu Search, dynamic routing, ATM

1. Introduction

The general problem of real time operation in telecommunication networks involves a collection of different complex problems. Consequently, optimisation in real time operation in telecommunication networks arises when communications have to be routed across the network from the origin point to the destination. In this context soft computing can offer additional capabilities with respect to other traditional approaches.

The basic and more traditional problem of real time operation is known as routing and can be undertaken in either a static or dynamic way. Static routing makes use of the routing tables that are constructed by means of the capacity assignment and multicommodity flow problem. In this case the network is analysed offline using deterministic and static demands of traffic. **Dynamic routing** arises when the messages are routed in real time trying to minimise the losses or to minimise the delays experienced by the messages. Several specific problems following on from the basic problem have been tackled. Medhi *et al.* (2000) have considered solution approaches to a multi-hour combined capacity design and routing problem, which arises in the design of dynamically reconfigurable broadband communication networks that use the virtual path concept. They compared a genetic algorithm, a Lagrangian relaxation based subgradient optimisation method, a generalised proximal point algorithm with subgradient optimisation, and a hybrid approach where the subgradient based method is combined with a genetic algorithm. Also, Xu *et al.* (1997) presented a tabu search approach that optimises the link capacities in a dynamic routing telecommunications network. The traffic between any pair of nodes in the network is routed over a one-link direct path or, if no direct capacity is available, over a two-link alternate path. Later, Shyur *et al.* (2001) presented a non-time dependent model to solve the problem of optimising the system of virtual paths using a tabu search approach. Although these two papers make use of tabu search approaches, as ours does, they cannot be directly compared to our paper because they do not use a time horizon with different time scales, as is considered here. More recently, Cortes *et al.* (2002) have presented a genetic algorithm for dynamic routing in ATM networks. The model objective was to minimise cell loss subject to strict delay conditions. Alternatively, Wille *et al.* (2006) showed models and algorithms considering at the same time the dynamics of packet network and the grade of service (GoS) user-layer requirements.

Another problem in real time operation corresponds to *communication admission control*. This admission control is a deterministic and informed decision that is made before a communication is established and is based on whether the required network resources are available to provide suitable GoS for the new communication. Yener *et al.* (1997) presented a paper along these lines, dealing with the cellular call admission control. They define local policies based on genetic algorithms. For the mobile network cases, Sheng *et al.* (2006) successfully tested a threshold-adaptive call admission control based on genetic algorithms for multiclass services.

The *dynamic bandwidth allocation* is a problem by which traffic bandwidth in a shared telecommunications medium can be allocated on demand and evenly between different users of that bandwidth. Pitsillides *et al.* (2000) analysed the performance of the traditional constrained optimisation algorithm compared to a new constrained optimisation genetic algorithm for solving the bandwidth allocation for the virtual paths problem. Comparisons were carried out with respect to throughput, fairness and time complexity. More recently, El-Madbouly (2005) has proposed a genetic algorithm for the design of an optimisation algorithm to achieve the bandwidth allocation of ATM networks.

Recently, the problem of *wavelength-division multiplexing* has arisen as a very efficient method of transmission that consists of the simultaneous transmission of more than one information-carrying channel on a single fibre light side, using two or more light sources of different wavelengths. This is currently one of the most efficient methods for the transmission of mass communication. Specifically, WDM technology has emerged as a promising technology for backbone networks. The set of all-optical communication channels (lightpaths) in the optical layer defines the virtual topology for the upper layer applications. Since the traffic demand of upper layer applications fluctuates from time to time, it is required to reconfigure the underlying virtual topology in the optical layer accordingly. Din (2004, 2007) has used genetic algorithms to solve the optimal multiple multicast problem on wavelength division multiplexing ring networks, without wavelength conversion and the virtual topology configuration transition problem. Also, Le *et al.* (2006) have dealt with the dynamic survivable routing problem, both in optical networks without wavelength conversion and in optical networks with sparse wavelength conversion. They make use of a hybrid algorithm based on the combination of mobile agents technique and genetic algorithms. Finally, Tak *et al.* (2007) have proposed restoration techniques incorporating the concept of k -shortest disjoint paths in survivable WDM optical networks, considering network performance and network costs simultaneously, making use of several soft computing algorithms to do so.

Here, we focus our research on the first set of problems described, specifically on the dynamic routing problem in an ATM cell-switching network. The rest of the paper deals with a brief description of the Asynchronous Transfer Mode (ATM) special characteristics, included in the second section. Once the technological background of the paper has been stated, the third section is devoted to the mathematical programming model of the problem, which is based on a cell loss model with bounded delay conditions. The fourth section presents the Tabu Search used to solve the problem. The fifth section outlines the results and the comparison between Tabu Search, Genetic Algorithms, and a fast heuristic algorithm. Finally the last section describes the main conclusions and our further research.

2. A description of the Asynchronous Transfer Mode

ATM was created to provide faster communications in both public and private networks. ATM can be characterised as a hybrid transfer mode between circuit and packet switched networks. ATM combines circuit and packet switched networks maintaining the channel structures, by means of timeslot division, as in circuit switched, and the packet structure, by means of ATM cells, as in packet switched.

ATM cells are 53 bytes long. These bytes are divided into different length fixed fields. The Virtual Channel Identifier (VCI) and Virtual Path Identifier (VPI) fields specify the circuit and path that each cell should take. The Payload Type Identifier (PTI) field is used to specify the type of information contained in the payload section of the cell. The Cell Loss Priority (CLP) field gives an indication of the severity of the consequences of the loss of this cell in the event that there is insufficient bandwidth in the switch to process all incoming cells. The Header Error Check (HEC) field function is to detect and correct possible errors in the transmission process into any field of the cell header. The payload field is a 48-byte area that contains the user data to be transmitted over the virtual circuit. Figure 1 depicts a Network-Network interface (NNI) cell, i.e., a cell flowing between network switches. Unlike NNI cells, User-Network Interface (UNI) cells substitute the 12-byte VPI field with an 8-byte field, due to the remaining 4 bytes being used for the Generic Flow Control (GFC) field, allowing the flow control that is transmitted from the UNI interface.

LOCATION FOR FIGURE 1

The VPI and VCI are transmitted into the header. These identifiers are only valid for each connection link and identify a logic circuit into the physical link. When the cell arrives at the switch, the VPI and VCI are set and the cell is routed through the output port, changing the identifier when necessary. The virtual circuit (or channel) concept is an abstraction representing the cell unidirectional transport associated to an identifier (VCI). The VCI and the VPI indicate an ordered cell flow associated with a concrete connection. The virtual connection channel (VCC) concept means the connection of different virtual paths between the origin of communication and the destination of communication. The concepts are graphically described in Figure 2.

LOCATION FOR FIGURE 2

Once the first cell arrives at a switch, the communication is routed from an inflow logic channel onto an outflow logic channel. The ATM switches are characterised by the information volume, the rate of bit error, the delay, the cell loss probability and the delay fluctuation. In ATM the information volume being processed and the rate of bit error are conditioned by the technology and the system dimension. The use of high-speed technologies, such as BICMOS or ECL, easily provides hundreds of Mb/s with acceptable error rates.

Cell loss in ATM switches appears when many cells are routed onto the same link, being greater than the buffer size of the switch. The switching delay is the time used by one cell to switch to its destination.

The switches can be classified in accordance with the structure of the switching fabric architecture. The matrix switch is one of the most recommended switches, where a connection matrix establishes the interconnection of any input point with any output point. One of the better delay-performance relations is obtained for matrix switches with queues at the end. This option allows more than one cell to be transmitted in the same timeslot. In order to ensure no cell loss, the transference must be N (number of input ports) times faster than the cell homing rate. Moreover, the queue location at the end prevents the line header from blocking.

The ATM contract must specify the requested bandwidth, as well as the quality of service. The resultant parameters will be negotiated between the network and the user as a function of the network resources. The service categories are: Constant Bit Rate service (CBR), Variable Bit Rate service (VBR), Available Bit Rate service (ABR) and Unspecified Bit Rate service (UBR). CBR service is chosen to maintain real time communications as video or audio, with a very strict delay requirement. CBR service provides a connection with large bandwidth and very low cell loss probability. VBR service is indicated for frame relay traffic. ABR service is used for

unknown characteristics traffic and must be moderately restrictive attending to cell loss and without delay requirement as real time applications. Finally, UBR service uses as much bandwidth as is available.

One additional specific characteristic of the ATM switching mode that very much determines the mathematical modelling of the problem refers to the time scales in ATM. The traditional and widely used Erlang model considers one unique time scale: the connection scale or call scale. Bandwidth traffic requires additional scales. Consequently, ATM needs to consider at least two time scale levels: the call time scale to ensure the call connection level and the cell time scale to ensure the quality specifications with respect to cell loss and message integrity. Thus, flow variables are needed for the cell level scale, and path variables for call routing. In fact, once the first cell is routed onto a virtual circuit, the remaining cells follow the first one through the same circuit, so the path variables must be referred to as the call level scale. Figure 3 depicts the different levels and their interrelations.

LOCATION FOR FIGURE 3

3. A mathematical programming model for the dynamic routing problem in ATM cell-switching networks

ATM networks can be well suited by graphs $G = (N,E)$, with N being the set of switches (nodes) and E the set of links. Messages are sent from communication origin nodes to destination nodes. The messages are divided into cells with message information (payload) and management information (header). These cells are routed onto different switches with the objective of reaching the final destination.

In order to construct a tractable mathematical programming model, we consider some limitations and hypotheses, which are described below:

- We have considered that the cell loss takes place within the switch. In practice, ATM cells can be lost due to line transmission failures, the Usage Parameter Control in the switching network, buffer sizes or switching fabric limitation. The consideration of cell loss only at the switch is not a strong limitation because it represents the very largest part of cell loss in ATM networks.
- We have considered only CBR traffic. In fact, this is not a strong restriction in relation to our objective function. VBR, ABR and UBR traffic are much less restrictive according to cell loss (our objective function) than CBR traffic.
- We have not considered capacity constraints supposing links of extra capacity. This is a very realistic supposition because telecommunication links are over dimensioned due to the prevision of future traffic increments at the planning stage.
- The delay limitations are considered as a heavy constraint of the mathematical model. Most of the research only considers the delay that is taking place within the switch due to the queue and the real switching process, because the remaining sources of delay, such as electric-optic conversion or signal propagation, are second order factors.

The mathematical programming model is defined by the following parameters, variables, data and constraints.

□ Parameters:

- **N** Set of switches (nodes).
- **E** Set of links.

- **M** Set of communication origin-destination pairs, being $O(m)$ origin of communication m and $D(m)$ destination of communication m .
- **H(m)** Set of feasible paths to establish a connection between each origin-destination pair m . This includes only those paths with a feasible number of switches attending to the delay specifications.
- **T** Temporal horizon at call scale. This horizon is composed by call time division $t \in T$. Subsequently, the call time scale is divided into the cell time scale, $\tau \in t$.
- **N(h)** Set of nodes belonging to path h .
- **E(h)** Set of links belonging to path h .
- **B(i)** Set of nodes located before node i .
- **A(i)** Set of nodes located after node i .

□ **Variables:**

- $\mathbf{P}_h^{m,t}$ Binary variable indicating whether the connection between the origin of communication $O(m)$ and the destination of communication $D(m)$ is established by the path h , referred to the call scale $t \in T$.
- $\mathbf{X}_{h,ij}^{m,\tau}$ Continuous flow variable associated to the pair m over the link (i,j) in that specific direction (from i to j), being (i,j) a link of the path h , referred to the cell scale $\tau \in t$. Expressed in cells/slot.
- $\mathbf{l}_{h,ij}^{m,\tau}$ Continuous variable determining the cell loss at the link (i,j) due to the port j into the switch i in the connection path h used by the pair of communication m , in the timeslot $\tau \in t$ (so it is referred to the cell scale). The variable is expressed in cells/slot.
- \mathbf{F}_{ij}^τ Total cell flow variable that should be routed from node i to node j . It is referred to the cell scale $\tau \in t$. Expressed in cells/slot.

It is important to note the difference between the variables F_{ij}^τ and $X_{h,ij}^{m,\tau}$. The former refers to the total flow of cells that should ideally go through the link (i,j) , supposing no cell loss, meanwhile the second variable takes into account the cell loss possibility. Figure 4 depicts an example. In Figure 4 the switches v, w, y, z are before node i , whereas nodes j, k are located after node i . For an instant τ there is a flow of cells from nodes v, w, y, z to node i , the flow from nodes v and z must be routed to node j and the flow from nodes w and y must be routed to the switch k .

LOCATION FOR FIGURE 4

□ **Data:**

- \mathbf{A}_m^τ Demand of the origin-destination pair m in the timeslot $\tau \in t$. It is referred to the cell scale and expressed in cell/slot.
- \mathbf{GoS}^m Grade of service. Maximum number of lost cells for each connection m , measured as cells/slot.

Finally, the mathematical programming model can be formulated as follows:

$$MIN \sum_{t \in T} \sum_{\tau \in t} \sum_{(i,j) \in E} \sum_{m \in M} \sum_{h \in H(m)} P_{h,ij}^{m,t}$$

s.t.

$$(3.1) \quad \sum_{h \in H(m)} P_h^{m,t} = 1 \quad \forall m \in M, \forall t \in T$$

$$(3.2) \quad X_{h,ij}^{m,\tau} + I_{h,ij}^{m,\tau} = \begin{cases} A_m^\tau P_h^{m,t} & \text{if } i = O(m) \\ X_{h,ki}^{m,\tau-1} & \text{if } i \neq O(m) \end{cases}$$

$$\forall \tau \subset t, \forall t \in T, \forall h \in H(m), \forall m \in M, \forall i \in N: \forall (i,j) \in E(h), \forall (k,i) \in E(h), /i,j,k \in N(h)$$

$$(3.3) \quad F_{ij}^\tau = \sum_{m \in M} \sum_{\substack{h \in H(m), \\ (q,i) \in E(h): j \in N(h)}} \sum_{q \in B(i)} X_{h,qi}^{m,\tau-1} + \sum_{m \in M / i=O(m)} A_m^\tau \quad \forall (i,j) \in E, \forall \tau \subset t, \forall t \in T$$

$$(3.4) \quad \sum_{m \in M} \sum_{h \in H(m)} I_{h,ij}^{m,\tau} \geq L_{ij}^\tau (F_{ij}^\tau) \quad \forall i \in N, \forall (i,j) \in E, \forall \tau \subset t, \forall t \in T$$

$$(3.5) \quad \sum_{\tau \in t} \sum_{h \in H(m)} \sum_{(i,j) \in E(h): j \in A(i)} I_{h,ij}^{m,\tau} \leq GoS^m \quad \forall m \in M, \forall t \in T$$

$$(3.6) \quad I_{h,ij}^{m,\tau} \geq 0 \quad \forall (i,j) \in E, \forall m \in M, \forall \tau \subset t, \forall t \in T$$

$$(3.7) \quad X_{h,ij}^{m,\tau} \geq 0 \quad \forall (i,j) \in E, \forall m \in M, \forall \tau \subset t, \forall t \in T$$

$$(3.8) \quad P_h^{m,t} \in (0,1) \quad \forall m \in M, h \in H(m), t \in T$$

Constraint 3.1 is referred to the call scale stating that pair m communication must be routed onto only one path, h . Constraint 3.2 refers to the flow balance equation; the term on the left shows the flow through the link (i,j) belonging to path h used to connect the pair m plus the loss in the buffer j of the switch i . The term on the right shows two situations: in the first situation, node i is the origin of communication, so all the generated traffic is routed onto the path h ; in the second situation, node i is an intermediate node of the path h , so the expression must include the flow sent by the node before the switch i in the path h , i.e. the link (k,i) . This traffic will carry a timeslot delay due to the switching process and includes the losses into the previous switches. This balance equation is imposed at the cell scale. Constraint 3.3 determines the total flow including all the flows homing from nodes before node i and being routed onto the switch j . In constraint 3.4, the cell loss is characterised as $L_{ij}^\tau (F_{ij}^\tau)$ for the output port j relative to the switch i in the instant τ when the arrival rate is the total flow homing to the switch i and flowing to port j , that is F_{ij}^τ . The term on the left reflects the sum for all the origin-destination pairs and their feasible paths for the connection including the link (i,j) , and the term on the right states the losses at the output port j . Finally, constraint 3.5 is bound on the grade of service, so a maximum cell loss is imposed for each path of communication.

The objective function includes the total cell loss for all the pairs of communication, for each path, for each link and for the entire horizon.

4. The Tabu Search algorithm

Tabu Search was first introduced by Glover (1989, 1990). The simple Tabu Search algorithm uses the best improvement local search as the basis for its movements and uses a short-term memory to escape from local minima. Starting from its solution, the algorithm tries to find a better one in its neighbourhood (defined as a set of near solutions that are found by applying an appropriate transformation of the current solution). The algorithm permits moves that result in a degradation of the objective function value to escape from a local minimum, thus preventing

them from being trapped into a local optimum. The algorithm considers some of the previously visited solutions as “tabu” preventing the searching process falling into cycles. Tabu search has been largely used to solve transport problems; it is considered one of the most suitable algorithms to solve a large range of vehicle routing problems. Its applicability to telecommunications has recently been gaining support in scientific literature, as the literature review in Section 1 describes.

In this paper, we propose a Tabu Search algorithm for the dynamic routing problem in ATM cell-switching networks. The Tabu Search algorithm is initialised with a minimum switching path algorithm (MSwP), which assigns the communication to that path with a minimum number of switches where no over-buffering is taking place (the MSwP pseudocode is described in the appendix).

4.1. Solutions encoding

The feasible solutions are encoded, as is shown in Figure 5. The solution is therefore determined by a binary matrix with the following considerations:

- Each solution must contain as many lines as periods of connection. We consider t as the time for an ATM communication (call scale).
- Each of the lines is composed by a number of fields, M , representing the different origin-destination pairs of communication.
- The field associated to each pair is depicted by so many elements as feasible paths exist between that pair of communication. Only one route is feasible for each communication, setting the value to 1.

LOCATION FOR FIGURE 5

Figure 6 illustrates an example regarding the encoding of solutions. It corresponds to the network status for a feasible solution in a hypothetical call time division (period of connection) t_x . The network considers four nodes and all the possible connections between them, consequently giving 12 possible connections between pairs of nodes. Each one has a set of feasible paths. It has to be taken into account that paths could be forbidden due to the delay condition, so not all the connections will necessarily have the same number of feasible paths. We address two specific connections in the example, showing three feasible paths for each of them, whose links (pair of nodes) are detailed. The figure only shows the feasible set of paths for the first two connections, in order to gain clarity. The proposal of the example gives a row for the time period, t_x , corresponding to the specific row of the binary matrix detailed in Figure 5, which should be given as is described in Figure 6. That is, the binary register corresponding to paths h_2 and h_6 must be set to 1, while the remaining registers are set to 0.

LOCATION FOR FIGURE 6

4.2. Cell loss evaluation

The quality of each solution is assessed in terms of the total cell loss. To do so, each feasible solution can be evaluated according to the process described below.

Once the variable $P_h^{m,t}$ has been fixed (following the Figure 5 encoding) the conditions and constraints of the model can be rewritten as follows:

$$(4.1) \quad X_{ij}^{m,\tau} + l_{ij}^{m,\tau} = \begin{cases} A_m^\tau & \text{if } i = O(m) \\ X_{ki}^{m,\tau-1} & \text{if } i \neq O(m) \end{cases}$$

$$\forall \tau < t, \forall m \in M, \forall i \in N: \forall (i,j) \in E(h), \forall (k,i) \in E(h), /i,j,k \in N(h)$$

$$(4.2) \quad F_{ij}^\tau = \sum_{m \in M} \sum_{q \in B(i)} X_{qi}^{m,\tau-1} + \sum_{m \in M / i=O(m)} A_m^\tau \quad \forall (i,j) \in E(h): q \in A(i), \forall \tau < t$$

$$(4.3) \quad \sum_{m \in M} l_{ij}^{m,\tau} \geq L_{ij}^\tau (F_{ij}^\tau) \quad \forall i \in N, \forall (i,j) \in E(h), \forall \tau < t$$

$$(4.4) \quad l_{ij}^{m,\tau} \geq 0 \quad \forall (i,j) \in E, \forall m \in M, \forall \tau < t$$

$$(4.4) \quad X_{ij}^{m,\tau} \geq 0 \quad \forall (i,j) \in E, \forall m \in M, \forall \tau < t$$

By grouping constraints 4.2 and 4.3 into one equation (4.6), the set of constraints of the model is reduced to:

$$(4.5) \quad X_{ij}^{m,\tau} + l_{ij}^{m,\tau} = \begin{cases} A_m^\tau & \text{if } i = O(m) \\ X_{ki}^{m,\tau} & \text{if } i \neq O(m) \end{cases}$$

$$\forall \tau < t, \forall m \in M, \forall i \in N: \forall (i,j) \in E(h), \forall (k,i) \in E(h), /i,j,k \in N(h)$$

$$(4.6) \quad \sum_{m \in M} l_{ij}^{m,\tau} \geq L_{ij}^\tau \left(\sum_{m \in M} \sum_{q \in B(i)} X_{qi}^{m,\tau-1} + \sum_{m \in M / i=O(m)} A_m^\tau \right) \quad \forall i \in N, \forall (i,j) \in E(h): q \in A(i), \forall \tau < t$$

$$(4.7) \quad l_{ij}^{m,\tau} \geq 0 \quad \forall (i,j) \in E, \forall m \in M, \forall \tau < t$$

$$(4.8) \quad X_{ij}^{m,\tau} \geq 0 \quad \forall (i,j) \in E, \forall m \in M, \forall \tau < t$$

The grade of service constraint is removed from the model and is considered as a feasible control of the solutions, discarding those solutions that do not satisfy the grade of service control.

Once the variable $P_h^{m,t}$ is specified, the model evaluation is much easier and the total cell loss can be calculated by a simple evaluation of the objective function. To do so we introduce the following procedure.

Cell Loss evaluation procedure

1. For each τ in t :

1.1. Calculate:
$$F_{ij}^\tau = \sum_{m \in M} \sum_{\substack{q \in B(i) \\ q,i,j \in h}} X_{qi}^{m,\tau-1} + \sum_{m \in M / i=O(m)} A_m^\tau$$

1.2. Calculate:
$$l_{ij}^\tau = \sum_{m \in M} l_{ij}^{m,\tau} = L_{ij}^\tau (F_{ij}^\tau)$$

1.3. The cell loss for each connection is calculated by distributing the total l_{ij}^τ proportionality to the demand of the pair over the link (i,j) . It is therefore calculated as:

$$l_{ij}^{m,\tau} = l_{ij}^\tau \left(\frac{A_m^\tau}{\sum_{m:(i,j) \in h(m)} A_m^\tau} \right)$$

- 1.4. $X_{ij}^{m,\tau}$ is calculated by means of the flow balance constraint (4.5), by substituting the cell loss values obtained in step 1.3.

$$X_{ij}^{m,\tau} + l_{ij}^{m,\tau} = \begin{cases} A_m^\tau & \text{if } i = O(m) \\ X_{ki}^{m,\tau-1} & \text{if } i \neq O(m) \end{cases}$$

$$\forall i \in N: \forall (i,j) \in E(h), \forall (k,i) \in E(h), /i,j,k \in N(h)$$

2. Calculate cell loss:
$$\sum_{t \in T} \sum_{\tau \in t} \sum_{(i,j) \in E(h)} \sum_{m \in M} l_{ij}^{m,\tau}$$
3. Test GoS feasibility:
$$\sum_{\tau \in t} \sum_{(i,j) \in E(h)} l_{ij}^{m,\tau} \geq GoS^m$$

4.3. Tabu Search algorithm characteristics

The Tabu Search implementation is described in this section. Figure 7 depicts the Tabu Search algorithm.

LOCATION FOR FIGURE 7

The algorithm starts from a feasible solution (current solution) that is supplied by the MSwP algorithm. This is a first promising solution.

The algorithm consists of *short-term memory* (STM) and *long-term memory* (LTM) procedures. Both procedures are iteratively run during a maximum number of iterations. STM is processed every iteration while LTM is considered for processing every iteration although it is not always run (as Figure 7 depicts).

The exploration of feasible solutions is carried out by searching the neighbourhood and making a “movement” from one feasible solution to another. Figure 5 depicted the solution encoding structure, and Figure 6 an encoding example. For example of Figure 6, a neighbour solution of the current solution is generated by changing the communication route (path) in a specific pair of communication ($m \in M$) for a specific period of connection ($t \in T$). Figure 8 shows a neighbour solution swapping the bit corresponding to the first connection ($m = I$) and the period t_x . The *neighbourhood* is constructed of all the neighbour solutions of the current solution, that is, for every connection (set M), every possible swap of paths of such connection (set $H(m)$), and every call period (set T).

LOCATION FOR FIGURE 8

Each iteration provides a new solution that is found by making a definitive local movement over the current solution within the neighbourhood. The next solution is selected as the best one within all the possible solutions in the neighbourhood, according to the previous description of the neighbourhood.

In this sense a *chain of solutions* is constituted by all the enchainned adjacent solutions. STM is responsible for the optimal enchainning of these solutions, depicting a *trajectory*. The trajectory is a consequence of the quality of the solutions and a tabu filter depicting an intelligent exploration, meanwhile the LTM periodically analyses whether the trajectory is acceptable or the process should not follow this path and the algorithm should move back to take another more promising path to configure a new trajectory.

STM carries out the generation of a neighbourhood shaped by a set of similar solutions (with only one different route in a specific time instant) and its quality analysis, as well as the identification of the new generated solution shaping a specific trajectory. STM also checks the validity of such a solution, moving to it in case of not having any route in the *Tabu List* or any forbidden route (for example with an expected delay higher than feasible). In case of non-validation, the next most promising neighbour is selected.

The *attributes* of each solution are defined by all the routes linking the origin-destination pairs. The tabu list is composed of all the solutions with a tabu status. The *tabu status* is determined by the set of routes that are assigned as tabu routes. This set is dynamic and can change during the procedure run. The purpose of the tabu list is to memorise a certain number of the most recent moves and to prohibit any repetition or cycling. The process memory is based on recency and frequency.

To state if an attribute is tabu or not, we make use of the *recency* and *frequency* tabu. Each time the best candidate is selected from the neighbourhood, the tabu list is modified. The communication routes of the selected solutions are recorded and their frequencies are accordingly increased. The tabu frequency of an attribute (route) is a parameter stating the number of times that the attribute can appear before being stated as tabu. If a route surpasses the threshold, it is labelled as tabu and is included in the tabu status (a time for the tabu recency parameter). The recency for the remaining routes with tabu status not appearing in the new solution is reduced by one unit. The tabu recency of an attribute is a parameter stating the number of iterations that the attribute is maintained in the tabu list. When recency is equal to zero, the route is erased from the status tabu, and the new solution could make use of it.

When the chain of solutions reaches a specified length (checked after the STM procedure), the quality of solutions is assessed to test if it is acceptable. In the case of a negative result, the LTM procedure changes the trajectory, selecting another solution from the *elite list* (which includes the best found solutions), and a new search procedure is initiated from this new seed, thus shaping a new trajectory. The selection of a solution from the elite list is carried out randomly.

The *explicit memory* saves the better solutions that have been visited; they are in the *elite list*. The *aspiration criterion* determines when the tabu status of a solution can be ignored and the algorithm permits the move to that solution. Every solution with a global cell loss lower than the cell loss of the elite list solutions can be adopted and the movement is allowed even if it has been made very recently or frequently.

STM and LTM are the most important procedures of the algorithm. Both procedures are necessarily combined in order to gain equilibrium between the exploration of the feasibility region (STM procedure) and the exploitation of a promising region (LTM procedure).

In this sense, STM constitutes a form of aggressive exploration (exploitation) that seeks to make the best possible movement within the neighbourhood. The use of STM together with the recency and frequency tabu and the aspiration criterion allows a very efficient and quick exploration of the neighbourhood. Meanwhile, LTM permits the moving back to the best-stored solutions, in order to continue searching in those promising areas after the number of visited solutions (length of the chain of solutions) exceeds a certain value and the quality of the solutions being explored starts to worsen.

Finally, Figure 9 depicts an example of the exploration search in the solution space. The figure addresses the concepts of solution, neighbour solution, neighbourhood, chain of solutions, solution from the elite list, STM procedure and LTM procedure.

LOCATION FOR FIGURE 9

4.4. Parameters calibration

Several tests were carried out to calibrate the parameters of the Tabu Search algorithm. Figure 10 illustrates the sample networks that were used for the analysis.

LOCATION FOR FIGURE 10

Tests were carried out by increasing the maximum number of iterations from 50 to 150. Through balancing the computational effort with the quality of the results, satisfactory values were obtained for 50 to 80 iterations. Figure 11 depicts how the improved results are obtained at earlier moments in the area of the MSwP initial solution, due to the high richness of such an area of the solutions space. The neighbourhood size is another critical parameter of every TS algorithm. The figure reveals a decreasing trend, although with certain fluctuations, due to the random character of the TS algorithm. In general, the selection of a size equal to 50 provided good solutions, although values in the order of 100 could be a better selection for specific larger problems. Neighbourhood sizes higher than 120 slows down the computation considerably, without providing clear improvements.

LTM determines when the search in a specific direction must be stopped and the algorithm must change the search in the neighbourhood of another solution stored in the elite list. The minimum improvement threshold states when the LTM procedure must be activated. Low values of such a threshold allow for a more exhaustive search in the previous zone, while high values allow other zones to be explored, probably crossing bad zones but with the hope of reaching new rich zones. Figure 8 represents the cell loss versus the number of solutions that must be visited before LTM checks the quality of the solutions and decides whether to change to another solution in the elite list or not. After testing different alternatives we selected a value of 6 to 8 solutions to be visited previously, in order to check the LTM procedure (this is LTM_threshold parameter in the flow chart) and an improvement threshold equal to 20% for changing the search direction to another solution of the elite list (this is Impr_threshold parameter in the flow chart).

The tabu recency states the number of iterations that a tabu route must have to be tabu-active. The results are affected by the random process of the TS algorithm, and similarly by LTM. Low recency allows an exhaustive exploration of the neighbourhood and high recency allows faster movements among feasible zones of the solutions space. The best values were obtained for low recency. We selected a value equal to 2 for the parameter.

The tabu frequency was fixed to 6 with the objective of not discarding good solutions of the initial zone in the area of the MSwP algorithm solution (this is Freq_threshold parameter in the flow chart). Finally, an elite list equal to 5 was selected in order to maintain only the particularly good solutions.

LOCATION FOR FIGURE 11

Although the size of the buffer and the GoS are parameters that depend on the technological aspects of the network, we analysed the behaviour of the network performance with respect to them in order to select a good set of technical parameters. Figure 12 depicts the cell loss as a function of the buffer size and the GoS required.

LOCATION FOR FIGURE 12

The size of the buffer was fixed to 8 cells and the GoS was determined to allow a delay equal to 5×10^{-7} (sec), equivalent to 5 switching processes in nodes.

5. Computational results

The Tabu Search algorithm was compared to the Genetic Algorithm first described in Cortés *et al.* (2002), whose pseudocode is shown in the appendix. The solutions provided by the MSwP algorithm are also presented, showing the improvement reached by the TS and GA heuristics respectively.

Tests were carried out with 25 trial problems, which can be seen via the Internet at <http://io.us.es/RedTeltrials/DynamicRouting>. Table 1 shows the results for the Tabu Search algorithm compared to the genetic algorithm and the MSwP heuristic.

LOCATION FOR TABLE 1

The table shows that both algorithms (TS and GA) allow the solution provided by MSwP to be improved significantly. Results also reveal that TS consistently provides better solutions than GA, practically speaking. Furthermore, TS successes can be better appreciated in the more complex cases (with a larger size in terms of nodes and links). Figure 13 shows the behaviour of the metaheuristics. The size of the green bar rises dramatically when the size of the problem is increased. For networks that are larger than 25 nodes, TS provided improvements in the order of 50% with respect to GA.

With regards to time consumption, MSwP is very fast but does not provide particularly high quality solutions. On the contrary, TS and GA, which are iterative algorithms, take up more time. For networks that are smaller than 19 nodes, TS makes use of more time than GA. However, for bigger networks, especially for networks that are bigger than 24 nodes, TS clearly takes less time than GA with the reduction being in the order of 50%.

So, TS offers better results for the larger networks, with regards to the quality of the solution and the computational time.

LOCATION FOR FIGURE 13

6. Conclusions and further research

In this paper, we have dealt with the dynamic routing problem in ATM cell-switching networks, considering matrix switches and queues at the end. We present a model based on cell loss and a Tabu Search algorithm to solve it using short-term memory, which is reinforced with a long-term memory. The estimation of the quality of the solutions (in terms of cell loss) is rapid due to the specific encoding of the feasible solutions. Several tests were done to calibrate the algorithm parameters adequately.

The procedure showed an adequate behaviour relating to the cell loss and CPU time consumption. The algorithm outperformed other metaheuristics approaches such as the genetic algorithm and the Minimum Switching Path heuristic. Furthermore, this approach offered better results for those more complex networks with a higher number of nodes (switches) and links. The results allow us to confirm that Tabu Search in general, and our proposal in particular, are valuable tools with great potential in the operation of real time telecommunication networks. However, the implementation of such algorithms in real network controllers has to be carried out carefully in order to maintain the response time of the algorithm bounded. Tabu Search algorithms are iterative and can therefore take a long time for implementation, when a long neighbourhood and a great number of iterations are used. The selection of these parameters has to be based looking not so much at the algorithm accuracy but rather at the available time for routing messages in real time. In real cases, an alternative could be to stop the algorithm before it reaches the next event, which would occur after a known time interval. Of course all these

decisions are very much dependant on the computation speed and capabilities of real life equipment.

Our further research is now focused on considering other traffic conditions such as VBR, ABR or UBR for other types of services, as well as other switches options with non-FIFO queues.

Appendix. Genetic Algorithm and Minimum Switching Path pseudocodes

LOCATION FOR TABLE 2

LOCATION FOR TABLE 3

LOCATION FOR TABLE 4

References

- [1] D. Medhi, D. Tipper, "Some approaches to solving a multihour broadband network capacity design problem with single-path routing," *Telecommunication Systems* 13 (2-4) (2000) 269-291.
- [2] J. Xu, S.Y. Chiu, F. Glover, "Tabu search for dynamic routing communications network design," *Telecommunication Systems* 8 (1) (1997) 55-77.
- [3] C-C Shyur, E-P Wen, "Optimizing the system of virtual paths by tabu search," *European Journal of Operational Research* 129 (3) (2001) 650-662.
- [4] C.G. Wille, M. Mellia, E. Leonardi, M. Ajmone, "Algorithms for IP Network Design with End-to-End QoS Constraints," *Computer Networks* 50(8) (2006) 1086-1103.
- [5] P. Cortés, J. Muñuzuri, J. Larrañeta, L. Onieva, "A genetic algorithm based on cell loss for dynamic routing in ATM networks," in: R. Roy, M. Köppen, S. Ovaska, T. Furuhashi, F. Hoffman (Ed.) *Soft Computing in Industry - Recent Applications*. Springer-Verlag, Berlin, 2002, pp. 627-640.
- [6] A. Yener, C. Rose, "Genetic Algorithms Applied to Cellular Call Admission: Local Policies", *IEEE Transactions on Vehicular Technology* 46 (1) (1997) 72-79.
- [7] S-L. Wang, Y-B. Hou, J-H. Huang, Z-Q. Huang, "Adaptive Call Admission Control Based on Enhanced Genetic Algorithm," in: *Wireless/Mobile Network*. 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), 2006, 3-9.
- [8] A. Pitsillides, G. Stylianou, C.S. Pattichis, A. Sekercioglu, A. Vasilakos, "Bandwidth allocation for virtual paths (BAVP): investigation of performance of classical constrained and genetic algorithm based optimisation techniques," in: *Proc. of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2000* 3 (26-30), 2000, pp 1501-1510.
- [9] H. El-Madbouly, "Design and Bandwidth Allocation of Embedded ATM Networks using Genetic Algorithm, Transactions on Engineering." *Computing and Technology* 8 (2005) 249-252.
- [10] D-R. Din, "Genetic algorithms for multiple multicast on WDM ring network," *Computer Communications* 27 (9) (2004) 840-856.
- [11] D-R. Din, "A genetic algorithm for solving virtual topology configuration transition problem in WDM network," *Computer Communications* 30 (4) (2007) 767-781.
- [12] V.T. Le, X. Jiang, S.H. Ngo, S. Horiguchi, Y. Inoguchi, "A novel dynamic survivable routing in WDM optical networks with/without sparse wavelength conversion," *Optical Switching and Networking* 3 (3-4) (2006) 173-190.
- [13] S. Tak, E.K. Park, "Restoration performance study of k-shortest disjoint paths in WDM optical networks," *Telecommunication Systems*, 34 (1-2) (2007) 13-26.
- [14] F. Glover, "Tabu search I," *ORSA Journal on Computing*, 2(1) (1989) 4-3.
- [15] F. Glover, "Tabu search II," *ORSA Journal on Computing*, 1(3) (1990) 190-206.