16

Genetic Algorithms for Semi-Static Wavelength-Routed Optical Networks

R.J. Durán, I. de Miguel, N. Merayo, P. Fernández, J.C. Aguado,
A. Bahillo, R. de la Rosa and A. Alonso
Optical Communications Group, University of Valladolid,
Spain

1. Introduction

Optical communication networks are an excellent option to establish backbone or transport networks due to the high-bandwidth provided by the optical fibres. In this kind of networks the information is transmitted through optical fibres in the infrared domain. Although fibres provide high bandwidth for the transmission of information (around 50 THz), the current electronic technology cannot work at these bit rates. Therefore, a new technology was developed to exploit the bandwidth of the fibre: Wavelength Division Multiplexing (WDM). In this technology, the part of the spectrum used to transmit information is divided into different channels, each one centred in a different wavelength (or frequency). Then, it allows the transmission of several channels through the same fibre by using a different wavelength for each of the channels.

In first optical communication networks, the optical technology was only used for transmission between nodes that were directly connected by means of one or more optical fibres, i.e., adjacent nodes in the physical network. Therefore, when data should be transmitted between two non-adjacent nodes, the information should transverse intermediate nodes in which the information is converted into electrical domain in order to route the traffic through the appropriate output fibre and then transmit the information in the optical domain. In this scenario, the nodes become the bottleneck of the network.

To solve this problem, new optical networks emerged with the capacity of using the wavelength to perform routing functions (Mukherjee, 1997). Thus, a network node can distinguish between traffic destined for other nodes and for itself without any processing, as it is determined by the wavelength and the input port of incoming data. This kind of networks are called Wavelength-Routed Optical Networks (WRONs) and they are based in circuit switching. WRONs allows the establishment of optical circuits between two network nodes (not necessarily adjacent in the physical topology) for the transmission of information. These circuits are called lightpaths. The lightpaths can be permanently (or semi-permanently) established in static (or semi-static) WRON, or established and released on demand in dynamic WRONs.

The objective of this chapter is to show a set of single-objective and multi-objective genetic algorithms, designed by the Optical Communications Group at the University of Valladolid,

to optimize the performance of semi-static Wavelength-Routed Optical Networks. The fundamentals of those algorithms, i.e., the chromosome structures, their translation, the optimization goals and the genetic operators employed are described. Moreover, a number of simulation results are also included to show the efficiency of genetic algorithms when designing WRONs.

2. Wavelength-routed optical networks

Wavelength Division Multiplexing (WDM) allows the transmission of several data channels through an optical fibre. Thanks to this technique it is possible not only to use better the huge bandwidth of the optical fibre but also to perform routing in the network nodes in order to avoid the bottleneck that can appear due to the electronic processing. In this context, the concept of Wavelength-Routed Optical Networks (WRONs) was introduced. The basic element in WRON is the lightpath, i.e., an all-optical connection between two network nodes even when they are not adjacent in the physical topology (i.e., when there is not any fibre connecting the two nodes). In this way, the transmission between the two end nodes does not require any electronic processing in the intermediate nodes.

A lightpath should fulfil two conditions:

1. Each lightpath should use the same wavelength in all the fibres that it transverses, unless the network is equipped with wavelength-converters in the nodes, but this is not considered in this work. This restriction is known as the *wavelength continuity constraint*.
2. Two lightpaths cannot use the same wavelength in the same fibre. However, it is possible to reuse wavelengths in different network fibres.
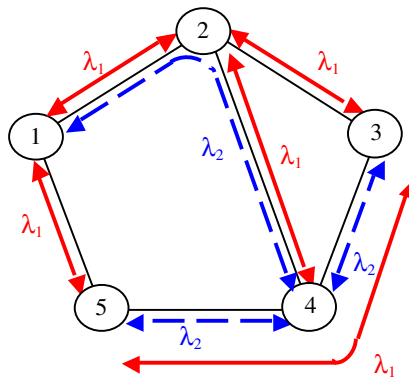

Fig. 1. Example of a wavelength-routed optical network (WRON). Eight bidirectional lightpaths have been established using only two wavelengths

In Fig. 1, a WRON with eight bidirectional lightpaths using only two wavelengths is shown. Moreover, it is shown that a lightpath can be established between adjacent source and destination nodes (for instance, between nodes 5 and 4), or traversing one or more intermediate nodes. For instance, the lightpath between nodes 5 and 3 traverses node 4, which is in charge of properly routing the optical signal with no need to perform conversion to the electronic domain. Note that a lightpath can also be established in a unidirectional

way, or even being bidirectional but following different routes and wavelengths for each direction.

In principle, we could think about establishing one lightpath between each source-destination pair (or even more), but this option is not always appropriate. In a network with N nodes, N–1 transmitters (and receivers) are required at each node, and $N(N$–1) unidirectional lightpaths should be established, so that the number of wavelengths required can be high as well as the cost of the network, due to the number of transmission and reception equipments needed, and to the requirements on wavelength-handling capacity of the optical crossconnects (OXC) and optical add and drop multiplexers (OADM). Moreover, the utilisation of a high number of wavelengths may degrade the quality of the transmission due to physical impairments. This effect can provoke that some of the connections cannot be established due to not reaching enough quality level.

Therefore, instead establishing lightpaths in a permanent o semi-permanent way between each source-destination pair of the network, one solution consists in establishing only a subset of them. In this way, a lower number of transceivers per node and a lower number of wavelengths are required. The drawback is that not all the traffic can be transmitted directly from the source node to the destination node (single-hop communication), but traffic between pairs of nodes which are not directly connected by a lightpath must traverse one or more intermediate nodes where conversion to the electronic domain is performed. Therefore, it is a multihop scenario. Moreover, this solution usually implies a static or semi-static situation as lightpaths are permanently or semipermanently established, although in some cases some of them can be deleted and new ones can be established, for instance to adapt to changes in the traffic offered to the network or to react to network failures.

3. Genetic algorithms

Genetic algorithms (GAs) (Goldberg, 1989; Man et al., 1999) are search algorithms based on the mechanics of natural selection and natural genetics, where stronger individuals are the likely winners in a competing environment. Genetic algorithms represent each solution to any problem as an individual represented by a set of parameters. These parameters are regarded as genes and can be structured on a string or chromosome. The fitness of each individual is the objective parameter that we want to optimize.

An initial population is created, randomly most of the times, and then evolved by means of genetic operators, such as reproduction, crossover and mutation, to form a new population (the next generation) that is hoped to be fitter than the previous one. The reproduction operator creates a literal copy of selected individuals from the parent population in the descendant generation. The crossover operator is applied to pairs of individuals in order to interchange their genetic material. To generate a good offspring, a good mechanism for selecting parents is necessary. Roulette wheel selection is one of the most common techniques. When using this method, the probability of selecting an individual is proportional to its health (or fitness). In this way, good properties should propagate down the generations. On the other hand, the mutation operator makes a random change in the genetic material of a single individual, allowing the GA to explore new corners of the search space and hence avoiding the risk of being trapped in a local optimum. The evolution process is repeated a predefined number of iterations or until another criterion is met. Since

individuals from the population become fitter throughout the generations, it is hoped that the final population will contain an optimal or near optimal solution.

Finally, it should be remarked that GAs are generic methods that have to be customized to the particular problems that we attempt to solve. Therefore, the design of the encoding mechanism of the individuals and the different operators of GAs must be adapted to the characteristics of the problem being tackled, since they have a significant impact on the performance of the algorithm.

4. Multi-objective optimization: Pareto optimality with genetic algorithms

Pareto optimality (Man et al., 1999) is based on the concept of the dominant individual. In an n-objective optimisation problem, where $f_i(u)$ is the result of evaluating the individual u according to the objective function i, and assuming that lowest values of $f_i(u)$ are preferred, u is said to be dominated by v if

$$f_i(u) \geq f_i(v) \quad \forall i = 1, 2, ..., n \tag{1}$$

and if

$$\exists j = 1, 2, ..., n \text{ such that } f_j(u) > f_j(v) \tag{2}$$

Then, the ranking of an individual is defined as the number of individuals for which it is dominated. The set of optimal individuals is called the Pareto optimal set and it is defined as the set of individuals with ranking zero, i.e., those that are do not dominated by anyone. Hence, each solution that belongs to the Pareto optimal set is characterized because it cannot be simultaneously improved in terms of all the optimization objectives.

Fig. 2 shows an example of a set of solutions provided by a multi-objective method in which each solution is represented in terms of two optimization objectives f_1 and f_2. In both optimization objectives lower values are preferable than higher ones. The number that it is represented near each point (solution) is the ranking of the individual. Moreover, the Pareto optimal set is also represented.
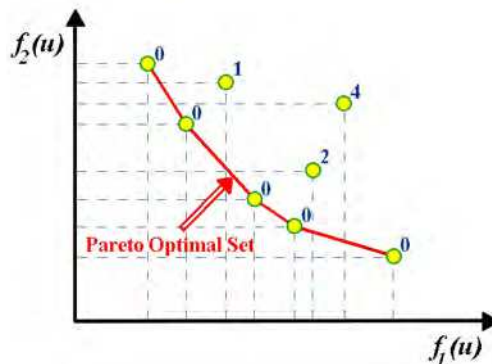

Fig. 2. Examples of the set of solution obtained by a multi-objective method represented in terms of two optimization objectives f_1 and f_2. Lower values are desirable in both objectives.

Pareto optimality can be combined with genetic algorithms in order to solve multi-objective optimization problems. In single-objective problems the roulette wheel method, according to the individual fitness, is utilized as selection process, but when Pareto optimality is combined with genetic algorithms in order to solve multi-objective optimization problem, one possible method to perform the selection is that all the individuals in the Pareto optimal set have the same probability of being selected for a reproduction or crossover stage. Moreover, the selection of the individuals that will take part in the next generation is done by selecting only those solutions from the descendant population which belong to the Pareto optimal set. Then, every individual in the population belong to the Pareto optimal set and all of them have the same probability of being selected for a reproduction or crossover stage.

5. Optimization of semi-static WRON

As it was explained in the previous section, the establishment of a lightpath between each pair of the nodes of the network can be a technical or economical unfeasible option. Therefore, a common solution consists in establishing only a subset of connections. This subset of connections is called the virtual (or logical) topology, and it is the network topology seen by the protocols from the upper layer in the communication protocol tower (Mukherjee, 1997). In Fig. 3 it is possible to see the physical topology (i.e., the physical network where each link of the topology consists of two pair of fibres between a pair of nodes, each one in a direction) from the network shown in Fig. 1. The virtual topology of the network shown in Fig 1 is represented in Fig. 4.
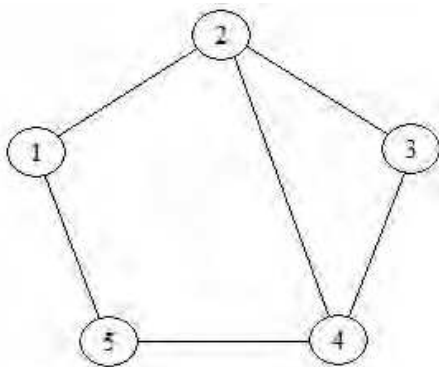

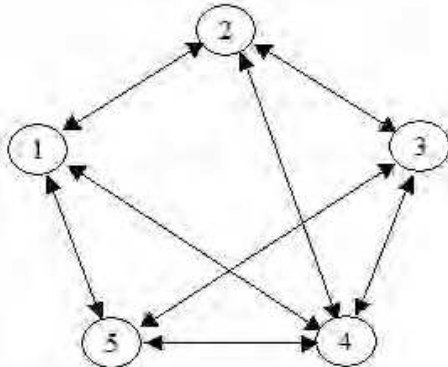Fig. 3. Physical topology from the network shown in Fig. 1

Fig. 4. Virtual topology from the network shown in Fig. 1

5.1 Virtual topology design problem

The optimization problem of designing a virtual topology, given a physical topology and a matrix of traffic to be carried by the network, is shown to be NP-complete. It is usually divided into three subproblems (Mukherjee, 1996):

1. To determine which network nodes should be connected by lightpaths, i.e., to determine the interconnection pattern in the virtual topology.

2. To find a route for each of those lightpaths in the physical network, and to assign an available wavelength to it. This subproblem is known as the static routing and wavelength assignment (RWA) problem.
3. To route traffic on the virtual topology.

Due to the relationship between the three subproblems, some of them are usually solved jointly. The most common method consists in adopting an intermediate solution, and not trying to solve all subproblems jointly nor each subproblem separately. The method consists in solving some of the subproblems jointly. There are a number of possibilities but the most usual one is to solve subproblems (1) and (3) jointly, and then subproblem (2).

Now, we present the mathematical definition of the problem of designing a virtual topology for a WRON, taking into account the three subproblems into which it is divided. The problem can be described as a Mixed Integer Linear Programming formulation (MILP). The notation followed is borrowed from (Dutta & Rouskas, 2000). As an example, the objective of the formulation will be the minimization of the congestion (i.e., the traffic carried by the most loaded lightpath). We describe here all the equations for completeness and then add a constraint for lightpath routing that reduces the computing time required to solve the problem. The following notation is used:

* s and d are used as subscript or superscript and denote the source and the destination of a traffic flow, respectively.
* q is an index to distinguish between the lightpaths between the same pair of nodes.
* i and j denote source and destination nodes, respectively, of a lightpath.
* m and n denote endpoints (nodes) of a physical link.
* f is an index to identify the number of the fibre used among the fibres between two nodes.
* k is the wavelength assigned to a lightpath

Given:

* N is the number of nodes in the network.
* W is the number of wavelengths per fibre.
* Δ is the logical degree of the nodes, i.e., the number of transmitters placed in each node. For simplicity we assume that each node is also equipped with the same number of receivers.
* F is the maximum number of fibres between two nodes.
* Q is the maximum number of lightpaths that can be established between a pair of nodes. In order to do not exceed the logical degree:

$$Q \le \Delta \tag{3}$$

* $\Lambda^{s,d}$ is the traffic from node s to node d.
* $p_{m,n,f}$ is a binary variable that denotes whether there is a fibre between nodes m and n. $p_{m,n,f}$ takes the value 1 if there are at least f fibres between nodes n and m and takes the value 0 otherwise.

$$p_{m,n,f} \ge p_{m,n,(f+1)} \qquad f \in [1, F-1] \tag{4}$$

* $d_{m,n,f}$ is the propagation delay in the fibre f between the nodes m and n.

- $D_{\min}^{i,j}$ is the propagation delay between the nodes i and j following the shortest path in terms of propagation delay. This value can be estimated using the Dijkstra algorithm modified with the use of weights in the different links (Johnsonbaugh & Schaefer, 2004) and each weight is the propagation delay in that fibre ($d_{m,n,f}$).

Variables:

- $b_{i,j,q}$ is a binary variable that denotes if the lightpath q between node i and node j is established. Note that in this nomenclature i is the source node for this lightpath and j is the destination node. It is important to remark this point since in our definition the lightpaths are unidirectional. $b_{i,j,q}$ = 0 means that the lightpath q between nodes i and j is not established. $b_{i,j,q}$ takes the value 1 when this lightpath between these two nodes is established.
- $\lambda_{i,j,q}^{s,d}$ is a variable to represent how much traffic from node s to node d uses the lightpath q between nodes i and j.
- $\lambda_{i,j,q}$ is the traffic that is carried by the lightpath q between nodes i and j.
- $\lambda_{\max}$ is the traffic carried by the most loaded lightpath, i.e., the network congestion.
- $c_{i,j,q}^{k}$ is a binary variable that takes the value 1 if the lightpath between i and j uses the wavelength k, and 0 otherwise.
- $c_{i,j,q}^{k}(m,n,f)$ takes the value 1 if the lightpath between nodes i and j uses the fibre between nodes m and n and uses the wavelength k, and 0 otherwise.

Objective:

$$\text{Minimize } \lambda_{\max} \tag{5}$$

Constraints:

- Degree constraints:

$$\sum_{j=1}^{N}\sum_{q=1}^{Q} b_{i,j,q} \leq \Delta \qquad \forall i \tag{6}$$

$$\sum_{i=1}^{N}\sum_{q=1}^{Q} b_{i,j,q} \leq \Delta \qquad \forall j \tag{7}$$

Both restrictions limit the number of lightpaths that can be established depending on the maximum number of transmitters, eq. (6) and receivers, eq. (7).

- Routing and wavelength assignment constraints:

$$\sum_{k=1}^{W} c_{i,j,q}^{k} \leq b_{i,j,q} \qquad \forall i,j,q \tag{8}$$

$$c_{i,j,q}^{k}(m,n,f) \leq c_{i,j,q}^{k} \qquad \forall i,j,q,k,m,n,f \tag{9}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{q=1}^{Q}c_{i,j,q}^{k}(m,n,f)\le p_{m,n,f}\qquad\forall k,m,n,f \tag{10}$$

$$\sum_{f=1}^{F}\sum_{k=1}^{W}\sum_{l=1}^{N}c_{i,j,q}^{k}(m,l,f)-\sum_{f=1}^{F}\sum_{k=1}^{W}\sum_{l=1}^{N}c_{i,j,q}^{k}(l,m,f)=\begin{cases}b_{i,j,q} & m=i\\ -b_{i,j,q} & m=j\\ 0 & m\ne i,m\ne j\end{cases}\qquad\forall i,j,q,m \tag{11}$$

Eq. (8) ensures that each lightpath uses one, and only one, wavelength. Eq. (9) makes that the same wavelength is used in all the fibres that each lightpath traverses. Eq. (10) avoids collisions, i.e., avoids the possibility that different lightpaths use the same wavelength in the same fibre. Eq. (11) is based on the flow conservation constraints given in (Dutta & Rouskas, 2000) and it is the equation to ensure that the routing of the lightpath over the physical network is continuous along the route.

- Traffic constraints:

$$\lambda_{i,j,q}=\sum_{s=1}^{N}\sum_{d=1}^{N}\lambda_{i,j,q}^{s,d}\qquad\forall i,j,q \tag{12}$$

$$\lambda_{i,j,q}\le\lambda_{\max}\qquad\forall i,j \tag{13}$$

$$\lambda_{i,j,q}^{s,d}\le b_{i,j,q}\cdot\Lambda^{s,d}\qquad\forall i,j,q,s,d \tag{14}$$

$$\sum_{j=1}^{N}\sum_{q=1}^{Q}\lambda_{i,j,q}^{s,d}-\sum_{j=1}^{N}\sum_{q=1}^{Q}\lambda_{j,i,q}^{s,d}=\begin{cases}\Lambda^{s,d} & s=i\\ -\Lambda^{s,d} & d=i\\ 0 & s\ne i,d\ne i\end{cases}\qquad\forall s,d,i \tag{15}$$

Eq. (12) defines the traffic load assigned to the lightpath q between node i and j. The congestion is defined in the Eq. (13). The restriction of eq. (14) avoids that traffic can be routed over a non-existing lightpath. Traffic flow conservation constraints given by Eq. (15) guarantee the conservation of the traffic demand $\Lambda^{s,d}$ along the routes between the source and destination.

In order to speed up the process, one more constraint is added.

$$\sum_{m=1}^{N}\sum_{n=1}^{N}\sum_{k=1}^{W}c_{i,j,q}^{k}(m,n,f)\cdot d_{m,n,f}\le D_{\min}^{i,j}\qquad\forall i,j,q \tag{16}$$

By the Eq. (16) a lightpath can only be established using one of the shortest paths between the source and destination of the lightpath.

5.2 Network provisioning problem

Most network operators have already deployed their fibre networks and they are currently using a set of locations (network nodes) to aggregate and disaggregate the traffic that goes

through it. As the cost of the physical installation is the main capital expenditure when a fibre network is built, the optimization methods to be developed in order to upgrade these first generation optical networks to WRONs (while being cost efficient), should use the current physical topologies as an input parameter. Hence, the problem of the design of a WRON starts with the decision about the number of transmitters and receivers that will be used in it and their placement among the network nodes.

On the other hand, WRONs rely on Wavelength Division Multiplexing (WDM) to increase the capacity of the network and also to make the optical routing of the lightpaths. Hence, it is also necessary to determine the number of wavelengths that will be utilized in the network. In this sense, the use of less wavelengths is preferred because it is cheaper as it allows to utilize simpler tunable transmitters and receivers and reduces the complexity of the OXCs.

Therefore, the network provisioning consists of deciding the number of wavelengths, transmitters and receivers and choosing how many transmitters and receivers will be placed in each network node. Note that the optimal solution could require equipping each node with different numbers of these devices. The design of a virtual topology depends on the network provisioning. Hence, in order to design a proper wavelength-routed optical network, the network provisioning and the design of the virtual topology should be considered jointly. However, most of the current methods do not solve both issues together as it is even more complex that only designing the virtual topology. Therefore, it is possible to modify the mathematical definition shown in the previous section to include the problem of network provisioning together with that of virtual topology design. Considering that:

- $Tx_{\max}$ is the maximum number of transmitters that can be used in the network.
- $Rx_{\max}$ is the maximum number of receivers that can be used in the network. As no lightpath protection is introduced in this formulation and as each lightpath requires a transmitter and a receiver, the number of these to parameters should be equal in order not to waste money.

$$Tx_{\max} = Rx_{\max} \tag{17}$$

- Tx_i is the number of transmitters that should be placed at node i.

$$Tx_i = \sum_{j=1}^{N} \sum_{q=1}^{Q} b_{i,j,q} \tag{18}$$

- Rx_j is the number of transmitters that should be placed at node j.

$$Rx_j = \sum_{i=1}^{N} \sum_{q=1}^{Q} b_{i,j,q} \tag{19}$$

Then, it is necessary to change Eqs. (6) and (7) by the next two equations:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{q=1}^{Q} b_{i,j,q} \leq Tx_{\max} \tag{19}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{q=1}^{Q} b_{i,j,q} \leq Rx_{\max} \tag{20}$$

5.3 Genetic algorithms to design virtual topologies

In (Durán et al., 2009a) Durán *et al.* propose GALD (Genetic Algorithm for Logical topology Design). GALD is a method designed to solve the full problem of virtual topology design, i.e., to solve the three subproblems shown in section 5.1 jointly. In this section, this work will be presented and complemented with the different versions of the algorithm proposed by the authors. Its chromosome encoding, its translation, the evolution operators used and a summary of performance results will be presented.

5.3.1 Chromosome encoding and translation

The chromosome structure in GALD consists of a sequence of genes where each gene represents a source-destination node pair.

When the chromosome is interpreted in the translation stage, GALD takes each gene in order and tries to establish a unidirectional lightpath between the source and the destination node, subject to the available set of resources. Any of the methods proposed in the literature to solve the RWA problem can be used but a fast method is preferred, hence, we use pre-calculated shortest paths (in terms of propagation delay) in order to solve the routing, and the first-fit heuristic (Zang, 2000) to assign a wavelength to each lightpath. Every time that a gene in the chromosome is read, GALD looks for the lowest available wavelength in any of the shortest paths between the source node and the destination node represented by the gene. If an available wavelength is found, the lightpath is established. Otherwise, the lightpath is not established.

Fig. 5 shows an example of both the chromosome encoding and the translation stage. Each node in the sample network has two transmitters and two receivers, each cable consists of two fibres of the same length (one for each direction), and each fiber is equipped with only one wavelength. When the translation stage takes place, each pair of source-destination nodes (gene) is read from the chromosome. The first three genes are converted into lightpaths because there are enough free resources for their establishment. However, when the algorithm tries to establish the lightpath represented by the fourth gene, it discovers that there is no free wavelength in the shortest path between D and B, therefore this connection cannot be established and is not included in the logical topology. After that, the algorithm tries to establish the lightpaths represented by the final two genes of the chromosome and, as there are enough resources, both of them are established. The resulting logical topology is shown in the figure.

When the chromosome is completely translated, GALD checks whether the logical topology obtained is connected, i.e., there should be at least one possible path in the virtual topology between all the network nodes. If the topology is not connected, the logical topology is rebuilt by first establishing a set of lightpaths forming a Hamiltonian circuit (Johnsonbaugh & Schaefer, 2004, pp. 75-77) and then it continues by trying to establish additional lightpaths according to the information in the chromosome following the method presented above. Then, the traffic is routed in this virtual topology following the shortest paths in terms of hops, similarly as it is done in IP routing protocols.
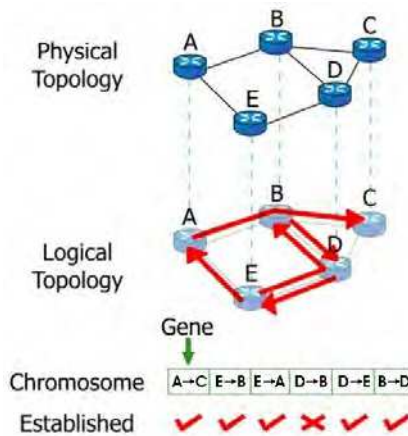
Fig. 5. Example of the proposed chromosome structure and the translation stage.

5.3.2 Initial population

When a new request arrives at the centralized control node to design a virtual topology, it considers the matrix of traffic demands and launches the GALD algorithm. The initial population of GALD is created randomly except one individual that is created following the ideas of a very fast heuristic to design virtual topologies with low congestion: HLDA (Heuristic for Logical topology Design Algorithm) (Ramaswami & Sivarajan, 1996). Hence, the matrix of traffic demand is ordered from the highest to the lowest one. Then, this first individual is created selecting as the first gene the source-destination with the highest traffic demand, the second gene should represents the source-destination pair with the second highest demand and so on. This chromosome will ensure that the health of the fittest individual of the initial population will be better, in most of cases, than the health of the fittest individual of a completely randomly created initial population.

5.3.3 Genetic evolution and operators

During the evolution process (i.e., when the genetic operators are applied), GALD uses the roulette wheel selection mechanism to select the parent individual. Once the first parent (parent 1) has been chosen, it is necessary to decide if a reproduction stage or a crossover stage will take place. The former stage is selected with probability 1– p_{cross} and the later with probability p_{cross}. In the reproduction stage, the individual selected will be copied to the new population without changes. In the crossover stage, it is necessary to select another individual (parent 2) to make the combination and interchange their genetic material. This individual (parent 2) is also selected with the roulette wheel selection mechanism. However, this time the first selected individual (parent 1) cannot be picked out again in order to avoid a veiled reproduction stage. Once both parents have been selected, the crossover operator randomly selects a gene position. Then, the parent chromosomes are divided into two halves by that gene and children are made by interchanging the second halves of their parents. These processes of selection, reproduction and crossover are repeated until the number of individuals in the descendant population (DP) is reached.

The mutation operator selects the genes one by one in each chromosome from the descendant population and mutates them according to a probability $p_{mutation}$. If mutation is necessary, the gene is interchanged by another one randomly chosen among the rest of possible values that a gene can take.

After applying the evolution operators, the P fittest individuals from the descendant population are selected to be the parent population for the next generation.

The chromosome structure used for GALD and its translation process makes that each gene position is dominant over the subsequent ones, so it has greater impact on the resulting logical topology. Hence, the crossover and mutation operators are improved by making that there is more probability to choose as crossover point or as gene to be mutated a gene from the first positions in the chromosome than from the latest ones. This is made because when these changes are done in the first genes, the outcomes of applying these operators will be more perceptible.

The stopping criterion is set on the maximum number of generations or on the computation time that the algorithm is allowed to find the solution. One advantage of GALD is that it can be used in a reconfigurable WDM network as the optimization process can be stopped at any time (if required) in order to give the best virtual topology found up to the moment.

5.3.4 Fitness function

As it is explained in section 5.4.1, when the chromosome is translated, the set of lightpaths that will be established are determined, the route and the wavelength used by them is decided, and the traffic is routed over this virtual topology. Then, once the traffic is routed on the virtual topology, the congestion and the delay can be estimated. Different versions of GALD have been presented depending on its fitness function:

- **Traffic Optimization:**
  - **C-GALD** (Congestion optimized – GALD) (Durán et al., 2009a): It is a single-objective version of GALD in which the objective is the minimization of the network congestion (i.e., the traffic carried by the most loaded lightpath).
  - **F-GALD** (Fairness optimized – GALD) (Durán et al., 2009a): a variation to optimize the fairness of the network (whether traffic is uniformly distributed over the established lightpaths). The fairness can be measured by means of the Jain index, which is defined as

$$Jain\ index = \frac{\left(\sum_{i=1}^{L} \lambda_i\right)^2}{L\sum_{i=1}^{L} \lambda_i^2},$$

  where L is the number of lightpaths in the virtual topology and λ_i is the traffic carried by lightpath i. Hence, this index can take values from $1/L$ in the worst case to 1 in the fairest case. So, if the Jain index is close to 1, all lightpaths will carry a similar amount of traffic.
  - **CF-GALD** (Congestion and Fairness optimized – GALD) (Durán et al., 2009a): a version of GALD that optimizes both congestion and fairness. When using this

method, new individuals need to improve both congestion and fairness to be considered as the best topology found until that moment.

- **Delay Optimization:**
  - **MD-GALD** (Mean Delay optimized – GALD) (Durán et al., 2009b): It is a single-objective version of GALD in which the mean end-to-end delay is the parameter employed to evaluate the fitness of an individual. This parameter is estimated using method shown in ((Durán et al., 2009b).
  - **MxD-GALD** (Maximum Delay optimized – GALD) (Durán et al., 2009b): It is a single-objective version of GALD in which the parameter to be optimized is the average end-to-end delay of the most delayed traffic flow.
- **Traffic and Delay Optimization:**
  - **DC-GALD** (Delay and Congestion – GALD) (Durán et al., 2009c): It is a multi-objective version of GALD that employs the classic technique of making a random choice of the fitness criterion (end-to-end delay or congestion, each with 0.5 probability) in order to determine which individuals survive in the next generation.
  - **PDC-GALD** (Pareto optimality of Delay and Congestion – GALD) (Durán et al., 2009c): It is a multi-objective version of GALD, based on Pareto optimality techniques, which jointly optimizes congestion and end-to-end delay.

5.3.5 Results obtained with the methods

Firstly, in Table 1, the performance of C-GALD is compared in terms of network congestion and computing time with that obtained by the MILP formulation shown in section 5.1 with both the basic MILP formulation (Eq. 3 to 15), and with an extension which considers an additional constraint that only allows routing lightpaths through the shortest paths (Eq. 16). The former formulation is denoted by MILP and the latter by SP-MILP (Shortest-Paths MILP). Two networks with different numbers of nodes were used as physical topology and each node was equipped with three transmitters and three receivers. The traffic load between each pair of nodes was set to 0.3 (i.e., the 30% of the capacity of a lightpath). Simulations were made on a computer with a 1.6 GHz Intel Centrino processor and 512 MB of RAM memory, and ILOG CPLEX 10.0 was used to solve the MILP formulations.
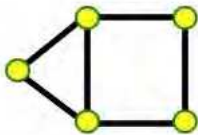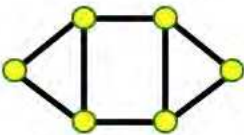
	Five-node network		Six-node network	
	Congestion	Execution Time (s)	Congestion	Execution Time (s)
MILP	0.5	15.88	0.7	38060.66
SP-MILP	0.5	1.86	0.7	775.39
C-GALD	0.5	0.01	0.7	1.39

Table 1. Comparison of the congestion of the virtual topologies and the corresponding execution time with the MILP formulations and C-GALD

The results show that all the methods achieve the same results in terms of congestion but C-GALD is the fastest method. It has a computing time which is 180 and 550 times lower than that of SP-MILP in the networks with five and six nodes, respectively. Then, if a network with seven nodes is used, the MILP formulations are unable to find the optimal solution in less than a week using the computer described above. The problem with the MILP formulations is that realistic wide area networks usually have more than seven nodes; hence, the MILP formulations become computationally intractable. On the contrary, C-GALD designs a logical topology in a relatively short period of time and has the advantage of always having a suboptimal solution during the evolution process, thus, this solution can be utilized (if necessary) in a reconfigurable WRON.

In order to evaluate the performance of the different versions of GALD, a simulation study was developed in a simulator based on OMNeT++ using the 14-node NSFNet (Ghose et al., 2005) as network topology and assuming that cables consisted of two unidirectional fibers (one for each direction). The capacity of a wavelength was set to 10 Gbps and the traffic load was normalized by the lightpath capacity. Each node is equipped with five transmitters and five receivers and three wavelength per fibre. Although GALD can be used in networks with or without wavelength converters, we assumed that there were no wavelength converters in the network. The performance achieved by the different versions of GALD is compared with HLDA (Ramaswami & Sivarajan, 1996) as it designs the logical topology with the aim of minimizing network congestion, and it takes into account both the traffic matrix and the availability of physical resources, like GALD does.

In Fig. 6 we compare the congestion obtained when using HLDA (Ramaswami & Sivarajan, 1996) and the tree versions of GALD to optimize the network capacity: C-GALD, F-GALD and CF-GALD. In this last version of GALD, the probability of selecting congestion as fitness function in each generation is set to 0.8, so the probability of choosing fairness is 0.2. We employed uniformly distributed traffic matrixes with different values of the mean traffic load, the creation of 1,000 generations (which takes less than a minute) was selected as the stopping criterion for the three versions of GALD, and the population size was set to two and the descendant size to 12. The corresponding values of fairness are represented in Fig. 7.
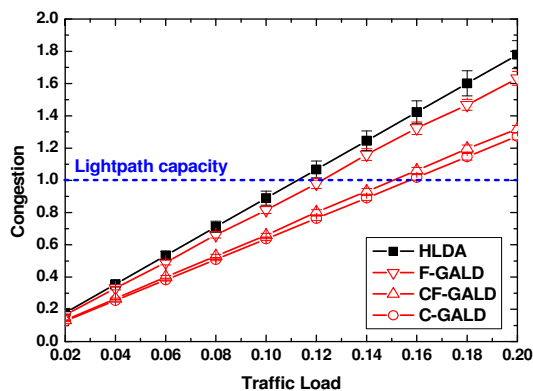

Fig. 6. Congestion value of the logical topologies designed by using HLDA, C-GALD. F-GALD and CF-GALD.
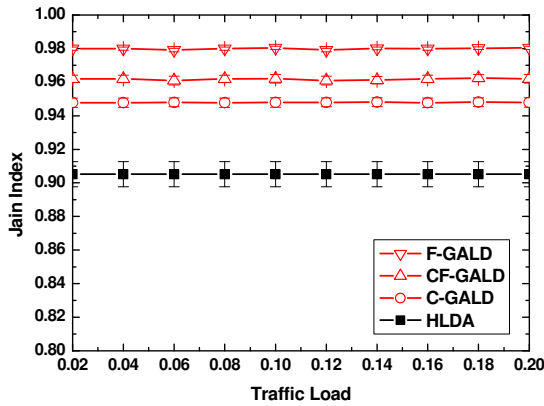
Fig. 7. Fairness of the logical topologies designed by using HLDA, HLDA, C-GALD. F-GALD and CF-GALD .

Fig. 6 show that the congestion of the virtual topologies designed by the four methods grows as the traffic load increases. However, GALD methods always obtain virtual topologies with lower congestion (Fig. 7) and higher fairness (Fig. 8) than those designed with HLDA. In this way, when the mean traffic load is over the 11% of the capacity of a lightpath, the virtual topologies designed by HLDA show a congestion higher than the lightpath capacity and this will cause packet losses. Nevertheless, C-GALD supports traffic loads of up to 0.15 without exceeding the lightpath capacity. Then, the virtual topologies designed by F-GALD, i.e., the version to increase the fairness of the virtual topologies, are the ones that present a higher value of Jain index. Finally, when using the multipurpose version, CF-GALD, there is a trade-off between congestion and fairness, and although the resulting congestion is a little higher than the one achieved with C-GALD, the fairness is better than that obtained with C-GALD.
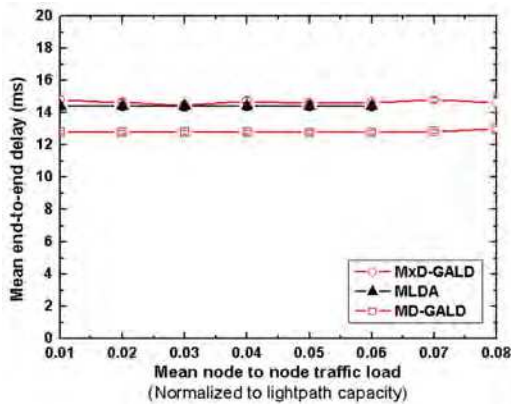

Fig. 8. Mean end-to-end delay of the logical topologies designed by using MLDA, MD-GALD and MxD-GALD. The creation of 500 individuals was the stopping criterion for GALD methods.

Comparing the performance of the versions of GALD to optimize the delay in the same network conditions than the previous study, in Fig. 8, the mean end-to-end delays of the logical topologies obtained with MD-GALD and MxD-GALD are compared with those achieved with MLDA (Ramaswami & Sivarajan, 1996) (an algorithm to design virtual topologies with the objective of minimize the end-to-end delay). The corresponding values of end-to-end delay of the most delayed traffic flow are represented in Fig. 9.
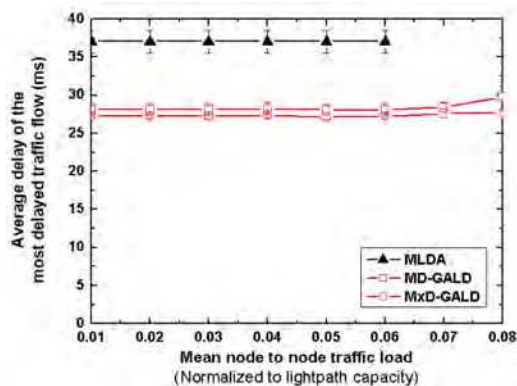

Fig. 9. Average delay of the most delayed traffic flow in the logical topologies designed by MLDA MLDA, MD-GALD and MxD-GALD. The creation of 500 individuals was the stopping criterion for GALD methods.

As shown in Fig. 8, the mean end-to-end delay suffered by the packets in the logical topology designed by all the algorithms has almost the same value for different node-to-node traffic loads because the main component of this delay is due to propagation rather than to processing delays. However, MD-GALD designs virtual topologies that have 10% less average delay than those obtained with MLDA. Moreover, although MxD-GALD is not designed to minimize the average end-to-end delay, it achieves almost the same values than MLDA. Finally, using the network configuration detailed above, the virtual topologies designed by MLDA do not support traffic loads higher than 0.06 (and so the results are not plotted in the figures), while the virtual topologies designed by MD-GALD and MxD-GALD support traffic loads higher than 0.08.

Regarding the average end-to-end delay of the most delayed flow (Fig. 9), both MD-GALD and MxD-GALD design virtual topologies with a significant reduction of this parameter when compared with the one produced by MLDA. In particular, MxD-GALD, which is the most effective algorithm when minimizing this parameter, obtains around 30% reduction when compared to MLDA.

Finally, the advantages of using the multi-objective versions of GALD, mainly PDC-GAPD (i.e., the one with Pareto optimality), will be studied. In Fig. 10 it is shown the results of the virtual topologies obtained by PDC-GALD, DC-GALD, C-GALD, MD-GALD and MLDA assuming a traffic matrix with uniform load of 0.07 between each pair of nodes. Each virtual topology is plotted in terms of its congestion and its mean end-to-end delay. For the genetic algorithms the creation of 1,000,000 individuals was set as the stopping criterion, except for

PDC-GALD. For PDC-GALD we show the results of the initial population (two individuals), the results when 1,000 individuals have been created, and the results when 1,000,000 individuals have been generated
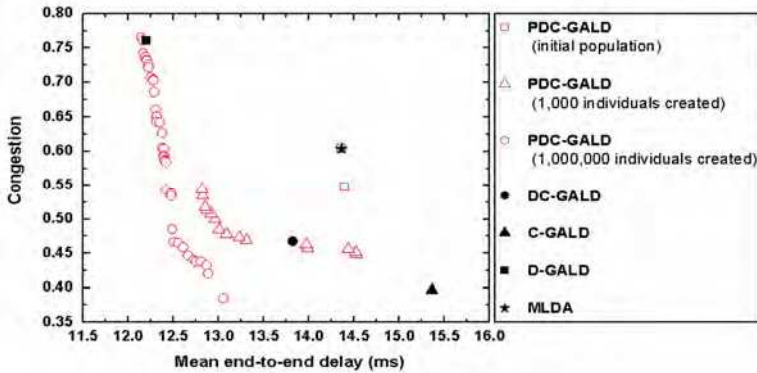

Fig. 10. Mean end-to-end delay and congestion of the logical topologies designed by PDC-GALD, DC-GALD, C-GALD, MD-GALD and MLDA

The first advantage of the method that uses the Pareto optimality in combination with genetic algorithms is that it provides a set of possible virtual topologies (an estimate of the Pareto optimal set) while the other methods only design one (the best topology found until that moment according to the optimization target). This feature of PDC-GALD is very attractive in reconfigurable WRONs, since the network operator can select the virtual topology that better adapts to current network.

Since the first individual of the initial population of PDC-GALD is based on MLDA, obviously PDC-GALD always improves or at least obtains equal results than that algorithm. Moreover, as the number of individuals created increases, the virtual topologies designed by PDC-GALD improve, leading to lower values of congestion and delay. In particular, the virtual topologies found after producing 1,000,000 individuals, do not only get lower values of either congestion or end-to-end delay than the other methods, but they even improve both values, so that PDC-GALD is a very effective method.

Finally, in Fig. 11 and Fig. 12 the congestion and the mean end-to-end delay obtained are compared, respectively, for different traffic loads (randomly generated according to a uniform distribution) for each source-destination pair in the NSFNet with the same configuration than that used in the studies of C-GALD and MD-GALD. The solutions shown for PDC-GALD, are those ones presenting the lowest end-to-end delay (therefore, with the highest congestion) and the ones showing the lowest congestion (hence, with the highest end-to-end delay). The results of the other methods are not shown in order to simplify the figure, but they again got worse results than PDC-GALD.

Fig. 11 and Fig. 12 show that the virtual topologies with lower value of congestion designed by PDC-GALD present a value of this parameter 20% lower than that obtained by the classic multi-objective method, while the end-to-end delay is almost the same for all traffic loads. On the other hand, the solutions with the lowest delay designed by PDC-GALD achieve a

reduction of up to 10% in the mean end-to-end delay while the congestion value of this solution is almost the same than the solution found by DC-GALD. Moreover, PDC-GALD designs other logical topologies with congestions and delays between those values.

Summing up, in multi-objective problems, like the one shown in this book chapter, the combination of Pareto optimality with genetic algorithms can give advantages as it does not only provide a set of solution, but the solutions found by it can be better than those obtained with a classic multi-objective genetic algorithm.
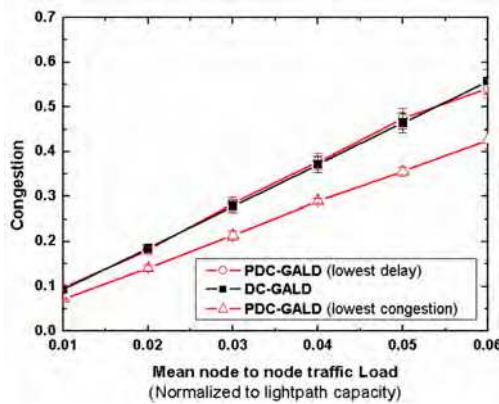
Fig. 11. Congestion of the logical topologies designed by PDC-GALD and DC-GALD
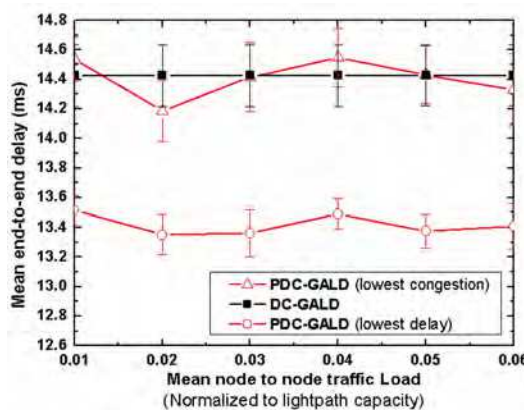

Fig. 12. Mean end-to-end delay of the logical topologies designed by PDC-GALD and DC-GALD

5.4 Genetic algorithms to provision the network and design virtual topologies

In this section, we describe GAPDELT (Genetic Algorithm to Provision the network and to DEsign the Logical Topology) (Durán et al. 2007), a multipurpose method based on the combination of Pareto optimality with genetic algorithms to provision the network and

design the virtual topology using the congestion, the number of transmitters/receivers and the number of wavelengths as optimization criteria.

5.4.1 Chromosome encoding and translation

Each chromosome in GAPDELT is composed of a set of $N(N–1)$ genes, where N is the number of nodes in the network. Each gene is a real number between zero and one and is used to determine how many lightpaths will be established between a pair of nodes. For instance, the gene 0 is used to determine the number of lightpaths that will be established from node zero to node one (see Fig. 13) and so on. Furthermore, the genetic information is used to determine in which order the lightpaths will be established.
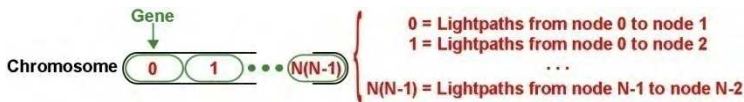

Fig. 13. GAPDELT chromosome structure.

The procedure to translate the genetic information of a chromosome into a network provision scheme and a logical topology follows this pseudo-code:

Step 1.   Search for the gene with the highest value in the chromosome.
Step 2.   Try to establish the lightpath represented by the gene. Precalculated shortest paths (measured in terms of number of hops in the physical topology) are used for routing the lightpaths and the first-fit heuristic (Zang et al., 1996) is employed to assign a wavelength to each lightpath.
　　Step 2.a: If the lightpath cannot be established due to lack of free resources, the value of the gene is set to zero.
　　Step 2.b: If the lightpath is established, a fixed quantity, δ, will be subtracted from the gene. Moreover, a transmitter is assigned to the source node and a receiver to the destination node.
Step 3.   Repeat the process from Step 1 as long as there are free transmitters and receivers and as long as there is any gene in the chromosome with value higher than zero. (Thus, note that δ determines the maximum number of lightpaths allowed between a source-destination pair). Otherwise, the translation process finishes, and the resulting resource distribution and the logical topology are the solution represented by the chromosome.

Moreover another version of GAPDELT is presented to solve both the problem of network provisioning and virtual topology design but also ensuring that all the lightpaths of the virtual topology are protected against network failures. Hence, the optimization objectives of P-GAPDELT are also the same ones that GAPDELT but it modifies the step 2 and it does not only offer a lightpath but it also reserves a set of resources to establish a backup lightpath in case the primary one fails. As it is obvious, the primary and the backup lightpath are link disjoint. In order to minimize the resources that will be reserved, P-GAPDELT uses the technique of backup multiplexing in which the reserved resources for a backup lightpath can be also used to provide the redundancy to additional lightpaths if the primary lightpaths are link disjoint.

5.4.2 Initial population

GAPDELT uses as initial population three chromosomes that are designed to ensure that they show either low congestion or low resource utilization and the rest of individuals of the initial population are randomly generated.

The chromosome of the first individual contains the corresponding values of the traffic matrix normalised by the highest value. This individual is inspired in HLDA and has the same objective as the one used in GALD methods. The second individual is a chromosome in which genes that represent lightpaths between nodes that are adjacent in the physical topology are set to δ, and zero otherwise. The third special individual is designed to create a Hamiltonian cycle as the logical topology. These two individuals ensure solutions requiring few network resources.

Then, the rest of individuals are randomly generated setting each value of a gene to a number randomly generated between zero and one. However another modification has been done to find the optimal solutions with different number of resources quickly. The interval between zero and one is divided into σ intervals (parameter defined by the user) and as many chromosomes as desired are generated in the interval $\left[\dfrac{r}{\sigma}, \dfrac{r+1}{\sigma}\right]$. considering $r = 0, 1, \cdots, (\sigma - 1)$.

5.4.3 Genetic evolution and operators

In GAPDELT, following the principles of GAs, an initial population of individuals of size P is randomly created. The selection of the two individuals that will act as parents in each crossover process is made randomly among the individuals in the population. In single-objective problems the roulette wheel method according to the individual fitness is utilized as selection process but in GAPDELT, every individual in the population belongs to the Pareto optimal set. The crossover operator is applied to pairs of individuals to interchange a part of their genetic material. It uses the same method than GALD: the chromosomes are divided into two parts by a gene randomly selected (the same in the two chromosome) and the second parts of the chromosomes are interchanged to create two new individuals. The process of crossover is repeated until the descendant population reaches a size of DP that will be proportional to the size of the population ($DP = a \cdot P$). When the process of creating the descendant population is completed, the mutation operator is applied to the individuals in the new population. The mutation operator goes through the genes of the new individuals and it changes the value of the genes randomly according to a probability $p_{mutation}$.

Each time that an individual (or virtual topology) is generated, it is necessary to calculate its goodness according to the fitness functions. Then, the solutions that belong to the Pareto optimal set among the individuals from the descendant and the parent populations will form the parent population for the next generation. When using this technique, solutions which exhibit good performance in many, if not all, objectives are more likely to be produced. Note that the size of the population P, and so DP, will dynamically change depending on the number of individuals that belong to the Pareto optimal set. The evolution process is repeated a number of iterations or until another criterion is met.

A new operator, called vitalization, is designed for GAPDELT. The vitalization tries to increase the health of the individuals similarly as it is done when an individual receives a vitamin. The vitalization is only applied in certain generations, e.g., when the number of individuals generated reaches the user-defined values. When the vitalization is applied, the size of the population is set to three times the current size. In the first third of the new population the current population is copied. Then in the second third, the current population is copied but a certain value between zero and one, called *vitamin*, is subtracted from the genes. If the result is lower than zero, the value of the gene is set to zero. This set of chromosomes can increase the health in terms of resources used (less resources). The third part of the new population is composed by the current population but adding the value *vitamin* to the genes of the current population. If this leads to a gene with a value greater than one, the value of that gene is set to one. The objective of this third part of population is to build solutions with many lightpaths established and so the congestion value of the virtual topologies should decrease. Then, the health of all the population is calculated and the individuals belonging to the Pareto optimal set are considered to be the parents for the new generation.

5.4.4 Fitness function

Each time that an individual is generated and its genetic information is translated into a network provision scheme and a logical topology, its fitness has to be calculated in terms of congestion and number of resources required (transmitters/receivers and wavelengths in operation). To estimate the network congestion, the traffic is routed in the logical topology following the shortest paths measured in number of hops, similar to IP routing protocols. Only those solutions with congestion lower than the lightpath capacity are considered as feasible solutions.

5.4.5 Results obtained with the methods

In order to evaluate and measure the accuracy of both GAPDELT and P-GAPDELT, a simulation study is done using the NSFNet as the physical topology, like in the simulations done to evaluate the GALD method. The maximum number of wavelengths was set to eight and the maximum number of transmitters and receivers to 182. The maximum number of lightpaths between two nodes was fixed to four.

The number of individuals in the initial population was set to 13, three of them were calculated as explained in Section 5.5.2, and the other ones were randomly generated using σ = 9 and a chromosome per interval. The size of the descendant population was two times the size of the parent population. The vitalization was applied each time that 500 individuals were generated and the value of the *vitamin* was set to 0.25. Other configurations for each parameter were analysed, but we finally adopted this one since it led to the best results.

As happens with PCR-GALD, GAPDELT provides as solutions all the dominant individuals in the current population, i.e., an estimate of the Pareto optimal set. However, in this case, the solutions can be represented by the group of three parameters: congestion, number of wavelengths and number of transmitters and receivers used. Hence, in Fig. 14 a three-dimensional colour graph has been used to show the fitness of the set of logical topologies

found by the algorithm. Each point in the graph is a solution represented in terms of its number of wavelength, transmitters and receivers in operation and its congestion. In order to show this qualitative result, a random traffic matrix was used in which the traffic demands between a pair of nodes was randomly generated using an uniform distribution with load of 0.1, i.e., 1 Gbps. The creation of one million of individuals was set as stopping criterion.
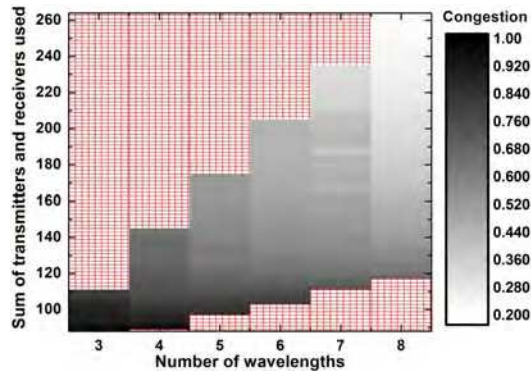

Fig. 14. Fitness of the solutions found by GAPDELT

As it is shown in Fig. 14, GAPDELT obtains in a single execution a high number of feasible solutions. In contrast, if traditional virtual topology design methods were used, it would have been necessary to repeat the execution of these methods a number of times varying the number of transmitters, receivers and wavelengths and also varying the distribution of the transmitters and receivers among the nodes to obtain the same result that GAPDELT can obtain in only one execution.

Moreover, each solution of GAPDELT brings the following information: the number of transmitters in each node of the network (it can be heterogeneously distributed), the number of receivers in each node of the network (it can also be heterogeneously distributed), the number of wavelengths used in the network, the set of lightpaths that should be established in the network, the route and the wavelength that each lightpath has to use and the traffic routing over the virtual topology.

Regarding the results illustrated in Fig. 14, obviously, as the number of resources employed to design the virtual topology increases, the congestion obtained is generally lower. However, as shown by the grated areas in the figure, there are regions where increasing the number of resources does not lead to reductions in congestion, i.e., some resources are wasted in these areas with the corresponding increase in the cost of the network. For instance, in some cases, if the number of wavelengths is increased, but it is not accompanied by an increase on the number of transmitters/receivers, there is no reduction in congestion. GAPDELT detects these situations, leading to solutions with a balanced use of transmitters/receivers and wavelengths, hence optimizing the utilization of resources and improving the cost efficiency of the network.

Fig. 15 shows the number of solutions found, in average, by GAPDELT and P-GAPDELT in a single execution depending on the traffic load. Each value shown has been obtained using 250 traffic matrixes where each value of the matrix has been randomly selected following a

uniform distribution with the mean of the traffic load shown in the figures. Results are shown with 95% confidence intervals. The stopping criterion in GAPDELT was set to the creation of 100,000 individual during the evolution process. As can be seen in the figure, as the traffic load increases, the number of solutions found by both version of GAPDELT decreases because there are less solutions with a congestion value under the maximum lightpath capacity. Moreover, GAPDELT achieves to design more virtual topologies than P-GAPDELT as it can use more resources to establish primary lightpaths.
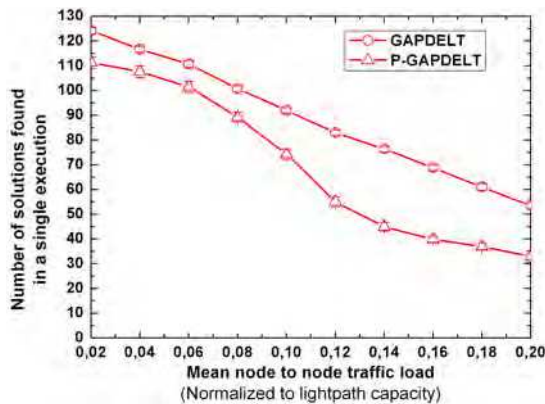
Fig. 15. Number of solutions found, in average, by GAPDELT and P-GALDELT in a single execution

Fig. 16 compares the congestion obtained by GAPDELT, P-GAPDELT, HLDA (when each node is equipped with 13 transmitters and 13 receivers) and HLDA again but using the resource distribution provided by GAPDELT. As GAPDELT methods design solutions with different number of wavelengths used, only the solutions which use eight wavelengths are shown in the figure, and from these solutions, two of them are selected to be plotted: the one with lowest congestion and the one with lowest number of transmitters and receivers. Each value shown has been obtained using 250 traffic matrixes where each value of the matrix has been randomly selected following a uniform distribution with the mean of the traffic load shown in the figures. Results are shown with 95% confidence intervals. The stopping criterion in GAPDELT was set to the creation of 100,000 individual during the evolution process.

The objective of this comparison is to study not only the performance of GAPDELT as a method to design the virtual topology but also to analyse the advantages obtained as a method to solve the network provisioning problem. For higher loads, HLDA cannot find a solution with a value of congestion lower or equal than one (i.e., without exceeding the lightpath capacity) and these are not plotted in the figure.

Firstly, it is possible to see in Fig. 16 that GAPDELT can design virtual topologies for higher traffic loads when HLDA cannot, even when using the resource distribution found by GAPDELT. Moreover, GAPDELT does not only design a set of solutions per execution instead of only one, but it also designs virtual topologies that can reduce in 50% the value of congestion when compared with HLDA. Moreover, the results from Fig. 16 show that the

problem of network provisioning and the design of logical topologies are strongly related. In this sense, in Fig. 16 it is shown that when HLDA uses the resource distributions obtained by GAPDELT, it designs virtual topologies with lower congestion than if a uniform distribution of resources among the network nodes is used. Hence, GAPDELT also offers advantages if it is only used to solve the problem of network provisioning.
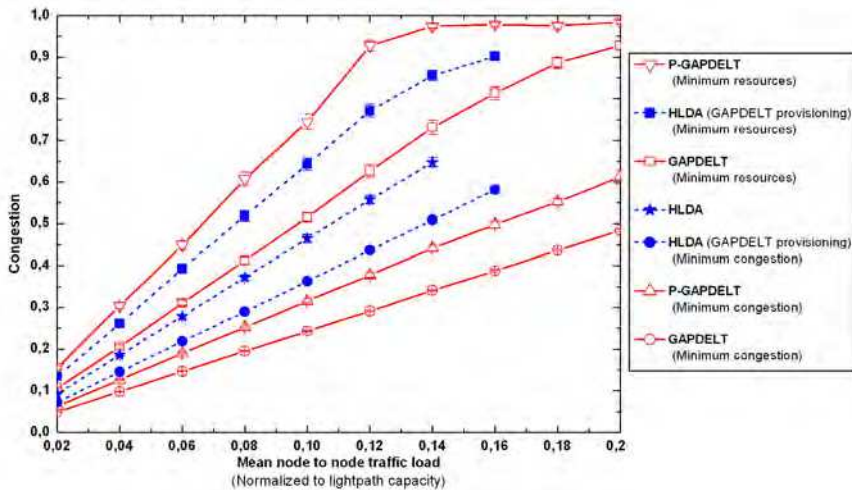

Fig. 16. Comparison of the congestion of the virtual topologies designed by GAPDELT and P-GAPDELT with HLDA for different network loads

As it is obvious, the congestion of the solutions obtained by GAPDELT are lower than those of P-GAPDELT because the former does not need to reserve resources for backup lightpaths and it can establish a higher number of primary lightpaths reducing the congestion but without providing failure protection. However, an interesting result to show the efficiency of the GAPDELT family is that P-GAPDELT can also reduce the congestion obtained by HLDA when the latter is a method to design virtual topologies reducing the congestion and it does offer failure protection.

6. Conclusion

In this chapter, the optimization problems of designing a semi-static wavelength-routed optical network have been described. In this kind of optical networks two problems should be solved: the network provisioning and the virtual topology design. A set of genetic algorithms proposed by the Optical Communications Group at the University of Valladolid to solve these two problem problems have been presented. In this way, GALD methods were proposed to design the virtual topology while GAPDELT methods were created to solve both the network provisioning and the virtual topology design problems. The chromosome structures, their translation, the optimization goals and the genetic operators utilized by each method have been described.

A number of versions of GALD have been developed, each one to optimize a network parameter related with the network capacity (C-GALD, F-GALD) or the delay (MD-GALD,

MxD-GALD). Moreover, a multi-objective method, PCR-GALD, based on the combination of genetic algorithms with Pareto optimality has also been presented. The results from a simulation study show that GALD methods are very good techniques to design virtual topologies as they obtain virtual topologies with better performance than those designed by other methods from the literature.

GAPDELT is a multi-objective algorithm to design both the network provisioning and the virtual topology that minimises the congestion and the number of required resources (in terms of number of transmitters, receivers and wavelengths) and thus the network cost. The method is based on the combination of genetic algorithms and Pareto optimality. Moreover, another version of the method called P-GAPDELT has also been presented to solve the same problems than GAPDELT, but reserving resources to establish backup lightpaths to replace the primary ones when facing network failures. By means of a simulation study, we have shown that both GAPDELT and P-GAPDELT provide a set of solutions, the set of optimal or near-optimal solutions, in only one execution. Moreover, the simulation study done shows that the performance of the solutions found by GAPDELT methods outperform the ones obtained with other methods previously proposed in the literature.

7. Acknowledgment

8. References

Durán, R.J. et al. (2007). Multiobjective genetic algorithm to design cost-efficient wavelength-routed optical networks. *9th International Conference on Transparent Optical Networks Proceedings*, vol. 4, (Rome, Italy), pp. 96−99.

Durán, R.J. et al. (2009a). Genetic algorithm to design logical topologies in reconfigurable WDM networks. *Photonic Network Communications*, vol. 17, no. 1, pp. 21-33.

Durán, R.J. et al. (2009b). Minimization of end to end delay in reconfigurable WDM networks using genetic algorithms. *European Transactions on Telecommunications*, vol. 20, no. 8, pp. 722-733.

Durán, R.J. et al. (2009c). Joint Optimization of Delay and Congestion in Wavelength-Routed Optical Networks Using Genetic Algorithms. *Photonic Network Communications*, vol 18, no. 3, pp. 233-244.

Dutta, R & Rouskas, G.N (2000). A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks Magazine*, vol. 1, pp. 73-89.

Ghose, S. et al. (2005). Multihop virtual topology design in WDM optical networks for self-similar traffic. *Photonic Network Communications*, vol. 10, no. 2, pp. 199−214.

Goldberg, D.E. (1989). *Genetic algorithm in search, optimization and machine learning*, Addison-Wesley Professional.

Johnsonbaugh, R. & M. Schaefer, Algorithms, Pearson Education, Inc., 2004.

Man, K.F. et. Al. (1999). *Genetic algorithms*, Springer-Verlag.

Mukherjee, B. et al. (1996). Some principles for designing a wide-area WDM optical network. *IEEE/ACM Transactions on Networking*, vol. 4, no. 5, pp. 684−696.

Mukherjee, B. (1997). *Optical Communication Networks*, McGraw-Hill, 1997.

Ramaswami, R. & Sivarajan, K.N.(1996) Design of logical topologies for wavelength-routed optical networks. *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 840-851

Zang, H. et al. (2000). A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks, *Optical Networks Magazine*, vol. 1, no. 1, pp. 47-60.
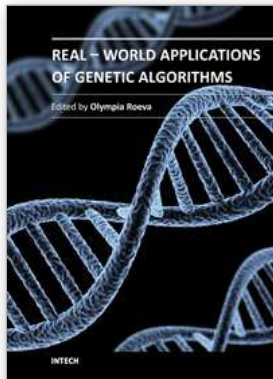
Real-World Applications of Genetic Algorithms

Edited by Dr. Olympia Roeva

The book addresses some of the most recent issues, with the theoretical and methodological aspects, of evolutionary multi-objective optimization problems and the various design challenges using different hybrid intelligent approaches. Multi-objective optimization has been available for about two decades, and its application in real-world problems is continuously increasing. Furthermore, many applications function more effectively using a hybrid systems approach. The book presents hybrid techniques based on Artificial Neural Network, Fuzzy Sets, Automata Theory, other metaheuristic or classical algorithms, etc. The book examines various examples of algorithms in different real-world application domains as graph growing problem, speech synthesis, traveling salesman problem, scheduling problems, antenna design, genes design, modeling of chemical and biochemical processes etc.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

R.J. Durán, I. de Miguel, N. Merayo, P. Fernández, J.C. Aguado, A. Bahillo, R. de la Rosa and A. Alonso (2012). Genetic Algorithms for Semi-Static Wavelength-Routed Optical Networks, Real-World Applications of Genetic Algorithms, Dr. Olympia Roeva (Ed.), ISBN: 978-953-51-0146-8, InTech, Available from: http://www.intechopen.com/books/real-world-applications-of-genetic-algorithms/genetic-algorithms-for-semi-static-wavelength-routed-optical-networks

INTECH

open science | open minds