Survivability on Optical Networks: Protecting the Protection

Alisson Barbosa de Souza, Ana Luiza de B. de P. Barros,
Antônio Sérgio de S. Vieira, Jéssyca Alencar L. e Silva, Joaquim Celestino Júnior
Computer Networks and Security Laboratory (LARCES)
State University of Ceará (UECE)
$\{$*alisson, analuiza, sergiosvieira, jessyca, celestino*$\}$*@larces.uece.br*

***Abstract*—Survivability in WDM optical networks has been a subject of great interest recently. The idea used over the years is to establish a protection path so that, in case of failure, this path can be used. However, these proposals take into account neither the real situation of these paths nor the requirements of the applications. In this article an approach is proposed that takes said requirements into account, using genetic algorithms and fuzzy logic, additionally considering wavelength and SRLG (Shared Risk Link Group) constraints.**

***Keywords*-optical networks; genetic algorithms; fuzzy logic; srlg;**

I. INTRODUCTION

Optical fiber is a revolutionary means of transmission. Its low error rates, its transmission capacity, and its immunity to noise are considered to be responsible for its growing popularity. More and more optical fiber networks are linking to other networks and being used in big corporate networks. Networks like these must work $24/7$ to support applications that require unbroken processing, such as applications in airline companies, hospitals, banks, supermarkets, and many others.

WDM (Wavelength Division Multiplexing) technology allows optical networks to support speeds close to terabits per second, and consequently any interruption could lead to the loss of enormous quantities of data, thus harming the applications. In order to avoid these problems safe and reliable protection techniques are necessary.

One of the techniques in use on optical networks to try and guarantee that their services are uninterrupted is the use of protection paths. In this technique, previously defined, alternative routes (protection paths) are employed so that, in case of any failure along the main path, the flow remains undamaged because traffic is quickly redirected to the protection path.

Another known technique, restoration, does not previously allocate network resources, and a protection path is not created prior to failure. After detecting failure in the main path, alternative routing is requested by means of specific protocols. Then traffic can be redirected to the new route established.

Both techniques have advantages and disadvantages, and it is up to the administrator to choose the one best for his needs. This article proposes a method to try to assure that a pre-established alternative protection route is adequate in case of failure in the main path. Traffic must be detoured quickly, and the links in the protection path must have low bit error rates (BER) and also be different from the links in the main path (link-disjoint).

Since WDM optical networks without lambda converters were used in this article, another question to be considered is the continuity of the wavelength, where a lambda must be available end-to-end.

Commonly, network operators consider recovery time of a failure to be 50 ms or less [1], raising difficulties in the search for an adequate solution. In the worst case a breakdown may occur in the main route, and at the same time a deficiency is detected in the protection path. Thus, the method must locate a good solution within a maximum of 50 ms.

To solve this problem, exact algorithms, such as Dijkstra, may be inadequate because they only obtain a solution in reasonable computational time for small instances [2], characterized as NP-complete problems. In a network with many links, where each link may have m fibers, and each fiber has n lambdas, the time necessary to determine the best solution using an exact algorithm may surpass 50 ms by far. In general, to achieve the solution to this type of problem, metaheuristics are employed, and while these do not guarantee the optimal, they offer low resolution time

[3].

This work uses Genetic Algorithms with the support of Fuzzy Logic, allowing the method to quickly reach a solution respecting the limit of 50 ms, at the same time that it tries to overcome the constraints posed by the problem.

The work is organized as follows: in Section 2 related works are presented. Definition and formulation of the problem are in sections 3 and 4, respectively. Description of the method is presented in Section 5. Next, analyses and results are presented in Section 6. Finally, the article comes to a conclusion in Section 7.

II. RELATED WORKS

Due to the NP-complete characteristic of these problems, several ways of solving them have been proposed. In most of the works found, maintenance of the protection path in not considered.

The work undertaken by [4] developed a heuristic to allocate pre-planned protection paths in a WDM network, minimizing excessive resource capacity. The scheme proposed in our article also takes into account minimization in the use of network resources, however, metaheuristics are used to allocate new protection paths considering class of service.

[5] analyzes the calculation of pairs of disjoint paths with wavelength constraint. This study shows that the link-disjoint and node-disjoint versions of the problem are NP-complete. The article uses two algorithms, one of which is approximative and the other exact, to find the paths. In contrast, our scheme uses a metaheuristic, GA, and also uses lambda continuity constraints, BER, type of protection and shared risk link group constraints.

[6] proposes a protection scheme considering SRLG constraint, which develops a heuristic based on APF (Active-Path-First) on a WDM optical network with lambda conversion, and an improvement is observed in the blocking probability rate. However, neither the quality of the paths nor their types of protection are analyzed. Our work also considers SRLG constraint, however, there is no lambda conversion.

III. DEFINITION OF THE PROBLEM

The problem to be dealt with consists of ensuring, for a WDM optical network without lambda conversion, an SRLG-disjoint and link-disjoint pathway that serves the requirements of the classes of service regarding low error rates and types of protection.

In a WDM optical network each fiber has a set Λ with n wavelengths $\Lambda = \{\lambda_1, \lambda_2, \lambda_3, ..., \lambda_n\}$. In a network with no Lambda converters, an origin-destination path will only be established if the same λ_i is available end-to-end. Furthermore, to try and ensure survivability of a WDM optical network with no lambda conversion, the main path must be link-disjoint from the protection path.

The SRLG value identifies the risk group of a link. A failure in an element of the group causes an interruption in the functions of the other elements. Therefore, it is also necessary that the protection path be link-disjoint from the risk groups of the links in the primary path.

Consider a scenario where there are several paths established through an optical network, and one of them has an exclusive protection path $(1 : 1)$. In this case, although it uses more network resources, traffic will be quickly rerouted to its protection path should there be any failure along the main path (a broken fiber, failure in the switch energy source, etc.). However, there is no assurance that the protection path will be capable, at that instant, of receiving the rerouted traffic, since, at that moment, this route may be presenting problems in the bit error rate, for example.

We consider a protection path adequate when it attends the requirements of the applications and is capable of carrying out the SLA - Service Level Agreements. For example, a client who contracts service for VoIP traffic must have assurances that even if there is a breakdown on the main route, a protection route is capable of taking on the delay and BER restrictions peculiar to a voice traffic application.

The solution to this problem requires that a new protection route be disjoint from the primary route, and disjoint from the risk groups of the main path.

If we consider E to be the set of links on the protection path, P to be the set of links on the main path, and S to be the set of links that are part of the same risk group as the links on the primary path, then $E \cap (P \cup S) = \emptyset$. Figure 1 shows, as an example, a secondary route link-disjoint and SRLG-disjoint from the primary route. In this example, $E = \{a{-}b, b{-}c, c{-}d\}$, $P = \{a - e, e - d\}$ and $S = \{b - e, e - g\}$, where risk groups $R1 = \{e - d, e - g\}$ and $R2 = \{a - e, b - e\}$ have links in sets P and S.

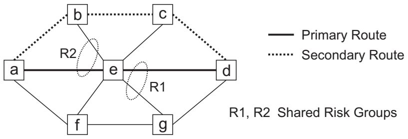


Figure 1. Example of link-disjoint and SRLG-disjoint Primary and Secondary Routes

BER values and types of protection are used to identify a SLA. Following what was proposed in [7], four classes of service were defined, and a range of BER was assigned to each one. Table 1 shows the values chosen for each class.

Table I
CLASSES OF SERVICE ACCORDING TO BIT ERROR RATE

Class of Service	Gold	Silver	Bronze	Best-Effort
Bit Error Rate	$\leq 10^{-8}$	$\leq 10^{-7}$	$\leq 10^{-6}$	$> 10^{-6}$

According to [5], the problem of selecting a disjoint route, taking the lambda continuity constraint into consideration, is NP-complete. In this article, apart from the same constraints, SRLG constraints are also considered. One can therefore conclude that the problem approached here can also be considered NP-complete.

IV. FORMULATING THE PROBLEM

A WDM network is modeled as a non-oriented connected graph $G = (V, E, \Lambda, S)$, where the set of vertices $V = \{v_i | i = 1, 2, 3, ..., n\}$ represents the nodes of the net, the set of edges$E = \{e_{ij} | v_i, v_j \in V\}$ represents the links, and e_{ij}^k is a fiber k that connects nodes v_i and v_j . Each link can have m fibers, in each fiber there is a set of lambdas $\Lambda = \{\lambda_1, \lambda_2, \lambda_3, ..., \lambda_n\}$, and for each lambda a BER value $BER \in [0, 1]$ is defined, as well as a type of protection and whether or not it is available.

The type of protection defines characteristics related to the use of a lambda. Three types are defined: (1) Never, (2) Shared and (3) Only. A lambda with a Never type of protection is never used in any secondary route, and can be fully utilized in a new protection path ($1 : 1$, $1 + 1$ ou $1 : n$). A lambda with Shared type protection is used in a shared secondary route, and may also be used in the solution (1 : n). An Only type lambda is used exclusively by a primary route and may only be chosen if no other solution is possible.

Last, $S_{ij} = \{s_1, s_2, s_3, .., s_n\}$ defines the set of risk groups e_{ij}, where S_{ij} identifies the risk groups to which e_{ij} belongs.

A solution is considered more adequate if it consists of lambdas with a low BER value and Never type protection, and its links must be link-disjoint and SRLG-disjoint from the primary route.

V. DESCRIBING THE PROPOSAL OF A SOLUTION

In order to solve the problem of choosing a new protection route, taking the mentioned restrictions into consideration, we opted for using Genetic Algorithms (GA) and Fuzzy Logic. The use of metaheuristics, in this case, is justified by the fact that it is an NP-complete problem [5], and therefore cannot be resolved in polynomial time. The use of GA provides a relatively simple manner of representing several metrics intrinsic to the problem, and comparing the solutions using a fitness value. Furthermore, several GA configuration parameters (Mutation Rate, Crossover Rate, Stop Criterion, and Maximum Generations) can be used to find a solution that meets the requirements of our problem, such as a time period below 50 ms. Opting for the use of Fuzzy Logic is appropriate since it better represents the contribution value of the BER to the gene, as previously demonstrated in [8].

The first measure taken is to reduce the space for the search of a solution, in other words, to find a subset of the set of all possible routes that comply with the constraints of the problem.

Allow U to be the set of all the links in the topology, $P \subseteq U$ to be a subset of links in the primary route, and $R \subseteq U$ to be a subset of links SRLG-non-disjoint from P. Before starting the search, all the elements of U that are part of P and R are removed, as are all lambdas of the elements with a high BER value. This ensures that the solution will be link-disjoint and SRLG-disjoint from the main path. Therefore, the set $U' = U - (P \cup U)$ represents the new search space.

A gene, in this proposal, is a tuple composed of an edge$< u, v > \in U'$ and of a λ that belongs to the set of lambdas of the edge $< u, v >$. Therefore, a chromosome is a set of genes $G = \{g_1, g_2, g_3, ..., g_n\}$. A chromosome will be considered invalid, that is, its fitness value will equal zero, if it does not possess the same lambda in all its genes (restriction for networks without lambda conversion), or if it does not form a valid origin-destination route.

In this article, the fitness value of a chromosome c with lambda j, for the valid cases, will be calculated according to the formulations below:

$$fitness_c^j = \frac{\dfrac{\sum\limits_{i=1}^{n} \alpha.f(ber_j^i) + (1-\alpha).g(prot_j^i)}{f(0) + g(Never)}}{n^2} \quad (1)$$

$$f(x) = defuzzy(x) \quad (2)$$

$$g(x) = \begin{cases} 0,50, & \text{se } x = Never \\ 0,49, & \text{se } x = Shared \\ 0,01, & \text{se } x = Only. \end{cases} \quad (3)$$

In (1) n is the quantity of genes in chromosome c. The value of $\alpha \in [0,1]$ defines the weight of the BER and type of protection in the problem. The expression $f(0)+g(Never)$ represents the highest contribution value of a gene (optimal link). Thus, the result of $\frac{\sum_{i=1}^{n} \alpha.f(ber_j^i)+(1-\alpha).g(prot_j^i)}{f(0)+g(Never)}$ is the contribution value relative to the gene's maximum contribution value. Dividing the sum by n^2 is used to give the smaller chromosomes a larger fitness value, minimizing the use of network resources.

Function (2) returns a value for the type of protection of lambda j of gene i, according to [8].

Function (3) returns a value for the type of protection of lambda j of gene i. The value of $g(x)$ influences the result of the problem by giving a higher contribution value to the genes that have Never or Shared type protection.

A. Genetic Algorithm

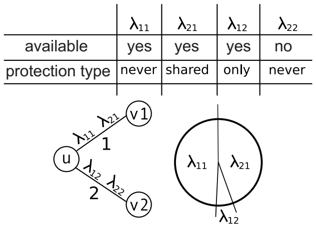
This subsection presents the operation of the mechanisms that make up the GA, each of which was developed considering the constraints imposed by the problem. The mechanisms developed are: (I) Initial Population, (II) Natural Selection, (III) Crossover, and (IV) Mutation.

The initial population (I) has an important role in this problem. To trust just mutation and crossover does not ensure that the search will converge on a good solution because, as previously mentioned, converge time cannot surpass 50 ms. An initial population that contains good solutions can help the GA to find the best solution more rapidly than using a random initial population [3] [9] [10]. Therefore, the method of creating the initial population must generate the largest quantity of good solutions in the shortest possible period of time. In this method two approaches were used. The first uses a biased roulette wheel and the second a fair roulette. In both, V is the set of vertices adjacent to vertex u.

In the biased roulette the value of $\Gamma_{v \in V}(u, v_j)$ (4) represents the percentage of the area of the roulette where, from vertex u, adjacent vertex v_j can be chosen. Still in (4), the value of n represents the quantity of lambdas of the fiber (edge) $< u, v_j >$, and m the quantity of edges incidental to u. Function $h(x)$ (5) will return 1 if λ_i is available, and 0 otherwise. The value returned will be multiplied by the weight given to the type of protection (3) of λ_i, thus, a larger area on the roulette will be given to the links that have more available lambdas and the type of protection with greater weight. Figure 2 shows the idea of the biased roulette's setup.

$$\Gamma_{v \in V}(u, v_i) = \frac{\sum\limits_{i=1}^{n} h(\lambda_i).g(\lambda_i)}{\sum\limits_{j=1}^{m}\sum\limits_{i=1}^{n} h(\lambda_{ij}).g(\lambda_{ij})} \quad (4)$$

$$h(x) = \begin{cases} 1, & \text{if } x \quad \text{is available} \\ 0, & \text{if } x \quad \text{is not available} \end{cases} \quad (5)$$



	λ_{11}	λ_{21}	λ_{12}	λ_{22}
available	yes	yes	yes	no
protection type	never	shared	only	never

Figure 2. Biased Roulette Wheel

The operation of the fair roulette is when the roulette is partitioned into equal areas for each lambda available between vertices u and v_j. Therefore, if two fibers (links) exist, containing two lambdas each, the roulette wheel will be divided into four equal parts. Figure 3 shows the idea of the fair roulette's setup.

A route in the initial population is created starting from the vertex of origin, and the adjacent vertex is chosen by one of the roulettes. For each new vertex chosen, the prior procedure is conducted until the route

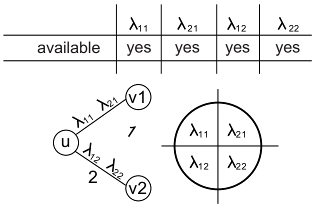


Figure 3. Fair Roulette Wheel

to the destination is found, or until there are no more vertices to choose. Last, a lambda is attributed to the route created, using an attribution algorithm (FirstFit or MostUsed).

When the initial population has been totally created it will be composed of valid and non-valid routes. Among those that are valid there will be solutions (chromosomes) with different lambdas. To better organize the chromosomes in the initial population, and to guarantee that a crossover will only happen between chromosomes with the same lambda, lambda groups were created. A group is created for each lambda in the network. Thus, if there are 4 lambdas, there will be 4 groups, and each group will be made up of solutions with lambda identical to the group lambda. Algorithm (1) shows how the chromosomes in the initial population are created.

The method of natural selection uses the lambda groups formed. For each selection the best solution and another random solution from the same group

are chosen. For n groups and m selections, with $m > n$, the first selection uses group $\lambda_{i=1}$, the second uses group m selees, com $m > n$, a primeira seleo utiliza o grupo $\lambda_{i=1}$, a segunda o grupo $\lambda_{i=2}$, and so on, successively, until group $\lambda_{i=n}$. For selections where $i > n$, groups from $\lambda_{i=1}$ to $\lambda_{i=n}$ will be used again. This method is responsible for choosing the best chromosomes to be used in crossover and mutation. This ensures that the best solutions will not be lost as new generations are created.

Mutation is carried out in the best chromosome (route) of a group. An alternate route is attempted from vertex b of chromosome 1, to vertex c, destination of the route, thus generating a new solution: chromosome 2. Next, one tries to attribute a new wavelength to chromosome 2. If the fitness of the new chromosome is greater than that of the original chromosome, the former is inserted in the next generation group, if not, the original chromosome is included. Figure 4 shows an example of mutation.

Algorithm 1 Initial Population

```
Input: $U'$, source, destiny, population size
Output: set $C$ of chromosomes - initial population
  createGroups($U'$)
  $u \Leftarrow$ source
  $V \Leftarrow \{u\}$
  for $i \Leftarrow 1$ to population size do
    while $u$ is different of destiny or don't have available adjacent vertex of $u$ do
      roulette $\Leftarrow$ createRoulette($u$)
      the incidents edges of $u$ not may be choosen at create a roulette
      $u \Leftarrow$ nextVertex(roulette)
      $V \Leftarrow V \cup \{u\}$
    end while
    $c \Leftarrow$ createChromosome($V$)
    assignLambda($c$)
    addAtGroup($c$)
  end for
  return groups
```

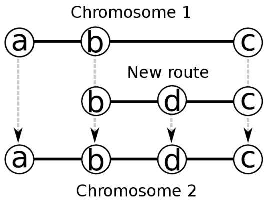


Figure 4. Example of Mutation

Crossover is carried out between the best individual and another random one from the same group returned by natural selection. First, one tries to locate identical vertices in both chromosomes to carry out the crossover. In the case of Figure 5, chromosomes 1 and 2 have vertex b in their genes. Son 1 will be made up of the vertices of origin as far as vertex b of chromosome 1, and by the subsequent vertices from b to the destination of chromosome 2. Similarly, son 2 will be made up of the vertices of chromosome 2, from the origin as far as b, and by the vertices of chromosome 1 after b as far as the destination. If there are no identical vertices in the chromosomes, the best individual is passed to the next generation group. Note that because they are chromosomes of the same group, a new attribution of λ is not necessary.

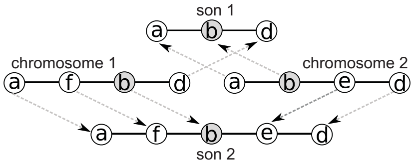


Figure 5. Example of Crossover

VI. ANALYSES AND CONCLUSIONS

This section presents the results and analyses of the tests conducted. The experiments were carried out using the hypothetical scenario of a network for South America [11], which is composed of 52 nodes and 99 links. Each link has a specified type of protection (Only, Shared, Never), a corresponding BER value, and whether or not it is available.

In this scenario, close to 70% of the lambdas have adequate BER values (approximately 10^{-9}) to provide quality of service for the Gold, Silver and Bronze classes, and the remainder have a high BER value ($> 10^{-3}$). Furthermore, some links are SRLG conjunct to the links of the main path to make it more difficult to choose sub-optimal solutions.

In the tests conducted in this article four factors were taken into consideration: solution search time, solution fitness, the type of roulette used to create the individuals, and blocking probability.

Furthermore, a comparison is made between the quality of the route selected by this algorithm, and the quality of the route calculated by the Dijkstra algorithm.

Initially the breach of a prerequisite, high BER, was simulated in the protection path of a certain route. To accomplish this, at a certain moment of the simulation a value of BER $> 10^{-3}$ is specifically attributed for the lambda of the protection path of Silver traffic (which requires BER $< 10^{-7}$). Thus, the algorithm will search for a new protection route.

For the experiments involving search time for a new protection route, some parameters that affect the algorithm's response time were analyzed: population size and type of roulette. In Figure 6 one can see that in a population of 512 individuals average time is about 50 ms, and for a population of 256 individuals this average is about 25 ms. To ensure that the total time of the algorithm does not surpass 50 ms, a population of 256 individuals was opted for.

In the experiments two types of roulette were used, biased and fair, as detailed in Section 5.1. It was verified that the biased roulette takes less time. Since the fair roulette randomly chooses the links that will be part of the population, the individuals created may be made up not only of shorter routes that will take less time, but also of very long routes that will take longer to be set up, so it takes longer to calculate the route. However, the biased roulette will tend to choose the same individuals more often, with the same setup time.

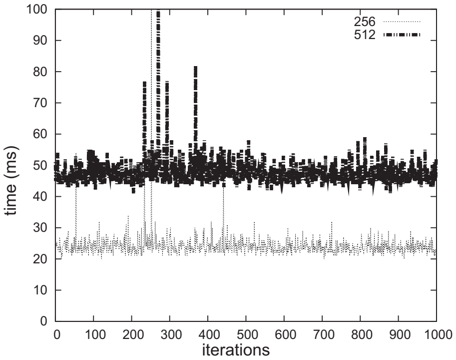


Figure 6. Times for different population sizes

Furthermore, it was noted that the biased roulette is a few milliseconds (approximately 3 ms) slower than the fair roulette. This is due to the fact that with the biased roulette, at the moment of setting up the route, for each current node the parameters of the adjacent lambdas are considered, and the roulette is created based on said parameters, while with the fair roulette no parameter is analyzed and the route is created quickly. However, using the biased roulette to create the chromosomes of the initial population obtained a gain in the quality of the solutions found, as described below.

To state that a solution is the best or the worst of all, one needs to compare this solution to all the other solutions. This is not carried out in polynomial time; there is an exponential number of comparisons of possible route combinations similar to the problem of the travelling salesman [12]. In order to evaluate the quality of the routes in polynomial time, a new criterion was applied that uses a sample of the possible solutions. The algorithm is executed several times, and at each execution the solution is stored in a vector. At the end the vector is queued by fitness value. In this manner, the solutions with the lower rates in the vector are those with the smaller fitness values. Since the definitions of best, average and worst are somewhat imprecise when one cannot compare all the possible solution combinations, we decided to divide the vector into three equal parts. A solution is said to be worst if it is in the initial part of the vector, average in the second part, and best in the final part of the vector.

In fact, analyzing the results obtained, the individuals placed in the final portion of the vector, with the best fitness values, have the lowest BER values, good types of protection, and are short routes.

As seen in Figures 7 and 8, 1000 experiments were conducted for each configuration of parameters, and the quantities of best, average and worst solutions were tallied. It was noted that in the configuration used by the GA, the algorithm turns out to be efficient, and most of the times returns the best solutions. Apart from that, using a biased roulette there is a larger rate of choice of the best routes.

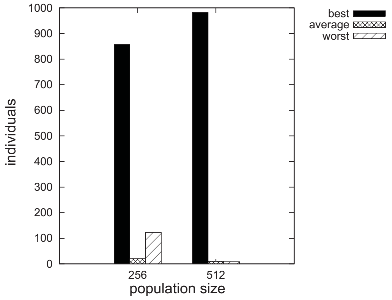


Figure 7. Quality of returned solutions (Biased Roulette)

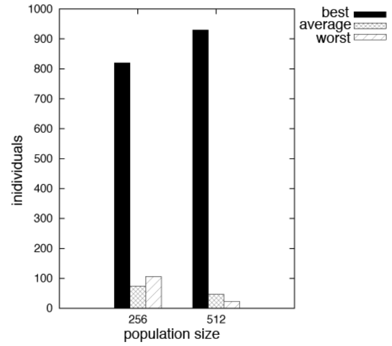


Figure 8. Quality of returned solutions (Fair Roulette)

In order to validate the effectiveness of the algorithm, a comparison was carried out between the solution returned by the algorithm developed in this work (DisjointSchema) and the solution obtained by the Dijkstra algorithm. Figures 9 and 10 show some of the results.

In the graph in Figures 9, the x axis represents the quantity of iterations carried out, and the y axis is the fitness value of the solution found.

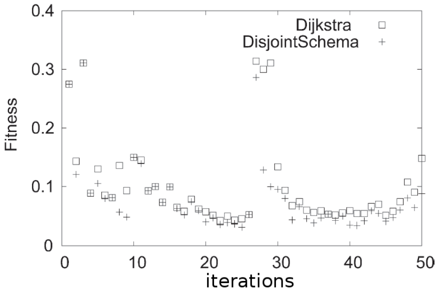


Figure 9. Dijkstra x DisjointSchema

The results in the graph in Figure 10 were obtained by randomly choosing a destination for a certain fixed origin, and the quantity of lambdas in each link varied from 10 to 100. As one can see, the Dijkstra algorithm takes longer to return the route than the DisjointSchema, regardless of the quantity of lambdas in the links. Furthermore, the DisjointSchema is capable of finding a route in less than 50 ms for all quantities of lambdas, while Dijkstra surpasses this time already in tests with 40 lambdas.

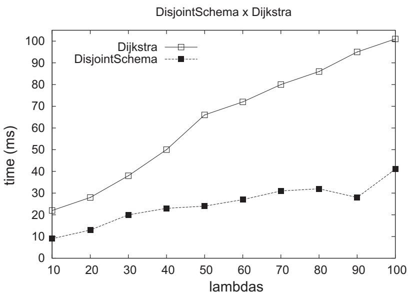


Figure 10. Dijkstra x DisjointSchema - Time

The graphs display the optimal values obtained by the Dijkstra algorithm in comparison with the solutions achieved by the DisjointSchema. On average, the algorithm proposed in this article returns solutions 81.40%

of optimal, however with the advantage of returning the result in less computational time than the time required by the Dijkstra algorithm.

VII. CONCLUSION

The article describes a scheme for choosing the best link-disjoint and SRLG-disjoint protection route in optical networks with lambda continuity constraints, by using Genetic Algorithms with Fuzzy Logic support.

The proposed scheme achieves quite satisfactory results, contributing to network survivability, at the same time that it seeks to satisfy the requirements of the applications. The protection routes found by the method possess links with low BER values and Never type protection. Furthermore, they do not share the same risk group and the links of the main path. It was also demonstrated, through comparisons, that the results are very close to the optimal solution found by the Dijkstra algorithm, with the advantage of returning the result in less computational time.

Another important contribution of this work is in the form that the routing is carried out with GA. The success of the method is directly related to the method of creating the initial population, since it is able to rapidly create good solutions for the problem. Furthermore, the group scheme used in the initial population allows comparing the fitness of any type of route, even routes that have the same links but use different lambdas, in order to choose the best solution.

FUTURE WORKS

Future works include the use of other metaheuristics to solve the problem. Apart from that, other parameters characteristic of optical networks will be used. Finally, new optical transmission technologies, such as OBS (Optical Burst Switching), will be used.

REFERENCES

[1] J.P. Vasseur, M. Pickavet, P. Demeester, **Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS**, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2004.

[2] R.Sedgewick, **Algorithms**, Addison-Wesley, 1988.

[3] F. Glover and G. Kochenberger, **Handbook of Metaheuristics, volume 57 of International Series in Operations Research & Management Science**, Kluwer Academic Publishers, 2003.

[4] G. Maier, A. Pattavina, S. De Patre, M. and Martinelli, **Optical Network Survivability: Protection Techniques in the WDM Layer**, Photonic Network Communications, v. 4, number 3, pp. 251–269, Springer, 2002.

[5] R. Andersen, F. Chung, A. Sen, and G. Xue, **On disjoint path pairs with wavelength continuity constraint in WDM networks**, INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, v. 1, 2004.

[6] Z.L. Tang and X.M. Li, **A Mixed Shared and Multi Paths Protection Scheme with SRLG Constraints**, Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)-Volume 01, pp. 60–65, 2007,IEEE Computer Society Washington, DC, USA.

[7] C. Pinart, **Alternatives for in-service BER estimation in all-optical networks: towards minimum intrusion**,JOURNAL OF COMPUTERS, v. 2, number 3, pp. 55, 2007.

[8] J. Celestino Jr., A.B. Souza, A.S.S Vieira, J.A.L. Silva, A.L.B.P. Barros, G.A.L. Campos, and L.W.N. Mendouga, **GAFUPA: Uma Proposta para Proteo em Redes pticas**,XXVI Simpsio Brasileiro de Telecomunicaes - SBrT, Rio de Janeiro, RJ, 2008.

[9] C.R. Reeves, **A genetic algorithm for flowshop sequencing**, Computers and Operations Research, v. 22, pp. 5–13, Elsevier, 1995.

[10] R.K. Ahuja, and J.B. Orlin, **Developing Fitter Genetic Algorithms**, INFORMS Journal on Computing, v. 9, pages 251–253, 1997.

[11] J. Celestino Jr., A.B. Souza, A.S.S Vieira, J.A.L. Silva, A.L.B.P. Barros, **A Management Scheme of SRLG-Disjoint Protection Path**, International Symposium on Integrated Network Management, New York, USA, 2009.

[12] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, **Introduction to Algorithms**, MIT Press, 2001.

[13] A. Wason and R.S. Kaler, **Wavelength Assignment Problem in Optical WDM Networks**, IJCSNS, v.7, pages 27, 2007.