Heuristic Algorithms: Novel Solution for RWA Problem in WDM Optical Networks

A. Rodríguez[1], L. Ramirez[2] and J. Martinez[3]
[1]Universidad Santiago de Chile, Department of Industrial Technology, Research Group New Technologies (GINT), Santiago, Chile.
[2]Universidad Militar de Nueva Granada, Division of Technology Development and Innovation Research Group on Telemedicine (TIGUM), Bogotá, Colombia.
[3]Universidad Santiago de Chile, Vicerrectory of Research, Development and Innovation, Santiago, Chile.
arturo.rodriguez@usach.cl

***Abstract*—A new metaheuristic based on the Snake One algorithm is presented and it is compared with different heuristics for the NSFNET network. It is based on monitoring the congested nodes, but with sufficient coverage to satisfy the request for service. That condition does not improve essentially the blocking probability, but it does so in terms of the use of the network's resources. In the case of the blocking probability, this metaheuristic performs better in optical networks that support traffic loads up to 120 erlangs, and in the case of the use of the network it operates much better than its predecessors Snake One and Snake Two, using fewer network resources for the same traffic load.**

***Index Terms*—Heuristics Algorithm; NSFNET; Optical Networks; RWA; Snake Algorithms.**

I. Introduction

From their beginnings, optical networks have required a different treatment than that of conventional networks, due mainly to their high speed (2.5 Gbps), which contrasts with electronic speeds, which reach the order of Mbps. Electronic routing devices developed a series of algorithms such a Dikstra, Bellman-Ford, Floyd Warshall, Ford-Fulkerson, etc., which can be called conventional from the optical perspective; those algorithms searched route optimization based on some metric like the number of jumps, the time delay of the connection, etc. However, these algorithms are unable to achieve optimization if the solutions universe has high variability [1-3].

The scenarios whose universe of solutions is dynamic cannot be solved by conventional optimizing algorithms. That would require a heuristic that allows finding several solutions. In the literature there are heuristic algorithms such as Genetic, Simulated Annealing, Tabu Search, Firefly, among others. Furthermore, different meta-heuristics have been developed that modify or improve the heuristics by means of combinations or inclusion of processes in the original algorithms which allow an improvement of the results at the time of getting the solution, a greater number of solutions, or directing the solutions to a space of the pre-established or desirable universe. In the field of WDM (Wavelength Division Multiplexing) optical networks, the predominant technology is the switching that allows commuting different channels between wavelengths, wavelength bands, fibers, time channels, etc. [4-6].

Heuristic algorithms are the ones that best respond to the characteristics of these networks and their traffic in transit. The modeling of this traffic is of the Poisson type, and the load is measured in erlangs: it can be static or dynamic, and that difference is given by the relation that there is between the average time between the arrivals of service requests and the average time of connection requests. The present research shows various heuristics and develops the comparison of simulations up to 180 erlangs in the National Science Foundation Network (NSFNET), showing a new family of heuristics based on the Snake One and Snake Two algorithms, and shows an advance of the Snake Three metaheuristic.

II. Routing In Optical Networks

The RWA (Routing and Wavelength Assignment) problem appears in optical networks, and it is defined as the problem of finding a route and wavelength for every link of the route; if for the network it is necessary for the wavelength to be the same for every link, they are called networks with restriction of wavelength continuity, or wavelength continuity constraint (WCC), otherwise they can have a different wavelength and they would be said to be networks with wavelength reuse or WR. In the literature, light path is defined as the set of route and associated wavelength.

For example, in CCW it would be LP = {Route-R, Lambda-λ} and in WR it would be LP = {Link 1, λ 1); (Link 2, λ 2); ……; (Link n, λ n)}, where Route = (Link1-Link2-… -Link n). The WDM-WCC networks have the advantage of a low latency in the commutation in the OXC (Optical Cross Connect), but the disadvantage that the solutions universe is exhausted rapidly when the traffic increases, generating a high congestion in the network, but the WDM-WR networks have the advantage of an efficient use of the network's resources, but the disadvantage of a high latency in the commutation of the OXC. The strategies used to solve these problems are to divide the problem into two parts (two consecutive algorithms) and the integral solution (one algorithm), i.e., in the first case the route and wavelength are found in two algorithmic processes, and in the second case they are found in a single algorithmic process. On the other hand, use is made of the maximization or minimization criteria of the aptitude function (fitness). For example, the delay in the link, the ASE noise of the amplifiers in the route, the number of channels used in the link, congestion in the link, congestion in the most widely used link, etc., and the metrics which in the case of the optical networks is what is called cost, which is characterized by a whole number that represents the criteria mentioned above, in other networks, mainly electronic. Other metrics are the delay in the link, the number of jumps, etc.

III. Heuristics and Metaheuristics

To explain the heuristics a topology of N optical commuters (OXC) labelled from 0 to N-1 is defined, with NW wavelengths for each optical fiber link, where the request for incoming service reaching the nodes is:

$$\mathrm{R}_{\mathrm{i}}^{\mathrm{j}} = (D, \mathrm{n_C}, \mathrm{t_C}) \tag{1}$$

where:
$\mathrm{R}_{\mathrm{i}}^{\mathrm{j}}$: i-th request from the j-th node
D : Destiny of the request from the j-th node
$\mathrm{n_C}$: Number of connections requested $\mathrm{p_{ijk}}/\mathrm{p_{ijk}}$
$\mathrm{t_C}$: Connection time requested

Figure 1 shows an optical network with 8 nodes and 10 links where node 5 is requested with R through a route to node 3 with 2 channels for a time of 500 ms. The network uses 8 wavelengths, each labelled with numbers from 0 to 7. Therefore, a route and a wavelength must be sought that allow satisfying the demand under wavelength continuity conditions, i.e, the same wavelength in each link of the route.

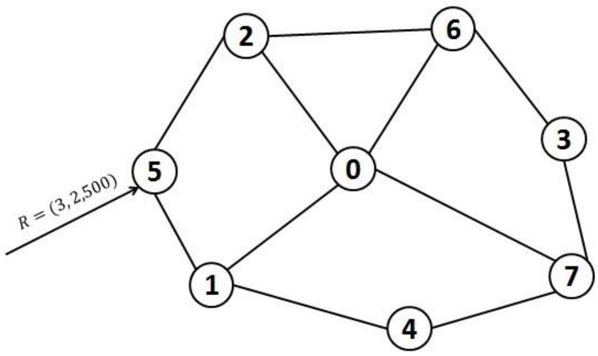

Figure 1: Optical network showing the incoming request

The heuristics find the solutions from a random group of elements obtained randomly and then procedures belonging to each methodology are followed. This group is called population, and it is defined by:

$$\mathrm{P_0} = \left\{ \begin{array}{c} \mathrm{p_{ijk}}/\mathrm{p_{ijk}} = \mathrm{Random}(0, \mathrm{N}-1)\ \forall\ \mathrm{i} \in [0, \mathrm{N}-1] \\ \mathrm{j} \in [0, \mathrm{N}-1]\ \cap\ \mathrm{k} \in [0, \mathrm{n_W}-1] \end{array} \right\} \tag{2}$$

where:
$\mathrm{P_0}$: Initial population
$\mathrm{p_{ijk}}$: j-th population element of the i-th solution for the k-th wavelength

Example:
In the population matrix the elements $\mathrm{p_{i02}}$ and $\mathrm{p_{i(N-1)2}}$ correspond to the origin node and destination node of the RWA problem where $\mathrm{i} \in [0,7]$. Equation 3 shows the initial population for wavelength 2, where a node 5 receives the service request toward node 3.

$$\mathrm{P_0} = \left\{ \begin{matrix} 5 & 0 & 1 & 4 & 2 & 5 & 7 & 3 \\ 5 & 3 & 1 & 7 & 4 & 2 & 1 & 3 \\ 5 & 6 & 0 & 2 & 7 & 5 & 6 & 3 \\ 5 & 2 & 1 & 0 & 4 & 3 & 2 & 3 \\ 5 & 7 & 3 & 1 & 5 & 4 & 3 & 3 \\ 5 & 6 & 4 & 0 & 3 & 1 & 6 & 3 \\ 5 & 2 & 7 & 2 & 3 & 0 & 5 & 3 \\ 5 & 5 & 3 & 6 & 4 & 1 & 7 & 3 \end{matrix} \right\} \tag{3}$$

For the case of the population of Equation 3 it is expected to find a route from node 5 to node 3 through wavelength 2. Looking at the routes (rows), initially they do not have a finite cost, but rather infinite as there are no links like 5-0 for the case of row 1 or 5-6 for the case of row 3. However, following a predetermined process (heuristic) these rows can improve in aptitude converting them into finite, thereby determining their possible application as solution.

A. *Genetics Algorithms (GA)*

The GA are one of the most extensively studied processes in the literature, and their application dates back to 1970 through John Henry Holland. They have been applied in different areas in which optimization is required, however since the insertion of optical fibers in the transport networks their use has increased at the research level. The algorithm is based on genetic combination to obtain populations with better characteristics at each evolution, in this way getting genotypes with increased aptitude for the solution, which generally takes place through a function called aptitude or fitness function. It also incorporates reproduction, mortality and mutation processes similar to those that occur in genetics. From Equation 4, we can extract the consecutive chromosomes (rows) with all their elements (genes), such as:

$$\left\{ \begin{matrix} 5 & 0 & 1 & 4 & 2 & 5 & 7 & 3 \\ 5 & 3 & 1 & 7 & 4 & 2 & 1 & 3 \end{matrix} \right\} \tag{4}$$

Both chromosomes will reproduce when combined, as shown in Equation 5.

$$\left\{ \begin{matrix} 5 & 0 & 1 & 4 & 4 & 2 & 1 & 3 \\ 5 & 3 & 1 & 7 & 2 & 5 & 7 & 3 \end{matrix} \right\} \tag{5}$$

These new rows have new aptitudes, i.e., an aptitude function different from the rows that generated them, producing a new population. Then we proceed to make a descending arrangement on the FF (Fitness Function), and the rows with a greater value will remain under the population, and they can be eliminated (mortality) and replaced by new random rows (natality), and an element in particular can be replaced randomly (mutation), to be able to minimize the FF until the proper aptitude is reached. Work like that shown in [4, 10-12] gives results that guarantee excellent operation under dynamic traffic and with high stress load.

B. *Simulated Annealing (SA)*

The SA algorithm is a heuristic that performs a process similar to the slow heating and cooling technique with the purpose of changing the physical characteristics of materials. Who introduced the method is still being discussed. In 1983 Scott Kirkpatrick, C.

Daniel Gelatt, and Mario P. Vecchi mention that there is a deep connection between statistical mechanics and multivariable combinatorial optimization, and in 1985 Vlado Černý mentioned that by analogy with statistical thermodynamics, and using the probability given by the Boltzmann-Gibbs distribution, we can reach or get very close to the optimal solution of a problem, and even the true optimum can be obtained [13-14].

For this case the population is treated as a set of molecules that turn like the gases in a control volume, where the walls are elements p_{i02} and $p_{i(N-1)2}$, in such a way that the rest of the elements rotate in layers at different speeds, with the slowest outside and the fastest inside, and that speed decreases according to a cooling function as seen in Equation 6.

$$v_i = v_{i0}(1 - \propto T) \tag{6}$$

where:

v_i : Speed of the i-th layer at temperature T
v_{i0} : Initial speed of the i-th layer
$\propto$: Cooling coefficient
T : Temperature

There are plenty of papers on the application of the algorithm in different fields, but in telecommunications, and in particular in optical networks, the work has not improved the results significantly [4, 6].

Rotating Layers

5	0	1	4	2	5	7	3
5	3	1	7	4	2	1	3
5	6	0	2	7	5	6	3
5	2	1	0	4	3	2	3
5	7	3	1	5	4	3	3
5	6	4	0	3	1	6	3
5	2	7	2	3	0	5	3
5	5	3	6	4	1	7	3

Direction of rotation

Static walls

Figure 2: Population matrix in Simulated Annealing

C. Tabu Search (TS)

This algorithm was presented for the first time by Dr. Fnetwork Glover in 1989, but it was in 1990 that it was recognized as a powerful heuristic [15-16]. It is based on the exchange of actions that improve the FF or make worse, in such a way that by means of a register (memory) of those actions, those that improve it are rewarded and those that make it worse are punished, and in this way it is aimed to converge rapidly toward the local optimum. The main advantage of the algorithm is that when a possible solution is localized, it converges rapidly. It is based on the idea that more information is provided by the errors than by the the correct answers, and TS has set records in the search for better solutions to the problems of planning production and programming, resource assignment, network design, routing, financial analysis, telecommunications, portfolio planning, supply chain management, modeling based on agents, business process design, forecasting, automatic learning, data mining, etc. Work like that shown in [6] make visible the robustness of the method when dealing with high traffic loads in dynamic scenarios for WDM optical networks.

D. Snake One

The Snake One algorithm is based on the use of the cost matrix of the network, concentrating its process on the horizontal and vertical displacement within that matrix, and to avoid the closed cycles it makes random jumps in each process where the maximum number of jumps is less than the maximum number in the nodes. This algorithm computes the route and wavelength assignment more efficiently than the rest of the heuristics, but the routes that it finds tend to use the edge of the network, oversaturating those resources rapidly, so their performance under stress is not as good as that of the other heuristics. For calculating the route it uses a matrix called Snake (S) that is constructed based on the cost matrix.

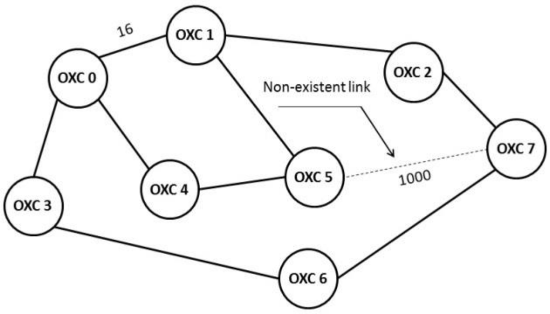

Figure 3: Network example for using Snake

Figure 3 shows an example of an optical network with 8 nodes that indicates that the non-existing links have a very high associated cost (1000), so the routes used by those links will have a large associated total cost and they will therefore appear at the end of the matrix when they are arranged in descending order (see Figure 4). In this way a low cost route is guaranteed, but not necessarily the optimum one.

N	0	1	2	3	4	5	6	7	C
1	16	1000	16	~~1000~~	~~1000~~	16	1000	1000	5048
0	1000	16	1000	16	16	1000	1000	1000	5048
4	16	1000	1000	1000	1000	16	1000	1000	6032
5	1000	16	1000	1000	16	1000	1000	1000	6032
2	1000	16	1000	1000	1000	1000	1000	16	6032
3	16	1000	1000	1000	1000	1000	16	1000	6032
7	~~1000~~	~~1000~~	16	1000	1000	1000	16	1000	6032
6	1000	1000	1000	16	1000	1000	1000	16	6032

Figure 4: Snake One algorithm

The results presented in [6] show that it improves the blocking probability indicators, but at the expense of a high use of the network. Its performance under stress (high traffic) is quite better than that of the other heuristics.

E. Snake Two

The Snake Two algorithm is a composition of the Snake One algorithm and a strategy to achieve the use of the links at their saturation point, and in this way direct traffic toward some zones and leaving others with some freedom to take care of the incoming demand. This is achieved by applying the Snake One algorithm from the origin of the request to the link chosen as the most congested that can take care of the request, and from the link to the request's destination. The most congested link is chosen from an ordered matrix that monitors the load of the links and is periodically arranged in ascending order with the aim of having the most congested links always available (see Figure 5).

The results shown in [x] and [y] indicate that the average indicators of the blocking probability and use of the network are improved.

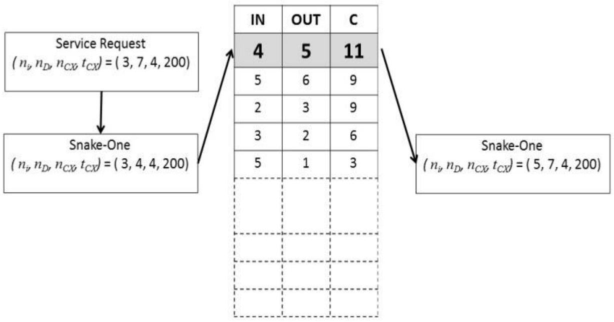

Figure 5: Matrix of congested links (MEC)

Figure 6 shows the arrival of the request R to OXC 3, and with MEC and the link 4-5 available, the origin route 3 to node 4 of the congested link is calculated with Snake One, getting 3-0-4; then the route from node 5 of the congested link to the destination 7 is calculated, getting 5-1-2-7; in this way the route 3-0-4-5-1-2-7 is composed.

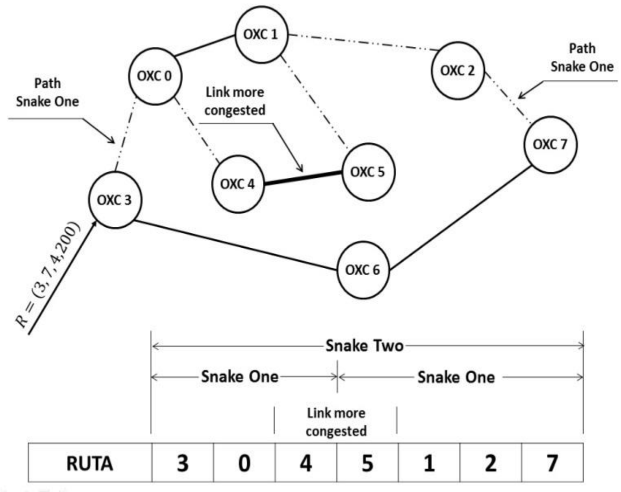

Figure 6: Snake Two algorithm

F. Other Heuristics

The PSO (Particle Swarm Optimization) algorithm was introduced in 1995 by Reynolds and Heppner to model social behavior such as, for example, the movement of fish and birds, making observations in which the movment of one or several predominates over the rest [18]. The algorithm looks for the solution position through the variation of the positions of the particles whose direction is corrected by their velocity, and in each iteration the particles correct their position with their best position found and their best position evaluated in the swarm, so the total movement is the composition of all the particles, but with the direction of the leader in a manner similar to how the swarms behave. Multiple applications of this algorithm have been made in fields like antennas, biomedicine, communication networks, clustering and classification, combinatorial optimization, control, design, faults, finances, robotics, and many more. The results reported in these papers show the great use of this algorithm, whose advantage is the rapid convergence of the solution, but at very high costs in terms of the use of the resources, and their disadvantage is the improbability of finding always the optimum solution [19-20].

The Firefly algorithm was introduced by Xin-She Yang in 2008, based on the attraction processes of fireflies due to their brightness. The conditions of the algorithm are three: the first one indicates that all the fireflies are attracted, i.e., there is no gender difference, the intensity of the attraction is proportional to the brightness, so if there is no brightness the fireflies move randomly. This algorithm is relatively young, and under some conditions it is similar to the PSO algorithm. Its application is multiple and in the same settings as those mentioned for PSO [21].

The Bat algorithm was introduced by Xin-She Yang (2010) and is based on the echolocation of microbats, and their performance has variable emission and amplitude pulse indices. In practical terms, the bats search for their prey, and depending on their echolocation they emit waves with different frequencies and amplitudes which get more intense as they approach their objective, and in this way the group communicates, succeeding in getting one or more preys (solutions) in a single process. Even though it is very young, different implementations have been made with the purpose of improving its performance [22-25].

IV. Simulation Scenario And Traffic Demand

The optical network used most widely to develop WDM network simulations is the NSFNET, or National Science Foundation Network, which was simulated with 14 nodes and 21 links in a manner similar to that of the work shown in [4-6] with the purpose of comparing the heuristic algorithms through two indicators, blocking probability and use of the network, varying the traffic load from 0 to 180 erlangs.

Equation 7 defines the request with three parameters that determine the traffic, where S_m^i is the m-th service request of the i-th node, n_d is the requested destination node, n_c is the number of requested connections, and t_c is the requested connection time.

$$S_m^i = (\mathrm{n}_d, \mathrm{n}_c, t_c) \tag{7}$$

If N is the number of nodes of the network (N=14), the total number of simulation requests is 10^8. The parameters are chosen randomly according to a uniform distribution. The time between the request arrivals t_s has an exponential distribution, so the number of requests that arrive at the node has a Poisson distribution.

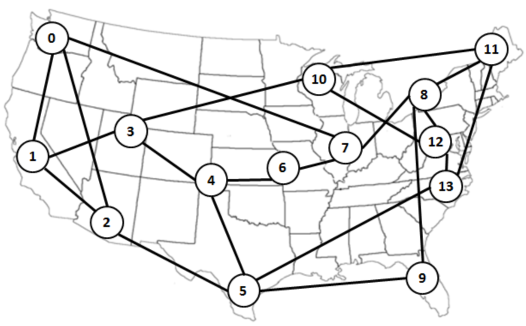

Figure 7: NSFNET network with 14 nodes and 21 links

The mean values of these random variables are similar to the values used in [4-6]. For comparison purposes, such as the number of wavelengths $\mathrm{n_w} = 8$, the traffic is uniformly distributed, and the mean connection request time is 100 ms. If the ${}^{\mathrm{t_s}}/_{\mathrm{t_c}}$ ratio is less than 1 the demand is static, otherwise it is dynamic. The present work develops the simulation for the dynamic scenario and develops a demand that stresses the network from 0 to 180 erlangs.

The blocking probability indicator (P_b) measures the probability of an incoming request not being attended. If it is impossible to wait for a request, it is blocked because it is not attended. On the other hand, the network utilization indicator (U_r) measures the used capacity of the network as the load in Erlangs is increased. This load is calculated multiplying the average rate of incoming requests by the requested mean connection time rate (100 ms).

V. SNAKE THREE ALGORITHM

The Snake Three algorithm is a new composition of the Snake One algorithm with a change in the strategy. In this case the most congested node that is available to attend the request is used, with the aim of directing traffic similarly to Snake Two, but this time through the nodal congestion indicator (NCI) (see Equation 8).

$$\mathrm{ICN}_i = \sum_{j=0}^{n} e_{ij} \quad \forall \; e_{ij} \; \exists \qquad (8)$$

Equation 6 shows the nodal congestion indicator of node i, which is the sum of the costs of the existing adjacent links and at least one of the links satisfies the request.

The most congested node is chosen from an ordered matrix that monitors the load of the nodes and is arranged periodically in ascending order with the purpose of always having available the nodes with the largest load (see Figure 8).

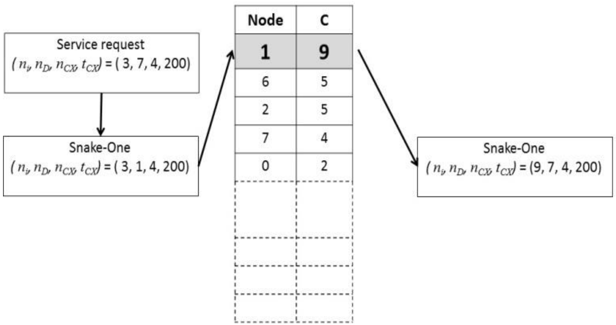

Figure 8: Congested Nodes Matrix (CNM)

Figure 8 shows the use of the Congested Nodes Matrix (CNM) for λ_0 of a total of n_W lambdas, for the purpose of the simulation $n_W = 8$, so therefore ($\lambda_0 = 0$, $\lambda_1 = 1$, $\lambda_2 = 2$, $\lambda_3 = 3$, $\lambda_4 = 4$, $\lambda_5 = 5$, $\lambda_6 = 6$, $\lambda_7 = 7$) is the set of available lambdas per link represented by whole numbers and the increasing arrangement of the costs, where the cost of one node is the sum of the costs associated with their adjacent links.

Figure 9 shows the route obtained using the Snake One algorithm to calculate the 3-0-1 route to the chosen congested node, and from the congested node to the destination of the request calculating the route 1-2-7, in this way composing the route 3-0-1-2-7. In contrast with Snake Two, this algorithm develops shorter routes but with the same effect as Snake Two, implying a smaller use of the network.

Figure 10 shows the flow sheet of the Snake Three metaheuristic, where the use of Snake One (SO) and the congested nodes matrix (CNM) for each available lambda is seen. On the other hand, matrices C, T, and λ are updated with the purpose of keeping CNM updated. Only when the route SO is obtained and it agrees with the available wavelength, the lightpath (LP) is established, otherwise the request is blocked.

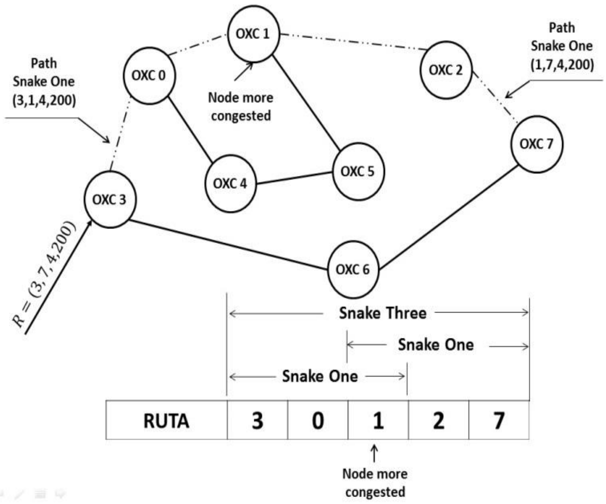

Figure 9: Snake Two algorithm

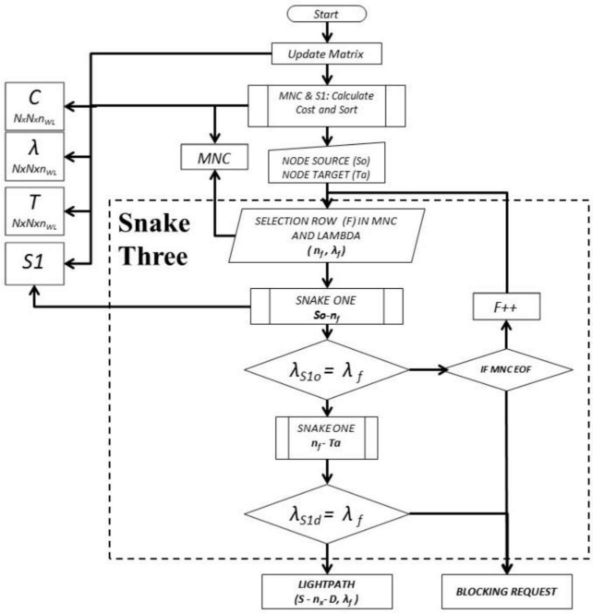

Figure 10: Flow sheet of Snake Three

VI. RESULTS

The simulation scenario used was applied to compare the heuristics Simulated Annealing (SA), Genetic Algorithm (GA), Tabu Search (TS), Snake One (SO), Snake Two (STW), and the algorithm shown as reference (RF) [26]. The previously defined indicators Blocking Probability (P_b) and Use of the Network (U_r) used as comparison elements of the algorithmic performance were applied.

Figure 11 shows the blocking probability and the performance of the algorithms for the different simulation loads varying from 0 to 180 erlangs. Three load intervals can be distinguished (Load Range of Blocking Probability - LRBP) that allow explaining better the performance: the first interval (LRBP1: 0-80 erlangs), the second interval (LRBP2: 80-120 erlangs), and the third interval (LRBP3: 120-180 erlangs). LRBP1 represents a low traffic load, LRBP2 represents a transitory traffic, and LRBP3 represents a high traffic or stress. It is seen that LRBP1, SNK2, and SNK3 do not offer great differences, performing similarly and surpassing the rest of the heuristics, including the reference algorithm, while in LRBP2 the performance starts balancing, with TS, SNK1, SNK2, and SNK3 very similar. However, when the traffic reaches LRBP3 ranges, the heuristics do not respond better than the reference algorithm, which is the one that has the best performance. It should be highlighted that the performance of AS for high traffic levels is very similar to that of the reference algorithm RF.

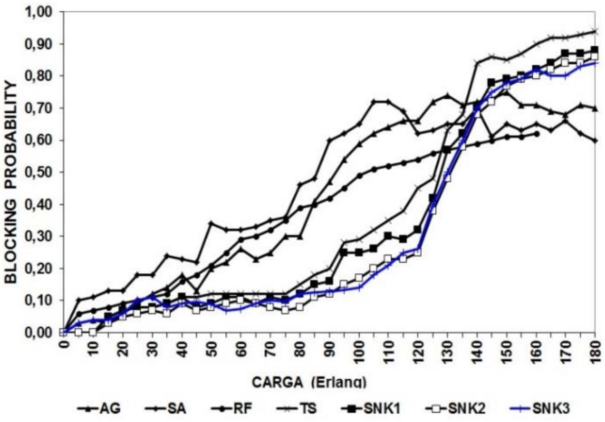

Figure 11: Comparison of the Blocking Probability in the NSFNET

Figure 12 shows the percentage use of the network, and two Load Range of Network Utilization (LRNU) intervals are seen that allow to explain the performance better: the first interval (LRNU1: 0-40 erlangs) and the second interval (LRNU2: 40-180 erlangs). It is seen that in LRNU1 the studied heuristic algorithms use up more resources than the reference algorithm and in LRNU2 the use of the resources is increasing and constant. As expected, SNK3 finds shorter routes and uses up fewer network resources than its SNK1 and SNK2 homologs. However, it does not get to be better than GA and SA, and it is the reference algorithm the one that uses the fewest resources. It should be pointed out that the reference algorithm is not heuristic and it is used only for comparative purposes.

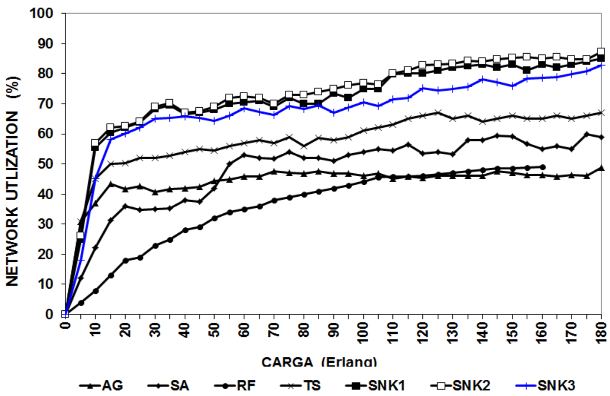

Figure 12: Comparison of network utilization in the NSFNET

VII. CONCLUSIONS

The reference algorithm found in [26] searches optimum solutions according to given indications and criteria, while the heuristics only find functional solutions, i.e., they achieve the objective of transporting without being necessarily the optimum. From this perspective, the work shown and the SNK3 metaheuristic have competitive advantages when we look at the blocking probability, turning as the best with respect to the reference algorithm. However, when we look at the use of the network's resources this is inverted, and it is the reference algorithm that is more efficient in terms of the use of resources.

It can therefore be concluded that for low ranges of traffic (LRBP1 and LRNU1) the heuristics respond with a low blocking probability, but with a high consumption of the network's resources, and this performance gets worse as the traffic increases, getting to use up more resources than the reference algorithm. New simulations must be established in a wavelength conversion scenario that allows decreasing the blocked requests without a large increase of the network's resources.

ACKNOWLEDGMENTS

The important support of the DYCYT Program of the Universidad de Santiago de Chile – USACH (DICYT Project 061572RG), Proyecto Basal USA1555 and Research Group in New Technologies (GINT-DTI-USACH) is acknowledged.

REFERENCES

[1] A. Rodriguez, A. Gutierrez, L. Rivera and L. Ramirez, "RWA: comparison of genetic algorithms and simulated annealing in dynamic traffic". LNEE Springer, Vol. 315, 2014, pp. 3-14.
[2] A. Rodriguez, L. Ramirez, L. Rivera and A. Gutierrez, "Routing wavelength assignment: a solution with tabu search in dynamic traffic", Ingeniare. Rev. chil. Ing, Vol. 22, Number: 4, 2014, pp. 495-503.
[3] A. Rodriguez, L. Ramirez and J. Travieso, "New heuristic algorithm for dynamic traffic in WDM optical networks", Revista Ingeniería e Investigación, vol. 35, Number: 3, 2015, pp. 100-106
[4] A. Rodriguez, A. Gutierrez, L. Rivera and L. Ramirez, "RWA: Comparison of Genetic Algorithms and Simulated Annealing in Dynamic Traffic". LNEE Springer, Vol. 315, 2014, pp. 3-14.
[5] Z. Zheng, F. Zhang, D. Wang, E. Sun, Y. Zhu, K. Zou and Z. Chen, "Fiber nonlinearity mitigation in 32-gbaud 16QAM nyquist-WDM systems", Journal of Lightwave Technology, Vol. 34, Number: 9 2016, pp. 2182-2187.
[6] M. Sadiq, H. Zhang, J. O'Callaghan, B. Roycroft, N. Kavanagh, K. Thomas, A. Gocalinska, Y. Chen, T. Bradley and J. Hayes, "40 Gb/s WDM transmission over 1.15-km HC-PBGF using an InP-Based mach-zehnder modulator", Journal of Lightwave Technology, Vol. 34 Number: 8, pp. 1706-1711, DOI: 10.1109/JLT.2015.2508941, Fecha de publicación: APR 15 2016.
[7] F. Gutierrez, E. Martin, P. Perry, A. Ellis, P. Anandarajah and L. Barry, "WDM orthogonal subcarrier multiplexing", Journal of Lightwave Technology, Vol. 34, Number: 8, 2016, pp. 1815-1823.
[8] F. Ferrucci and S. Bock, "Pro-active real-time routing in applications with multiple request patterns", Expert Systems with Applications, Vol. 56, 2016, pp. 320-334.
[9] F. Lima, D. Pereira, S. Conceicao and N. Nunes, "A mixed load capacitated rural school bus routing problem with heterogeneous fleet: algorithms for the Brazilian context", European Journal of Operational Research, Vol. 253, Number: 2, 2016, pp. 356-371.
[10] Y. Chen, J. Hao and F. Glover, "A hybrid metaheuristic approach for the capacitated arc routing problem", European Journal of Operational Research, Vol. 253, Number: 1, 2016, pp. 25-39.
[11] Z. Zahedi, R. Akbari, M. Shokouhifar, F. Safaei and A. Jalali, "Swarm intelligence based fuzzy routing protocol for clustered wireless sensor networks", Expert Systems with Applications, Vol. 55, 2016, pp. 313-328.
[12] W. Lefever, E. Aghezzaf and K. Hadj-Hamou, "A convex optimization approach for solving the single-vehicle cyclic inventory routing problem", Computers & Operations Research, Vol. 72, 2016, pp. 97-106.
[13] N. Siddique and H. Adeli, "simulated annealing, its variants and engineering applications". BMC Genomics, Vol. 25, Number: 6, 2016, Article. 1630001. DOI: 10.1186/s12864-016-3210-4
[14] Y. Susuki and J. Cortés, "A tabu search with gradual evolution process". Computers & Industrial Engineering, Vol. 100, 2016, pp. 52-57.

[15] H. Li and B. Alidaee, "Tabu search for solving the black-and-white travelling salesman problem", Journal of the Operational Research Society, Vol. 67, Number: 8, 2016, pp. 1061-1079.
[16] Y. Deng, J. Wu and Y. Tan, "Optimal attack strategy of complex networks based on tabu search ", Physica A-Statistical Mechanics and Its Applications, Vol. 442, 2016, pp. 74-81.
[17] J. Galan-Jimenez and A. Gazo-Cervero, “Using bio-inspired algorithms for energy levels assessment in energy efficient wired communication networks”, Journal of Network And Computer Applications, Vol. 37, 2014, pp. 171-185.
[18] J. Kennedy and R. Eberhart, “Particle swarm optimization”. Proceedings of IEEE International Conference on Neural Networks, Vol. IV, 1995, pp. 1942–1948.
[19] R. Poli, “An analysis of publications on particle swarm optimisation applications”, Department of Computer Science, University of Essex, UK. Technical Report CSM-469, 2007.
[20] R. Poli, “Analysis of the publications on the applications of particle swarm optimization”, Journal of Artificial Evolution and Applications, 2008, pp. 1-10.
[21] X. Yang, “Nature-inspired meta-heuristic algorithms”, Luniver Press, Beckington, 2010, 2nd ed.
[22] S. Vanegas, I. Amaya and R. Correa, “Algoritmo del murciélago virtual en el desarrollo de la Integral de Duhamel para sistemas estructurales con un grado de libertad”. Revista ingeniería de construcción, Vol. 28, Number: 3, 2013, pp. 278-289.
[23] X. Yang, “A new metaheuristic bat-inspired algorithm”, International Workshop on Nature Inspired cooperative Strategies for Optimization, Granada, Springer, 2010, pp. 65-74.
[24] X. Yang, “Bat algorithm for multiobjective optimization”, International Journal of Bio-Inspired Computation, Vol. 3, Number: 5, 2011, pp. 267-274.
[25] X. Yang and A. Gandomi, “Bat algorithm: A novel approach for global engineering optimization”, Engineering Computations, Vol. 29, Number: 5, 2012, pp.464-483.
[26] H. Zang, and J. Jue, “Dynamic lightpath establishment in wavelength-routed WDM networks”, IEEE Communications Magazine, Vol. 39, Number: 1, 2001, pp. 100-108.