A Hybrid Genetic Algorithm / Variable Neighborhood Search Approach to Maximizing Residual Bandwidth of Links for Route Planning

Gajaruban Kandavanam[1], Dmitri Botvich[1], Sasitharan Balasubramaniam[1],
Brendan Jennings[1]

TSSG, Waterford Institute of Technology, Ireland

Abstract. This paper proposes a novel approach to performing residual bandwidth optimization with QoS guarantees in multi-class networks. The approach combines the use of a new highly scalable hybrid GA-VNS algorithm (Genetic Algorithm with Variable Neighborhood Search) with the efficient and accurate estimation of QoS requirements using empirical effective bandwidth estimations. Given a QoS-aware demand matrix, experimental results indicate that the GA-VNS algorithm shows significantly higher success rate in terms of converging to optimum/near optimum solution in comparison to pure GA and another combination of GA and local search heuristic, and also exhibits better scalability and performance. Additional results also show that the proposed solution performs significantly better than OSPF in optimizing residual bandwidth in a medium to large sized network.

1 Introduction

QoS-aware network planning remains critically important for Internet Service Providers (ISP) given the challenges they are faced with due to the heterogeneity in network infrastructure and dynamism in network traffic. There have been many research efforts to address the route planning problem. Kodialam and Lakshman [1] proposed an integer programming formulation to find two disjoint paths for each demand pair during route planning. One path is used as the primary and the other as secondary. The secondary paths are selected in such a way they share the link capacity when their corresponding primaries do not have any links in common. They showed through experiments that the complete information regarding the allocated paths is not necessary to find a near optimal bandwidth allocation. Riedl and Schupke [2] proposed a routing algorithm that takes into account a concave link metric in addition to an additive one, showing that better utilization can be achieved when both metrics are used to perform the routing. They presented a mixed integer programming model to work on the metrics for small networks and a GA based technique for large networks. Applegate and Cohen [3] proposed a routing model based on linear programming. The maximum link utilization was taken as the metric to optimize routing. They further showed that a perfect knowledge of the traffic matrix was not required to perform robust routing under dynamic traffic conditions.

Kodialam et al [4] proposed an on-line multi-cast routing algorithm which minimizes a cost function. Their heuristic algorithm uses the knowledge of the ingress and egress points which are potential future demands to avoid the use of the loaded links that may be required for future demands. They also presented results that showed reduction in call rejection rate. The solution proposed by Yaiche et al [5] for optimal bandwidth allocation of elastic demands is based on a game theoretic framework. The Nash equilibrium is used as the bargaining point of the bandwidth requirement. The solution was presented in both centralized and distributed manners. In previous work [9] we proposed a heuristic solution based on a GA to support bandwidth guaranteed routing for clustered topologies. We showed through experiments that the proposed algorithm, which is distributed in the sense that individual clusters are treated in parallel outperformed similar algorithms that do not take cognizance of the clustered nature of the topology.

However, the above problem formulations of the well known routing problem do not produce optimum desirable route plans always. For example, choosing to minimize maximum link utilization does not guarantee minimal load on each link in the network. We formulate the said routing problem as maximizing the residual bandwidth of all links in a network. This is a significant multi-modal constrained optimization problem that deterministic heuristic techniques are incapable of addressing. We revisit the use of GA [13] as a means of addressing the above bandwidth optimization problem in network planning. GA have been proposed for various non-linear optimization problems in communication networks [2, 14], where deterministic optimization techniques such as linear programming are not applicable. However, it is well known that standard GA approaches often do not guarantee converging to feasible solutions in the context of constrained optimization problems [12]. Failure to guarantee convergence is an important limitation, especially in the context of route planning for large networks, where application of these techniques can be computationally expensive.

In this paper, we present a novel algorithm (denoted GA-VNS) to maximize residual bandwidth of links for route planning that is based on augmenting a standard GA approach with the VNS approach proposed by Tasgetiren et al [11]. We assume that the input to the algorithm is a demand matrix for the network that is QoS-aware in the sense that the estimated demand for a given source-destination pair takes cognizance of the effective bandwidth [10] of the traffic expected to flow from that source to that destination. In previous work [8], we outlined a cost-effective approach for QoS-aware demand matrix preparation based on a measurement-based approach to effective bandwidth estimation. Using traffic matrices prepared following this approach, we investigate the performance of GA-VNS in comparison to a standard GA algorithm and a GA algorithm augmented with another local search heuristic called Fixed Neighborhood Search (denoted GA-FNS), for a number of randomized and real network topologies. Our results show GA-VNS always finds an optimal or near-optimal solution, whereas the other two algorithms fail to find a feasible solution for a significant proportion of the attempts. Furthermore, the algorithm is shown to scale well to large networks and to out-perform the commonly deployed OSPF intra-AS routing protocol in terms of maximizing link residual bandwidth.

2 Residual Bandwidth Optimization Problem

Our residual bandwidth optimization problem can be stated informally as: identify a routing plan for a network that maximizes the residual bandwidth of all links in the network given that all links have defined maximum capacities and that the per traffic class effective bandwidth required between source-destination pairs for expected traffic flows is that defined in the provided traffic matrix. Provision of a traffic matrix that is QoS-aware in the sense that the demands for given source/destination pairs reflect effective bandwidth estimations means that the routing plan identified as a solution to the optimization problem is one that minimizes the risk of QoS violations if actual traffic carried on the network is equal to the estimated values. In this context choosing to maximize residual bandwidth on all links appears a natural choice, since the larger the residual bandwidth is on a given link the less the probability of traffic carried over that link incurring a QoS violation.

Our objective is to maximize the residual bandwidth on all links in the multi-class network. It can be represented as maximizing $\frac{X_l}{C_l}$, over all $l \in L$ satisfying the constraint $X_l \geq \lambda_l$, $\forall\ l \in L$, where X_l is the residual bandwidth on the link l, C_l is the capacity of the link l and L is the set of links. Route allocations are calculated to satisfy the QoS requirements of the estimated traffic demand, which is represented by the QoS-aware demand matrix. The metric used to compare the quality of different paths in forming routes is an asymptotic convex function, defined as:

$$F(X_l) = \frac{C_l}{(X_l - \lambda_l)^{\alpha}} - \frac{C_l}{(C_l - \lambda_l)^{\alpha}} \tag{1}$$

where α is a constant. The cost function used in our hybrid GA algorithm to guide the search, is based on $F(X_l)$. The function is motivated by the condition that if $X_l = C_l$ then $F(X_l) = 0$ and should go to infinity when the available bandwidth on link l approaches the reserved bandwidth λ_l. Note that Riedl and Schupke [2] found that the use of a convex metric significantly improves routing optimization.

3 Approach Outline

The first step in QoS-aware route planning using our approach is the preparation of a QoS-aware demand matrix. One possible process for doing this is outlined by Davy et al [8]. We then proceed to outline three separate GA-based algorithms to solve the residual bandwidth optimization problem. The first is a “pure” GA algorithm and forms the basis for the other two algorithms. These are the hybrid GA with Fixed Neighborhood Search (GA-FNS) and our proposed solution: hybrid GA with Variable Neighborhood Search (VNS).

3.1 Genetic Algorithm

We now summarize the GA technique and outline the "pure" GA we apply to the residual bandwidth optimization problem. For our residual bandwidth optimization GA we seek to populate a route table for our network which contains a sub-table for each source/destination pair, which in turn contains n shortest paths between that source/destination pair. A breadth first search based algorithm can be used to initially find the routes to build the route table. Given this starting point we now outline the GA in terms of the nature of Chromosomes, the Initial Population, the Selection mechanism, the Crossover mechanism, the mutation mechanism, and the Cost Function.

An array structure is used as the chromosome structure. The selected path for each source/destination pair is encoded in an array. For example $Chromosome[i][j]$ is the gene representing the path selected for the i^{th} traffic class in j^{th} source-destination pair.

The initial population is generated by randomly selecting the paths for the required source destination pairs for different traffic classes from the route table to form chromosomes. The probabilities of selecting given paths for each source-destination pair follows a uniform distribution.

A simple tournament selection mechanism is employed. Every chromosome in the odd position in the ordered population is mated with the next chromosome to produce offspring. This operation produces a population twice the size the actual population size. The best half is selected for the next generation and the rest is discarded. But, this may produce duplicates in the population as the generations go on. The duplicates are replaced with newly generated chromosomes. This process increases the diversity in the population.

A two point crossover is used to produce offspring. Crossover is applied separately for arrays representing each traffic class by selecting different crossover points for different arrays representing the paths selected for different classes of traffic. Once more, a uniform distribution is used to randomly select the crossover points. Mutation is performed by randomly changing one gene in the chromosome. Uniform random distribution is used to select the gene to be replaced.

A cost function based on the metric that is presented in the previous section for residual bandwidth calculation is used in this paper to measure the quality of the solution. This is given by: $\Phi(X_l) = \sum_{l \in L} F(X_l)$.

3.2 The GA-FNS Hybrid Algorithm

The same GA that is described in the previous section is used here. But the difference is that a greedy local search algorithm with fixed neighborhood is used along with the GA. Therefore, we use the term Fixed Neighborhood Search (FNS) to make it easier for the comparisons with the VNS algorithm that is described in the next section. The FNS is applied to all the chromosomes in the population with 10% probability except the best. The best chromosome in the population is always selected for applying local search. The FNS is applied after every generation of GA. The best chromosome in the population always undergoes FNS in order to accelerate the convergence to a global optimum/near optimum value. The search is also probabilistically applied to the other chromosomes to guide the search from different regions in the solution space. This is crucial, because a solution with high cost can be close to the global optimum solution. This process improves fairness and prevents the potential solutions being neglected.

The FNS performs the search by comparing the cost with the other solutions in the immediate neighborhood. The cost is calculated using the same cost function as the one used in GA. The neighborhood of a solution is the set of solutions which differ from the current solution by at most one bit. One gene in the chromosome is randomly selected and replaced by another random gene to find the neighbor of the solution that a chromosome represents. If the neighbor is found to be having lower cost than the current solution, the neighbor is made the current solution and the search proceeds from there. Another neighbor is examined for improvement otherwise. This process continues until no better solution is found in the neighborhood. But it is very likely that the search proceeds in the wrong direction in a multi-model solution space and leads to a very long convergence time. Therefore, we have restricted the number of hops allowed to search for a better solution, to a finite number. This process helps to improve the solution that the GA finds in a more deterministic fashion.

3.3 The GA-VNS Hybrid Algorithm

In this algorithm, VNS is applied as opposed to FNS. All other aspects of this algorithm are the same as that of the GA-FNS. The Variable Neighborhood search switches the neighborhood while searching for a better solution. Therefore, there are two neighborhoods defined in this algorithm. The first neighborhood is the set of solutions which have at most one bit difference against the current solution. This is the same as that of FNS. The second neighborhood is comparatively bigger. This is defined as the set of solutions that differ from the current solution by exactly two bits. A neighbor in the second neighborhood is defined by replacing two randomly selected genes from the chromosome that represents the current solution, by two random genes.

Initially the search is performed in the first neighborhood to find a better solution as described in the previous section. If a solution with lower cost cannot be found in the first neighborhood, the search switches to the second neighborhood. The neighbors are examined for a solution with lower cost in the new neighborhood. If the neighbor is having lower cost, the neighbor is made the current solution. The other neighbors are examined one by one for an improved solution otherwise. The search terminates when a better solution cannot be found in the new neighborhood. The number of maximum hops allowed is restricted to a fixed number for the same reasons as it is in FNS. The switching of neighborhoods prevents the search being stuck at the local minimum. When there is no better solution found in the first neighborhood, it can be a local minimum. But when the neighborhood changes, it is probable that a better solution can be found and thus the local minimum is skipped.

Table 1. The Range of Mean Rate Values

Traffic Class	Rate	γ_i	α_i
$Class_1$	$100 - 500$ Mb/s	142.640	0.6
$Class_2$	$100 - 200$ Mb/s	31.548	0.4

4 Experimental Setup

The problem scenarios and the different contexts in which the evaluations of the proposed techniques are carried out are explained in this section. Demand matrices with two degrees of load impact are generated to perform routing, where the first type has an impact of $30 - 40\%$ average link utilization and the other has $50 - 60\%$. This is to evaluate the performance of the proposed algorithm under different difficulty levels.

The experiments are carried out on different sizes of network topologies, both randomized and real, to show the applicability and scalability of the proposed solution. The random network topologies are built with 20% connectivity, ie. there exist links between 20% of node pairs in the network. The random topologies have 40% of their nodes as edge nodes. For certain experiments two real topologies were used: the Telstra Australia topology described in [6] and the $N20$ topology described in [7].

We simulate the 95^{th} percentile Effective bandwidth-Mean (EM) coefficient values $K_{95,i}$ to be as close as possible to the actual values. The EM coefficient $K_{95,i}$ is given by the following expression: $K_{95,i} = \frac{\gamma_i}{mean_i^{\alpha_i}} + 1$. The mean rate $mean_i$, γ_i and α_i values for the traffic class i are selected as shown in Table 1. The values for mean rate are generated using uniform distribution. Then the required effective bandwidth R_{eff} per source-destination $x - y$ per traffic class i is given by: $R_{eff} = K_{95,i} \times mean_{i,x-y}$.

5 Results and Analysis

This section discusses the results of performance evaluation carried out using the proposed solution. The results are presented in three sub-sections. The first set of experiments are performed on randomly generated network topologies of different sizes. The second and third sets of experiments are performed on Telstra Australia backbone topology as described in [6] and the $N20$ topology described in [7]. The GA have the population size of 60 and mutation rate of 5% as they were found to be optimum in our experiments [14].

5.1 Randomized Topologies

This section presents the results of the experiments carried out on randomly generated network topologies. The results of first set of experiments are illustrated in Fig 1. This shows the scalability of the proposed GA-VNS algorithm.

The results show that the number of fitness evaluations required to find the optimum/near optimum solution does not change significantly when the network grows in size. The generated demand matrices for this experiment had $(30-40\%)$ average link utilization. The experiment is carried out for a single class of traffic and the cost function has the α value of 1. Fig 2 illustrates the scalability of the proposed algorithm for two traffic classes based on the QoS-aware demand matrix.

Fig 3 illustrates how the load on the network makes an impact on the convergence of the proposed solution. The experiments are carried out under approximately $30-40\%$ and $50-60\%$ average link utilization scenarios. A randomly generated network topology with 50 nodes is used. The other parameters remain the same as that of previous experiment.

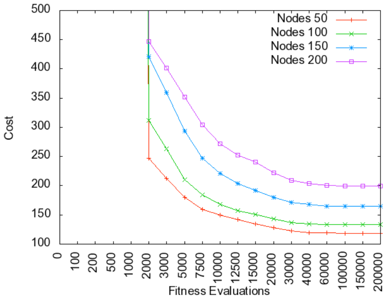

Fig. 1. The Scalability of the Algorithm for a Single Traffic Class

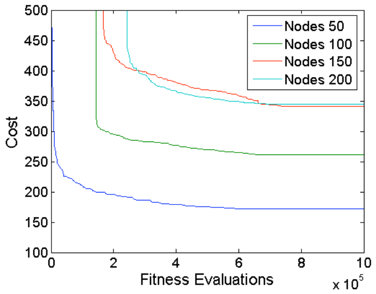

Fig. 2. The Scalability of the Algorithm for Multiple Traffic Classes

The next set of experiments are to evaluate the suitability of the other algorithms like Pure GA and GA-FNS in comparison to the proposed GA-VNS algorithm. The parameters of the experiments are the same as that of previous experiment. As it can be seen from the results shown in Table 2, the Pure GA and GA-FNS do not converge to an optimal solution always, that is, they often fail to drive the search into the feasible region. The percentage of successfully converging to an optimal solution for pure GA and GA-FNS are 17% and 60% respectively. Significantly, the GA-VNS algorithm has 100% success.

The results illustrated in Fig 4 show how GA-VNS converges with respect to different number of traffic classes. The load on the network is approximately the same for both scenarios. As it can be seen when there are multiple classes of traffic, it takes longer to converge to a feasible solution in comparison to single class traffic. This is due to the variation in effective bandwidth requirements for different classes of traffic. The average link utilization caused by the demand matrices used in this experiment are shown in Table 3.

The results illustrated in Fig 5 show the convergence of the GA-VNS algorithm for different sizes of network topologies when the cost function with $\alpha = 2$ is used. All other parameters associated with this experiment are the same as

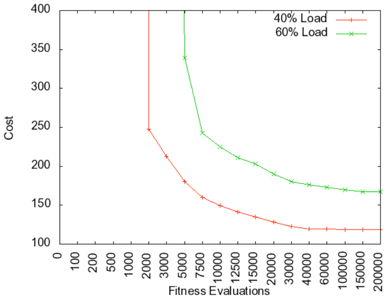

Fig. 3. The Effect of Load on Convergence

Table 2. The Success Rate of the Algorithms

The Algorithm	The Percentage of Success
GA-VNS	100%
GA-FNS	60%
Pure GA	17%

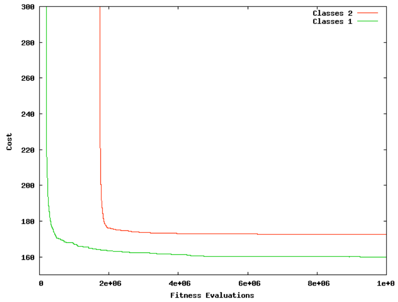

Fig. 4. Single Class of Traffic vs Two Classes of Traffic

Table 3. The Average Link Utilization

Average Link Utilization for single class	0.42
Average Link Utilization for two classes	0.33

that of the first experiment. It can be observed that the α value does not have an impact on converging to a feasible solution. However, it does have an impact on converging to an optimal solution, with the cost function with $\alpha = 1$ converges slightly faster than that with $\alpha = 2$.

Fig 6 illustrates the distribution of link utilization across the network. A topology with 50 nodes, 20 edge nodes and 20% connectivity is generated to perform the simulations. The generated demand matrix has 160 source-destination pairs. The link capacities and the bandwidth requirements are randomly generated using uniform distribution in the range of $150 - 200$ Mb/s and of $40 - 80$ Mb/s respectively. It can be seen that the maximum link utilization is kept below 0.6. The results show that there are many links that are having < 0.1 link utilization, because they are not used to route the traffic. This is due to the fact that the network topology is randomly generated and so are the demand matrix and edge routers. The maximum link utilization and the average link utilization for the above scenario are 0.59 and 0.33 respectively.

As it can be seen from all previous experiments, the cost of the best solution is infinity at the initial stages. This means the available bandwidth on at least one link l in the network is below the reserved bandwidth λ_l. Therefore, the solutions in the initial population as well as in the first few generations are infeasible, ie. the used bandwidth on the links is higher than the allowed bandwidth limit. This happened, because the initial population is randomly generated.

The size of the feasible region depends on the difficulty of the problem, as dictated by the size of the demand matrix and the bandwidth requirements. As the bandwidth requirement grows, the feasible region shrinks and becomes

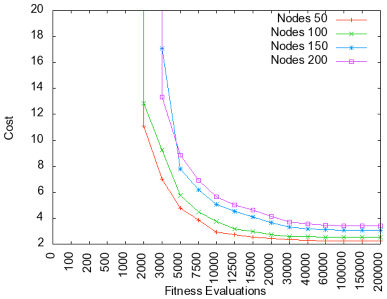

Fig. 5. The Scalability of the Algorithm for $\alpha = 2$

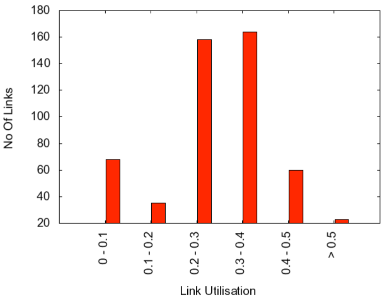

Fig. 6. The Distribution of Link Utilisation Across the Network

disconnected. At some stage, the feasible region completely vanishes. However, in our experiments, we considered the scenarios where all the demands can be satisfied. Searching for the global optimum in many disconnected feasible regions, each having a number of maxima and minima is a very challenging task. In fact, the size of the complete feasible region is much smaller than that of the infeasible region for a problem with moderate difficulty. Therefore, it is crucial that a successful algorithm must have the ability to drive the search from the infeasible region towards the feasible region. For the results reported here it is clear that GA-VNS has this ability, at least for problems of the scale and difficulty addressed here. To illustrate this point the results shown in Table 2 helps to compare the ability of different algorithms to find feasible solutions. In case of pure GA, the offspring of infeasible solutions are very likely to be infeasible. GA is a random guided search technique. Therefore, it fails to direct the search towards the feasible region. But for GA-FNS, the neighbor of the infeasible solution is also infeasible in most of the cases. Therefore GA-FNS also fails to search, in spite of having the local search along with GA.

Typically non-linear optimization problems are addressed by enforcing the feasibility restriction on initial population generation. But this cannot be done in complex problems like the one addressed in this paper due to the very small size of the feasible region in comparison to the infeasible region. When the feasible region is very small, generation of feasible results requires extensive searching, which makes it impractical to initiate the GA with a set of feasible solutions.

5.2 Telstra Australia Topology

The experiments for the results presented in this section are carried out on the Telstra Australia topology, as detailed by the Rocketfuel project [6]. Although the actual capacities of the topology that are needed for our study are not known, this project provides the derived OSPF/IS-IS link weights. In the absence of any other information on capacities, they are assumed to be inversely proportional to the link weights, as per the Cisco recommended default setting of link weights.

The first set of experiments performed to compare the maximum link utilization obtained in the route plan using the proposed solution in comparison to OSPF is as follows: Proposed solution produced a maximum link utilization of 0.87 in comparison to the maximum link utilization of 1.06 (infeasible) produced by OSPF. Fig 7 illustrates how the proposed GA-VNS algorithm performs in comparison to GA and GA-FNS algorithms. We have performed this experiment at comparatively low network load to start with a population having feasible solutions. This was necessary as GA and GA-FNS algorithms fail to converge on certain proportion of attempts. As it can be seen, GA-VNS algorithm converges faster in comparison to the rest. The next set of results illustrated in Fig 8 is to show how the α value of the cost function influences the performance of the solution. The costs of the solutions are normalized to make them comparable, since two different cost functions are used.

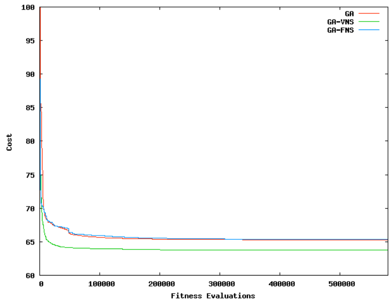

Fig. 7. The Performance of GA-VNS in Comparison to GA and GA-FNS

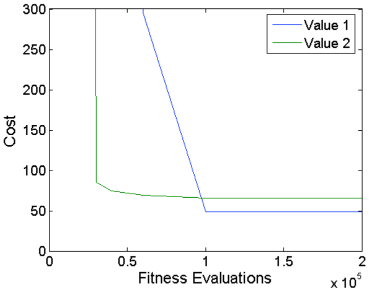

Fig. 8. The Comparison of Performance for α Values 1 and 2

5.3 Comparison with OSPF for $N20$ Topology

The results illustrated in Fig 9 are evaluated on $N20$ topology given in [7]. The demand matrix used is having 1.4 times the bandwidth requirements given in [7]. The distribution of link utilization is compared with that of OSPF. We have set the OSPF weights using two criteria. The first criterion is link capacity, where the link weights are inversely proportional to the capacity of the corresponding links. The second is the available bandwidth on links, where the weights are inversely proportional to the available bandwidth of the corresponding links. As it can be seen, OSPF weight setting using the available bandwidth as a metric shows better distribution of link utilization as expected. The proposed solution produces significantly better distribution in comparison to the two versions of OSPF. In this experiment, the average link utilization produced is approximately 0.4. The maximum link utilization produced by GA-VNS, OSPF-Capacity and OSPF-Free Bandwidth are 0.66, 1.33 and 0.81 respectively. The superiority of the performance of GA-VNS over Pure GA and GA-FNS is illustrated in Fig 10.

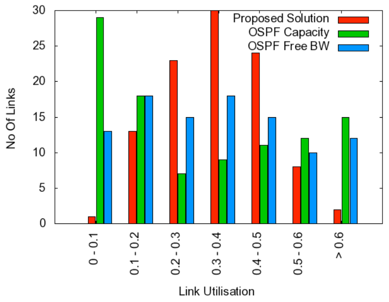

Fig. 9. The Distribution of Load in comparison to OSPF

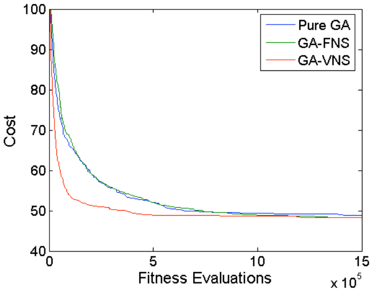

Fig. 10. The Performance of GA-VNS in comparison to GA-FNS and Pure GA

6 Summary and Future Work

In this paper we proposed a solution to support QoS-aware network planning for medium to large networks by performing residual bandwidth optimization for all links in the network, assuming the availability of a QoS-aware demand matrix. This is a constrained optimization problem that standard GA typically fails to converge for when we start from a set of infeasible solutions. To address this limitation we outlined a hybrid GA-VNS algorithm to perform residual bandwidth optimization on all links for multiple traffic classes in medium to large network topologies.

Experiments were performed on randomized networks of different sizes, real topologies and specific network topologies to evaluate the GA-VNS algorithm under different scenarios. The results demonstrated that, for the tests carried out, the algorithm always converges to an optimal or near optimal solution. Our performance evaluations further demonstrated that GA-VNS performs and scales better than standard GA and a representative combination of GA-local search algorithms. Additionally the algorithm performed significantly better in terms of optimizing residual bandwidth in comparison to OSPF.

The suitability of the proposed GA-VNS algorithm for residual bandwidth optimization is evident from the experimental results we have provided. However the scope of the proposed algorithm is not restricted to this problem alone. For future work we plan to investigate the application of GA-VNS and extensions thereof to other network optimization problems.

7 Acknowledgments

This work was funded by Science Foundation Ireland via grant number 08/SRC/I1403 ("Federated, Autonomic Management of End-to-End Communications Services").

References

1. M. Kodialam and T. V. Lakshman. Dynamic Routing of Restorable Bandwidth-Guaranteed Tunnels using Aggregated Network Resource Usage Information. *IEEE/ACM Transactions on Networking*, Vol-11, No. 3, June 2003.
2. A. Riedl and D. A. Schupke. Routing Optimization in IP Networks Utilizing Additive and Concave Link Metrics. *IEEE/ACM Transactions on Networking*, Vol-15, No. 5, October 2007.
3. D. Applegate and E. Cohen. Making Routing Robust to Changing Traffic Demands: Algorithm and Evaluation. *IEEE/ACM Transactions on Networking*, Vol-14, No-6, December 2006.
4. M. Kodialam, T. V. Lakshman and S. Sengupta. On-line Multicast Routing With Bandwidth Guarantees: A New Approach Using Multicast Network Flow. *IEEE/ACM Transactions on Networking*, Vol-11, No. 4, August 2003.
5. H. Yaiche, R. R. Mazumdar and C. Rosenberg. A Game Theoretic Framework for Bandwidth Allocation and Pricing in Broadband Networks. *IEEE/ACM Transactions on Networking*, Vol-8, No. 5, October 2000.
6. N. Spring, R. Mahajan and D. Whetherall. Measuring ISP Topologies with Rocketfuel. *Proc ACM SIGCOMM*, pp 133-145, 2002.
7. S. Kohler and A. Binzenhofer. MPLS Traffic Engineering in OSPF Networks-A Combined Approach. *Univ. Wurzburg, Germany*, Tech. Rep. 304, Feb 2003.
8. A. Davy, D. Botvich and B. Jennings. On the use of Accounting Data for QoS-aware IP Network Planning. *Proc. 20th International Teletraffic Congress - ITC-20*, eds L. Mason, T. Drweiga & J. Yan, LNCS 4516, Springer, Heidelberg, pp. 348-260, September 2007.
9. G. Kandavanam, D. Botvich, S. Balasubramaniam, P. N. Suganthan and M. F. Tasgetiren. A Dynamic Bandwidth Guaranteed Routing Using Heuristic Search for Clustered Topology. *IEEE Advanced Networks and Telecommunication Systems*, December 2008.
10. F. Kelly. Notes on Effective Bandwidth. *Stochastic Networks: Theory and Application*, Eds F. P. Kelly, S. Zachary and I. B. Ziedins. Royal Statistical Society Lecture Notes Series, vol-4, pp. 141-168, Oxford University Press, ISBN 0-19-852399-8, 1996.
11. M. F. Tasgetiren, M. Sevkli, Y. C. Liang and G. Gencyilmaz. Particle Swarm Optimization Algorithm for Permutation Flowshop Sequencing Problem. *Lecture Notes in Computer Science*, 2004.
12. C. A. Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, Vol-191, No. 11-12, pp 1245-1287, January 2002.
13. D. E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning, 1st edition. *Addison-Wesley Longman Publishing Co., Inc*, ISBN:0201157675, Boston, MA, 1989.
14. G. Kandavanam, D. Botvich, S. Balasubramaniam, P. N. Suganthan and W. Donnelly. A Multi Layered Solution for Supporting ISP Traffic Demand using Genetic Algorithm. *In the proc of IEEE Congress on Evolutionary Computation*, September 2007.