Qiao Lu PhD Thesis Defence Questions

Prof. Pon-Han Ho

1. What is the relationship between Chapters 3 and 4?

   In Chapter 3, we propose an analytical model to estimate the performance of general virtual network embedding algorithms. We propose two virtual network embedding (VNE) algorithms: SBGA and GAOne in Chapter 4. The models we developed in these two chapters can be independent existence. Our analytical model can estimate the performance and give a benchmark solution for SBGA and GAOne. The results are shown in Chapter 5.

   We add the above relationship in Chapter 4, page 58, first paragraph, "In Chapter 3, we propose an analytical model to estimate the performance of general virtual network embedding algorithms. In this chapter, we introduce our VNE approaches based on GA. The models we developed in these two chapters can be independent existence. Our analytical model can estimate the performance and give a benchmark solution for SBGA and GAOne."

2. Are the comparisons among the state-of-arts?

   We compared our VNE solutions with four related algorithms. The criteria we used to select algorithms for comparisons are best in performance and best in speed. We choose G-SP that is considered the fastest algorithm due to its simplicity. We select D-ViNE and R-ViNE that are known as benchmarks due to their mixed integer linear programming (MILP) approaches. We also compared our algorithms with another novel genetic algorithm (GA) based solution—PBGA, which operates the crossover procedure without reconstructing genes. That is to say, PBGA only considers the genes already in the original path pool.

   In Section 5.2, we described our compared algorithms and the reasons we choose the algorithms from page 102, last paragraph, "The criteria we used to select algorithms for comparisons are best in performance and best in speed. Specifically, we selected G-SP [41] for comparison because it uses the shortest path algorithm for link mapping, which is widely used by other metaheuristic algorithms as mentioned above, and also because it is considered the fastest algorithm due to its simplicity. To make it a fair comparison, our node mapping algorithm of SBGA is exactly the same as G-SP. We have demonstrated that our SBGA outperforms this shortest path based algorithm not only in performance but also in speed, thanks to the parallel processing structure enabled by our algorithm. We selected D-ViNE and R-ViNE [50] for comparisons because they are considered the best in performance due to their MILP-based approaches for both node and link mapping. We have demonstrated that the performance of our algorithm is either close or better than D-ViNE and R-ViNE while our execution speed is significantly superior to D-ViNE and R-ViNE. Another algorithm we used to compare is Path Based Genetic Algorithm (PBGA) which proposes a GA crossover and mutation based on chromosomes in VNE link mapping [84]. PBGA only operates the genes that are in the original path pool. While, our SBGA is expected to generate new differential genes which may not exist in the original path pool. We expect that SBGA can have better performance than the simpler PBGA algorithm.

Furthermore, [50] is the most popular research paper for VNE. It is not only because it produces a good embedding performance which is considered as a benchmark for many research papers in this field, but also because it publicly provides the source codes, which allows us to reproduce the evaluation results exactly. It is observed that algorithms in [50] are also used as a benchmark algorithm in many VNE papers where metaheuristics are deployed. In addition, all metaheuristic-based VNE approaches have not disclosed their source codes, which certainly prevent us from reproducing the correct simulation results exactly. In fact, due to the high complexity and difficulty of optimization problems under uncertainty and randomness, it is very difficult to reproduce the results fairly without implementation details".

3. What is the nonoverlapping path?

   If there are no overlapped links among paths, we define the paths as nonoverlapping paths. To avoid misunderstanding, we change the "nonoverlapping paths" to "link-disjoint paths" in our thesis.

4. What is the potential issue about independent arrival rates?

   We assume that the virtual network arrivals follow the Poisson distribution. At a substrate link, we assume the requests are subject to Poisson distribution too for the following reasons. To our best knowledge, in most VNE simulation setups, the virtual network (VN) requests arrive following the Poisson distribution. We consider the same scenario as other general VNE research. At the substrate link level, since we consider a random topology and all the substrate links are undifferentiated, we can treat the arrivals at a substrate link are random samples from the Poisson arrivals of the virtual network. Therefore, we assume that the arrivals at the substrate link level still obey Poisson distribution. Our simulation results justify that our analytical model can have a good estimation of the effective load of GAOne.

   We add the above arguments in Section 3.1, page 40, last paragraph, "We consider the same scenario as other general VNE research. At the substrate link level, since we consider a random topology and all the substrate links are undifferentiated, we can treat the arrivals at a substrate link are random samples from the Poisson arrivals of the virtual network. Therefore, we assume that the arrivals at the substrate link level still obey Poisson distribution with a mean rate $\lambda_{s,l}$."
   We justify our Poisson arrival assumption in Section 5.1.2, page 100, first paragraph, from "As shown in Fig. 5.1 , the results are very close at different VN arrival rates. This justifies that the assumption of (3.22) is acceptable and the (3.22) result is accurate."

5. Why don’t you consider node limitation?

   First, we consider the node mapping limitation at the virtual node level. At the virtual link level, we do not consider the intermediate substrate nodes’ capacity since the intermediate substrate nodes are only responsible for routing. There is a very little capacity requirement on the intermediate substrate nodes. Therefore, we assume the intermediate nodes with infinite capacity.

   We add the above discussion in Section 2.2, page 24, first paragraph, from “At the virtual link level, we do not consider the intermediate substrate nodes’ capacity since the intermediate substrate nodes are only responsible for routing. There is a very little capacity requirement on the intermediate substrate nodes. Therefore, we assume the intermediate nodes with infinite capacity.”

6. Why link-disjoint paths? Why are they independent?

   We only consider link-disjoint paths for the following reasons. With non-link-disjoint paths, if one link is blocked, multiple paths may be blocked at the same time. This tends to make blocking probability higher as shown in Section 5.1. Another reason is that the calculation of blocking probability is more complex due to the complex correlation structure. Therefore, we only consider selection candidate paths from link-disjoint paths.

   We assume that the blocking probability of the link-disjoint paths is independent in this thesis. This assumption is based on the assumption that the blocking probability of substrate links is independent. We first discuss the reason that we can assume the blocking probability of substrate links are independent. When we consider a substrate network with bus topology with a small number of nodes, the probability that two paths have joint links is extremely high due to limited substrate links and nodes. Specifically, the traffic on one substrate link is highly dependent on the one on another substrate links since the path routing options are small. Therefore, in this situation, the blocking probability of substrate links is dependent. However, if we have a large-size substrate network with mesh topology, the dependency between two paths is quite low. First, the probability of two paths having joint links is low. Second, even if two paths share one substrate link, the probability that they continue to share multiple links is very small. In [99], the authors proposed an algorithm to estimate the correlation between two node-joint links with a fixed routing scenario in a ring network with the same bandwidth capacity. In our case, we use a more complex and random topology with dynamic routing, it is complicated to calculate the dependency between two links. Therefore, we consider the blocking probability among substrate links is independent in this scenario. If the substrate links are independent, we can assume link-disjoint paths are independent because there are no shared substrate links among link-disjoint paths. Our simulation is not based on the independent assumption. The results show that our model could estimate our GAOne model accurately. This also verifies our independent assumption is reasonable.

We add the above discussion into our thesis in Section 3.2, from page 43, last paragraph, "While there are many paths that may exist between a source-destination pair, shorter paths are more favored for VNE due to the fact that they use less link resources. However, it is not difficult to see that shorter paths between a source-destination tend to have many joint links. …" to page 45, first paragraph, "…If the substrate links are independent, we can assume link-disjoint paths are independent because there are no shared substrate links among link-disjoint paths."

7. How do you select the K and E?

   K and Em are the parameters in our analytical model, where K is the number of shortest paths and Em is the maximum path length. To make a fair comparison, we select the K and Em referring to the VNE algorithm that we would like to compare. In this thesis, we compare the simulation performance of GAOne algorithms with our analytical model. Therefore, we use the K and Em same as the settings in the GAOne simulation.

   We emphasize how we choose K and Em in Section 5.1.1, from page 97, last paragraph, "the value of Em is a major factor contributing to the computing complexity…" to page 98, second paragraph, "…The purpose of maintaining dynamic path pools was to increase path diversity with a genetic algorithm. This dynamic pool makes it closer to our random topology assumption."

8. What are the assumptions taken by the proposed methods for substrate link blocking probability and for the evaluation of the maximum number of nonoverlapping paths between two nodes? Under which circumstances the derived analytical results will approach to ones of real scenarios?

   As we discussed in **Theorem 3.1**, the maximum number of potential link-disjoint paths that have e links is $K_{I,e} = \tilde{n}_s - 2$, where $\tilde{n}_s$ is the number of the substrate nodes. In our analytical model, we consider a random substrate topology. The maximum number of potential link-disjoint paths is only dependent on the number of substrate nodes. Therefore, for any real scenario, **Theorem 3.1** is always satisfied. We approximate the number of link-disjoint paths in a random topology in (3.7) and (3.9). Eq. (3.7) will be more accurate when there are less number of nodes and lower average degree of nodes because there will be less alternatives. When there are more nodes and higher degrees, Eq. (3.9) will be more accurate because the intermediate nodes will not matter very much when there are many alternatives.

   We add the above arguments in Section 3.2, page 49, "Both (3.7) and (3.9) can be used to estimate the probability that there exist exactly k_e link-disjoint paths with length e. (3.7) will be more accurate when there are less number of nodes and lower average degree of nodes because there will be less alternatives. When there are more nodes and higher degrees, (3.9) will be more accurate because the intermediate nodes will not matter very much when there are many alternatives."

9. Corollary 3.1 provides the probability that there exist exactly ke nonoverlapping paths with length e. It's also similar in Theorem 3.1. What is the sample space when talking about "probability" in this case?

   The sample space is any source destination pair in any substrate network with $\tilde{n}_s$ substrate nodes. We modify our theorems to make it clear, in Section 3.2, Lemma 3.1 to Corollary 3.5, "For any source-destination pair in any SN with $\tilde{n}_s$ nodes…"

10. Considered maximum flows of the sd pair in the evaluations of the above?

    In a splittable and static scenario, the maximum flow of a source-destination pair could be calculated and considered as the maximum acceptable demand between the source-destination pair. However, in an online problem, the capacity of each substrate link may fluctuate due to the virtual network requests arriving and departing. Furthermore, for an online problem, demands are given dynamically. Without future knowledge, typically heuristic fitness functions are defined to guide the resource allocation. Besides, in this thesis, we focus on unsplittable mapping, each virtual link is mapped into one substrate path. Therefore, the maximum flows of a source-destination pair can not help to estimate the performance of online, unsplittable mapping scenarios.

    We modify our thesis by emphasizing our unsplittable and online requirements in Section 3.2, page 42. We describe the details of unsplittable mapping and splittable mapping in Section 2.2.1.3, page 18.

11. What is the KSP algorithm used in the research? How do you determine K here? What is the inference of K toward your research results?

    For the K shortest paths with joint links, we use Yen's algorithm [110] to find K shortest paths with joint links. Yen's algorithm finds the shortest path first using the Dijkstra algorithm. Then, a recursive approach is used to set one link of shortest paths with infinite cost temporarily each time. In terms of the K shortest link-disjoint paths, if the shortest path is found by the Dijkstra algorithm, all links in the shortest path are removed until K paths are found.

    To make a fair comparison, we select the K referring to the VNE algorithm that we would like to compare. In this thesis, we compare the simulation performance of GAOne algorithms with our analytical model. Therefore, we use the K same as the setting in the GAOne simulation.

We add the K shortest paths algorithms used in our model in Section 5.1, page 98, second paragraph, "The path pools are obtained by finding K shortest paths between any source-destination pair. For the K shortest paths with joint links, we use Yen's algorithm [110] to find K shortest paths with joint links. Yen's algorithm finds the shortest path first using the Dijkstra algorithm. Then, a recursive approach is used to set one link of shortest paths with infinite cost temporarily each time. In terms of the K shortest link-disjoint paths, if the shortest path is found by the Dijkstra algorithm, all links in the shortest path are removed until K paths are found." And Sectin 5.1 also describes how we set the K parameter, at page 98, second paragraph.

12. Have you considered using ILP for the proposed problem for optimal solution? It may be hard to position a scheme without a "true optimal" case.

    We select D-ViNE and R-ViNE [50] as the compared algorithms that are known as benchmarks due to their mixed integer linear programming (MILP) approaches. We introduced our compared algorithms in point 2.

Prof. Minyi Huang

13. Is the general Erlang your contribution?

    No. The generalized Erlang model is not our contribution. We use the generalized Erlang model in our analytical model to estimate the blocking probability at the substrate level. We quantize the capacity of substrate networks and the demand of virtual networks, so that the capacity of substrate networks can be considered as multiple servers in the generalized Erlang model and the demand of virtual networks can be treated as different classes of requests in the generalized Erlang model.

    We cite the generalized Erlang model in Section 2.3.2, page 35 and describe the background of the generalized Erlang model in Section 2.3. We also explain how we apply the generalized Erlang model in our analytical model in Section 3.1, page 41, third paragraph, from "There is no existing way to calculate the blocking probability when $b(e_v)$ is a random real number. Generalized Erlang loss model [1] is the closest one that can be used to approximate this blocking probability. However, the generalized Erlang model requires requested bandwidth and link capacity to be discrete. In order to use generalized Erlang model, we have to quantize bandwidth requests into a fixed number of intervals denoted as R…."

14. Is the assumption satisfied?

    In the generalized Erlang model, the arrival requests follow the Poisson distribution with an exponentially distributed holding time. This arrival requests assumption is also used in general VNE problems. Besides, the generalized Erlang model requires the requests and servers to be discrete. We quantize the capacity of the substrate network and demand of the virtual networks into a fixed number of intervals. Therefore, all assumptions of the generalized Erlang model are satisfied in our analytical model. Our simulation result at the substrate level justifies that our assumption is reasonable and acceptable.

We modify our paper in Section 3.1, from page 40, last paragraph, "In most VNE simulation setups, the VN requests arrive following the Poisson distribution. We consider the same scenario as other general VNE research. At the substrate link level, since we consider a random topology and all the substrate links are undifferentiated, we can treat the arrivals at a substrate link are random samples from the Poisson arrivals of the virtual network…" to page 41, third paragraph, "… In order to use generalized Erlang model, we have to quantize bandwidth requests into a fixed number of intervals denoted as R".
And in Section 5.1.2, page 100, first paragraph, "As shown in Fig. 5.2, the maximum difference, which happens at arrival rate 4, is around 5%. This result justifies our assumption of (3.2) is acceptable."

15. In some applications, overlapping paths are preferred. Why do you prefer nonoverlapping paths?

    In some applications, multiple flows share the same resource to increase the efficiency of the communication system. It is also called multiplexing. Multiplexing is used in the case that there are multiple flows or multiple paths. However, in our DRART model, we deal with the blocking probability of one path. We choose link-disjoint paths as the candidate paths of one virtual link. Therefore, in our model, it is not the multiplexing scenario. We discussed the reason that we prefer the link-disjoint paths at point 6.

16. Page 52, how do you justify node and link mappings are independent?

    In **Theorem 3.4**, we assume the blocking probability of virtual nodes and virtual links are independent. Therefore, we can get a product form solution of the virtual network acceptance ratio. In our simulation, our GAOne maps the virtual nodes and virtual links dependently and coordinately. The result of the virtual network acceptance ratio justifies our assumption is reasonable.

    We justify our results in Section 5.1.4, page 102, second paragraph, "Also shown in Fig. 5.5 are our analytical results in comparison with simulation results. We can see that our analytical results are very close to the simulation results with link-disjoint paths. This confirms that the independence assumptions we made in Section 3.2 are acceptable."

17. The number of substrate network is only 3, why is that?

    In our simulation, we generate random substrate networks by the Waxman model with parameters $\alpha = 0.5, \beta = 0.2$. Each substrate network has 50 nodes. Therefore, there are $C_2^{50} = 1225$ number of source and destination pairs in one substrate network. Besides, for each substrate network, we generate 12,000 to 24,000 virtual network requests to get steady-state results. Therefore, it is not necessary to increase the number of substrate networks since our simulation size is large enough.

We modified our thesis and add the above discussion in Section 5.1.1, page 96, third paragraph, "To introduce randomness, we randomly generated three SNs in our simulation. The reason that we limited the number of SNs to 3. Therefore, there are $C_2^{50} = 1225$ number of source and destination pairs in one substrate network. Besides, we had to generate a very large number of VNs for each SN to get steady state results.
On average, we generated around 18,000 VNs for each SN and each load, which was a very time-consuming process.

We generated random VN requests following the Poisson process with λ_v ranging from 4 to 8 requests per 100 time units. "

Prof. Wei Shi

18. Which category is your approach?

    Our proposed GAOne algorithm is an online, coordinated, unsplittable and concise approach. Our SBGA algorithm is in the same category. The difference between GAOne and SBGA is that SBGA is not a fully coordinated solution since it solves the virtual node mapping and virtual link mapping in two separate stages.

    We add the category of our proposed algorithms in Section 4.2, page 59, last paragraph, "In this section, we discuss our virtual link mapping solution called Segment Based Genetic Algorithm (SBGA). Our proposed GAOne algorithm is an online, coordinated, unsplittable and concise approach. SBGA is not a fully coordinated solution since it solves the virtual node mapping and virtual link mapping in two separate stages. We talk about the details in this section." And in Section 4.3, page 75, last paragraph, "Our proposed GAOne algorithm is an online, coordinated, unsplittable and concise approach."

19. What is hard mapping and soft mapping?

    We proposed hard mapping and soft mapping algorithms in the proposal. We finally changed our topic to the analytical model in our thesis. We add flexible mapping idea to future work.

    We add future work of the flexible mapping in Section 6.2.1, page 114-115.

20. Justify why not reinforcement learning?

    There are some research papers discussing reinforcement learning (RL) solutions with exhilarating results. However, it is challenging to apply RL to an online resource allocation scenario. Specifically, training in RL is parametric based, which is usually implemented in a stationary environment. In a dynamic environment, RL requires training and building a model for each specific scenario. This means RL has to handle a large number of models. Therefore, the research of RL is confronted with computational complexity problems.

When we deal with online resource allocation, we have to face a highly dynamic environment with highly changed parameters. To be specific, the VN demands are unpredictable with different required holding times, different arrival rates, different request topologies, and different requested resources of virtual nodes/links. A new model should be trained if any parameter is changed. This makes offline pre-training hard to achieve since a large number of models have been trained and stored before all environments happen. If we choose to train the model online, it can hardly meet the delay requirements due to the time-consuming training process. Therefore, RL is hard to meet the online resource allocation requirements.

This above discussion could be found in Section 2.2.3.5, page 34, second paragraph, from "There are many research papers [91-94] discussing RL solutions with exhilarating results. However, it is challenging to apply RL to an online resource allocation scenario. Specifically, training in RL is parametric based, which is usually implemented in a stationary environment. In a dynamic environment, RL requires training and building a model for each specific scenario. This means RL has to handle a large number of models. Therefore, the research of RL is confronted with computational complexity problems.

When we deal with online resource allocation, we have to face a highly dynamic environment with highly changed parameters. To be specific, the VN demands are unpredictable with different required holding times, different arrival rates, different request topologies, and different requested resources of virtual nodes/links. A new model should be trained if any parameter is changed. This makes offline pre-training hard to achieve since a large number of models have been trained and stored before all environments happen. If we choose to train the model online, it can hardly meet the delay requirements due to the time-consuming training process. Therefore, RL is hard to meet the online resource allocation requirements.

21. What your system can handle, one-by-one, or multiple at a time?

    In our simulation, both of GAOne and SBGA handle the request one by one. It is a nice suggestion to serve multiple requests at the same time in future work.

    We add the description in Section 5.1.1, page 96, last paragraph, "We assume the requests arrive one by one, and our VNE algorithms handle one request at a time."

22. Is 50-node network a large network?

    As we discuss at point 12, the combination of source and destination pairs is 1225, which is a large number. We generate three random substrate networks. In total, there are 3675 combinations. Overall, the virtual node mapping is selected from the substrate nodes randomly. In our DRART model, a virtual link mapping solution is selected from these source-destination combinations randomly. We also consider the different link-disjoint paths between one source-destination pair. Therefore, we claim our substrate network topology size is large enough.

We modified our thesis and add the above discussion in Section 5.1.1, page 96, second paragraph.

23. Your references are very dated.

Thank you for the suggestion. We have added 33 more references in the thesis.

24. Can you describe the difference between your work and reference 6, 39, 40, 44, 45?

The previous research [6] proposes a one-stage VNE algorithm. In NAL, the virtual nodes are ranked and mapped using a heuristic approach one by one. A virtual link is mapped by the shortest path algorithm after the connected two virtual nodes are mapped successfully. The authors in the paper [49] claim they propose a one-stage algorithm called VIMS to map virtual nodes and virtual links jointly. VIMS maps the virtual node with the highest node degree. And then allocates virtual nodes adjacent to the mapped virtual nodes. Similar to [6], the virtual link is mapped immediately by the shortest path algorithm after the connected two virtual nodes are mapped. Different from two-stage VNE algorithms that solve all the virtual nodes first, and then map the virtual links.
both [6] and [49] solve the virtual node and links alternately, which are not fully coordinated solutions. In our GAOne, we use a genetic algorithm to generate virtual nodes and virtual links solutions at the same time in the crossover procedure. Therefore, our GAOne is a full one-stage coordinate solution.

[39] and [40] propose genetic algorithms to solve VNE. However, the genetic algorithm is only used in the virtual node mapping stage. After virtual node mapping, the virtual link mapping is solved by shortest paths. Our SBGA algorithm focuses on the virtual link mapping stage, which deals with paths. And Our GAOne works on the virtual node mapping and virtual link mapping at the same time.

The research [44] and [45] develop reinforcement solutions to solve VNE. The authors in [44] propose an encoding method to automatically extract the allocation problem features with no manual intervention by using a convolutional deep neural network. Another research [45] proposes a light-weight but efficient Deep RL framework, which formalizes the allocation problem as a Markov Decision Process with appropriate states and actions. Combining with a heuristic algorithm, [45] simplifies and converts unfeasible solutions to feasible solutions.

We modify our thesis by the arguments. The details in the thesis could be found in Section 2.2.1.2, page 17, second paragraph, from “Even though there is …, which are not fully coordinate solutions", in Section 2.2.3.3, page 31, last paragraph, "Above GA approaches pay more attention to node mapping…", in Section 2.2.3.5, page 33, second paragraph, "The authors in [87] propose an encoding method… [88] simplifies and converts unfeasible solutions to feasible solutions".

25. Add more after 2019.

Thank you for this comment. We added 32 more references after 2019. There are [3], [5], [10], [19], [33], [34], [38], [39], [44]-[47], [51], [55], [57]-[61], [67], [68], [74]-[76], [79], [89]- [94], [107], [114],[115].

26. How generic is your approach for serving as benchmark?

In **Theorem 3.4**, we can get the acceptance probability of the virtual network by the product form of the acceptance probability of all virtual nodes and the acceptance probability of all virtual links. In other words, our analytical model is sampling the source-destination pairs from all possible combinations of virtual node and virtual link mapping. Therefore, our analytical model considers all the combinations of source and destination pairs. In terms of different VNE algorithms, some VNE algorithms propose uncoordinated solutions that map the virtual nodes and virtual links in different stages. These uncoordinated solutions only take a subset of all possible combinations into account. In this case, our analytical model could provide a benchmark for these two-stage solutions. For some coordinate solutions, our analytical model aims to estimate the performance. As shown in Fig. 5.5, our analytical results are very close to our GAOne with link-disjoint paths, which is considered a fully coordinated solution.

Our model also covers all topologies, which is very generic than any simulation. A model developed under a specific substrate topology cannot be used to evaluate the performance of various embedding algorithms due to the lack of generality. In our analytical model, we assume the substrate network can be any topology.

The above arguments can be found in Section 5.1, page 94, last paragraph, "Most of the existing allocation algorithms are the so-called two-stage algorithms, where all virtual nodes are mapped first and the mapped virtual nodes provide a source-destination pair for each virtual link in the SN, which makes virtual link mapping in the second stage much easier…" to page 95, second paragraph, "…However, as (3.20) shown, our analytical model gets the acceptance probability from the product of virtual node and virtual link acceptance probability, which includes all possible combinations." And in Section 1.1, page 3, second paragraph, "Furthermore, a model developed under a specific substrate topology cannot be used to evaluate the performance of various embedding algorithms due to the lack of generality. By assuming random substrate network topology, the difficulty for calculating blocking probability increases dramatically."

27. Why do you choose other performance evaluation metrics?

Except for the acceptance ratio, we also pay attention to the remaining bandwidth ratio and Execution time. The remaining bandwidth ratio shows the bandwidth utilization. In our VNE algorithms GAOne and SBGA, one of our goals is to make a sensible allocation on the link mapping to save substrate bandwidth resources and benefit future requests. On the other hand, in an online scenario, the execution time is another important metric to measure the algorithms' efficiency. Therefore, we also choose execution time as one performance metric.

The performance metrics used in this thesis could be found in Section 5.2.1, page 105-106.

28. How often do you do request allocation in parallel to reflect the reality?

    Our parallel framework aims to accelerate the embedding speed and provides a fast and efficient VNE solution. For users, they may accept a slow mapping solution. However, it is undoubtedly that a fast response time increases users' satisfaction and improves the quality of service, especially in an online scenario. Moreover, a fast allocation encourages users to choose this service more often. On the other hand, if service providers could allocate users' virtual network requests fast and efficiently, they can serve more users and increase profits. Therefore, a parallel framework that can provide fast and efficient solutions for online VNE is very promising.

    We modify our thesis in Chapter 1, page 2, last paragraph, "With the increasing growth of application traffic and the increasing demands of various network services, developing a dynamic, fast and efficient solution for resource allocation in NV environment has attracted compelling attention. For users, they may accept a slow mapping solution. However, it is undoubtedly that a fast response time increases users' satisfaction and improves the quality of service, especially in an online scenario. Moreover, a fast allocation encourages users to choose this service more often. On the other hand, if service providers could allocate users' virtual network requests fast and efficiently, they can serve more users and increase profits. Therefore, the dynamic, fast and efficient solution also becomes promising for supporting future network technologies.

29. Future work

    We add Future work in Section 6.2, page 115-117.

Prof. Hussein Mouftah

30. Why didn't you propose any future work for other people to work on?

    We add Future work in Section 6.2, page 115-117.

31. Chapter 5, what simulation tool did you use?

For our analytical model, we use python. When it comes to our proposed VNE algorithms, GAOne, SBGA and other compared algorithms are simulated by C++. The linear program solver used in compared algorithms is glpk.

The simulation tools are introduced in Section 5.2.1 Performance metrics, page 105.

32. Have you done any validation and verification for your simulation results?

    Our simulation results of the analytical model can be mutually verified by theoretical derivation.

    We add some justifications in Chapter 5, which are already illustrated at point 4,6,9, and 11.

33. How can other people reproduce your results?

    To make others understand our VNE algorithms clearly, we use the pseudo-codes for some important steps in Section 4.2 page 66 and Section 4.4 page 88. We also give figures of our framework (Fig. 4.1 and Fig.4.2) to show how the procedures work together. We gave all the parameter setups in Section 5.1.1 page 95-98. Besides, all the source codes will be submitted, so that, other people could reproduce our results.

34. In conclusion, is quite accurate or accurate?

    Thanks for this comment. As shown in Fig.5.5, the acceptance ratio of our analytical model is close to our GAOne model with link-disjoint paths. We modify our conclusion, in Section 6.1, page 115, "Our numerical results show that the model we created is reasonable and acceptable. Our analytical model can serve as a benchmark for the performance prediction of VNE algorithms."

35. How many of your bibliography are cited in text? Why don't you call it references?

    We increase our references from 61 to 115. All these papers are cited in our thesis, Therefore, we change the "bibliography" to "references" in our revised version.

Prof. Changcheng Huang

36. What are other potential applications for DRART?

    Our DRART model can be applied to the scenario where dynamic routing is required. For example, in the vehicular ad hoc network (VANET), the communication performance between two vehicles can be estimated by DRART. However, there are some differences between the scenarios in our simulation and the VANET. VANET is a heterogeneous, dynamic moving network. It is a good extension of our DRART model.

Another future work is on drone networks. The routing in drone network includes node routing and arc routing [115]. Node routing aims to solve the delivery issues and arc routing focuses on the inspection of ground infrastructure. Different routing types of requests have different types of demand and constraints. An analytical model derived from our DRART is promising to estimate the performance of allocating the request to a drone device. Therefore, the performance model in drone networks is an interesting topic to explore in the future.

We add the application for DRART in Section 6.2.2, page 117.

37. Do you have any justification why nonoverlapping is adopted? Exactly why is there a gain?

As we talked about in Section 2.2.1.3, most unsplittable VNE algorithms are based on the shortest path algorithm. Without considering nonoverlapping(link-disjoint) paths, if the shortest path is blocked, the second shortest path has a high probability to be blocked. Specifically, the links in the shortest path are also favored by other short paths. Therefore, under a K shortest path algorithm without the nonoverlapping(link-disjoint) constraint, these K paths are likely overlapped with each other. Consequently, if one path is blocked, these K paths may be blocked at the same time. It tends to increase the blocking probability of mapping a virtual link. As we discussed at point 6, the nonoverlapping(link-disjoint) paths can be treated as independent. Therefore, we believe that VNE algorithms with nonoverlapping(link-disjoint) paths outperform the ones with overlapping(non-link-disjoint) paths. The results shown in Fig. 5.5 also verify our argument on the overlapping paths.

We add above arguments in Section 3.2, page 43, last paragraph "While there are many paths that may exist between a source-destination pair, shorter paths are more favored for VNE due to the fact that they use less link resources. However, it is not difficult to see that shorter paths between a source-destination tend to have many joint links. Without taking any extra measures, if one short path is blocked, other short paths will likely be blocked too. This will make blocking probability higher as shown later in Section 5.1 and make the calculation of blocking probability more complex due to complex correlation structure".