Journal of Network and Systems Management (2022) 30:5
https://doi.org/10.1007/s10922-021-09613-9

Joint Reliability-Aware and Cost Efficient Path Allocation and VNF Placement using Sharing Scheme

Abolfazl Ghazizadeh[2] · Behzad Akbari[2] · Mohammad M. Tajiki[1]

Received: 22 September 2020 / Revised: 12 April 2021 / Accepted: 11 June 2021 /
Published online: 14 September 2021

Abstract

Network Function Virtualization (NFV) is a vital player of modern networks providing different types of services such as traffic optimization, content filtering, and load balancing. More precisely, NFV is a provisioning technology aims at reducing the large Capital Expenditure (CapEx) of network providers by moving services from dedicated hardware to commodity servers using Virtualized Network Functions (VNF). A sequence of VNFs/services following a logical goal is referred to as a *Service Function Chain (SFC)*. The movement toward SFC introduces new challenges to those network services which require high reliability. To address this challenge, redundancy schemes are introduced. Existing redundancy schemes using dedicated protection enhance the reliability of services, however, they do not consider the cost of redundant VNFs. In this paper, we propose a novel reliability enhancement method using a shared protection scheme to reduce the cost of redundant VNFs. To this end, we mathematically formulate the problem as a Mixed Integer Linear Programming (MILP). The objective is to determine optimal reliability that could be achieved with minimum cost. Although the corresponding optimization problem can be solved using existing MILP solvers, the computational complexity is not rational for realistic scenarios. Thereafter, we propose a Reliability-aware and minimum-Cost based Genetic (RCG) algorithm to solve this problem with low computational complexity. In order to evaluate the proposed solution, we have compared it with four different solutions. Simulation results show that RCG achieves near-optimal performance at a much lower complexity compared with the optimal solution.

Keywords Software defined network (SDN) · Network function virtualization (NFV) · Service function chaining (SFC) · Fault tolerance · Redundancy scheme · Resource reallocation

✉ Abolfazl Ghazizadeh
abolfazlghazizadeh@modares.ac.ir

Extended author information available on the last page of the article

1 Introduction

Network traffic flows may need to be served or screened through different hardware middle-boxes while passing the network; as an example of such middle-boxes consider HTTP proxies, Intrusion Detection Systems (IDSs), Network Address Translators (NATs), and firewalls. In order to reduce the capital and operational expenditure of using middle-boxes and to increase the flexibility and scalability of services provided by them, Network Function Virtualization (NFV) replaces hardware middle-boxes with more flexible software applications known as *Virtual Network Functions (VNFs)*. On the other hand, the Software Defined Networking (SDN) paradigm offers the possibility to control the forwarding of packets from a logically centralized point of view, thus easing the introduction of efficient and flexible algorithms to optimize the utilization of network and processing resources [1]. Motivated by the collaboration of SDN and NFV, the topic of VNF as a Service (VNFaaS) is currently under attentive study by both telecommunication and cloud stakeholders as a promising direction [2, 3].

Optimal resource allocation is an essential metric for network providers to reduce their costs and maximize their efficiency [4]. Besides, to increase customers' Quality of Experience (QoE) and minimize the energy consumption, the VNFs need to be be dynamically relocated between network nodes, i.e., a running VNFs may need to migrate from a server to another one. Consequently, the placement of VNFs is a fundamental issue to efficiently deploy NFV technology. On the other hand, recent works focus on optimizing the resource (both nodes and links) utilization and developing efficient algorithms for the joint problem of VNF placement and network traffic routing [5, 6].

Another important metric of choosing a service provider is the reliability of its services. This forces the service providers to seek for NFV deployment algorithms that keep the reliability above some standards. VNFs are usually executed on commercial-off-the-shelf (COTS) network elements. COTS elements are characterized as low reliable devices meaning their reliability is significantly lower than carrier-grade equipment. Additionally, the COTS's operation may be affected by increasing the computing load, hardware failures or malicious attacks [3]. To ensure a desired level of end-to-end (e2e) reliability, redundancy scheme is an efficient way that is used in many works. There are two types of redundancy: (1): with dedicated protection, (2): with shared protecting. Existing redundancy methods with dedicated protection, enhance the reliability of services without considering the cost of redundant network functions. On the other hand, existing redundancy methods with shared protecting use an On-demand scheme that increases preparation time up to 3 times [7].

Motivated by the aforementioned considerations, we address the joint problem of VNF placement and flow routing with reliability and QoS considerations. More precisely, we study the joint problem with the objective of maximizing the resource utilization while keeping the reliability in a desirable threshold using a minimum set of redundant functions. We only consider the reliability of the computational node, because link reliability issues can easily be converted to node

reliability. To this end, we exploit redundancy schemes by mathematically formulating the the problem of minimum resource consumption with respect to QoS constraints. Thereafter, we use an Mixed Integer Linear Programming (MILP) solver to optimally solve the corresponding optimization problem. Due to the high computational complexity of MILP solvers, we propose an efficient meta-heuristic algorithm to handle the scalability issue over large-scale networks. Our main contributions are summarized as follows:

- We propose a new reliability-aware resource allocation algorithm using shared protection scheme with Active-Standby redundancy. The algorithm is proposed for software defined networks to address the SFC problem with the objective of minimizing redundant VNFs without affecting the Quality of Service (QoS) parameters;
- Mathematical formulation of the joint problem of VNF placement and routing for the proposed protection scheme by considering QoS parameters. The corresponding optimization problem belongs to the class of mixed-integer quadratically constrained programming (MIQCP) in our first natural formulation;
- Linearization of the non-linear constraints in order to have the modeling in form of Mixed integer linear programming (MILP) which is solvable using existing ILP solvers such as IBM CPLEX;
- We propose a near optimal meta-heuristic algorithm to solve the mentioned problem in a reasonable execution time. The proposed algorithm is an scalable solution which can be used for large-scale networks;
- Comparison of the Genetic algorithm with state-of-the-art algorithms and the optimal solution through a set of various metrics, which includes: i) execution time, ii) bandwidth consumption, and iii) transmission latency.

The rest of this paper is organized as follows: Sect. 2 goes through literature and surveys related works. Section 3 discusses one of the most important reliability enhancement schemes called 'shared protection scheme' and compares it with the other schemes using illustrative examples. Section 4 then provides the system model and problem formulation. To solve the scalability issues a meta-heuristic algorithm is proposed which is described in Section 5. Besides, to evaluate the proposed solution, numerical results are presented in Sect. 6. Finally, the paper is concluded and Remarks and outlines regarding the open research problems are included in sect. 7.

2 Related Works

In the following, the main literature on NFV related to our work is discussed. From now on we refer to the Joint problem of path Allocation and VNF placement as *Service Function Chaining (SFC)*. Related works are divided into two different categories: i) SFC solutions focusing on minimizing the fault/failure probability [8–14], and ii) SFC solutions focusing on redundancy protection [3, 15–23]. We then describe the works falling in each category.

2.1 SFC Solutions Focusing on Minimizing the Fault/Failure Probability

The available literature ranges from the problem of fault detection and recovery solutions [9, 10] to the problem of fault-aware routing of the network traffic in SDN/NFV infrastructure [11]. More in detail, they discuss failure occurrence and fault tolerance in the OpenFlow-enabled networks. The main goal is to propose a node/link failure recovery and fault detection method in the data plane that can be controlled through the controller. However, they neither cover the SFC fault-awareness, nor consider the application plane side-effect.

The authors of [13] propose a cost-efficient solution to detect link failures in order to increase the fault tolerance by combining the flow retrieval which is achieved through analyzing the protection switching times and using a fast protection method. Interestingly, this paper supports the fault minimization over the links and addresses the end-to-end fault tolerance method per flow, but the solution is not secured against occurrence of failure. In fact, the system tries to minimize the probability of failure but it cannot handle the occurrence of failure. The authors in [8], present an architecture for Fault Prevention and Failure Recovery which is a multi-tier structure in which the network traffic flows pass through networking nodes to decrease the energy consumption and network side-effects of traffic engineering. Similar approach is taken in [12], to formulate the problems of flow routing, allocation of VNFs to flows, and VNF placement as Integer Linear Programming optimization problems. Since the formulated problems cannot be solved in acceptable timescales for real-world problems, they propose several cost-efficient and quick heuristic solutions. Both [8, 12] reduce the probability of failure in physical servers, however, they both expose the network unprotected in case of failure in a networking node.

2.2 SFC Solutions Focusing on Redundancy Protection

Numerous works focus on increasing the reliability of each service/VNF separately and do not take the advantages of considering the global information of the VNF Forwarding Graph (VNF-FG). The main drawback of focusing on services/VNFs separately is low utilization of networking resources. A survey of the recent works on SFC is presented in [15] classifying VNF/service protection into three groups: Active-Standby, Active-Active, and on-demand. In the following, some of the state-of-the-art solutions proposed for redundancy protection are discussed briefly.

The authors of [16] proposed a model for dynamic reliability-aware service placement based on the simultaneous allocation of the main and backup servers. Then, they formulate the dynamic reliability-aware service placement as an infinite horizon Markov decision process, which aims to minimize the placement cost and maximize the number of admitted services. Although the proposed brings a lot of benefits, it may end up with waste of resources (this has been discussed deeply in Sect. 3). In our solution we exploit share scheme for backups to prevent waste of resources.

In [17, 18] an approach for planning and deploying backup schemes for network functions suggested which guarantee high levels of survivability with significant reduction in resource consumption. In the suggested backup scheme, they take advantage of the flexibility and resource-sharing abilities of the NFV paradigm in order to reduce backup servers. In this article, authors describe different goals that network designers can consider when determining which functions to implement in each of the backup servers. The main advantage of our solution compared to this solution is that we consider the joint problem of path allocation and VNF placement. This improves the performance of our solution. Also, in [18] authors focus on the case where a small number of middleboxes fail simultaneously, and study the backup resources required for guaranteeing full recovery from any set of failures, of up to some limited size. In [17, 18] the authors used the server-level sharing method, that means they share backup servers while we are sharing backup functions. In fact, in their solution, server resources are shared but backup functions are considered as dedicated function. Although this method makes the network more resilient, due to exploitation of extra resources, it needs more physical resources. Similar to [16], the authors used a Dedicated Protection (DP) scheme Which can lead to higher demand on physical resources compared to Shared Protection (SP) scheme.

In [19], an algorithm for minimizing the physical resources consumption is proposed which guarantees the required reliability with polynomial time complexity. The proposed scheme ignores the global information of the VNF-FG and cost of backups, which leads to the VNF over-replication.

An on-demand scheme is a lazy approach of tackling the VNFs failure meaning that it postpones the resource allocation of the backup function to a later time when the failure has occurred. In [19, 20] authors used this method for enhancing reliability of services. This is an efficient way to improve the performance of resources, but increases the fault recovery time.

In Active-Active scheme, all node (including redundant nodes) are active are serving incoming requests [15]. This solution not only requires redirecting traffic in case of failure but also requires a load balancer to be deployed in front of several backups.

The authors of [21], study the the potential of VNFs replications to accelerate network load balancing. In this way, they consider the problem of VNF placement with replications. They mathematically formulate their problem and propose three solutions for the allocation and replication of services/VNFs: Genetic Algorithm (GA), LP solver, and Random Fit Placement Algorithm (RFPA). Similarly, in [22], the optimization problem of load balancing is formulated as a mixed integer linear program. Thereafter, in order to solve the online load balancing problem a fast algorithm is developed.

In both [21, 22] authors focus on increasing the reliability using a replication data flow method through migrating backup functions from low reliability nodes to more reliable nodes. In this method while recovery time is very low the performance is not comparable with other existing methods.

Active-Standby is a method where active VNFs provide specific services, and these active VNFs are protected by one or more standby VNF(s). These redundant

VNFs do not actively provide service and they require a mechanism to redirect traffic to them in case of failure. As an example, authors of [3] follows the Active-Standby method by seeking for a trade-off between end-to-end reliability and computational load over servers. In this way, they exploit the joint design of VNF Chain Composition (CC) and Forwarding Graph embedding (FGE) using a dedicated redundancy scheme. They model the problem in the form of Mixed-Integer LP (MLIP) and exploit existing toolbars to solve the problem. In the same way, authors of [23] propose a multi-path backup scheme to enhance reliability while minimizing the end-to-end delay. Although the aforementioned schemes have many benefits, they lead to wasted resources since they focus on increasing the reliability of each service/VNF individually instead of considering the whole nodes as an integrated entity. Similarly to [17, 18], we consider an Active-Standby method in order to protect SFCs from failures.

3 The Propose Shared Protection Scheme

Considering the fact that hardware components fail frequently due to various human and natural causes (earthquakes, malicious attacks, fibre cuts, etc.), network operators must use protection methods to provide reliable services/functions [24]. Dedicated Protection (DP) scheme is a traditional way to enhance the reliability of SFCs. In this scheme, one or more redundant VNFs will be kept reserved for a service/function that needs high reliability. If DP is provisioned, the reliability of given VNF can be obtained as follow:

$$r_i = 1 - (1 - r_i^p) \cdot (1 - r_b) \quad \forall i \in [1, |F|], b \in \left[1, |F'|\right] \tag{1}$$

where r_i^p and r_b are corresponding to the reliability of primary VNF and reliability of the backup VNF. Although DP can provide high reliability for services, it suffers from high usage of bandwidth and computational resources. In order to balance between resource utilization and reliability the Shared Protection (SP) is a well-known scheme. In this scheme, each backup function can be reserved for several primary functions. Using SP, the reliability of VNF i is:

$$\begin{aligned} & r_i = r_i^p + (1 - r_i^p) \cdot r_b \cdot \varphi_i \\ & \varphi_i = 1 - \sum_{i \neq j} \frac{MTTR_j}{MTTR_i + MTTR_j} \cdot (1 - r_j) \\ & \forall i, j \in [1, |F|], b \in \left[1, |F'|\right] \end{aligned} \tag{2}$$

where r_i^p and r_b are the initial reliability of primary and backup VNFs, respectively. $MTTR_i$ is Mean Time To Repair of VNF i and φ_i is the probability that the shared backup VNF can be assigned to VNF i [25].

In order to clarify this method, we have given an example that compare reliability and bandwidth consumption in SP and DP. Also to evaluate the performance of proposed scheme, we mentioned a No Protection (NP) scheme. Consider a sub-network

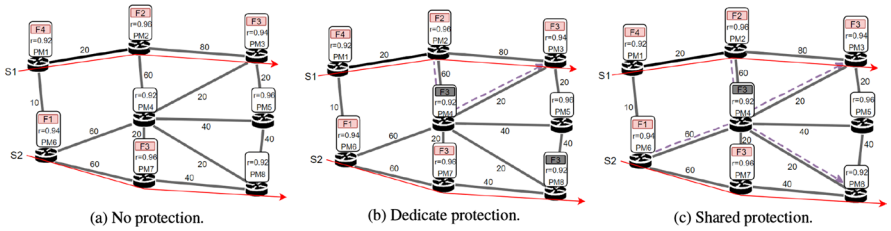


Fig. 1 Protection methods

consists of eight Physical Machine (PM), namely PM_1 through PM_8 as illustrated in Fig. 1a. The substrate network is assumed to host two service function chains, namely s_1 and s_2. s_1 requests for three functions consists of $\{f_4,f_2,f_3\}$ which are respectively hosted on $\{PM_1, PM_2, PM_3\}$ (initiated at PM_1 and destined to PM_3). Similarly, S_2 requests for two functions consist of $\{f_1,f_3\}$ which are respectively hosted on $\{PM_6, PM_7\}$ (initiated at PM_6 and destined to PM_8). Bandwidth requirements for each service is considered to be 20 units. Fig. 1a illustrates No Protection scheme where the reliability of s_1 and s_2 are $r_{s_1} = 0.94 \times 0.96 \times 0.92 = 0.83$ and $r_{s_2} = 0.96 \times 0.92 = 0.883$, respectively, and the consumed bandwidth is $b = 80$ unit. Another example is shown in Fig. 1b, where the f_3 of s_1 and f_3 of s_2 replicated into PM_4 and PM_8, respectively. In case of using DP, the reliability of s_1 and s_2 are $r_{s_1} = 0.94 \times 0.96 \times (1 - (1 - 0.94) \times (1 - 0.92)) = 0.898$ and $r_{s_2} = 0.96 \times (1 - (1 - 0.92) \times (1 - 0.94)) = 0.955$, respectively, as well as the consumed bandwidth is $b = 120$ unit. Figure 1b illustrates the same scenario when DP scheme is deployed to ensure high reliability. The main disadvantage of SP is that, despite higher reliability obtained, it considerably increases the amount of the required resources. In order to reduce the number of replicated VNFs while holding the level of reliability, we purpose a shared protection scheme with Active-Standby redundancy. An example of SP illustrated in Fig. 1c where one backup VNF type f_3 is placed on PM_4 and reserved for f_3 of s_1 and f_3 of s_2, simultaneously. According to Eq. 2 the achieved services reliability in this case is:

$$r_{s_1} = 0.94 \times 0.96 \times 0.992 = 0.895$$
$$r_{s_2} = 0.94 \times 0.992 = 0.932$$

The consumed bandwidth is $b = 160$ unit which is increased by 20% compared to DP.

4 Problem Formulation

In this section, the SFC-aware resource allocation with respect to system reliability is presented. The system model is for a joint problem of VNF placement and flow routing. Consequently, it guarantees the best possible end-to-end reliability for the assigned path to each flow. We also consider the cost of using redundant

resources by making a trade-off between reliability and cost. We have QoS as a fundamental metric in our system model. Hence, the propose system not only ensures the required service of each flow to be delivered via the selected path but also the QoS of the service to be kept in a proper range. In the following, we detail the formulation used in the proposed model. Table 1 defines the symbols with a brief description.

Consider a substrate network as a directed graph $G = (N, L)$, which consist of a set of physical machines N and directed links L. Let C_k be the processing capacity of PM_k where $k \in N$ and each PM can execute several VNFs, depend on its C_k. Let B_m donate the bandwidth capacity of link m where $m \in L$. We donate by S a set of demanded services. Each service $s_i \in S$ is specified by a required bandwidth b_i and accepted minimum reliability θ^i_{req} and source node σ_i and destination node δ_i also j-th function of service i required processing capacity $c_{i,j}$. Let F_i be the ordered chain of VNFs corresponding to service chain s_i.

Table 1 Main notation

Symbol	Definition
Input parameters	
N	The set of servers
F	The set of primary VNFs
F_i	An ordered chain of VNFs corresponding to service chain i
$F\prime$	The set of backup VNFs
S	The set of network services
L	The set of physical links
T	The set of VNF's Types
$c_{i,j}$	The processing capacity requirement of j-th function of service i
θ^i_{req}	The Minimum reliability accepted by service i
ϕ^i_{req}	The Maximum delay accepted by service i
ϕ_l	The delay of physical link l
θ_k	The reliability of server k
σ_i	The source of network service i
δ_i	The Destination of network service i
B_m	The bandwidth capacity of link m
C_k	The processing capacity of server k
$x_{i,j}$	An Integer Value that specifics the Type of j-th function of service i
x'_l	An Integer Value that specifics the Type of backup function l
Variables	
$y^k_{i,j}$	A binary variable that equals 1 if and only if j-th function of service i is located on server k
$y'k_i$	A binary variable that equals 1 if and only if i-th backup is located on server k
$U^l_{i,j}$	A binary variable that equals 1 if and only if l-th backup is assigned to j-th function of service i
$W^m_{i,j}$	A binary variable that equals 1 if and only if service i use link m for access to j-th VNF
$W'l_{i,j}$	A binary variable that equals 1 if and only if service i use link j for access to backup l

In the following, we develop a Mixed Integer Linear Programming (MILP) model to mathematically formulate the problem of reliability enhancement with shared protection scheme. We present the MILP model with all the notation specified in Table 1. In order to make the understanding of mathematical formulation easier, the model is divided into seven parts and each part is discussed separately.

4.1 Reliability Constraints

In this part, constraints related to reliability are discussed. The operation of each VNF may be affected by unexpected failure in its software or its physical machine (*PM*). Each *PM* has specific values for its Mean Time Between Failure (MTBF) and Mean Time To Repair (MTTR). For the sake of simplicity, we refer to the j'th VNF of service i as $VNF_{i,j}$. Let $r^p_{i,j}$ be the reliability of one instance of $VNF_{i,j}$. This value is highly influenced by the reliability of the *PM* hosting this VNF (Eq. 3). Similarly, $r_{i,j}$ is the reliability of $VNF_{i,j}$ considering all instances including the backups.

$$\begin{aligned} & r_{i,j} = r^p_{i,j} + \left(1 - r^p_{i,j}\right) \cdot r_l \cdot U^l_{i,j} \cdot \\ & \left[1 - \sum_{i \neq i'} \sum_{j' \in \left[1, |F'_{i'}|\right]} \frac{MTTR_{i'j'}}{MTTR_{i,j} + MTTR_{i'j'}} \cdot \left(1 - r_{i'j'}\right) \cdot U^l_{i'j'}\right] \\ & \forall i, i' \in [1, |S|], j, j' \in \left[1, |F_i|\right], l \in \left[1, |F'|\right] \end{aligned} \tag{3}$$

Let θ^i_{req} be the minimum reliability accepted by service i and $r^p_{i,j}$ and r_l are the initial reliability of the j'th VNF of service i and backup VNF l, respectively. according to 3, the achieved reliability θ^i of an arbitrary network service s_i is given by:

$$\theta^i = \prod_{j \in \left[1, |F_i|\right]} r_{i,j}, \ \forall i \in [1, |S|], j \in \left[1, |F_i|\right] \tag{4}$$

Where $r_{i,j}$ is the reliability of j^{th} VNF of service s_i. If achieved reliability $\theta^i < \theta^i_{req}$, then improve reliability of services i with add one or more backups to its primary VNFs. For all of network services, the number of redundant VNFs should be sufficient to satisfy its reliability requirement. Let F' denote the set of backup VNFs. Each redundant VNF may be shared with several primary VNF. If redundant VNF l is assigned to VNF j of service s_i, $U^l_{i,j}$ will be 1 otherwise 0. Let x'_l and $x_{i,j}$ respectively be the type of redundant VNF l and the type of j^{th} VNF of service s_i, as well as F' is set of backup VNFs and $U^l_{i,j}$ is a binary variable that equals 1 if and only if l^{th} backup is assigned to j^{th} function of service i The below constrain allow each primary VNF to use redundant VNFs which is the same type.

$$x'_l \cdot U^l_{i,j} = U^l_{i,j} \cdot x_{i,j}, \forall i \in [1, |S|], j \in \left[1, |F_i|\right], l \in \left[1, |F'|\right] \tag{5}$$

4.2 Routing Constraints

In the following, the constraints for flow routing with respect to QoS are discussed. In this formulation, m.head and m.tail respectively represent the first and the last node along link m.

$$\sum_{m \in L \,\&\, m.tail=\sigma_i} W_{i,j}^{m} = 1 \tag{6}$$

$$\sum_{m \in L \,\&\, m.head=\delta_i} W_{i,j}^{m} = 1 \quad \forall i \in [1, |S|], j \in \left[1, |F_i|\right] \tag{7}$$

Let $W_{i,j}^{m}$ be the binary variable that equals 1 if and only if service i use link m for access to j^{th} VNF. Eq. 6 and 7 make sure that the path of each service starts from σ_i and ends in δ_i, precisely.

$$\sum_{m \in L \,\&\, m.head=k} W_{i,j}^{m} \times b_{i,j} = \sum_{n \in L \,\&\, n.tail=k} W_{i,j}^{n} \times b_{i,j} \quad \forall i \in [1, |S|], j \in \left[1, |F_i|\right], k \in [1, |N|] - \{\sigma_i, \delta_i\} \tag{8}$$

Eq. 8 ensures that for each service, the amount of input load to each server is equal to the amount of its output load. Unless the server is the first node (start node) or the last node (end node) of that service.

$$\sum_{m \in L \,\&\, m.tail=k} W'^{l}_{i,m} \geq U_{i,j}^{l} \times y'^{k}_{l} \tag{9}$$

$$\sum_{m \in L \,\&\, m.head=k} W'^{l}_{i,m} \geq U_{i,j}^{l} \times y'^{k}_{l} \quad \forall i \in [1, |S|], j \in \left[1, |F_i|\right], k \in [1, |N|], l \in \left[1, |F'|\right] \tag{10}$$

Let y'^{k}_{l} be the binary variable that equals 1 if and only if i-th backup is located on server k. Eq. 9 and 10 make sure that if backup VNF l is assigned to one of the functions of the service i, there is a backup path that passes through the server which hosts the VNF l.

$$\sum_{m \in L \,\&\, m.tail=k} W'^{l}_{i,m} \geq U_{i,j}^{l} \times y_{i,j-1}^{k} \tag{11}$$

$$\sum_{m \in L \,\&\, m.head=k} W'^{l}_{i,m} \geq U_{i,j}^{l} \times y_{i,j+1}^{k} \tag{12}$$

$$\sum_{m \in L \,\&\, m.tail=k} W'^{l}_{i,m} + y_{i,j+1}^{k} \geq 1 \tag{13}$$

$$\sum_{m\in L\,\&\,m.head=k} {W'}^{l}_{i,m} + y^{k}_{i,j-1} \geq 1$$
$$\forall i \in [1,|S|], j \in \left[1,|F_i|\right], k \in [1,|N|], l \in \left[1,|F'|\right] \tag{14}$$

In Eq. 11-14 a path is marked as used to reach the backup l by j'th VNF of service i if the path Precisely starts from $(j-1)$ and ends in $(j+1)$.

$$\sum_{m,m'\in L\,\&\,m.head=m'.tail=k\,\&\,m'.head=m.tail=k'} {W'}^{l}_{i,m} + {W'}^{l}_{i,m'} \leq 1$$
$$\forall i \in [1,|S|], j \in \left[1,|F_i|\right], k,k' \in [1,|N|], l \in \left[1,|F'|\right] \tag{15}$$

Eq. 15 prevents the formation of the loops on the path and Eq. 16 prevents the path from being cut off.

$$\sum_{m\in L\,\&\,m.tail=m'.head} {W'}^{l}_{i,m'} + y^{m.tail}_{i,j-1} \geq {W'}^{l}_{i,m}$$
$$\forall i \in [1,|S|], j \in \left[1,|F_i|\right], k \in [1,|N|], l \in \left[1,|F'|\right] \tag{16}$$

4.3 The NFV Placement and Anti-Affinity Constraints

$$\sum_{k\in[1,|N|]} y^{k}_{i,j} = 1, \quad \forall i \in [1,|S|], j \in \left[1,|F_i|\right] \tag{17}$$

$$\sum_{k\in[1,|N|]} {y'}^{k}_{l} = 1, \quad \forall l \in \left[1,|F'|\right] \tag{18}$$

Eq. 17 and 18 make sure that Each VNF, such as backup or primary, is executed by one and only one *PM*. As such, the Anti-affinity constraints are formulated as:

$$y'^{k}_{l} + y^{k}_{i,j} + U^{l}_{i,j} \leq 2$$
$$\forall i \in [1,|S|], j \in \left[1,|F_i|\right], l \in \left[1,|F'|\right], k \in [1,|N|] \tag{19}$$

where Eq. 19 ensures $U^{l}_{i,j} \neq 1$ if and only if both primary VNF and selected backup VNF are hosted by same *PM*. Because in the event of a Fail for a *PM*, only one of the primary function or backup function of a service fails.

$$y^{k}_{i,j} + y^{k}_{i',j'} + U^{l}_{i,j} + U^{l}_{i',j'} \leq 3$$
$$\forall i,i' \in [1,|S|], j \in \left[1,|F_i|\right], j' \in \left[1,|F'_i|\right],$$
$$l \in \left[1,|F'|\right], k \in [1,|N|], i \neq i' \tag{20}$$

Eq. 20 ensures that if VNF j and VNF j'select one backup then the functions should be placed on different PMs. Because if they are located on the same PM, when the PM fails, they will need two backup functions at the same time.

4.4 Bandwidth Constraint

We formulated the allocated bandwidth problem as:

$$BW = \sum_{i\in[1,|S|]} \sum_{j\in\left[1,|F_i|\right]} \sum_{m\in[1,|L|]} W_{i,j}^{m} \times b_i + \sum_{l\in[1,|F'|]} \sum_{m\in[1,|L|]} \max_{i\in[1,|S|]} \left(W'^{l}_{i,m} \times U_{i,j}^{l} \times b_i \right) \tag{21}$$

where BW is the total allocated bandwidth and obtained from the sum of allocated bandwidth of each link which obtained from the sum of consuming bandwidth of services which selected the link and accumulate of maximum reserved bandwidth among the services that selected this link as a backup path to access the same backup VNF.

$$\sum_{i\in[1,|S|]} \sum_{j\in\left[1,|F_i|\right]} w_{i,j}^{m} \times b_i + \sum_{l\in[1,|F'|]} \max_{i\in[1,|S|]} \left(W'_{i,m} l \times U_{i,j}^{l} \times b_i \right) < B_m, \forall m \in [1,|L|] \tag{22}$$

Eq. 22 ensures that the total allocated bandwidth on any physical link l cannot exceed its bandwidth capacity B_m.

4.5 Computational Capacity Constraints

In the following, the constraints for computational capacity are discussed.

$$\sum_{i\in[1,|S|]} \sum_{j\in\left[1,|F_i|\right]} y_{i,j}^{k} \times C_{i,j} + \sum_{l\in[1,|F'|]} \max_{i\in[1,|S|]} \left(y'^{k}_{l} \times U_{i,j}^{l} \times c_{i,j} \right) < C_k, \ \forall k \in [1,|N|] \tag{23}$$

Equation (23) ensures that the total allocated computing resources on any PM_k cannot exceed its capacity C_k.

4.6 Delay Constraints

The delay constraints are formulated as follows:

$$\sum_{j'\in\left[1,|F_i|\right] \,\&\, j\neq j' \,\&\, j\neq(j'+1)} \sum_{m\in[1,|L|]} W_{i,j'}^{m} \times \phi_m + \sum_{m\in[1,|L|]} \sum_{l\in[1,|F'|]} W'^{l}_{i,m} \times U_{i,j}^{l} \times \phi_m \leq \phi^{i}{}_{req} \quad \forall i \in [1,|S|], j \in \left[1,|F_i|\right] \tag{24}$$

where Eq. 24 ensures the experienced delay for each service in any combination of primary path and backup path is less than the maximum delay accepted by the service (Table 2).

4.7 Objective Function

The objective function is establishing reliable service chains while minimizing the resource consumption. Our optimization problem is based on two objective:

- Minimizing the bandwidth usage caused by both primary and backup functions:

$$\min\left(\frac{BW}{\sum_{m\in[1,|L|]} B_m}\right)$$

  where BW is the total allocated bandwidth and obtained from Eq. 21 and B_m is the bandwidth capacity of link m.
- Minimizing the utilization of the processing capacity by minimizing number of backup VNFs:

$$\min\left(\frac{|F'|}{|F|}\right)$$

where $|F'|$ is number of backup functions and $|F|$ is number of primary functions. Our objective represents a Multi Criteria Decision Making. The comprehensive objective function is given by:

$$\min\left(\omega\cdot\frac{|F'|}{|F|}+(1-\omega)\cdot\frac{BW}{\sum_{m\in[1,|L|]} B_m}\right)$$

where ω is the preference weight of each sub-goal. Coefficient ω has a critical impact on the performance of the proposed solution. The selection of ω is determined by two criteria: computational consumption and bandwidths consumption. A higher ω implies that the of VNFs computational consumption of the solution is closer to its optimal value, whereas a lower ω implies that the bandwidths consumption is closer to its optimal value. Therefore, the resource utilization is parametric, this enables the datacenter owner to modify the minimization goal. For example, if the datacenter owner feels lack of available bandwidths, then a lower ω can be assigned to the algorithm, which results in a less bandwidths consumption. If lack of computational resource is more sensitive, then a higher ω can be assigned. This provides flexibility in respect to different perceptions about what needs to be more minimized. The objective function considers the minimization of two different costs: server utilization and link utilization.

Table 2 Chromosome 1: VNF's genes

Location	Function type	SFC id	Position in SFC	Used links to access assigned backup VNFs	List of assigned backup	Reliability

5 Genetic Algorithm

Since using existing MILP solvers for the proposed formation is quite complex and challenging even for medium-scale networks, we propose a genetic algorithm to practically solve it. In this section, we develop a reliability-aware placement based genetic algorithm that jointly optimizes node mapping and routing while Considering the reliability of the SFCs to achieve desirable reliability with minimum resource consumption. The pseudo code of the proposed genetic algorithm is provided in Algorithm 1. The algorithm finds solutions in the processes of initial population generation, fitness evaluation, selection, crossover, and mutation. First, generate random population of P individuals and evaluate the fitness of each individual in the population and select set of parent individual from the population according to their fitness (lines 1 through 5 of Algorithm 1). According to a crossover probability, crossover the parents to form new offspring and with a mutation probability mutate new offspring then checking for the new individuals satisfy the constraints and update people ranks (lines 6 through 21 of Algorithm 1). If it converges, provides fittest individual and terminate possess else this possess repeats as far as convergence occurs (lines 22 through 27 of Algorithm 1). In the following sub-sections we present the encoding mechanism, feasibility checking process, and the fitness function.

An example of the encoding mechanism and crossover illustrated in Fig. ??. Where individuals #1 and #2 assumed as parents. Parent #1 has no backup VNF and it is placed on physical machine *PM*3. On the other hand, parent #2 has one backup VNF and it is placed on physical machine *PM*4. The child has one backup VNF and is placed on physical machine *PM*3.

Table 3 Chromosome 1: SFC's genes

SFC id	Used links	Maximum tolerable delay	Minimum tolerable reliability

Table 4 Chromosome 2

Location	Function type	List of user VNF	Reliability

Algorithm 1 Pseudo-Code of Reliability-aware and minimum-Cost based Genetic (RCG)

INPUT: $\alpha, \beta, \gamma, \delta$
α: size of population
β: rate of elitism
γ: rate of mutation
δ: convergence threshold
OUTPUT: X
X: solution
//Initialization
generate α solutions randomly;
save them in the population Pop;
//Loop until the Convergence Condition
do
//Elitism based selection
number of elites $ne = \alpha \cdot \beta$;
select the best ne solutions in Pop and save them in Pop_1;
//Crossover
number of crossover $nc = (\alpha - ne)/2$;
for $j = 1$ to nc **do**
select two solutions X_A and X_B from Pop;
generate X_C and X_D by one-point crossover to X_A and X_B;
save X_C and X_D to Pop_2;
end for
//Mutation
for $j = 1$ to nc **do**
select a solutions X_j from Pop_2;
mutate each bit of X_j under the rate γ and generate a new solution X'_j;
if X'_j is unfeasible **then**
update X'_j with a feasible solution by repairing X'_j
end if
update X_j with X'_j in Pop_2;
end for
//Updating
calculate rank of Pop_2 individuals;
update $Pop = Pop_1 + Pop_2$;
//Convergence Condition
calculate D_c using Eq. 25;
if D_c value is less than δ **then**
break;
end if
while (*true*)
return best solution X in Pop;

5.1 Encoding Mechanism

In general, our NFV Network encodes two chromosomes: chromosome 1 (Tables 2 and 3) represents the location and assigned backups of VNFs of each service also used link for connecting VNFs of each service and chromosome 2 (Table 4) represents the location, function type, list of user VNFs and calculated reliability of each backup VNF in the network. Given a network with i services, j VNFs and k Backup VNFs, chromosome 1 consists of i+j genes, i genes for representing the path of each service and i gene for representing the properties and assigned backups of a primary VNF. Chromosome 2 consists of k genes, each of which represents the specifications of a Backup VNF.

5.2 Checking for Feasibility

Crossover phase and mutation phase can cause VNF mapping that cannot satisfy NFV Placement and Anti-affinity constraints in previous section. So the phase added after crossover and mutation to check the feasibility of mapping results. If there are invalid mapping result, we have to correct them so that they can meet all the constraints.

5.3 Selection

In selection strategy, we use a ranking scheme to avoid premature convergence. The ranking scheme is such that first the fitness value of each individual calculated then the individuals are sorted based on this value then the individuals based its rank selected for crossover phase, which means the individual with larger fitness value has a higher chance of taking part in crossover presses.

The penalty process is used to calculate points and rank. In this process the individuals who do not satisfy the constraints of the problem, are given a negative score depending on the importance of the violated constraint. This makes the individual violating constraints get lower scores, thus reducing the probability of selecting the individual in the next crossover.

5.4 Convergence Condition

To evaluate the performance of the algorithm we modify the degree of diversity [26] as follow:

$$D_c = \frac{2}{(P(P-1))} \sum_{p_1=1}^{P-1} \sum_{p_2=p_1+1}^{P} \frac{|F_{p_1} - F_{p_2}|}{F_{max}} \tag{25}$$

Parent #1

Chromosome 1: SFC's genes	
SFC id	1
Used links	{1, 2}
Maximum tolerable delay	10ms
Minimum tolerable reliability	0.98

Chromosome 1: VNF's genes - VNF #6	
Location	PM3
Function type	Type3
SFC id	1
Position in SFC	3
Used links to access assigned backup VNFs	-
List of assigned backup	-
Reliability	0.94

Chromosome 2: backup VNF #2	
Location	PM4
Function type	Type3
List of user VNF	{7, 10}
Reliability	0.96

Parent #2

Chromosome 1: SFC's genes	
SFC id	1
Used links	{1, 4, 5}
Maximum tolerable delay	10ms
Minimum tolerable reliability	0.98

Chromosome 1: VNF's genes - VNF #6	
Location	PM4
Function type	Type3
SFC id	1
Position in SFC	3
Used links to access assigned backup VNFs	{6, 7}
List of assigned backup	{2}
Reliability	0.986

Chromosome 2: backup VNF #2	
Location	PM7
Function type	Type3
List of user VNF	{6, 8}
Reliability	0.92

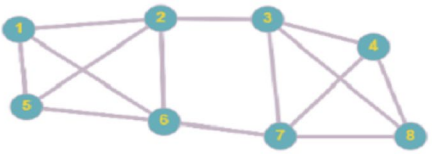
Child

Chromosome 1: SFC's genes	
SFC id	1
Used links	{1, 2}
Maximum tolerable delay	10ms
Minimum tolerable reliability	0.98

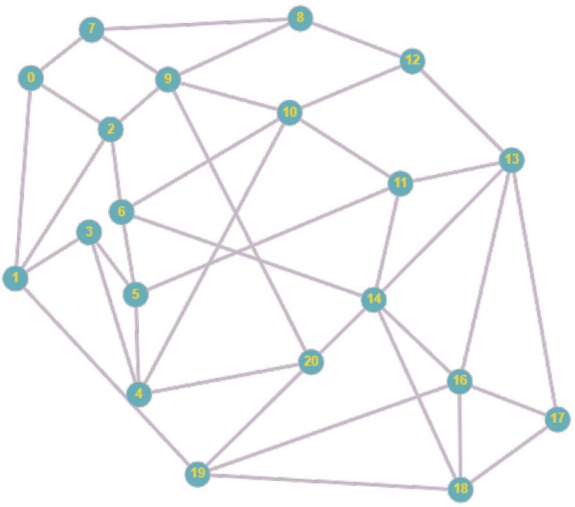
Chromosome 1: VNF's genes - VNF 6	
Location	PM3
Function type	Type3
SFC id	1
Position in SFC	3
Used links to access assigned backup VNFs	{3, 4, 8}
List of assigned backup	{2}
Reliability	0.985

Chromosome 2: backup VNF #2	
Location	PM7
Function type	Type3
List of user VNF	{6, 8, 10}
Reliability	0.92

Fig. 2 Example of crossover



(a) Sample network with 8-nodes (refereed as 8-node network).



(b) Sample network with 20-nodes (refereed as 20-node network).

Fig. 3 Network topologies

where $|F_{p_1} - F_{p_2}|$ is the absolute difference of the fitness of individual p_1 and p_2; and F_{max} is the maximum fitness value in the generation. For 5 generations or more, if D_c value is less than a given, we consider that the algorithm is converged.

5.5 Computational Complexity

In the following the computational complexity of the genetic algorithm is discussed. The computational complexity of executing the first line of Algorithm 1 is $\alpha \times n$ where α is the size of the population and n is the number of variables in each solution. The loop presented in line 3 is in order of o(threshold). Line 6-11 and 12-18 are similar and have similar computational complexity which is $\alpha \times NC \times n$. NC is equal to $\alpha(1-\beta)$ where β is the elitism ratio. Since β is has a negligible value, we consider NC is equal to α. Finally, the complexity order of lines 20-22 is $\alpha \times n \times p$. Where p is a small value (usually below 5). Therefore, the total execution time is in order of $O(\alpha^2 \times n)$ where α is the size of the population and n is $|F| + |F'|$.

We should also analyse the computational complexity of MILP formulation. It is stated in [27] that the complexity of MILP algorithms grows with increasing problem size and presents a table that contains the relationship between the number of variables and the computational complexity. According to the table, since the DP variables are 100 to 10,000, its computational complexity is $O(n^2)$ and also the SP-MILP variables are more than 10,000, its computational complexity is $O(n^4)$. Where variables size is $\sum_{i=1}^{|S|} |F_i| \times N + |F'| \times N + \sum_{i=1}^{|S|} |F_i| \times |F'| + \sum_{i=1}^{|S|} |F_i| \times |L| + |S| \times |F'| \times |L|$.

6 Performance Evaluation

In this section, we evaluate the performance of the proposed MILP model and the heuristic algorithms. IBM CPLEX is used to solve the mathematical formulation of problem and Python to implement the heuristic algorithms. All of the simulations are done using a machine with 2.30 GHz Intel Xeon CPU and 16 GB RAM. Two substrate networks are considered (Fig. 2):

- An 8-node and 14-link NSF network and hosting 4 network services (Fig. 3a).
- A 20-node and 40-link NSF network and hosting 5 to 30 network services. (Fig. 3b).

We assume there are four types of function in the network, and each physical node can execute up to 4 VNFs. The reliability of nodes are specified randomly using uniform distribution between 0.90 and 0.96. It is assumed that each SFC requires 3 VNFs and a minimum reliability of 0.98. This requirement is not hold in random placement. The link delay and bandwidth are respectively fixed to 10ms

Table 5 Routing Results (8-Nodes Network)

Algorithm	NS	Routing and VNFs assignment (VM (VNFs))	Reliability	Bandwidth Utilization	CPU time (s)
MILP	1	$5 \rightarrow \{6,2,1\}(f_3) \rightarrow \{5,4,7\}(f_2) \rightarrow \{8,7,2\}(f_1)$	0.994	49.28%	238.922
	2	$1 \rightarrow \{8,4,7\}(f_2) \rightarrow \{3,7,2\}(f_1) \rightarrow \{5,2,1\}(f_3) \rightarrow 3$	0.994		
	3	$1 \rightarrow \{3,4,7\}(f_2) \rightarrow \{8,2,1\}(f_3) \rightarrow \{6,7,2\}(f_1) \rightarrow 8$	0.995		
	4	$3 \rightarrow \{5,7,2\}(f_1) \rightarrow \{3,2,1\}(f_3) \rightarrow \{6,4,7\}(f_2) \rightarrow 5$	0.994		
Genetic	1	$8 \rightarrow 7 \rightarrow \{6,1,5\}(f_1), \{6,3\}(f_2) \rightarrow 7, \{4,6\}(f_3) \rightarrow 7 \rightarrow 6 \rightarrow 5$	0.998	66.71%	0.639
	2	$1 \rightarrow \{5,3,1\}(f_1) \rightarrow 2 \rightarrow \{3,6\}(f_2) \rightarrow \{7,5\}(f_3) \rightarrow 3$	0.998		
	3	$1,3(f_2) \rightarrow \{5,3\}(f_3), \{5,6,3\}(f_1) \rightarrow 6 \rightarrow 7 \rightarrow 8$	0.999		
	4	$3 \rightarrow \{8\}(f_1) \rightarrow 3 \rightarrow 2 \rightarrow \{1\}(f_2), \{1\}(f_3) \rightarrow 5$	0.999		
NO protection	1	$8 \rightarrow 3 \rightarrow 2(f_3) \rightarrow 1(f_2) \rightarrow 5(f_1)$	0.796	10.00%	0.031
	2	$1(f_2) \rightarrow 5(f_1) \rightarrow 2(f_3) \rightarrow 3$	0.810		
	3	$1(f_2) \rightarrow 2(f_3) \rightarrow 5(f_1) \rightarrow 6$	0.797		
	4	$3 \rightarrow 2(f_1) \rightarrow 1(f_3) \rightarrow 5(f_2)$	0.795		
Random	1	$8 \rightarrow 3 \rightarrow 2 \rightarrow 5(f_1), 5(f_2) \rightarrow 2 \rightarrow 3 \rightarrow 8(f_3) \rightarrow 3 \rightarrow 2 \rightarrow 5$	0.809	63.60%	0.023
	2	$1 \rightarrow 6 \rightarrow 7 \rightarrow \{4,7\}(f_1), \{4,8\}(f_2) \rightarrow \{3,2\}(f_3)$	0.998		
	3	$1 \rightarrow \{5,3\}(f_1)5 \rightarrow 1(f_2) \rightarrow \{2,3,8\}(f_3) \rightarrow 3 \rightarrow 8$	0.908		
	4	$3 \rightarrow 7 \rightarrow \{6,3\}(f_1) \rightarrow 5(f_2), 5(f_3)$	0.868		
Dedicate protection	1	$8 \rightarrow \{3,2\}(3) \rightarrow 2 \rightarrow \{7,4\}(2) \rightarrow \{5,6\}(1)$	0.982	62.85%	0.023
	2	$1 \rightarrow \{7,4\}(2) \rightarrow \{3,6\}(1) \rightarrow \{1,2\}(3) \rightarrow 2 \rightarrow 3$	0.986		
	3	$1 \rightarrow \{3,8\}(2) \rightarrow \{7,2\}(3) \rightarrow 2 \rightarrow \{1,6\}(1) \rightarrow 8$	0.986		
	4	$3 \rightarrow \{5,6\}(1) \rightarrow \{7,2\}(3) \rightarrow \{3,4\}(2) \rightarrow 5$	0.982		

and 20 (units). We empirically found that $\omega = 0.9$ jointly minimizes computational and bandwidths consumption across multiple datasets. We consider four different demand scenarios to evaluate the proposed solution:

1. $f_3 \rightarrow f_2 \rightarrow f_1(\sigma_1 = 8, \delta_1 = 5, b_1 = 2, \phi_1 = 50)$
2. $f_2 \rightarrow f_1 \rightarrow f_3(\sigma_2 = 1, \delta_2 = 3, b_2 = 4, \phi_2 = 50)$
3. $f_2 \rightarrow f_3 \rightarrow f_1(\sigma_3 = 1, \delta_3 = 8, b_3 = 2, \phi_3 = 60)$
4. $f_1 \rightarrow f_3 \rightarrow f_2(\sigma_4 = 3, \delta_4 = 5, b_4 = 4, \phi_4 = 60)$

For the purpose of performance comparison and bench-marking, three additional schemes are implemented. They are:

1. A dedicated protection (DP): as for DP, we exploit state-of-the-art solution proposed in [23] for resource allocation problem. The mentioned solution exploits dedicated protection to guarantee the end-to-end reliability of services.
2. A none protection (NP): this algorithm is reliability unaware, i.e., no any backup VNF should be defined through this solution. We removed the reliability constraints from our solution and use it as NP.
3. A Random placement (RP): this random placement algorithm neither satisfies the reliability requirements nor the end-to-end delay constraints. This makes the algorithm to have the optimal response time when deploying SFCs. RP algorithm only focuses on the routing constraints while randomly doing (both primary and backup) VNFs placement.

In the first step, all of the above-mentioned algorithms are tested on 8-nodes network and the results are shown in Table 5. According to the table, NP achieves the smallest bandwidth utilization 10.0% while RP has the smallest execution time. This happens because the main objective of NP is to minimize the bandwidth consumption

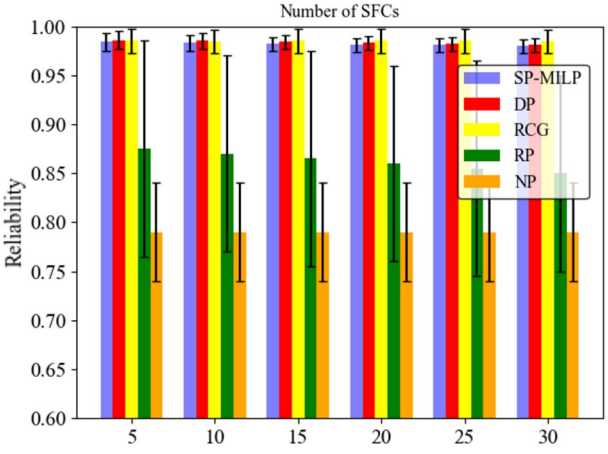


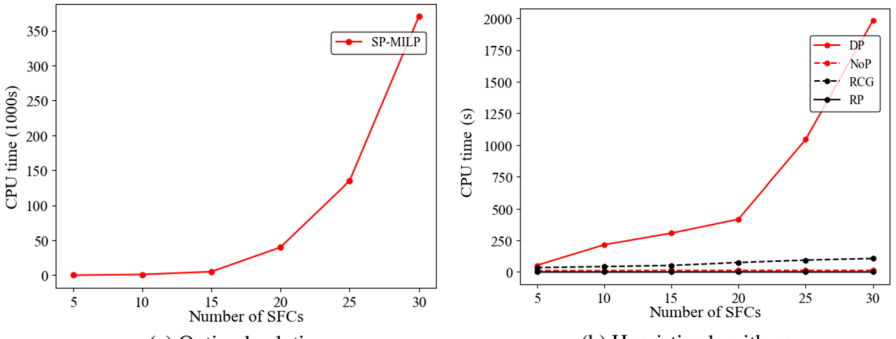
Fig. 4 Reliability versus the number of requested services

without considering any backup VNFs. Similarly, the main objective of RP is to simplify the management process. It should be mentioned that these two algorithms fail to satisfy the reliability constraints. On the other hand, SP-MILP, RCG, and DP satisfy the reliability constraints of the network services at the cost of increasing the bandwidth utilization. Due to this reason, SP-MILP achieves minimum bandwidths utilization 49.28% along with maximum execution time of 104 and 238s, respectively. Conversely, reliability of DP is the highest among all. The bandwidth utilization achieved by RCG are better than DP solution. As it is expected, the execution time of RCG is significantly lower than SP-MILP. In the next step, we test the algorithms on a 20-nodes network which is hosting 5 to 30 network services. We compare the performance of the algorithms through five metrics: I. Reliability, II. Execution time (CPU time), III. Computational resource consumption (CPU utilization), IV. Link utilization, and V. Computational complexity (order of complexity).

6.1 Reliability

Reliability is the ability of the network (including routing and processing devices) to consistently perform its intended or required function, on demand and without degradation or failure. It is critical in many case to reduce the probability of failure occurrence that could cause the entire service presence to come crashing down. In this part, we compare the reliability of the proposed resource allocation algorithms with state-of-the-art algorithms. The first group of carried out tests aims to evaluate and compare the achieved reliability of above-mentioned algorithms. In our emulation, we consider the minimum acceptable reliability to be 0.98 for each SFCs, however, RP and NP cannot manage to achieve this reliability. Based on simulation results, the achieved reliability of proposed algorithms are reported in Fig. 4.

In each scenario, we increase the number of available SFCs in the network from 5 SFCs up to 30 SFCs by adding 5 new service function chains in each iteration and calculating the average SFCs reliability. As can be seen, achieved reliability for NP and RP in all scenarios are lower than SP-MILP, RCG, and DP. This happens



(a) Optimal solution. (b) Heuristic algorithms.

Fig. 5 CPU time versus number of network services (20-nodes network)

because these algorithms do not consider the service reliability as metric so they cannot meet the reliability constraints. Considering the error bar in Fig. 4, NP and RP algorithms have a very high range of results in term of reliability. In contrast, the other algorithms (DP, SP-MILP, and RCG) not only satisfy the reliability constraint but also they achieve a system reliability higher than 0.98 in most cases. It should be mentioned that DP, RCG, and SP-MILP try to find the minimum reliability higher than acceptable reliability to reduce the total waste of resources. Therefore, the lower error band (distance from desirable reliability) is more intended. Based on our simulation results, SP-MILP has the most stable outcomes around the desirable reliability. This is due to the fact that SP-MILP finds the optimal solution for a system with a reliability higher than a pre-defined threshold but with lowest computational resource consumption.

6.2 Execution Time (CPU Time)

In a general sense, high-performance algorithm means getting the most out of the resources. This translates to utilizing the CPU as much as possible. Consequently, CPU utilization becomes a very important metric to determine how well an algorithm is using the computational resources. Talking about a predefined goal, high-performance algorithm uses less resource to achieve the goal in compare to less productive ones. In this way, CPU time (or execution time) is defined as the time spent by the system executing each algorithm, including the time spent executing run-time or system services on its behalf. In Fig. 4, we evaluate the execution time of preferred algorithms for different service chain request, i.e. execution time versus the number of requested services (VNFs) is depicted. It is worth mentioning that RP and NP are not reliability-aware while RCG, DP, and SP-MILP are. Since the execution time of SP-MILP is dramatically higher than other methods, we put it in

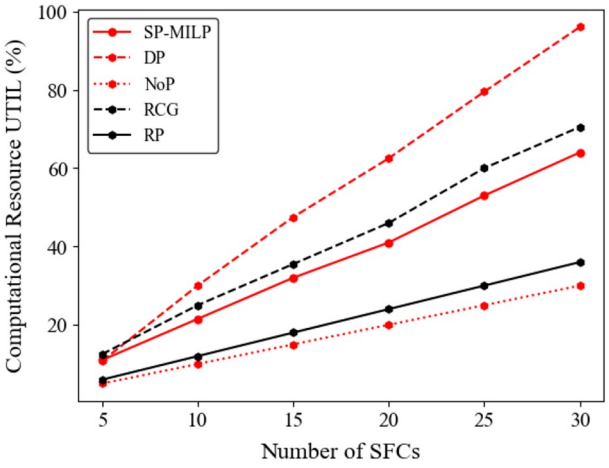


Fig. 6 CPU utilization (%) versus the number of requested services

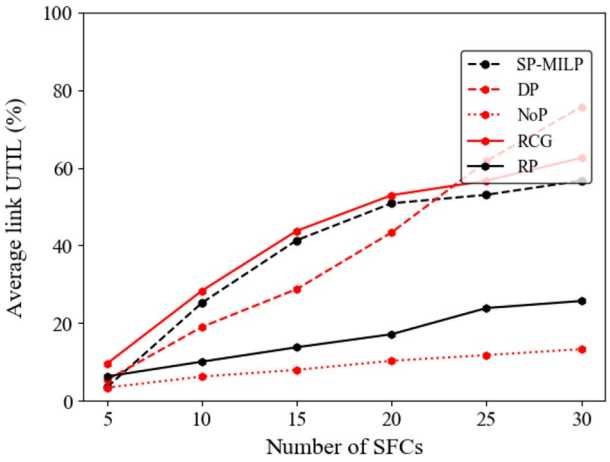


Fig. 7 Bandwidth utilization (%) versus number of network services (20-nodes network)

a separate plot to keep the plots clear and simple to read. In this way, the execution time of SP-MILP over 20-nodes network is reported in Fig. 5a while the other algorithms are measured in Fig. 5b. According to Fig. 5a, SP-MILP is too complex for even small-scale networks, therefore, it is not applicable for real-world scenarios. However, the reliability of the solution provided by SP-MILP is the optimal one. On the other hand, RP and NP methods have a very low CPU time which makes them applicable for real-world networks. However, both methods do not satisfy some of the constraints of the problem, and this allows them to respond quickly as compared to other methods.

Considering reliability-aware solutions, DP has a medium-high execution time meaning dramatically lower than SP-MILP but sufficiently higher than NP, RP, and RCG. Consequently, we can conclude that although DP considers the reliability in allocation of resources, due to its high execution time it is not practical for medium and large scale networks. Comparing the proposed genetic algorithm with DP, NP, RP, and MILP, not only RCG is reliability aware but also it is practical and could be used for real-world scenarios.

6.3 Computational Resource Consumption (CPU Utilization)

Although CPU time (execution time) is a very important metric to measure the performance of algorithms, it is not a comprehensive metric. In this sub-section, we evaluate the computational resource consumption of each preferred method. This includes the total resources required for the primary functions and the backup functions for each of the mentioned resource allocation algorithm. To this end, Fig. 6, shows the CPU utilization versus the number of requested services (VNFs).

Similar to previous sub-section, RP and NP have the lowest CPU utilization, however, they are reliability unaware. This is mean that they reduce consumption of the computational resource at the cost of an intense reduction in the system reliability.

Among reliability aware solutions, SP-MILP has the lowest CPU utilization with the sacrifice of CPU time. This means that although the CPU utilization is lower than DP and RCG, it is not practical due to its high execution time. Comparing RCG and SP-MILP, it is clear that the meta-heuristic method closely follows the optimal solution obtained via the mathematical optimization model. Comparing DP and RCG, dedicated protection scheme requires more computational resources than shared resource consumption. It also needs a higher CPU time which clearly explains why RCG is superior to DP in terms of both CPU time and utilization.

6.4 Link Utilization Results

Bandwidth utilization is one of the most basic and critical statistics available in assessing a network resource allocator. It shows the average traffic levels on links compared to total capacity of those lins. Fig. 7 shows the comparison of the average link utilization between the RCG and other algorithms. We can comprehend how RP introduces a lot of overload links. According to this plot, maximum link utilization of NP method is much lower than other methods. Sharing protection methods involving the SP-MILP and the genetic algorithm initially consume more bandwidth than DP method, but with increasing number of SFC, bandwidth consumption of the DP method will increase in sharing protection methods. Another issue that can be inferred from Fig. 7 is that bandwidth consumption in the RCG is very close to the SP-MILP. Initially, due to the small number of primary functions, the intensity of sharing the backup functions is low as result, dedicated methods less bandwidth consuming, But with the increase in the number of primary functions and the upgrading of the intensity of sharing the backup functions and decrease the need for more backup functions, the required bandwidth of shared methods reduced.

7 Conclusion

In order to provide reliable service function chains, a large number of backup functions are required. Although this redundancy is essential, it may sufficiently reduce the network resource efficiency if resources are not well assigned. To solve this problem, we exploited a Shared Protection (SP) algorithm with Active-Standby redundancy as an optimization issue to achieve the optimal network in the virtualization environment. We first formulated the problem as a mixed-integer linear programming (MILP) and found the optimal solution of the problem then we compared the SP-MILP method with three other methods: Dedicated Protection (DP), No Protection (NP) and Random Placement (RP). SP-MILP has a very high time complexity compared to the other approaches. But in terms of Computational resource consumption, it is about 33% lower than DP. Also, bandwidth consumption in the case of a high number of services is 25.2% lower than DP. To solve the complexity issue, we proposed a genetic algorithm. Based on simulation results, the proposed genetic algorithm with time complexity yields an optimize gap of approximately 9% bandwidth consumption and 9.3% optimize gap

in computational resource consumption, compared to the SP-MILP response. In this paper, we proposed a reliability enhancement method using a shared protection scheme to reduce the cost of redundant VNFs and implemented it as a mathematical model and a genetic algorithm. In this method, we focused on the different reliability of the computational nodes and the calculation of the reliability of the function based on their location. In future work, more features such as link reliability, network function application and etc can be considered. Also heuristic can be used rather than Genetic algorithm.

References

1. Zhang, B., Zhang, P., Zhao, Y., Wang, Y., Luo, X., Jin, Y.: Co-scaler: Cooperative scaling of software-defined nfv service function chain, in: Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE Conference on, IEEE, (2016), pp. 33–38
2. Casazza, M., Bouet, M., Secci, S.: Availability-driven nfv orchestration. Computer Networks **155**, 47–61 (2019)
3. Kang, J., Simeone, O., Kang, J.: On the trade-off between computational load and reliability for network function virtualization. IEEE Commun. Lett. **21**(8), 1767–1770 (2017)
4. Tajiki, M.M., Akbari, B., Mokari, N.: Qrtp: Qos-aware resource reallocation based on traffic prediction in software defined cloud networks, In: 8th International Symposium on Telecommunications (IST). IEEE **2016**, 527–532 (2016)
5. Nguyen, T.-M., Minoux, M., Fdida, S.: Optimizing resource utilization in nfv dynamic systems: New exact and heuristic approaches. Comput. Netw. **148**, 129–141 (2019)
6. Herrera, J.G., Botero, J.F.: Resource allocation in nfv: a comprehensive survey. IEEE Transact. Netw. Serv. Manag. **13**(3), 518–532 (2016)
7. Cziva, R., Pezaros, D.P.: Container network functions: bringing nfv to the network edge. IEEE Commun. Magaz. **55**(6), 24–31 (2017)
8. Tajiki, M.M., Shojafar, M., Akbari, B., Salsano, S., Conti, M., Singhal, M.: Joint failure recovery, fault prevention, and energy-efficient resource management for real-time sfc in fog-supported sdn. Comput. Netw. **162**, 106850 (2019)
9. Vilchez, J.M.S., Yahia, I.G.B., Crespi, N.: Self-healing mechanisms for software defined networks. In: 8th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2014), 2014, pp. 23–35
10. da Rocha Fonseca, P.C., Mota, E.S.: A survey on fault management in software-defined networks. IEEE Commun. Surv. Tutorials **19**(4), 2284–2321 (2017)
11. Sterbenz, J.P., Hutchison, D., Çetinkaya, E.K., Jabbar, A., Rohrer, J.P., Schöller, M., Smith, P.: Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. Comput. Netw. **54**(8), 1245–1265 (2010)
12. Tajiki, M.M., Shojafar, M., Akbari, B., Salsano, S., Conti, M.: Software defined service function chaining with failure consideration for fog computing. Concurr. Comput: Practice Exp. **31**(8), e4953 (2019)
13. Van Adrichem, N.L., Van Asten, B.J., Kuipers, F.A.: Fast recovery in software-defined networks. In: Software Defined Networks (EWSDN), 2014 Third European Workshop on, IEEE, (2014), pp. 61–66
14. Tajiki, M.M., Salsano, S., Shojafar, M., Chiaraviglio, L., Akbari, B.: Energy-efficient path allocation heuristic for service function chaining, in: 2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), IEEE, 2018, pp. 1–8
15. Virtualisation, N.F.: Reliability; report on models and features for end-to-end reliability. ETSI Standard GS NFV-REL **3**, V1 (2016)
16. Karimzadeh-Farshbafan, M., Shah-Mansouri, V., Niyato, D.: A dynamic reliability-aware service placement for network function virtualization (nfv). IEEE J. Select. Areas Commun. **38**(2), 318–333 (2020)
17. Kanizo, Y., Rottenstreich, O., Segall, I., Yallouz, J.: Optimizing virtual backup allocation for middleboxes. IEEE/ACM Transact. Netw. **25**(5), 2759–2772 (2017)
18. Kanizo, Y., Rottenstreich, O., Segall, I., Yallouz, J.: Designing optimal middlebox recovery schemes with performance guarantees. IEEE J. Select. Areas Commun. **36**(10), 2373–2383 (2018)

19. Fan, J., Ye, Z., Guan, C., Gao, X., Ren, K., Qiao, C.: Grep: Guaranteeing reliability with enhanced protection in nfv. In: Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization, ACM, pp. 13–18. (2015)
20. Ye, Z., Cao, X., Wang, J., Yu, H., Qiao, C.: Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization. IEEE Netw. **30**(3), 81–87 (2016)
21. Carpio, F., Dhahri, S., Jukan, A.: Vnf placement with replication for loac balancing in nfv networks. In: 2017 IEEE International Conference on Communications (ICC), IEEE, (2017), pp. 1–6
22. Pham, T.-M., Fdida, S., Binh, H.T.T., et al.: Online load balancing for network functions virtualization. In: 2017 IEEE International Conference on Communications (ICC), IEEE, (2017), pp. 1–6
23. Qu, L., Assi, C., Shaban, K., Khabbaz, M.J.: A reliability-aware network service chain provisioning with delay guarantees in nfv-enabled enterprise datacenter networks. IEEE Transact. Netw. Serv. Manag. **14**(3), 554–568 (2017)
24. Han, B., Gopalakrishnan, V., Ji, L., Lee, S.: Network function virtualization: Challenges and opportunities for innovations. IEEE Commun. Magaz. **53**(2), 90–97 (2015)
25. Zhang, H., Zheng, X., Li, Y., Zhang, H.: Availability analysis of shared backup path protection subject to srlg constraints in wdm mesh networks. In: 36th European conference and exhibition on optical communication, IEEE, (2010), pp. 1–3
26. Gao, X., Zhong, W., Ye, Z., Zhao, Y., Fan, J., Cao, X., Yu, H., Qiao, C.: Virtual network mapping for reliable multicast services with max-min fairness, in, : IEEE Global Communications Conference (GLOBECOM). IEEE **2015**, 1–6 (2015)
27. Till, J., Engell, S., Panek, S., Stursberg, O.: Empirical complexity analysis of a milp-approach for optimization of hybrid systems. IFAC Proceed. Vol. **36**(6), 129–134 (2003)

Abolfazl Ghazizadeh received the bachelor's degree in Computer engineering (Hardware) from Hakim Sabzevari University. He earned a master's degree at Tarbiat Modares University with a focus on optimization in Network function virtualization. His previous research mainly dealt with Computer Networks, Network QOS,Network optimization ,service function chaining, software-defined networking (SDN)

Behzad Akbari received the B.S., M.S., and PhD degree in computer engineering from the Sharif University of Technology, Tehran, Iran, in 1999, 2002, and 2008 respectively. His research interest includes Computer Networks, Multimedia Networking Overlay and Peer-to-Peer Networking, Peer-to-Peer Video Streaming, Network QOS, Network Performance Analysis, Network Security, Network Security Events Analysis and Correlation, Network Management, Cloud Computing and Networking, Software Defined Networks.

Mohammad M. Tajiki is a research associate at Queen Mary University of London, UK, holding two PhD degrees one from University of Rome Tor Vergata in Electrical Engineering and another one from Tarbiat Modares University in Computer Engineering. His main research interests are Network Monitoring, Network Function Virtualization, Network QoS, data centre networking, traffic engineering, service function chaining, IPv6 segment routing, and software-defined networking (SDN).

Authors and Affiliations

Abolfazl Ghazizadeh[2] · Behzad Akbari[2] · Mohammad M. Tajiki[1]

Behzad Akbari
b.akbari@modares.ac.ir

Mohammad M. Tajiki
m.tajiki@qmul.ac.uk

[1] School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

[2] Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran