Autonomous optimal composition and embedding of Service Function Chains using a hierarchy of control logics that are evolved by knowledge-based Genetic Algorithms

Service Function Chains (SFCs) consist of network functions, such as a firewall, that are virtually deployed on commercial servers and linked together using Software Defined Network. SFCs, while driving down capital and operational expenditure, make autonomous networks a possibility by making networks programmable.

However, this introduces new challenges such as the optimal ordering of network functions in an SFC (chain composition) and the optimal deployment of the network functions and the links between them on the underlying physical network (chain embedding).

Existing research offers solutions assuming the environment to be static. However, a real environment is dynamic as the amount of ingress traffic may vary, network links may fail, servers may crash and new SFCs may have to be deployed and existing ones removed. Consequently, a practical solution has to exhibit autonomous behaviours such as self-healing and self-optimisation to adapt to a dynamic environment.

My hypothesis to solve this problem is to use online evolution using a hierarchy of control logics. Online evolution consists of using an evolutionary algorithm like a Genetic Algorithm to evolve optimal solutions by evaluating them via experimentation. In this case, this involves randomly generating chain composition and embedding schemes, implementing them on a network, evaluating their performance, and iteratively improving their performance using a Genetic Algorithm. The Genetic Algorithm will be knowledge-based as it will use the knowledge gained during the evolution to improve its performance.

A hierarchy of control logics is used here so that low-level decision-making is carried out by the control logics at the lower levels of the hierarchy while high-level decisions are taken by the top-level control logics. For example, there can be a control logic each for chain composition and embedding, and these logics can be evolved by a high-level logic to meet an objective such as keeping the traffic latency below an agreed threshold. At the same time, there can be another control logic to optimise for a different objective such as maximising the number of SFCs that can be deployed. A higher-level control logic may try to maximise revenue by trying to find the optimal balance between minimising SFC latency (quality) and maximising SFC deployment (quantity).

I have developed an emulator to evaluate SFCs and two neural networks that act as control logics that optimally compose and embed SFCs. These neural networks are evolved using a Genetic Algorithm that tries to minimise traffic latency and maximise SFC deployment. However, optimising solutions by evaluating them on the emulator consumes a lot of time. To mitigate this issue, I developed a surrogate. A surrogate learns from past evaluations and their outcomes to make predictions on the performance of new solutions. This way I have managed to decrease the time needed to produce a solution. As the next step, I am planning to make it work in dynamic environments.

I am at an early stage in my research and I am looking for feedback on my research idea and the work I have already done.