添加另一片文章中的冗余？保证可靠性的基础上xxxx

在reward里面添加提前执行了多少+节约了多少贷款，多少资源xxxxx

section tips：

网络功能虚拟化（NFV）[1]已被提议作为电信业传统服务供应的替代方案。NFV将软件与硬件解耦，从而使它们能够独立发展。此外，这给网络带来了进一步的灵活性，因为它允许虚拟化的网络功能（VNF）及其互连，这些功能以图的形式表示，即VNF转发图（VNF-FG）[2]，可以按需部署在网络拓扑结构的任何地方。因此，NFV允许网络运营商使用与实际流量相匹配的资源对网络进行细粒度的控制[2]。

2.1. NFV架构框架概述

网络功能虚拟化（NFV）在2012年由领先的电信服务提供商（TSP）撰写的白皮书发表后，受到了极大的关注[1]。不久之后，欧洲电信标准协会（ETSI）被指定为主持NFV行业规范小组（ETSI ISG NFV）的实体，负责制定NFV标准。ETSI规定的NFV架构由两个主要部分组成[2]，[19]。(i)虚拟化网络功能（VNFs）；(ii)管理和协调（MANO）。

NFV的核心是一套已部署的VNFs，它是属于网络基础设施的功能块，具有明确定义的行为和外部接口。VNF是网络功能（NF）的虚拟化，它可以是，例如，动态主机配置协议（DHCP）服务器或防火墙。一个NF的功能行为通常与它是否被虚拟化无关。

NF被组合成代表端到端服务或网络服务的更高层次的功能。根据ETSI，"网络服务（NS）是一个由网络基础设施支持的NF互连的转发图"[2]。NS有通过虚拟链接连接的节点，除了NF之外，还可以包括嵌套的NF转发图。我们在图1中说明了一个端到端的NS，由VNF提供，它包括一个嵌套的转发图。一个VNF转发图（VNF-FG）是由虚拟化的NF组成的转发图。此外，VNF-FG和服务功能链（SFC）是构成端到端服务的服务功能的有序集合，在NFV文献中可以互换使用。这个术语是从软件定义的网络（SDN）的背景中借用的。最近，NFV和SDN被指出是网络切片的基础[20]，这是一种垂直化技术，允许多个切片（端到端逻辑网络）在共享物理网络上运行。

图1. 转发图和支持的基础设施[2]。在这个例子中，一个网络服务由两个VNF-FG提供，其中VNF-FG-2被嵌套到VNF-FG-1中。椭圆表示NFVI-PoP，粗线表示物理链路，虚线表示虚拟链路，VNF和NFVI-PoP之间的映射通过垂直虚线箭头来描述。

2.2. NFV的协调

尽管NFV中的功能行为是由VNFs提供的，但MANO组件是至关重要的，因为它对它们进行协调，以便形成NS，实现网络运营商的目标。VNF的协调基本上涉及三个互补的决定，这导致了三个需要解决的问题，即VNF链组成（VNF-CC）、VNF转发图嵌入（VNF-FGE）和VNF调度（VNF-SCH）。第一个，VNF-CC，或者说，选择问题[13], [21], [22]，包括选择一组VNF来共同实现一个特定运营商的目标。作为结果，我们得到一个VNF-FG。

VNF-FG是实现运营商目标的概念性解决方案，它必须被嵌入到物理网络中。这需要决定在哪里放置VNF，以及如何在物理上连锁VNF，这是与VNF-FGE相关的两个决定。因此，VNF-FGE解决方案由VNF与NFVI-PoP的映射（放置解决方案）和物理网络中连接每对VNF的适当路径（连锁解决方案）组成。VNF-FG解决方案必须保证NFVI-PoP和链路的可用资源量能够满足将要实例化的VNF-FG及其VNF的要求。图2显示了一个例子，其中两个VNF-FG被嵌入到一个NFV启用的基础设施中。

VNF-FGE问题类似于虚拟网络嵌入（VNE）问题，它是网络虚拟化领域的主要资源分配挑战[23]。在VNE中，虚拟路由器被映射到物理节点，而连接虚拟路由器的虚拟链接被映射到物理链接[24]。尽管VNE和VNF-FGE问题的目的都是寻找虚拟化组件的有效分配，但它们之间有两个显著的区别[11]，[22]。首先，VNF-FG中的VNF可能有不同的功能，即它们可以是不同的NF，而VNE中的节点则执行同等的功能，即路由。第二，VNF-FG中的VNF有特定的排序。在寻找适当的解决方案时，支持VNF之间具体排序的要求增加了额外的困难。

图2. 虚拟网络功能转发图嵌入实例。在左侧，有两个VNF-FG，它们有VNF和虚拟链接的要求，需要由VNF-FGE解决方案来满足。每个几何形状代表一个不同的VNF。物理网络及其可用资源位于右侧。VNF-FGE解决方案由位于物理节点（NFVI-PoPs）旁边的几何形状来说明，表明每个VNF的部署位置，同时还有虚线和虚线链接，突出了虚拟链接如何在物理链接中实例化。

最后，NFV协调器要解决的第三个问题是VNF-SCH问题[25]。这个问题的目标是在多个VNF-FG之间共享VNF。它建立在VNF并非一直被使用的想法上，因此可以在空闲时间段处理其他VNF-FG的数据包。最终，将任务分配给空闲的VNF，减少了部署的VNF数量。

这三个引入的问题，即VNF-CC、VNF-FGE和VNF-SCH，被统称为NFV资源分配（NFV-RA）问题[5]。所有这些问题都必须由处理VNFs的协调和生命周期管理的MANO组件来解决。在这项工作中，我们将注意力集中在VNF-FGE问题上，系统地调查和分析了现有的解决方案。

4.1. 分配任务

放置和连锁问题都是NP-hard问题[100]。然而，绝大多数回顾的工作都集中在解决安置和连锁问题，即VNF-FGE问题。

4.2. 目标

由于VNF-FGE是一个分配问题，解决方案通常是为了实现优化目标，它可以专注于一个或多个方面。例如，在应用关于服务请求时间的严格SLA的情况下，解决方案可能会分配VNF及其虚拟连接，以使用具有最低延迟的物理链接和具有最低处理延迟的VNF。我们将所审查的作品的分配目标归为四个选项之一。(i) 减少网络资源；(ii) 减少主机资源；(iii) 权衡网络和主机资源；以及(iv) 减少经济成本。

需要注意的是，这四个目标是相关的。例如，一个旨在改善网络和主机资源的算法，其输出的解决方案具有较低的网络和主机使用率，这就意味着经济成本较低。然而，要将一个目标归类为经济目标，它必须明确地考虑经济因素。有一些作品将网络成本或嵌入成本正式化，但没有将其转化为货币成本[54]。此外，还有一些研究工作引入了计算和网络成本等概念，这些成本被定义为，例如，CPU核的数量和带宽的数量，或者自由CPU和总CPU之间的比率以及链接的自由带宽和总带宽之间的比率，并优化其解决方案以最小化这些成本[27], [88], [99]。在这种情况下，分配目标不被认为是为了降低经济成本。只有指定定价政策[60]、引入收入和支付形式化[92]、[93]或使用货币价值作为成本单位[59]、[61]、[67]的方法，以及为降低成本或增加收入而进行的优化被归类为降低经济成本。最后，我们指出，这些目标并不详尽，因此未来的解决方案可能会使用本分类法中没有考虑到的创新和目前未预见的目标。

4.3. 因素

当设计一个算法来解决VNF-FGE问题时，我们必须选择哪些特征将被建模为问题的一部分，并因此在拟议的方法中使用。可以说，选择常见的因素，如处理能力（通常被形式化为CPU核的数量）和链路能力（通常被建模为链路的带宽），就足以开发出VNF-FGE的解决方案[50], [79]。解决方案可能包括VNF的处理延迟、VNF和NFVI-PoP的存储和内存（或者仅仅是NFVI-PoP支持的VNF数量）以及链路延迟和物理/虚拟链路的跳数等因素，以提供更适用和现实的解决方案。此外，从物理和虚拟网络中继承的因素往往与使用成本相平衡。例如，在公共NFVI-PoP中分配CPU的货币成本[67]或使用特定物理链接的货币成本[61]。

表6和图5对审查工作中发现的最常见因素进行了分组。经审查的文献所考虑的最小和最大的因素量分别为两个[9]、[51]、[79]和十个[60]。最常见的因素量（模式）是三个。我们把有兴趣的读者推荐到其他地方1，以了解经审查的文章中单独使用的因素的详细情况。

4.4. 技术风格

执行分配任务的算法会考虑不同的因素，旨在使用精确、启发式或元启发式技术来实现目标。鉴于这些分配任务的NP-hardness，VNF-FGE及其子问题的精确算法通常在非多项式时间内找到最佳解决方案。其中一个精确算法为树形拓扑结构提供了多项式时间解决方案[77]。

启发式算法和元启发式算法以优化换取更快的处理时间。绝大多数审查的工作将VNF-FGE问题建模为整数线性编程（ILP）[7], [48]或混合整数线性编程（MILP）[10]。然后，他们提出一个启发式或元启发式来有效地解决它，同时使用优化模型的精确解进行比较[8], [11], [60], [77]。在这种情况下，我们只考虑启发式或元启发式对文章分类的贡献。从表5中可以看出，高效的启发式和元启发式解决方案的数量超过了低效的精确解决方案。

在下一节，我们将讨论已经提出的解决VNF-FGE问题的各种方法。这些方法根据我们第一个引入的面（行为）进行分组，并在考虑其余面的情况下强调每个方法的关键方面。

5. 最先进的VNF-FGE

我们现在回顾一下选定的文献。我们的讨论分为离线和在线方法，因为它们在性质上有所不同。因此，它们不能互换使用，即使它们专注于相同的分配任务，因为离线方法不考虑先前部署在网络中的VNF-FG。首先，我们讨论离线方法（第5.1节），然后重点讨论那些在线方法（第5.2节）。包括离线和在线算法的建议（例如[81]）按其在线贡献分类。

在这两种情况下，大量解决VNF-FGE问题的现有文献都是基于线性编程的[6], [41]。通常，他们使用网络配置和约束条件以及选定的目标函数来指定要解决的优化问题。根据选定的问题变量，这项工作可能依赖于，例如ILP或MILP。可用的工具，如CPLEX2或YALMIP[101]，然后被用来解决指定的优化问题，给出精确（或最优）的解决方案。由于ILP是NP-Hard，精确解通常被用作提出替代技术的基线，该技术可能使用启发式或元启发式。

5.1. 离线

有27篇文章提出了VNF-FGE问题的离线解决方案。它们按分配任务分组，并在下文中单独描述。

5.1.1. 安置

有五项研究工作涉及安置问题，提出了使用启发式算法给出解决方案的离线算法[6]、[41]、[42]、[43]、[44]。虽然他们关注的是同一个问题，但他们的目的是找到一个实现不同目标的解决方案。Cohen等人[6]的目标是找到一个解决方案，使VNF的连接和设置成本最小。为了实现这一目标，指定了一个线性程序，对每个VNF的存储要求和设置成本以及NFVI-PoPs的可用资源进行建模。这是第一个考虑在多个VNF-FG之间共享实例化VNF的可能性的工作。指定的线性程序的解决是通过两种算法实现的；一种是不考虑NFVI-PoP资源限制的算法（无约束安置），另一种是考虑这种限制的算法（约束安置）。这两种算法都依赖于一种启发式算法，即根据存储量大小（降序）将VNF分配给NFVI-PoPs。

在不同的目标下，有两项工作专注于最小化VNF实例的数量[41], [42]。Chi等人[41]提出了一个替代的线性程序规范，以解决安置问题。他们的规范根据VNF的处理能力进行建模。Chi等人[41]还依靠启发式方法，将放置问题简化为垃圾箱包装问题[102]。其想法是将VNF分配到好仓（相对于坏仓），即NFVI-PoP，其满度被评估为大于某个阈值。仓的满度是由穿越NFVI-PoP的流量数量决定的。他们的工作只限于单一的VNF类型。Sang等人[42]则提出了一种贪婪的启发式方法，在具有未处理请求的NFVI-PoP上进行迭代，并将VNF的组合置于NFVI-PoP中，以处理最大数量的未处理流量。

Tomassilli等人[43]为安置问题创建了两个启发式方法和一个精确解决方案。他们认为所有的需求都是先验的，目的是最大限度地减少与主机有关的资源，如许可费或能源效率。他们引入了 (i) 一种动态编程的贪婪启发式方法，在预先计算的路径中选择成本最低的NFVI-PoP来承载VNF；(ii) 一种近似算法，从ILP公式中随机选择约束条件，以指导近似最佳解决方案；(iii) 一种针对树状拓扑结构的最佳动态编程技术，在所有流量在树中上游或下游的特殊情况下以多项式时间运行。

最后，Sallam等人[44]还开发了两种贪婪的启发式方法，旨在通过选择最佳的NFVI-PoP来承载VNF，使完全处理的流量总量最大化。此外，他们还考虑到了使用NFVI-PoP的货币成本及其处理能力。第一个启发式是为具有统一成本的NFVI-PoP量身定做的。它将每个VNF分配给与给定目标函数的最高值相关的NFVI-PoP。第二，针对异质价格的情况，引入了基于枚举的贪婪启发法[103]。它将满足预算约束的NFVI-PoP按已安装的VNF数量分组，然后将每个VNF部署在该组的NFVI-PoP中，该组呈现出关于目标函数的最大收益。此外，假设链式问题得到解决，提供了VNF-SCH问题的两个近似值。

有三项工作开发了元启发法[7]、[45]、[46]来解决具有减少网络资源分配任务的放置问题。两个元启发式方法[7], [45]旨在减少端到端的服务延迟，同时保证一定的可靠性[7]或可用性[45]水平。Chantre等人[7]制定了一个非线性混合整数程序，其中VNFs由其处理延迟和可靠性来描述。作者开发了一种元启发式，使用粒子群优化（PSO）[104]来寻找冗余VNF的最佳数量和它们各自的NFVI-PoPs。PSO是一种基于共享信息的粒子互动的搜索算法。基于局部和全局信息，它们向全局最优迁移。非线性混合整数编程公式被用作健身函数，指导搜索良好的解决方案。同样，Yala等人[45]开发了一种遗传算法（GA），在边缘云或中心云分配VNFs。前者的延迟低但资源稀缺，而后者的延迟高但计算资源充足。GA算法考虑到了链路延迟和实例化成本，尽管有目标函数，但它仍能找到接近最优的解决方案。Manias等人[46]没有使用LP来模拟网络，而是选择了一种机器学习算法--决策树[105]--来实现VNF的放置，目的是减少VNF之间的延迟。作者认为，这种方法大大减少了协调VNFs所需的计算量，因为学习阶段是在其他地方提出的近乎最优的启发式算法的基础上离线进行的[106]。他们的评估显示，他们的元启发式在分析的一半场景中优于用于训练模型的启发式。

5.1.2. 链式结构

有三个建议假定已经完成了VNF的放置，并集中于链式问题[8], [47], [107]。它们的目标是最大限度地提高流量吞吐量，即在网络中尽可能多地链合VNF-FG，同时保证满足端到端的延迟约束[8]，[47]，或者最小化数据包开销[107]。每个人都提出了一种启发式方法，考虑到链路的容量[8]、[47]、[107]或其延迟[8]、[47]，同时考虑到VNF的吞吐量[8]、[107]或数据包处理率[47]、[107]和VNF处理延迟[47]。

Jiao等人[8]首先将VNF-FG请求链化，而不考虑延时约束。通过平衡部署在网络中的每个能够执行所需工作的VNF的两个值，将VNF链化。(i) VNF的可用资源；以及(ii) 从当前NFVI-PoP到部署VNF的NFVI-PoP的最短路径的容量。然后，如果计算出的路径违反了VNF-FG的端到端延迟约束，则放弃所选的VNF及其NFVI-PoP，并评估另一个候选人。这个过程不断重复，直到找到一个不违反端到端延迟约束的路径。Xu等人[47]使用一种不同的方法，将VNF-FG请求分为子请求，使每个子请求具有与具有最低包速率的请求相同的包速率。这些子请求从它们的源节点链到目的节点，通过具有最低成本并满足端到端延迟约束的NFVI-PoP。这个约束是总的链路延迟加上请求的VNF的处理延迟，该VNF已经部署。最后，Wang等人[107]创建了一个链式启发法，将VNF的放置作为输入，对于VNF-FG请求中的每个VNF。(i) 计算从当前NFVI-PoP到承载请求的VNF的NFVI-PoP的最短路径；(ii) 丢弃对当前请求没有足够带宽容量的最短路径；(iii) 将请求链给最短路径有足够带宽的VNF。

5.1.3. Both

大多数研究工作为放置和连锁问题提供了综合解决方案，关注不同的目标并采用不同的技术风格。有三项工作引入了启发式方法来解决VNF-FGE问题，目标是最小化带宽消耗[22], [50], [51]。这些启发式方法使用了VNF和NFVI-PoPs的处理能力，以及链接的带宽。在Beck和Botero的[22]启发式中，VNF-FG请求中的每个VNF都带有它可以直接连接到哪些VNF的信息。这使得VNF-FG的VNF可以自由地交换位置，并有松散或无依赖性的约束。如果一个VNF不能被链起来，那么就使用另一个满足依赖性约束的VNF。启发式试图将一个VNF放置在一个直接连接到前任的物理节点中，以最大限度地减少带宽的利用。如果没有可能放置VNF的位置，启发式会回溯前一个VNF的分配，并尝试将其安装在其他地方。与此不同，Ye等人[50]开发了一种分组策略。他们评估了哪些映射到同一NFVI-PoP的VNF的组合能带来更高的带宽节省。这些VNF组合及其直接连接的VNF被称为VNF-FG的关键子图（CS）。首先，CS被放置并链入物理网络。接下来，VNF-FG请求的其余VNF被嵌入到基础设施中。最后，Gupta等人[51]引入了一种形式化，其中每个VNF-FG请求在物理网络中有特定的起点和终点。他们使用两步解决方案放置和连锁VNFs。首先，启发式对NFVI-PoP对进行分组，其链接符合多个请求的起点和终点之间的最短路径。用户输入启发式中使用的组的数量。接下来，一个列生成[108]算法使用上一步形成的组的NFVI-PoPs放置和连接请求。

与上面详述的依靠启发式算法提供VNF-FGE问题的解决方案以最小化带宽消耗的工作不同，已经提出了两种精确算法[48]、[49]和一种元启发式算法[62]来解决具有相同目标的问题。精确算法[48]、[49]使用VNF和NFVI-PoP的处理能力来决定VNF的分配，而元启发式[62]只允许每个NFVI-PoP分配一个VNF。所有这些方法还评估了虚拟和物理链接的带宽。Carpio等人[62]提出了一种GA算法，该算法通过创建几代可能的解决方案来提高每次迭代的适配度，其中对VNF的流量被分成不同的流量，这些流量对目标VNF的副本进行负载平衡。分割流量和使用多个VNF增加了VNF-FG可以使用的可能路径的数量，从而改善了虚拟化资源的分配。适配度函数背后的想法是评估一个给定问题的可能解决方案。它们被用来对GA生成的不同解决方案进行排序，GA从自然界中借鉴了诸如变异和交叉等技术[109]。Huin等人[48]为VNF-FGE问题创建了一个ILP公式，其变量数量随着输入的VNF-FG呈指数级增长。前面提到的一项工作[51]中也使用了列生成技术，该技术被用来寻找最优解[108]。一个互动过程开始于使用ILP的松弛，为每个请求找到最佳解决方案。然后，生成限制条件，并将其添加到每个解决方案中，如果它们改善了线性松弛的值。当无法找到改进的映射时，该过程就会停止。最后，Tomassilli等人[49]介绍了两种列生成[108]方法，考虑到两种备份模型，输出了VNF-FGE问题的精确解决方案。(i) VNF之间的不相交虚拟链路，即有一条连接给定请求的所有VNF的备份路径，与主路径不相交；(ii) 单一链路保护，只有两个VNF之间的路径是冗余的。它以交互方式放松ILP约束，直到找到最佳解决方案，与Huin等人[48]类似。

最大限度地减少带宽消耗的另一种方法是在放置和连锁VNFs时减少主机资源的使用。这是三项工作所追求的目标，它们提出了离线启发式方法[11]、[52]、[53]，使用不同种类的输入映射VNF和NFVI-PoP的处理能力以及虚拟和物理链接的带宽。Luizelli等人[11]旨在使用链路延迟、处理延迟和可用许可证的数量作为输入，减少为VNF-FG服务的VNF实例的数量。首先，他们的建议是发现满足需求的最小数量的VNFs，并将ILP模型的目标函数转化为约束条件，从而形成一个更有约束的模型。在发现最小数量的VNFs后，任何使用最低数量VNFs的解决方案都被认为是可行的解决方案。Kong等人[52]增加了冗余的VNF和路径，以保证嵌入式VNF-FG呈现出理想的可用性水平。所采用的目标函数是在考虑到VNF和链接的可用性的情况下最小化计算资源的使用。首先，它找到起点和终点之间的最短路径。如果该路径不符合使用最大数量的VNF复制的可用性约束，就会选择一个不相交的备份路径。然后，确定复制体的数量和位置。最后，Chen等人[53]开发了一种贪婪的启发式，试图将VNF-FG的VNF分配到NFVI-PoP上，而同一转发图的前一个VNF已经被分配到那里。这种启发式方法试图在向其邻居重复之前耗尽NFVI-PoP的可用资源。作者表明，与Carpio等人提出的GA算法相比，这种贪婪的方法具有类似（略差）的性能[62] 。

与之前讨论的旨在减少网络或主机资源的方法不同，我们现在讨论VNF-FGE的离线算法，其具体目的是共同最小化两者。有四个启发式算法[54]、[55]、[56]、[57]和一个精确的解决方案[58]具有这个目标。他们依靠处理和链接能力来制定问题的解决方案。此外，Song等人[54]将这些资源的重要性表述为用户的偏好。他们将VNF的放置视为隐性标记模型（HMM）中的一个状态[110]，将VNF-FG的VNF之间的连锁关系视为状态的隐藏过渡。虽然不能直接观察到过渡，但NFVI-PoP的计算资源和链接的带宽等因素被用来寻找良好的状态序列，这对应于资源效率高的路径。类似地，在[56]中使用了马尔科夫链，其中不同的状态代表VNF的不同安置和/或连锁。该链的构建使各状态收敛到所需的目标，在可接受的时间内达到接近最优的解决方案。在第三个提出的启发式方法[55]中，ILP公式的约束和变量随着VNF-FG的数量呈指数级增长，被替换掉，以获得一个宽松的线性程序来嵌入VNFs。启发式方法通过选择具有最大带宽的路径，将约束和变量降至多项式数量。然后，新的线性程序在多项式时间内得到最佳解决。考虑到链路和节点都不完全可靠，Zhong等人[57]引入了一个HMM，它嵌入了VNF-FG，并为分配不具有所需可靠性水平的VNF-FG创建备份VNF和链路。在HMM中，每个隐藏状态代表一个可以分配某些VNF的NFVI-PoP。两个状态之间的转换代表在两个NFVI-PoP上分配两个VNF的概率，并考虑到链路和节点的资源使用情况进行计算。然后使用基于维特比的算法[111]预测最可能的隐藏状态序列。维特比算法是一种寻找最可能的状态序列的方法，其中创建了一个多阶段图，节点代表可能的状态。Wang等人[58]提出了一个精确的解决方案，将VNF-FGE问题制定为一个没有主机和网络约束的线性编程问题，并使用Simplex[112]来输出VNF和虚拟链接的最佳分配。然后，一种算法对VNF进行迭代，并选择有足够资源来承载它们的最合适的NFVI-PoP，以及最适合的链接到下一个NFVI-PoP。最大限度地减少带宽消耗的另一种方法是在放置和连锁VNFs时减少主机资源的使用。这是三项工作所追求的目标，它们提出了离线启发式方法[11]、[52]、[53]，使用不同种类的输入映射VNF和NFVI-PoP的处理能力以及虚拟和物理链接的带宽。Luizelli等人[11]旨在使用链路延迟、处理延迟和可用许可证的数量作为输入，减少为VNF-FG服务的VNF实例的数量。首先，他们的建议是发现满足需求的最小数量的VNFs，并将ILP模型的目标函数转化为约束条件，从而形成一个更有约束的模型。在发现最小数量的VNFs后，任何使用最低数量VNFs的解决方案都被认为是可行的解决方案。Kong等人[52]增加了冗余的VNF和路径，以保证嵌入式VNF-FG呈现出理想的可用性水平。所采用的目标函数是在考虑到VNF和链接的可用性的情况下最小化计算资源的使用。首先，它找到起点和终点之间的最短路径。如果该路径不符合使用最大数量的VNF复制的可用性约束，就会选择一个不相交的备份路径。然后，确定复制体的数量和位置。最后，Chen等人[53]开发了一种贪婪的启发式，试图将VNF-FG的VNF分配到NFVI-PoP上，而同一转发图的前一个VNF已经被分配到那里。这种启发式方法试图在向其邻居重复之前耗尽NFVI-PoP的可用资源。作者表明，与Carpio等人提出的GA算法相比，这种贪婪的方法具有类似（略差）的性能[62] 。

与之前讨论的旨在减少网络或主机资源的方法不同，我们现在讨论VNF-FGE的离线算法，其具体目的是共同最小化两者。有四个启发式算法[54]、[55]、[56]、[57]和一个精确的解决方案[58]具有这个目标。他们依靠处理和链接能力来制定问题的解决方案。此外，Song等人[54]将这些资源的重要性表述为用户的偏好。他们将VNF的放置视为隐性标记模型（HMM）中的一个状态[110]，将VNF-FG的VNF之间的连锁关系视为状态的隐藏过渡。虽然不能直接观察到过渡，但NFVI-PoP的计算资源和链接的带宽等因素被用来寻找良好的状态序列，这对应于资源效率高的路径。类似地，在[56]中使用了马尔科夫链，其中不同的状态代表VNF的不同安置和/或连锁。该链的构建使各状态收敛到所需的目标，在可接受的时间内达到接近最优的解决方案。在第三个提出的启发式方法[55]中，ILP公式的约束和变量随着VNF-FG的数量呈指数级增长，被替换掉，以获得一个宽松的线性程序来嵌入VNFs。启发式方法通过选择具有最大带宽的路径，将约束和变量降至多项式数量。然后，新的线性程序在多项式时间内得到最佳解决。考虑到链路和节点都不完全可靠，Zhong等人[57]引入了一个HMM，它嵌入了VNF-FG，并为分配不具有所需可靠性水平的VNF-FG创建备份VNF和链路。在HMM中，每个隐藏状态代表一个可以分配某些VNF的NFVI-PoP。两个状态之间的转换代表在两个NFVI-PoP上分配两个VNF的概率，并考虑到链路和节点的资源使用情况进行计算。然后使用基于维特比的算法[111]预测最可能的隐藏状态序列。维特比算法是一种寻找最可能的状态序列的方法，其中创建了一个多阶段图，节点代表可能的状态。Wang等人[58]提出了一个精确的解决方案，将VNF-FGE问题制定为一个没有主机和网络约束的线性编程问题，并使用Simplex[112]来输出VNF和虚拟链接的最佳分配。然后，一种算法对VNF进行迭代，并选择有足够资源来承载它们的最合适的NFVI-PoP，以及最适合的链接到下一个NFVI-PoP。

最后，为了降低经济成本，已经提出了四种启发式方法[59], [60], [61]。它们都考虑到了使用NFVI-PoPs和物理链路的成本。然而，每个启发式都用其他不同的输入来补充这一信息：链路延迟[59]、[60]、存储容量[60]、[61]、处理延迟[59]或内存、许可成本、VNF和链路的可用性，以及用户对这些的偏好[60]。Chen等人[59]使用了一种类似于上面介绍的技术[54]，其中VNF的安置和连锁是通过分析HMM中隐藏状态的转换而给出的。选择一个NFVI-PoP来承载一个VNF是HMM的一个状态，而遍历VNF-FG的所有VNF被视为状态之间的转换，这些状态是隐藏的。在Chen等人的启发式方法中，NFVI-PoPs中分配VNF的成本和资源使用情况在状态的转换中被观察。如果获得的路径违反了请求的延迟约束，那么对于路径中的每个NFVI-PoP，将评估替代的NFVI-PoP，如果它们能减少路径的延迟，就使用它们。Vizarreta等人[60]在嵌入VNF-FG时使用了一种贪婪的启发式方法来确保QoS。为了嵌入一个请求，他们的启发式方法首先找到VNF-FG的物理端点之间满足QoS约束的最短路径。接下来，该路径被扩展以支持VNF-FG的所有VNF。托管VNF的最佳NFVI-PoPs是产生最小成本的，其中包括使用更多网络资源和额外软件许可的成本。最后，Tastevin等人提出的启发式方法[61]由三个步骤组成。首先，它找到承载请求的VNFs的最小数量的NFVI-PoPs。为此，它对VNF实例化要求进行排序，并贪婪地将最大数量的VNF分配给同一NFVI-PoP，直到所有VNF都被分配。然后，通过中心度量[113]对NFVI-PoP进行排名，中心度量评估NFVI-PoP在VNF-FG请求的起点和终点之间的最短路径中的次数。这一衡量标准用于按递减顺序对NFVI-PoP进行排名，在第一步中获得的NFVI-PoP的数量决定了在这一步中选择多少个NFVI-PoP。最后，VNFs在选定的NFVI-PoP中的放置和连锁是通过维特比启发算法[111]进行的，其中节点代表部署VNFs的可能位置，其连接的重量是两个节点之间的最短路径。通过减少分配的NFVI-PoP和物理路径的数量，解决方案的成本也会降低。

5.2. 在线方法

离线方法解决的是一个NP-hard[100]的问题，所以那些在线的。然而，开发一个在线算法意味着有能力处理在设计时传入的需求是未知的情况。因此，支持动态到达的请求，这是电信运营商的现实，为可行的在线解决方案的构思增加了额外的难度。与上面详述的离线工作一样，我们将在线方法按分配任务分组。

5.2.1. 安置

在在线方法中，有六种方法特别关注放置问题[63], [64], [65], [66], [67], [68]。尽管有这些工作，但有一个引入了精确的解决方案[63]，四个提出了启发式方法[64]、[65]、[66]、[67]，一个是元启发式方法[68]。唯一的精确解决方案[63]旨在将VNF放在使用边缘云（即小云）的运营商云基础设施上，考虑它们之间的最佳平衡。这是在小云的最大利用率最小化和云中计算资源分配最小化之间实现的，这两者是相互冲突的。一个拟议的MILP模型考虑了几个分配目标--VNF和NFVI-PoPs的处理能力、连接运营商云和云小站的物理链路的延迟、VNF-FG的到达率和VNF的虚拟化开销--以及用户对它们定义的偏好。