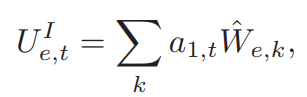
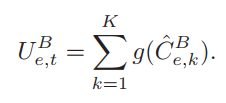
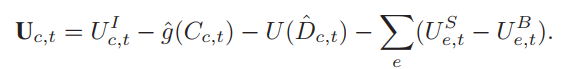
第一篇

基本架构： MEC服务器和云端通过有线核心网络连接，可以根据计算需求共享计算资源。具体来说，当MEC服务器的计算资源丰富时，可以将部分计算资源批发到云端，以提高盈利能力。相应地，云可以以较低的批发价从MEC服务器购买计算资源，减少云计算资源，以降低高昂的运营成本，提高QoS。当MEC服务器的计算需求较高时，可以以较高的回购价格从云端回购部分计算资源，以保证其服务质量。1相应地，云需要满足MEC服务器的回购需求，以提高盈利能力。这样，云可以通过低批发价的批发方案从MEC服务器获取计算资源，而MEC服务器可以通过高回购价的回购方案保证本地突发流量和突发事件的时延性能。

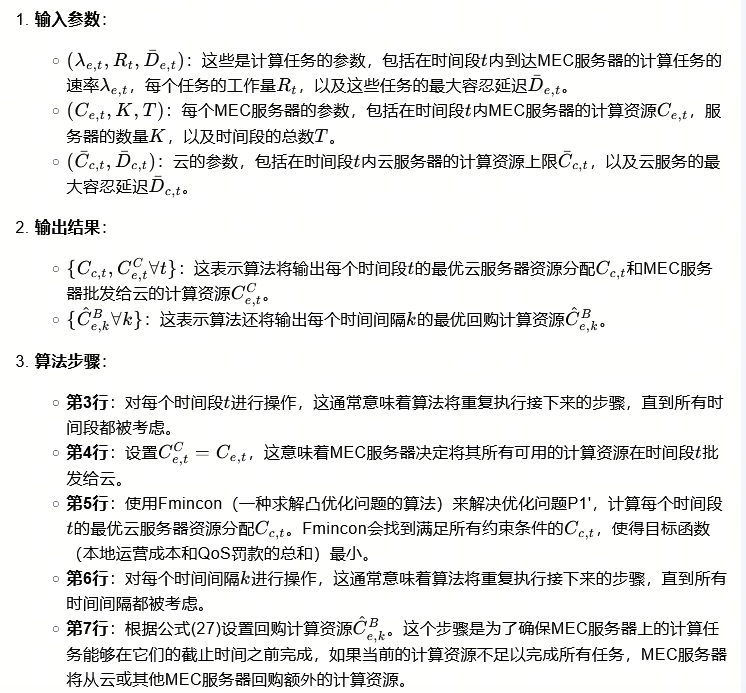
1. 两种不同的时间尺度：第一种是时隙，另一种是时间间隔。通常，一个时隙可以是几十分钟，而一个时间间隔可以是数百毫秒。每个时间间隔，MEC服务器可以将部分计算资源批发到云端，时隙内无法调整。在每个时隙，MEC服务器可以从云端买回一些计算资源，这些资源应该立即得到满足。这样，MEC服务器需要保证云端批发的计算资源，并可以根据计算需求调整回购计算资源。云可以在该时段内利用MEC批发的计算资源，并需要保证在每个时段内为MEC服务器回购计算资源。移动边缘-云计算网络的运行方案将实现如下：在前一个时隙结束时，云端需要向MEC服务器下发批发价格。然后，每个MEC服务器确定批发的计算资源并估算回购的计算资源。在此时段内，MEC服务器的批发计算资源由云端管理。如果一台MEC服务器预留的计算资源不能满足其计算需求，MEC服务器将在接下来的时间间隔内通过向云端发送回购请求来调整回购计算资源。然后，云端将回购的计算资源相应地分配给相应的MEC服务器。
2. 每个MEC服务器的计算资源可以分为两部分。第一部分由MEC服务器保留，作为计算服务器处理接收到的计算任务。第二部分批发到云端，作为灵活的计算服务器来处理来自云端的计算任务。由于计算任务的随机性，预留的计算资源可能不足以及时完成所有的计算任务，因此MEC服务器需要从云端买回一些计算资源。因此，MEC 服务器上的可用计算资源由其批发和回购方案决定。每个MEC服务器上的可用计算资源如图所示。
3. 云的可用计算资源也分为两部分，云服务器的计算资源由云进行管理，保留的计算资源取决于MEC的批发和回购计划，云服务器上的可用资源如图所示。
4. MEC和云的盈利模式
5. **MEC服务器的利润组成**：MEC服务器的利润由四部分组成：
   * **运营成本**：包括能源消耗、日常维护和日常开销等固定成本。
   * **处理用户计算任务的收入**：根据在时间段t内处理的计算工作量单位数，乘以服务费a(1,t)来计算。
   * **向云批发计算资源的收入**：根据批发给云的计算资源单位数，乘以批发价a(2,t)来计算。
   * **从云回购计算资源的成本**：是一个随着回购计算资源数量增加而增加的函数，具体形式为c1CeB,k+c2(CeB,k)2c1CeB​,k+c2(CeB​,k)2。
6. **MEC服务器的总利润**：是处理用户任务的收入、批发给云的收入减去从云回购的成本。
7. **云的利润组成**：云的利润也由四部分组成：
   * **处理计算任务的收入**：与MEC服务器类似，根据处理的计算工作量和云的服务费来计算。
   * **本地运营成本**：包括云服务器的运营成本，通常与云服务器的计算资源数量有关。
   * **与MEC服务器交易计算资源的成本**：是MEC服务器批发给云的收入减去MEC服务器从云回购的成本。
   * **由于计算延迟导致的服务质量(QoS)罚款**：计算延迟是影响服务质量的一个重要参数，罚款随着平均计算延迟的增加而增加。
8. **云的总利润**：是处理任务的收入减去本地运营成本、QoS罚款以及与MEC服务器交易的成本。
9. **决策权衡**：云在运营时需要在运营成本和服务质量之间做出权衡，以最大化其总利润。

分别目标：

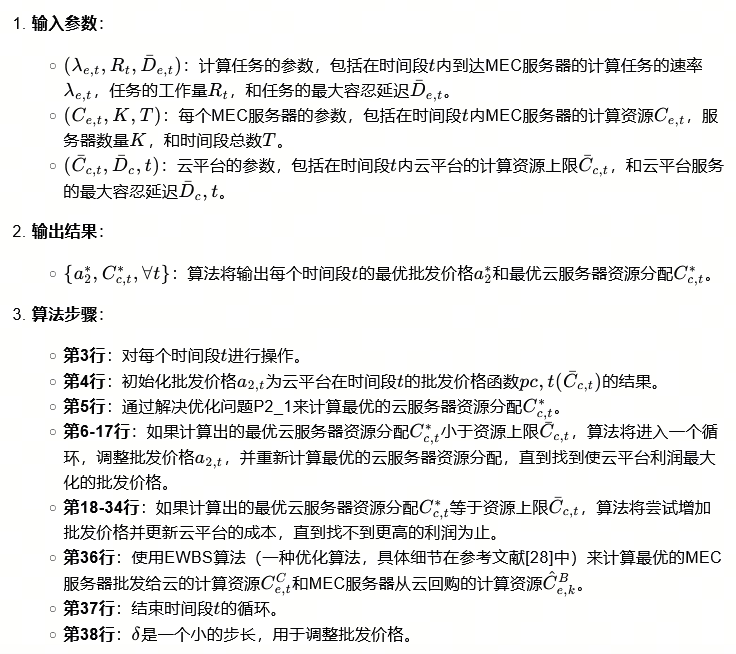
优化问题的两种情况：

第一种情况是MEC服务器和云之间没有利润转移，他们可以合并计算资源管理问题，目标是最大化整个系统的福利。 第二种情况是只有当发生利润转移时才进行计算资源共享，MEC和云需要基于对方的决策来设计自己的计算资源管理。

**无利润转移的批发回购计划**：讨论了当所有的MEC服务器和云属于同一个实体时，如何通过联合优化计算资源管理来最大化整个实体的总利润。那么MEC服务器和云之间的利润转移不会影响实体的总利润。



**含利润转移的批发回购计划**：为MEC服务器和云平台确定最优的批发价格和云计算资源管理。这个算法的目的是最大化云平台的总利润，同时考虑到计算任务的需求、MEC服务器的资源以及云平台的资源限制。



第二篇：

系统模型框架：云计算系统被建模成一个图G，包含有顶点V和边E。顶点V代表服务节点，这些节点是系统中提供计算服务的点，总共有K个。边E描述了这些服务节点之间的连接。在云计算中，需要有效的方法来将服务节点的可用资源分配给请求的任务。分配时要注意，任何服务节点上的资源总数都不能超过它的容量。同时，如果两个相邻的子任务被放置在不同的服务节点上，还需要考虑带宽成本。在系统模型中，资源分配的步骤是先获取请求任务和服务节点的信息，然后将所有子任务同时部署到多个服务节点上。邻近的子任务可以位于同一个服务节点上，也可以分布在不同的节点上。不同的资源分配方案会影响系统的性能。

图片栗子：图2展示了不同资源分配方案的示例。有五个服务节点，它们之间有通信链路，给出了两个任务t0和t1。在图2a中，任务t0的一些子任务被部署在服务节点v0上，其他子任务被部署在v1、v3和v4上。任务t1的子任务被分配在v2、v3和v4上。**由于t0的子任务t0,0和t0,1在同一个服务节点上，所以它们之间没有带宽成本。而任务t1的子任务t1,0和t1,1在v2上，t1,2和t1,3在v3和v4上，因此存在带宽成本。B展现了实现一个任务的算法路径**

**接下来两张图代表了本文提出的模型框架**

第一步：使用改进的GA来找到最大化 Z1(X)的最优解，每个代理代表一个个体。

第二步：应用多代理优化算法来寻找最小化 Z2(X)的最优解。共享代理保存所有服务节点的任务部署和资源分配信息，并支持服务代理实时访问和更新这些信息。

**右边图：**在遗传算法中，个体通常通过染色体表示，染色体由多个基因组成。每个基因代表一个特定的属性，例如在云计算资源分配问题中，一个基因可以代表一个子任务。

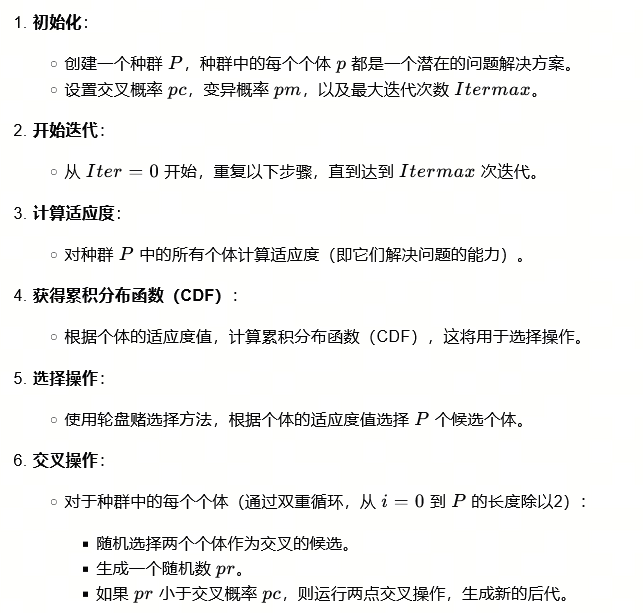
**解码步骤**

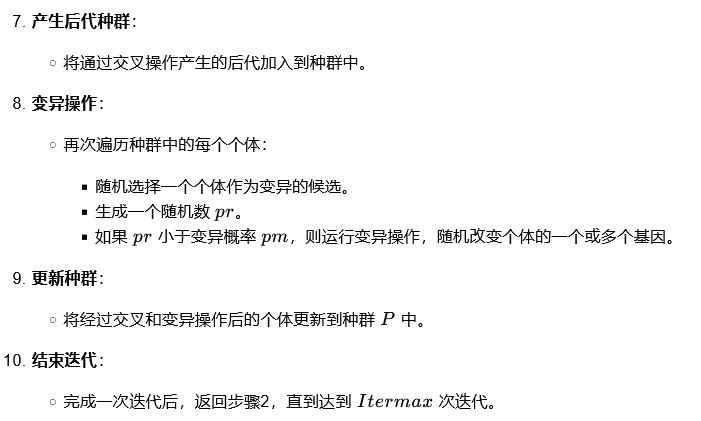
1. **初始化**：首先，我们有一个由多个服务节点组成的集合 V，每个节点 vi∈V 都有其资源限制，如CPU、内存、GPU和带宽。
2. **排序**：在解码开始前，所有服务节点根据索引进行排序，使得索引较低的节点具有较高的优先级。
3. **分配子任务**：对于个体中的每个基因（代表一个子任务），我们根据服务节点的优先级和资源限制来决定将该子任务分配到哪个服务节点上。
4. **满足资源要求**：在分配子任务时，必须确保分配到的节点有足够的资源来满足子任务的需求。
5. **处理资源冲突**：如果在尝试分配子任务时发现服务节点的资源不足以满足需求，就需要将该子任务分配到其他节点上，或者在可能的情况下，重新调整已有的分配方案。

以文中提到的图4为例，我们有一个个体 p={t00,t01,t02,t03,t04,t10,t11,t12,t13}，代表两个任务 T 的所有子任务。有三个服务节点 v0,v1,v2 ​，每个节点的资源有限。

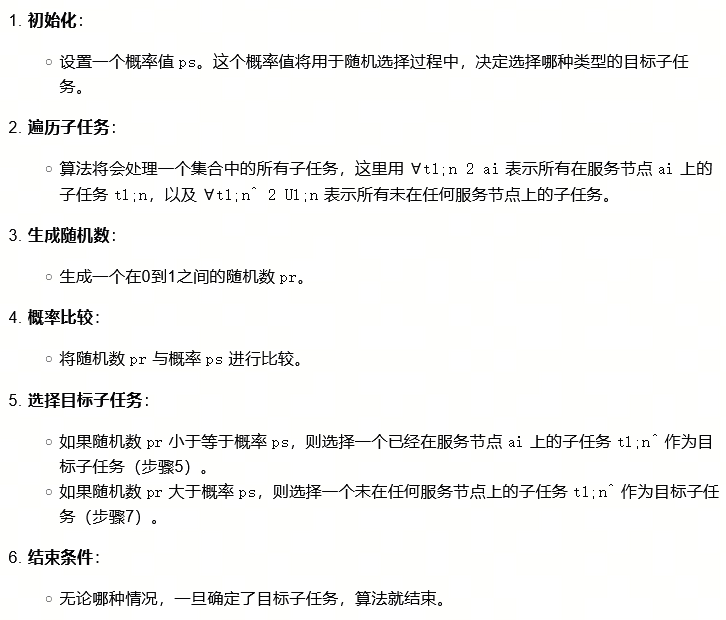
1. **开始解码**：从个体的第一个基因开始，尝试将对应的子任务分配到优先级最高的服务节点上。
2. **分配 t\_0^0 和 t\_0^1**：假设 v0​ 的资源足以支持 t00 和 t01​，这两个子任务就被分配到 v0​。
3. **资源不足**：当尝试将 t02分配给 v0 时，发现 v0​ 的资源不足以满足 t02的需求。
4. **重新分配 t\_0^2**：因此，t02​ 被分配到下一个优先级的服务节点 v1 上。
5. **继续分配**：重复上述过程，直到所有子任务都被分配完毕。例如，t03 可能因为 v0的优先级较高而被分配回 v0，而 t04​ 和 t10可能被分配到 v1，其余 t1​ 的子任务可能被分配到 v2​。
6. **完成解码**：当所有子任务都被分配到服务节点后，解码过程结束，得到的分配方案就是一个可行的解决方案。

**算法一：**

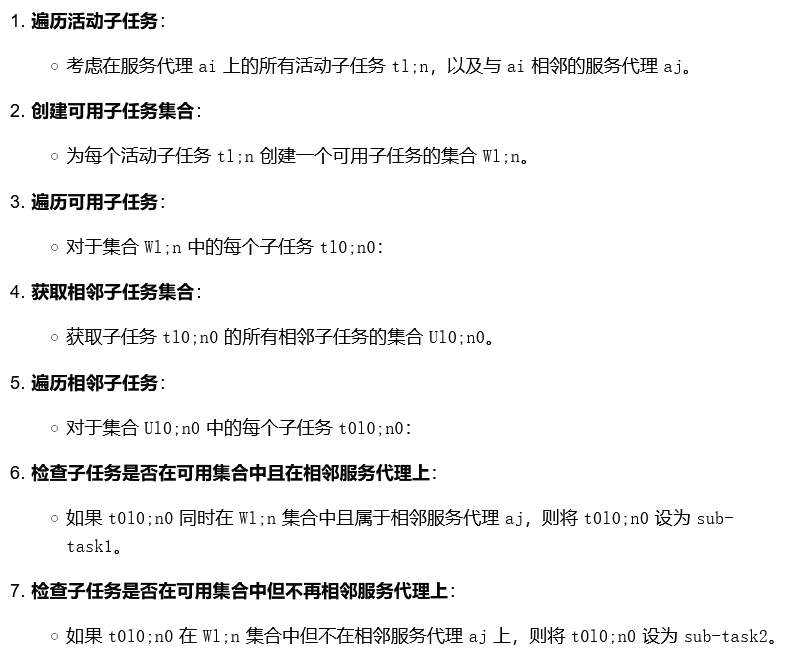


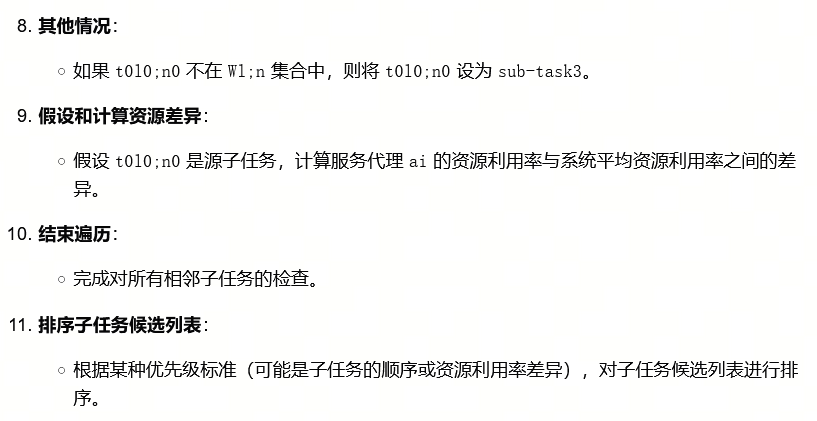


算法二：目标选择，MAO算法的主要部分之一，为了优化任务成本，尽可能把一个任务中相邻的子任务安排给相同代理。



算法三：源选择，对于子任务候选列表，可以根据优先级选择子任务，不同的源子任务也会影响总体性能。





最后一个算法：HMAO算法，

