仿真代码下载链接：
链接：https://pan.baidu.com/s/1eYIVBHiEFgIW9DQAGSC2pA
提取码：0805

本次仿真没有仿真 MLDF-GS 算法，因为这个算法必须要用到完全在本地计算的能耗，由于参数不足无法计算所以放弃仿真，但是思路和 MSDF-GS 是一样的套路，只不过前者是计算本地执行和卸载到最优容器的能耗差，后者是卸载到次优容器和卸载到最优容器的能耗差。

由于计算能量的参数未知，所以仿真时假设了 MEC 资源充足，然后是通过随机数生成 UEm 与 contianer n 关联时计算任务的能耗 m*n 矩阵。为了方便仿真，这里直接假设 container n 的数量和 UE m 的数量相等，即生成一个 m*m 大小的随机数矩阵代表能耗，行号为 UE m 的 m 值，列号为 container n 的 n 的值，这样可以比较方便的利用 KM 和 GS 算法实现匹配求解。

1.算法运行时间（在此处利用 tic toc 实现得到的是秒而不是 cpu 周期数）

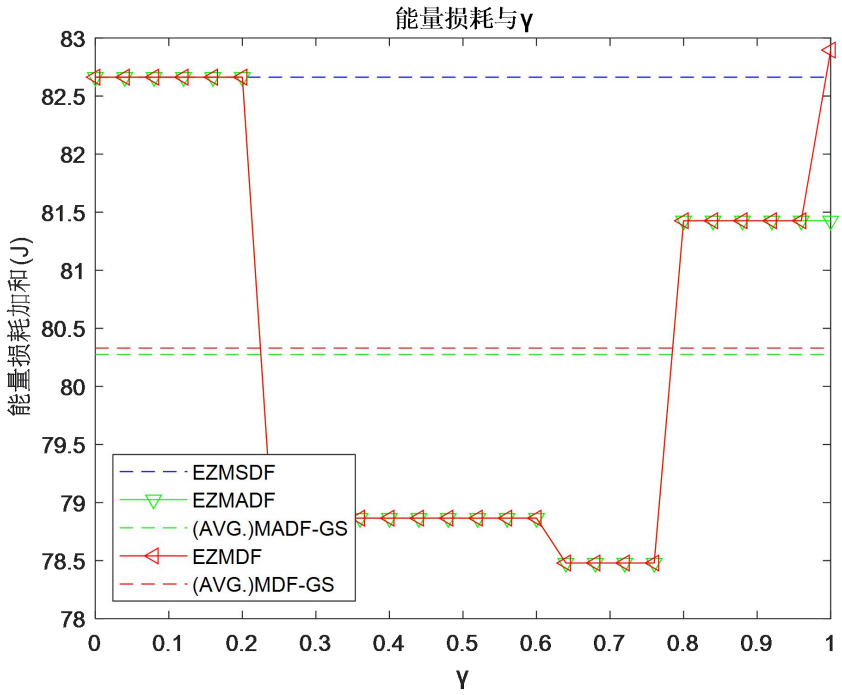
UE 数量	50	100	200	400
KM	0.058248	0.140413	0.471493	0.836457
MDF-GS	0.031465	0.035316	0.078191	0.320952
MADF-GS	0.026057	0.035621	0.082594	0.355270
MSDF-GS	0.028954	0.032171	0.072645	0.298832
GS	0.026331	0.053085	0.137064	0.481060

和论文中的运行时间不一样，应该是算法实现的方法不同。

可以观察到，GS 算法的平均运行时间比 KM 算法短得多，尤其是在 UEs 数量较大时（例如 400）。这是因为 KM 算法的时间复杂度比 GS 算法高一个数量级。因此，KM 算法糟糕的运行时间性能使其难以在大规模网络中广泛应用。

但是看到 MADF-GS 和 MDF-GS 并没有有更长的运行时间，和论文中结果不一致，根据我写的算法代码我认为即使它们更多地尝试接近全局最优，但是在每个 container n 中这两种算法中对 UE m 的排序是相同的，而在 GS 算法中对不同 container n 中对 m 的排序不同，所以 MADF-GS 和 MDF-GS 算法仅需对 UE m 排序 1 次，而 GS 算法需要对 UE m 排序 N 次，所以导致了运行时间更长，不同代码实现会有不同的运行性能时间。

仿真结果和解释如下：
2.



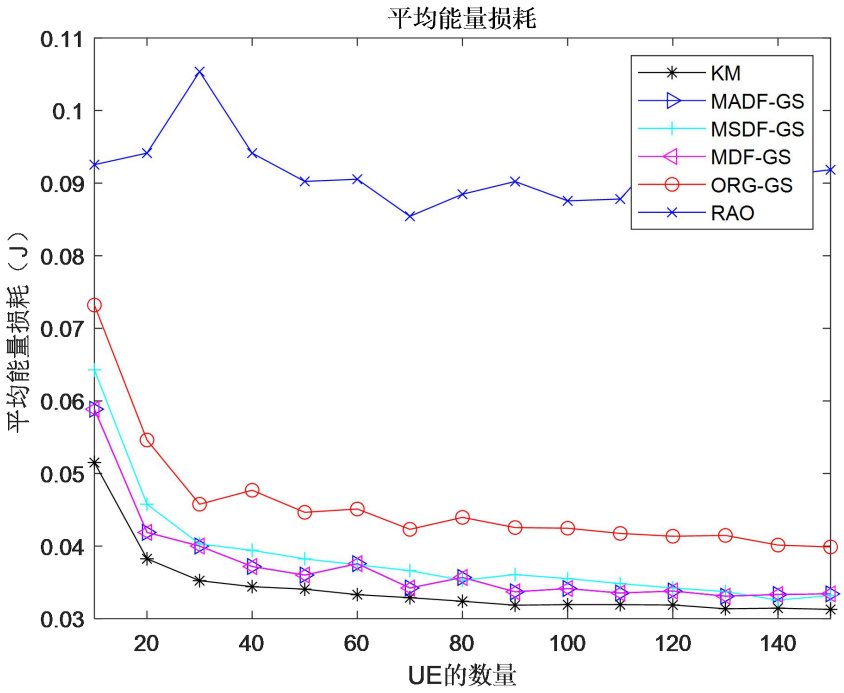
由于论文中参数没有给全面，所以利用随机数生成的 UEm 与 container n 相关联时能量损耗来测试不同算法的性能，所以和原论文中的仿真图不一致，但是仍然可以得到论文中的结论

仿真解释：

此处用 7~13 之间的随机数来模拟 UEm 与 container n 相关联时能量损耗

由于 γ 是以 0.04 微小步长变化的，所以 Q（m）排序函数得到在 container n 中对 UEm 的排名会在 γ 一定的范围内保持不变，当 γ 达到一定阈值后 UEm 的排名就会变化，这样根据 GS 算法，UE m 和 container n 的匹配也会发生改变，这样他们的能耗和就会发生突变，可以看到 γ 在[0.64,0.76]范围内的能耗是最小的，这仅是在仿真的条件下是这样的，实际情况可能会发生变化。在 γ 超过某个阈值后能耗变大是因为这两种算法高估了较差容器能耗的影响，γ 的 k 次方减去 γ 的 k-1 次方变大，靠后项的优先级增加了，与目标相悖，但是 MADF-GS 和 MDF-GS 算法所得的能耗都要比 MSDF-GS 小，这是因为他们都考虑了整体容器与 UE 相关联时的能耗，可以避免贪恋算法造成的局部最优，从而逼近全局最优解。

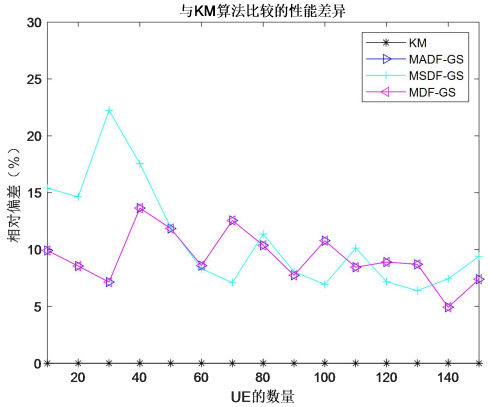
3.

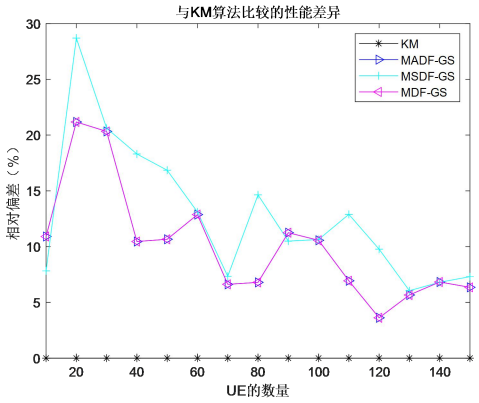


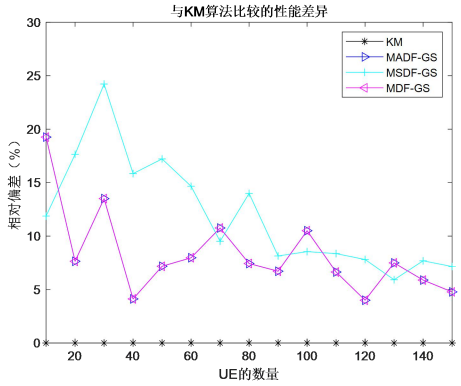
这里的仿真随机卸载策略中的随机匹配是将单位方阵的行打乱实现一个近似随机的匹配，可见随着 UE 的数量的变化平均能耗基本上没有发生什么变化。可以观察到，0RAO 表现最差，因为它不能充分利用网络状态和任务特征。ORG-GS 的性能远远落后于其他 GS 算法，因为它的排序函数过于贪婪，容易陷入局部最优。MSDF-GS 和 MLDF-GS 的表现是可以接受的，因为他们都开始考虑能量消耗的差异，这对于避免短视是有效的。MADF-GS 和 MDF-GS 表现更好，基本一致，因为可以证明在 UE 数量大的情况下，两者的排序函数实际上是等价的。

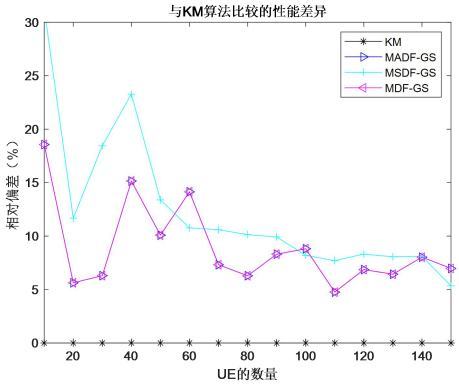
几乎所有曲线在 UE 数量增加时趋于稳定，这是因为这里假设了 MEC 资源充足，随着 UE 数量的变大，可供 UEm 选择卸载的 container n 也变多，竞争失败后仍然可以选择卸载到能耗消耗小的容器中，从而不同算法导致的差别不大，最终平均能耗趋于收敛。

4.









由于论文中参数没有给全面，所以利用随机数生成的 UEm 与 container n 相关联时能量来测试不同算法的性能，所以和原论文中的仿真图不一致，但是通过仿真图仍然可以得到论文中的结论。以上是重复四次随机实验所得到的结果，这些图由于每次随机的能耗是不同的所以得到的结果不同，但是都有一些共性：

此处用 0.03~0.15 之间的随机数来模拟 UEm 与 container n 相关联时能量损耗，可见 km 算法所得到的能耗是最小的，MADF-GS 和 MDF-GS 算法的性能一模一样，这是因为他们本质上式一致的，其他算法中可见 MSDF-GS 排序方式比 MADF-GS 和 MDF-GS 算法的性能更差，距离 KM 匹配算法的偏差更大，这是因为 MADF-GS 和 MDF-GS 都考虑了整体容器与 UE 相关联时的能耗，可以避免贪恋算法造成的局部最优，从而逼近全局最优解。

随着 UE 数量的变大，MSDF-GS，MADF-GS 和 MDF-GS 三种排序算法的性能趋于一致，这是因为这里假设了 MEC 资源充足，随着 UE 数量的变大，可供 UEm 选择卸载的 container n 也变多，竞争失败后仍然可以选择卸载到能耗消耗小的容器中，从而不同算法导致的差别不大。可以看出以上三种算法所产生的与 KM 算法的偏差随着 UE m 的数量增大而缓慢变小，最终应该趋于收敛。