算法输入：各业务的参数量和batch size（用来估算流量大小和单 GPU 训练用时），各业务的集合通信方案（ring、ps、alltoall,用来确定网络拓扑），网络资源限制；

算法输出：每轮迭代的两组拓扑（拓扑调度方案）、单轮迭代时长（优化目标）

算法思路：爬山法---只能找到局部最优解，但是怎样证明问题是凸优化还没有思路。

总算法：

1. 初始化两个集合A,B，分别存放两组业务，每一组对应一种调度模式，两种调度模式相互正交。将每一轮迭代分成两组，分别是α组和β组。其中α组进行A组业务的集合训练和B组业务的集合通信，β组进行A组业务的集合通信和B组业务的集合训练。初始化集合A，将所有业务放进集合A,初始化集合B为空集。此时代表无交叉情况，优化目标得到最差的解，通过子算法计算单次迭代用时，先令迭代解等于初始解；
2. 遍历每个业务，然后通过子算法，计算单业务交换组别后的单次迭代用时和对应的拓扑，如果每个业务对应的交换后用时都不小于迭代解，此时说明迭代解已经到达启发式算法可以得到的最优解，输出拓扑和业务分组即可。如果不是，找到小于迭代解并与迭代解相差最大的业务，执行这个业务的组别交换，其他业务不执行组别交换，然后进入下次迭代。

子算法：已知组别分配，求拓扑和单轮迭代时长。

1. 通过业务的并行度，确定每个业务在集合通信时的总流量大小。根据组别分配，假定每组用满服务器，且组内业务训练时间相同，确定最短训练时间和业务使用的服务器量；
2. 根据获得的业务服务器使用量和并行度，按流量大小降序处理业务，逐个部署业务并行单元，将并行单元部署进输出流量最小的有足够端口的pod。然后对使用ring通信的业务，通过环的连接算法，找到和已确定连接方案的业务共用pod对最多的成环方式，并按该算法连接各并行单元。如果单个并行单元无法完整部署，此时需要拆开并行单元，然后还是按照部署进输出流量最小pod的方式完成并行单元各部分的部署和连接。完成部署和连接后，可以获得流量矩阵；
3. 根据流量矩阵，找到相匹配的拓扑连接方式；
4. 通过流量矩阵和拓扑矩阵，计算集合通信用时，如果该组集合通信用时+重构时延短于最短训练用时，说明该组为训练瓶颈，直接按用满服务器的方式部署业务即可；反之，说明该组为通信瓶颈，此时无需用满服务器。使用通信时间反推业务服务器用量，然后按比例缩小各pod服务器分配量即可。计算此时的单轮迭代用时，输出两组各自的拓扑连接方案。