

智能算法及应用

期末作业

作业内容与要求：

- 1、从以下的题目中选择一题进行求解并写成论文
- 2、也可以另选相关内容、方向进行研究并写成论文
- 3、按照另附的模板进行论文写作
- 4、最终提交一份 “学号-姓名-智能算法期末论文-题目.doc”
或者 docx。同时提供对应的 PDF 文件，并做一份 “学号-
姓名-智能算法期末论文-题目.ppt” 或者 pptx
- 5、一共 3 个文件
- 6、作业提交时间：待定。

题目 1：飞机航班规划问题

航班的规划是指对航空公司的资源（如飞机、航线、资金、人员等）进行调度，并且规定正班飞行的航线、机型、飞行频率及班期时刻。航班规划的定义有两种，一种是广义上的解释，即市场分析和预测、航班频率和时刻的确定、机型的指派、飞机的排班和机组的排班。另一种是狭义的理解。狭义的航班规划只包括前三个方面的内容。根据市场分析和预测结果对航班的频率进行确定，并对机型进行指派。相应的优化问题即找出对有限资源进行合理配置的解，即找到合法解即每条航线应该投入什么机型或者是机型的组合以及其飞行频率为多少可以提高工作效率和经济利益。

在飞机调度问题的流程中，航班规划属于市场规划的范畴。通常由市场部根据机务部调度员提供的次日可用每种机型和飞机架数以及其它的具体的实际情况制定次日的航班计划。航班计划的制定在很多情况下依赖于工作人员的经验，但是随着科技的发展，航空公司的航线不断增加，飞机数也在不断增加，使用人工来进行排班便成了一件消耗时间和精力而且极难完成的事。所以，使用算法对飞机的航班进行规划是必然的时代进步，科学、高效的排班计划是算法可以满足的。

因此，飞机航班规划问题就是根据飞机的机型特性、航路的特性，对每种机型的飞机在每个航线上初次排班的安排，在满足约束条件的前提下，获得具有较大现实意义的航班班次安排。问题描述如下：

如图 1 所示，给出一个以广州为中心，分别飞往 3 个大城市（北京、上海和香港）的简单航线网络。

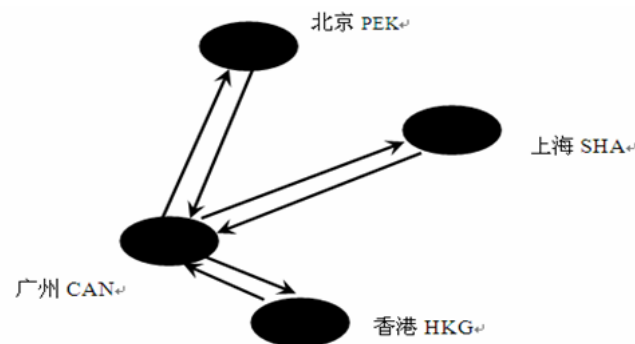


图 1 以广州为中心的简单航线网络

航空公司必须决定使用哪种机型来飞行以便满足需求，通过合理的航班规划，实现收益最大化。表 1 列出各种机型的相关特性，表 1 列出了出发城市与目的城市之间的飞行时间和票价。

表 1 飞机特性数据

机型	737-800	757-200
座位容量 S_i (个)	168	200
飞行速度 V_i (KM/小时)	700	800
小时运行成本 C_i (元)	30000	40000
日利用率 U_i (小时)	10	11
飞机架数 A_i (架)	3	5

表 2 飞机城市对飞行与票价数据【飞行时间：小时 票价：元】

城市对	737-800	757-200	票价
CAN-PEK	3.0	3.1	1500
CAN-HKG	0.56	0.5	600
CAN-SHA	2.3	2.0	1200

表 3 是在某个季度的一些市场信息。由于某种原因，737-800 机型飞机暂时不能飞往上海的航班，而 757-200 机型飞机暂时不能飞往香港的航班（这是实际问题中的机型可用性约束）。

表 3 客座率及需求情况

	北京	上海	香港
客座率	0.9	0.7	0.2
下限（班次）	5	4	3
上限（班次）	20	20	20

在完成以上问题的基础上，将问题规模扩大化，增加城市数量，增加飞机机型，并假设合理的数据，然后继续进行求解。

题目 2：飞机进港调度

随着民航事业的发展，空域拥挤的问题变得越来越严重，减少航班延误造成的经济损失已经成为当前亟待解决的问题，然而依靠扩充机场容量，增加跑道数目来减少飞机延误的方法常受到各种因素的制约，因此对到达航班进行排序和调度、优化进场飞机的次序，使其延误最小已经成为当前空中交通流量管理的一个重要内容。

简单的说，到达航班排序和调度问题（arrival sequencing and scheduling，简称ASS）的基本目标就是要在满足空域及机场容量和安全间隔约束的前提条件下对到达机场上空的飞机进行合理调度，安排降落时间，使得尽量减少航班的延误，从而减少给顾客和航空公司带来的损失。根据每架飞机的预计到达时间

（predicted landing time，简称PLT）的先后顺序安排飞机降落是解决ASS问题的一种简单常见的方法，简称先来先服务（first come first serve，FCFS）算法。尽管FCFS算法能够建立一种基于PLT的公平降落次序，但是它忽视了许多有效的提高机场空间利用率以及减少空中延迟的有效信息。例如根据不同飞机之间最小降落时间间隔（landing time interval，简称LTI）的不同，交换两架飞机的降落顺序，很大程度上可能减少飞机的空中延迟时间。

飞机之间的LTI是关于飞机的类型以及飞机之间的相对位置的函数，如表1所示，根据飞机的速度、载客量、机重以及其他技术指标，把待排序的飞机分为4种类型。从表中可以看出飞机之间的LTI是非对称的，例如一架波音747飞机降落后，至少需要等200秒才能让一架波音727的飞机降落，而一架波音727飞机降落后，只需要间隔72秒就可以让一架波音747的飞机降落。正是这种非对称性，使得可以通过交换两架飞机的降落次序来减少飞机在空中的总延迟时间，例如在0时刻有一架波音747飞机和一架波音727飞机同时到达，如果让B747先降落，再让B727降落，则总的空中延时为200s，而通过交换两架飞机的降落次序，则可以将总的空中延时降为72s。然而也正是因为这种非对称性的存在，使得基于位置交换进行优化的ASS问题变成一个NP难问题，不可能在多项式的时间内求得全局最优解。

表 1 最小时间间隔

S_{ij} (秒)		第 j 架降落飞机的类型			
		1	2	3	4
第 i 架降落飞机的类型	1	96	200	181	228
	2	72	80	70	110
	3	72	100	70	130
	4	72	80	70	90

注：1=B747；2=B727；3=B707；4=DC9

- 请根据表 2 的飞机类型和 PLT 降落情况
- (1) 使用 FCFS 方法得到所有飞机的平均等待时间。
 - (2) 进一步，设计一种更好的方法，对表 2 的飞机进行优化调度，再与 FCFS 的结果进行比较。
 - (3) 通过随机生成不同规模的测试数据，进一步验证（2）提出的方法比 FCFS 方法的优势。

表 2 飞机排序测试数据

飞机原始数据			飞机原始数据		
编号	类型	PLT	编号	类型	PLT
1	1	0	16	2	1166
2	1	79	17	2	1233
3	1	144	18	1	1642
4	2	204	19	1	1715
5	1	264	20	3	1770
6	1	320	21	1	2074
7	2	528	22	1	2168
8	1	635	23	4	2259
9	2	730	24	2	2427
10	2	766	25	1	2481
11	1	790	26	2	2679
12	1	920	27	3	2883
13	3	1046	28	2	2982
14	4	1106	29	1	3046
15	2	1136	30	1	3091

题目 3：无线传感器网络覆盖问题

无线传感网络(wireless sensor networks, WSN)已经被逐渐应用到许多实际应用中，如交通监控、移动计算、环境观测等等。在这些复杂的环境中，为了实现传感器的全覆盖从而获得更准确的结果，许多的传感器要部署在这些区域，从而造成了资源的浪费。由于传感器配备了有限的电池资源，低功耗已成为设计WSN时要考虑的关键因素。

一般而言，如果传感器可以实现全覆盖并且其中的一些传感器监视着同一区域，那这些传感器可以关闭从而节省能量。换句话说，如果我们只激活个别的传感器就能实现全覆盖，那么关闭其他的传感器就可以节省能量。因此，设计一种算法从所有传感器中找到尽可能少的传感器来实现全覆盖，就可以减少能量消耗。这也就是WSN中著名的能量有效覆盖问题(energy efficient coverage, EEC)。

对于一个给定的需要监测的 $L \times W$ 的长方形区域 A 和 N 个给定的传感器所构成的集合 $S = \{s_1, s_2, \dots, s_N\}$ ，我们需要找到一个全覆盖集合，使得它包含的传感器个数最少。

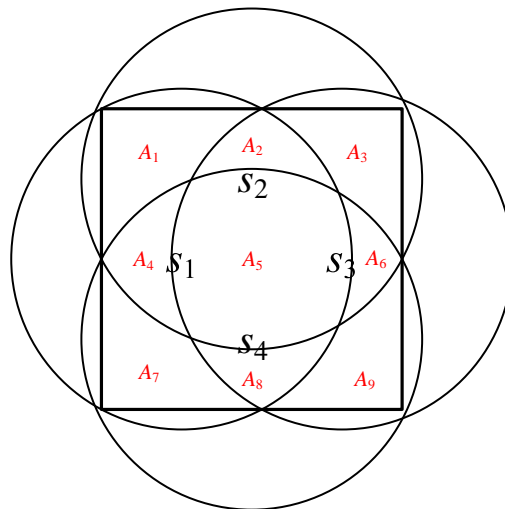


图1 传感器网络覆盖例子

如图1所示。假设我们有4个传感器，区域 A 为图1中的正方形区域。可以看到，集合 $\{s_1, s_3\}$ 和集合 $\{s_1, s_2, s_3\}$ 均可以实现全覆盖。但很明显集合 $\{s_1, s_3\}$ 比集合 $\{s_1, s_2, s_3\}$ 要好，因为它包含的传感器个数少。集合 $\{s_1, s_2\}$ 虽然也只有两个传感器，但是它并不能实现全覆盖。

为了知道集合是否充分覆盖了 A 区，我们假设传感器的位置是已知的，即每

一个传感器 s_i 位于坐标 (x_i, y_i) 处。对于区域 A 中的任意一点 $g(x, y) \in A$ ， s_i 与 g 的关系定义如下：

$$P(s_i, g) = \begin{cases} 1, & \text{if } (x - x_i)^2 + (y - y_i)^2 \leq R^2 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

其中1表示 s_i 已经覆盖了点 $g(x, y)$ ，而0表示 s_i 未覆盖点 $g(x, y)$ ， R 为传感器的传感半径。因此，对于区域 A 中的任意一点 $g(x, y)$ ，都至少存在一个 s_i 使得 $P(s_i, g)=1$ ，即可说明区域 A 被当前的传感网络覆盖。

现在，我们假设需要监控的区域 A 大小为 $50\text{m} \times 50\text{m}$ ，传感器节点 N 有100个，传感半径 R 为8m，完成如下工作：

- (1) 在 $50\text{m} \times 50\text{m}$ 范围内，随机生成100个坐标，作为每个传感器的位置；
- (2) 设计一种好的算法，找到一个全覆盖集，并使得其包含的传感器个数最少；
- (3) 扩大区域范围(如 $100\text{m} \times 100\text{m}$)以及增加传感器节点个数(如200个)，进一步验证算法的可行性。

题目 4：云计算虚拟机部署问题

面向计算资源需求的不断增长，云计算作为一个大规模的分布式计算模型，为用户提供即付即用的计算资源。云计算主要为用户提供三种不同服务：基础设施即服务，平台即服务，软件即服务。这些服务主要通过虚拟机来服务，虚拟化技术为多个用户在同一台服务器上同时独立运行提供了可能。根据研究发现，数据中心的大部分服务器的利用率只占总运行时间的11%-50%，这造成了计算资源和能源的浪费。如何提高资源的利用率，降低能耗成为了云计算数据中心管理的一个重大挑战。

当一个用户向云计算中心提交请求时，云计算中心会根据用户的环境需求，如操作系统、CPU核数、内存大小、存储大小、带宽需求等，在服务器上配置虚拟机。之后，用户再在虚拟机中运行应用。在部署虚拟机过程中，一个关键问题在于，如何在数据中心的众多服务器中选择一台服务器配置虚拟机，才能尽可能地降低成本。简单来说，如何将虚拟机部署到服务器上，降低电力消耗。当服务器处于空闲状态（即没有应用在上面运行）时，可以将服务器置于休眠状态，当资源不够时，再唤醒服务器，这可以大大降低电力的消耗。

问题描述如下：

假定某个区域内的云数据中心有1000台戴尔服务器，每台服务器都配备有8核的CPU，64GB RAM。假设这些服务器都是处于空闲状态。同一时间内1000个用户向数据中心提交请求。根据用户的请求，在服务器上部署虚拟机。这里采用的服务器的电力模型为：

$$P(u) = k \times P_{\max} + (1-k) \times P_{\max} \times u \quad (1)$$

其中 u 是服务器的CPU利用率（这里简化为已利用的核数与总核数的比值）， P_{\max} 为服务器利用率为峰值时的电力消耗，即其最大的电力消耗，设为215W； k 是服务器处于空闲状态时电力消耗与峰值电力消耗的比值，这里设为60%。如何将这些用户需求的虚拟机分配到这些服务器上，并最小化电力消耗？注意，这里不考虑机器唤醒和休眠过程消耗的电力。

- (1) 只考虑CPU核数需求。不同用户的CPU核数需求由均匀分布随机生成（处于区间[1,8]的整数）。建立模型并求解。如何在满足服务器的资源约束条

件下，在服务器上部署虚拟机，使得电力消耗最小？

- (2) 考虑多维的需求情况：包括CPU核数和RAM需求。不同的用户的需求由均匀分布随机生成（CPU核数是区间[1,8]的整数，RAM需求是区间[1,32]的整数）。建立模型并求解。同时考虑CPU核数和内存需求情况下，如何在满足服务器的资源约束条件下，在服务器上部署虚拟机，使得电力消耗最小？
- (3) 将上述两个问题的结果与FFD（First Fit Decreasing）[1] 算法进行比较。FFD算法是将虚拟机的需求降序排列，按照顺序将虚拟机放到服务器上。当已有的服务器满足当前的虚拟机资源需求时，重新开启一台服务器进行放置。

[1] Y. Ajiro and A. Tanaka, “Improving packing algorithms for server consolidation,” in *Proc. Int. Conf. the Computer Measurement Group*, 2007, pp. 399-406.

题目 5：物流选址规划问题

物流的选址规划是指为配送点选择合适的位置，以使配送点到零售商的配送距离最近或者配送费用最低。如果配送点选择离零售商近的位置能有效缩短配送距离，减少配送时间，及时满足客户需求并获得更多的订单。但是离零售商较近的位置通常处于繁华地段，建址费用和运输费用较高，增加了配送费用，因此在配送费用和配送距离之间，决策者需要选择一个折中点。此外，配送距离和配送费用不仅和配送点的选址相关，而且与配送点和零售商之间的合作关系相关。实际生活中，由于容量、地理位置、商品种类等限制，二者间的合作关系很难确定。

假设现在有5个配送点（P1，P2，P3，P4，P5），每个配送点的选址位置有3个（L1，L2，L3）。这5个配送点要给10个零售商（S1，S2，S3，S4，S5，S6，S7，S8，S9，S10）配送商品。简单起见，一个配送点只能为两个零售商服务，同时，一个零售商只接受一个配送点的商品。表3给出配送点各个选址位置到各个零售商间的距离。

表 3 配送点选址位置与零售商间的距离

	S1	S2	S3	S4	S5	S6	S7	S8	S9	10
P1.L1	58	96	91	77	94	68	89	73	68	96
P1.L2	72	99	90	83	79	68	81	82	54	52
P1.L3	65	96	98	60	92	59	83	61	67	99
P2.L1	66	57	52	53	52	87	90	80	65	76
P2.L2	90	84	78	100	61	55	98	91	72	94
P2.L3	62	50	63	53	93	87	86	90	76	57
P3.L1	91	51	83	70	63	81	85	90	83	76
P3.L2	62	85	76	81	98	69	91	52	84	86
P3.L3	70	65	52	65	69	64	81	69	81	77
P4.L1	74	52	61	70	71	53	72	52	80	79
P4.L2	55	73	95	75	62	90	96	80	76	97
P4.L3	80	88	83	83	75	58	57	74	94	62
P5.L1	68	59	77	65	91	81	70	96	65	68
P5.L2	96	86	71	90	69	86	95	84	56	92
P5.L3	97	65	75	96	78	63	51	83	83	75

请根据表3中的距离信息，求该问题最优的配送点选址位置、配送点及零售商的最优合作关系，使得配送距离最短。

题目 6：云任务调度问题

云计算（Cloud Computing）是分布式处理（Distributed Computing）、并行处理（Parallel Computing）和网格计算（Grid Computing）的发展，是基于互联网的相关服务的延伸、应用和交付模式，通常涉及通过互联网来提供动态易扩展的虚拟化的资源。

一个任务流中有许多的任务，而这些任务之间存在一个拓扑结构，即任务之间有着父子关系，这个结构可以形成一个无回路有向图（DAG）。我们定义一个任务流 $W=(T,E)$ ，其中 T 是所有任务的集合， E 是所有边的集合。 e_{ij} 表示任务 t_j 必须在 t_i 执行完之后才能执行，即 t_j 是 t_i 的子任务。

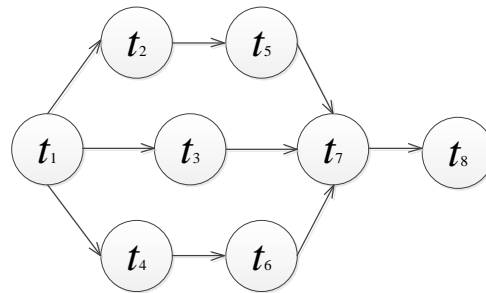


图 1 任务流实例

云计算资源调度问题是如何将任务流中的各个任务合理地分配到云计算平台的资源上，很好地满足用户的需求。图 2 是一个云计算资源调度的实例，其中 t_4 , t_5 以及 t_7 分配到计算资源 r_1 上运行， t_1 , t_2 以及 t_6 分配到计算资源 r_2 上运行， t_3 以及 t_8 分配到计算资源 r_3 上运行。

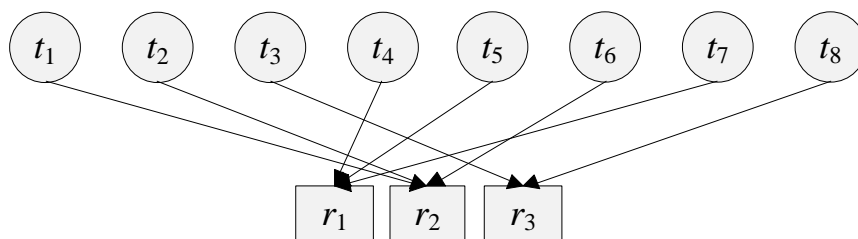


图 2 任务流调度实例

附件中包含一个 40 个任务 10 个资源的任务流测试样例，请根据以上描述，解决以下两个问题：

- （1）通过贪心算法进行求解，最小化任务流的运行时间。
- （2）进一步设计一种更科学的方法并与贪心算法进行比较。

题目 7：云任务调度问题

云计算（Cloud Computing）是分布式处理（Distributed Computing）、并行处理（Parallel Computing）和网格计算（Grid Computing）的发展，是基于互联网的相关服务的延伸、应用和交付模式，通常涉及通过互联网来提供动态易扩展的虚拟化的资源。

一个任务流中有许多的任务，而这些任务之间存在一个拓扑结构，即任务之间有着父子关系，这个结构可以形成一个无回路有向图（DAG）。我们定义一个任务流 $W=(T,E)$ ，其中 T 是所有任务的集合， E 是所有边的集合。 e_{ij} 表示任务 t_j 必须在 t_i 执行完之后才能执行，即 t_j 是 t_i 的子任务。

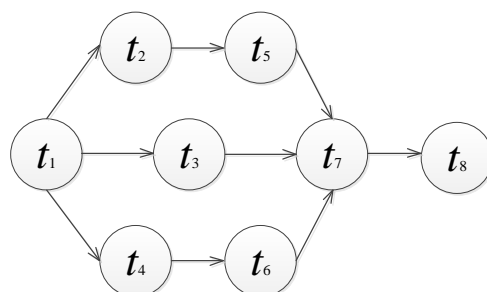


图 1 任务流实例

云计算资源调度问题是如何将任务流中的各个任务合理地分配到云计算平台的资源上，很好地满足用户的需求。图 2 是一个云计算资源调度的实例，其中 t_4 , t_5 以及 t_7 分配到计算资源 r_1 上运行， t_1 , t_2 以及 t_6 分配到计算资源 r_2 上运行， t_3 以及 t_8 分配到计算资源 r_3 上运行。

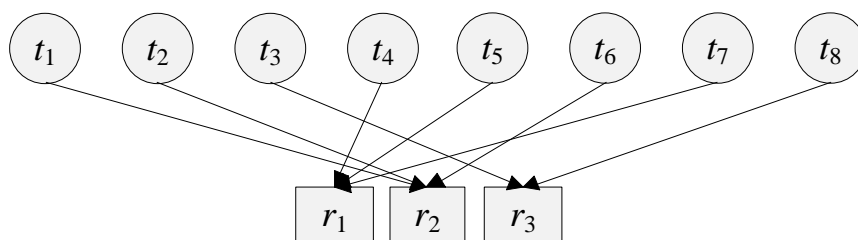


图 2 任务流调度实例

附件中包含一个 40 个任务 10 个资源的任务流测试样例，请根据以上描述，解决以下两个问题：

- （1）通过贪心算法进行求解，最小化任务流的运行时间。
- （2）进一步设计一种更科学的方法并与贪心算法进行比较。

题目 8：基于演化计算的覆盖半径可调无线传感器网络寿命优化方法研究

1. 研究内容

无线传感器网络（Wireless Sensor Network, WSN）被广泛应用于环境监测，入侵检测，智慧农业等领域。WSN 的许多功能都依赖于对目标对象的覆盖。常见的目标对象有一个区域，一条边界以及一组目标点。然而，传感器的电池通常是不可充电的，所以如何高效地利用传感器的电量延长 WSN 的寿命（lifetime）是一个富有挑战性的问题。其中，WSN 的寿命指的是 WSN 可以维持所有目标对象都被覆盖的时间。

Sensor Activity Scheduling 是一种常用的寿命优化方法。由于传感器往往是密集地分布在目标对象所在的区域，所以开启一部分传感器就可以实现对目标对象的完全覆盖，这时其他传感器就可以进入休眠状态以节省电量。这样一组可以覆盖所有目标对象的传感器就称为一个覆盖方案。假设 WSN 中所有传感器都是同一种并且只有一种覆盖半径，那么我们就可以通过找出一组相互之间没有共同传感器的覆盖方案，然后依次执行这些覆盖方案，这样就可以延长 WSN 的寿命。这一研究方向已有广泛研究。

然而，近年来出现越来越多覆盖半径可调的传感器，这对 Sensor Activity Scheduling 提出了新的挑战。一方面，传感器的可选模式不再只有两种（开/关），而是有多种覆盖半径可选，这极大地增大了搜索空间。另一方面，一个覆盖方案中各个传感器所开启的覆盖半径不同，所以在某些传感器耗尽电量后，另外一些传感器还有剩余电量可以继续工作，所以传感器的电量分配也必须妥善考虑。

我们将 WSN 中传感器的集合定义为 $S=\{s_1, s_2, s_3, \dots, s_{|S|}\}$ 。每一个传感器都是同一种类型的。每个传感器的可选覆盖半径定义为 $R=\{r_0, r_1, r_2, r_3, \dots, r_K\}$ ，其对应的电量消耗速率定义为 $E=\{e_0, e_1, e_2, e_3, \dots, e_K\}$ 。其中， R 中的覆盖半径满足 $r_i < r_{i+1}$ ($i = 0, 1, \dots, K-1$)。 r_0 表示不开启该传感器，其值为 0。覆盖半径和能量消耗速率的对应关系如公式（1）所示，其中 e_K （最大覆盖半径对应的能量消耗速率）是已知的。

$$e_i = e_K \times (r_i / r_K)^2 \quad (1)$$

基于上述这些变量定义，覆盖半径可调无线传感器网络寿命优化问题可以建模为公式（3）–（5）这个线性规划模型。公式（3）中的 X 表示一个大小为 M 的覆盖方案集合。 $X_{i,j}$ 表示传感器 S_j 在覆盖方案 X_i 中所开启的覆盖半径的下标，例如 $X_{i,j}=2$ 表示 S_j 被开启在 r_2 这个覆盖半径。 ε_i 表示覆盖方案 X_i 的开启时间， b_{ini} 指的是每个传感器的初始电量。在这个线性规划模型中，公式（2）为模型的优化目标，即 WSN 的寿命 NL 。 NL 等于所有覆盖方案的开启时间的总和，因为每个覆盖方案是依次开启的。公式（3）为传感器的电池电量约束，即每个传感器在各个覆盖方案中所分配的电量总和不能超过传感器的初始电量。公式（4）是一个边界约束，即每个覆盖方案的开启时间不能小于 0。

$$\max NL = \sum_{i=1}^M \varepsilon_i \quad (2)$$

$$s.t. \begin{cases} \sum_{i=1}^M e_{X_{i,j}} \times \varepsilon_i \leq b_{ini}, \forall j \in \{1, 2, 3, \dots, |S|\} \\ \varepsilon_i \geq 0, \forall i \in \{1, 2, 3, \dots, M\} \end{cases} \quad (3)$$

$$\varepsilon_i \geq 0, \forall i \in \{1, 2, 3, \dots, M\} \quad (4)$$

如果 X 包含 WSN 中所有可行的覆盖方案，那么这个线性规划模型就可以精确得到问题的最优解（即最长寿命）。但是，枚举出所有的可行覆盖方案无法在可接受的时间内实现，并且当 X 包含非常多覆盖方案时，线性规划模型的求解也会非常耗时。

针对这个问题，本项目提出基于演化计算的方法为上述的线性规划模型提供一组高效的覆盖方案。演化计算的种群中，每个个体就表征一个覆盖方案。通过种群的不断进化，可以提供一组可以使线性规划模型得到更长寿命的覆盖方案。如何根据该优化问题的特点，设计演化计算方法的适应值评估机制和种群演化策略，是本项目的核心研究内容。

2. 研究路线

1. 根据优化问题的特点，选择合适的演化计算方法进行求解，并结合问题特征设计算法策略，优化 WSN 的寿命。
2. 通过大量 WSN 的实例对所提出算法的性能进行验证，并与现有方法进行比较。

题目 9：面向城市场景的无线传感器网络部署优化和动态调度方法研究

1. 研究内容

无线传感器网络可以用在城市的多种应用场景中，例如交通流量监测，人流监测等。不同的应用场景对无线传感器网络的要求不同，例如为了提高监测的稳定性，有些场景要求同时有多个传感器对同一目标对象进行检测如何高效地部署无线传感器网络，是一个极具挑战性的问题。

a) 无线传感器网络的部署优化问题

通过最少量的无线传感器达到城市应用场景的需求，节约无线传感器网络的部署成本。

b) 无线传感器网络的动态调度问题

城市应用场景的需求不是固定的，比如举办大型活动或集会时就需要部署更多的无线传感器以提升监测精度和稳定性。另外，针对流动的无线传感器网络的路径优化也是一个可行的研究方向。

2. 研究路线

- a) 查阅与研究内容相关的国际期刊文献，进行整理并形成综述。一些检索的关键词：intelligent city, wireless sensor network 等。
- b) 瞄准一个城市应用场景，构建问题模型。
- c) 采用演化计算方法求解问题模型，并结合问题特征设计算法策略，提升算法性能。
- d) 通过大量测试样例对所提出算法的性能进行验证，并与现有方法进行比较。

题目 10：智能优化算法改进研究

认真研读以下论文（也可以找其他 IEEE Transactions 论文），然后重做一遍论文，得到和原文一致的结果。

或者提出一种改进了智能优化算法。

1. **Zhi-Hui Zhan (詹志辉)**, J. Zhang, Y. Li, and H. Chung, “[Adaptive particle swarm optimization](#),” *IEEE Transactions on Systems, Man, and Cybernetics--Part B*, vol. 39, no. 6, pp. 1362-1381, Dec. 2009.
2. **Zhi-Hui Zhan (詹志辉)**, J. Zhang, Y. Li, and Y. H. Shi, “[Orthogonal learning particle swarm optimization](#),” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 6, pp. 832-847, Dec. 2011.
3. Y. H. Li, **Zhi-Hui Zhan (詹志辉)(Corresponding Author)**, S. J. Lin, J. Zhang, and X. N. Luo, “[Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems](#),” *Information Sciences*, vol. 293, no. 1, pp. 370-382, Feb. 2015.