

Intermediate Report

|  |
| --- |
| Degree of Computer Science  SWJTU-Leeds Joint School |

Online scheduling of image satellites based on deep reinforcement learning

Guochen Zhao

Submitted in accordance with the requirements for the degree of  
<BSc Computer Science>

**<2020/2021>**

Type of Project:\_\_\_\_\_\_\_\_\_\_\_\_**ESw**\_Exploratory Software\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student)

© <2020> The University of Leeds and <Guochen Zhao>

**Contents**

[Chapter 1 Background Review 1](#_Toc59706177)

[Chapter 2 Introduction 2](#_Toc59706178)

[2.1 Aims and objectives 2](#_Toc59706179)

[2.2 Initial project plan 2](#_Toc59706180)

[2.3 Definition of Deep reinforcement learning 3](#_Toc59706181)

[2.4Deep reinforcement learning framework 3](#_Toc59706182)

[2.5 Ethical issues 4](#_Toc59706183)

[Chapter 3 Project Debriefing 5](#_Toc59706184)

[3.1Data generation 5](#_Toc59706185)

[References 10](#_Toc59706186)

(因为中期报告中已经写了相关的背景介绍和基本框架介绍，所以我这边会在这个基础上继续先用中文去写我现在已经做了的相关工作）

# Chapter 1 Background Review

#### 1.1 Background of the project

With the rapidly development of science and technology, our society has entered the information era. Satellite observation has become an essential part for this information era and people’s daily life. Due to satellite observation has a lot of advantages like wide observation scope, long observation time and smaller limiting conditions to access information compare to other observation method, satellite observation technology has been widely applied in military, medical, information technology and other fields. For example, satellite observation technology can help people prevent some natural disasters, besides this reason, people also need to get some important information by satellite observation technology. For satellite observation, satellites’ tasks scheduling is the core part of satellite observation. Reasonable satellite scheduling can help us get the needed information in a timely manner, make our life more convenient.

#### 1.2 Reasons to study online scheduling of image satellites

However, in traditional satellite observation, all the tasks that need to be observed are collected before the satellite start to observe, this observation method can realize most observation method. However, when some urgent tasks need to be dealt with immediately o when the tasks are arrive dynamically, the online satellite scheduling and real-time information services becomes very important. That’s the reason why online satellite scheduling method is important for the development of future. In this paper, some algorithms based on deep reinforcement learning is proposed to solve online scheduling of image satellites problem, besides this reason, in the future, the tasks will change rapidly and the collected tasks are not static, so satellites are needed to schedule and process related tasks in real time.

# Chapter 2 Introduction

## 2.1 Aims and objectives

The topic of the project is about online scheduling of image satellites based on neural networks and deep reinforcement learning. The aim of this project should be using neural networks and deep reinforcement learning algorithms to build a model which can find a better solution for image satellite scheduling. There is the specific problem description: The situation is based on a limited time span, there is an image satellite, which has a limited storage and observe tasks within time windows, the satellite’s time windows for the tasks can be got by STK, which is a professional software to get scientific data, the reward of the selected task will be added if the it’s selected successfully. The objective is that at the end of selected time span, maximize the total reward which comes from selected tasks.

## 2.2 Initial project plan

To achieve the aim there are six steps in total: Production of data, Data process, Problem modelling, Algorithm design, Setting of experiment and Results of experiment. There is a description in detail about these six steps:

**Data generation:** In this process, STK is used to produce scientific data: a concise time span about a satellite can access the selected places, which is labeled as T in this situation.

**Data process:** Consider the production of STK software is a report in txt format, which means the time span data need to be extracted and stored in a data structure so that it can be used directly in the next step.

**Problem modelling:** Complete a training environment file of this project, based on the knowledge of machine learning, in this environment file, the problem’s state, action, reward, termination conditions and so on need to be presented in suitable data structure. Besides this, the environment file also need to include update state, reset state and check current state.

**Algorithm design:** In this step, the scheduling algorithm need to be designed and improved.

**Setting of experiment:** In this step, some parameters and structure need to be changed to improve the performance of the training which might include the choices about the deep reinforcement leaning framework(DQN,A3C,DDPG),the structure of the neural network, and some specific parameters.

**Results of experiment:** The final step will display different experiment result, then get the best method to achieve the objective by comparing different experiment results.

## 2.3 Definition of Deep reinforcement learning

Deep reinforcement learning(DRL)is a kind of machine learning, combine deep learning and reinforcement learning together, which can maximize the profit for a specific problem by reasonable dynamic decision making.[[1]](#endnote-0)[F](IntermediateReport_ZhaoGuochen.docx)ig 2-1 gives a basic explanation about a single loop for DRL: There are four main elements in DRL , agent, environment, state and reward. First, the agent makes decision according to current state and reward, then the environment will update current state and reward and send it to the agent, the loop will continue until the environment satisfied termination conditions.

## 2.4Deep reinforcement learning framework

To find the best solution for online scheduling of image satellites based on neural networks and deep reinforcement learning, three deep reinforcement learning frameworks are used. There is a description of these three frameworks.

1. DQN(Deep Reinforcement Learning)

Reinforcement learning algorithms can be divided into three categories: value based, policy based and actor-critic. DQN belongs to value based algorithm, which means there is only one value function network, no policy network. Deep reinforcement learning is a method that integrates neural network and Q learning. Compared with Q Leaning, it does not need table to store Q values, but through repeated training of neural network to get the best solution. DQN has two main characteristics :(1) Experience replay. DQN stores the data obtained from the system explored environment, then randomly collects samples to update the parameters of the deep neural network. (2) Fixed Target Q-Network. There are two types network in DQN. The main-network is used to determine the action, the target-network is used to make a decision on action value when calculating the loss function. The DQN’s framework is similar to Q-learning.

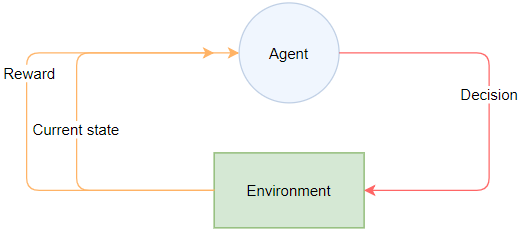


Figure 2-1 a simple figure of RL (one loop)

1. A3C(Asynchronous Advantage **Actor-Critic**)

A3C is based on Actor-Critic algorithm which is DRL combine valued based and policy based.A3C is mentioned by DeeoMind team to reduce compute skills and training time.[[2]](#endnote-1)The main idea of A3C is to learn and integrate all of its experiences through multiple agents in parallel.Fig2-2 shows the framework of A3C.Comparing to QDN,A3C doesn’t have experience,A3C has many agents which can Interact independently with the environment then update the parameters in global network. Fig 2-3 shows the network structure of A3C.The input is the current state, the output is the value of state V and the strategy π(s).

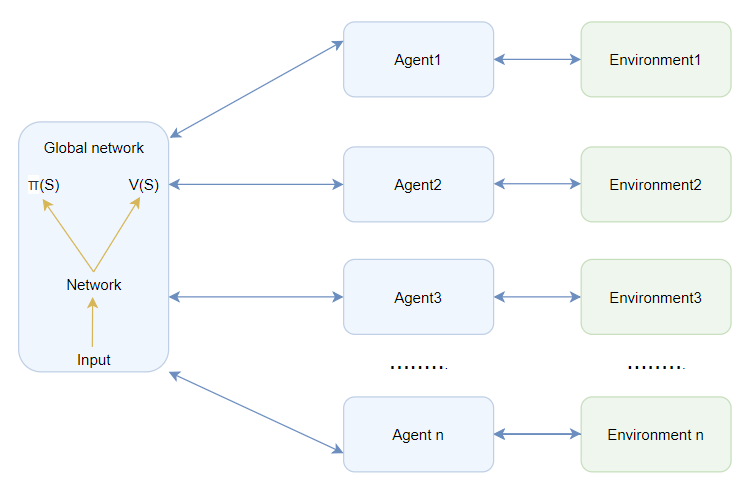
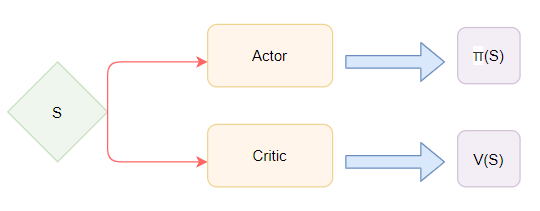
 

Figure 2-2: framework of A3C Figure2-3: network structure of A3C

1. DDPG(Deep Deterministic Policy Gradient)

DDPG is also based on Actor-Critic algorithm, comparing to A3C algorithm, similar to DDQN,DDPG use double network and replay experience to solve problem rather than use multi-threading to solve problems. In this algorithm, the actor doesn’t need to produce each action’s possibility, it will produce an action. For the DDPG’s network structure, it has 2 actor network and 2 critic network. The actor networks are used to produce specific action, the critic networks are used to produce value.

## 2.5 Ethical issues

There are no ethical issues involved in this project

# Chapter 3 Project Debriefing

Up to now, the first step is done, there are specific description of this step:

## 3.1Data generation

Consider the objective of our project is to use knowledge about deep reinforcement learning to solve satellite scheduling problem, it’s very important to get scientific data about satellite especially precise time windows data for each place need to be observed. As the first step of this project, STK, a professional software to solve celestial issue, is used to product data. There is a description in detail about how to production data using STK.

1. Start STK create a new scenario, set the start time and stop time of the scenario.

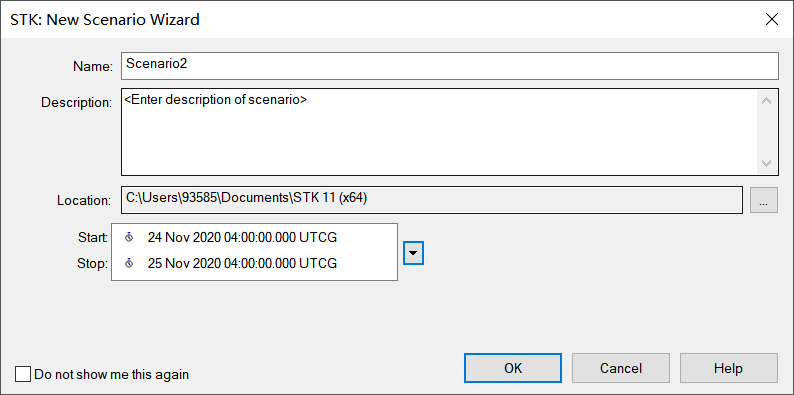


Figure 3-1-1 establish a new scenario

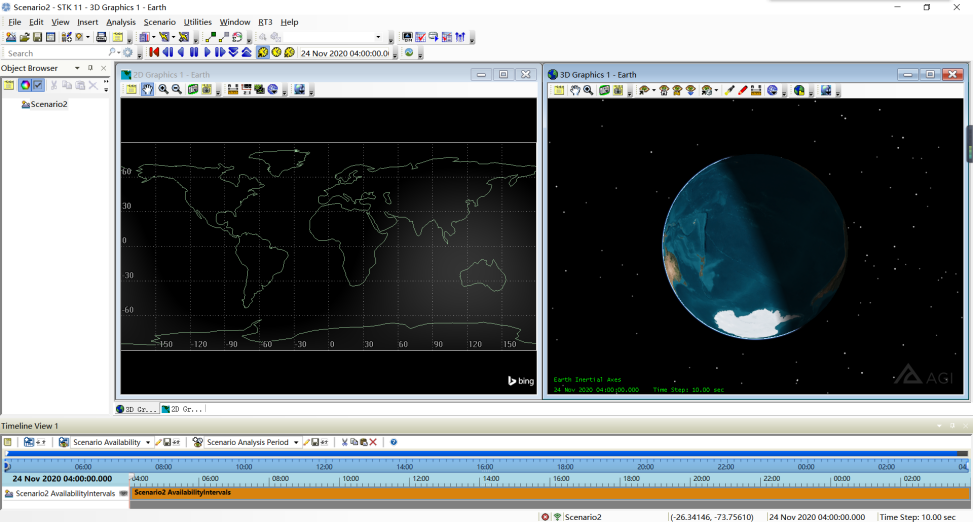


Figure 3-1-2 Main window

1. **Insert the places satellite need to observe.** First, click ‘insert’ in the top bar, click ‘New’ and select Insert Default In the right area, it will insert a default place in graph like figure 1-4.

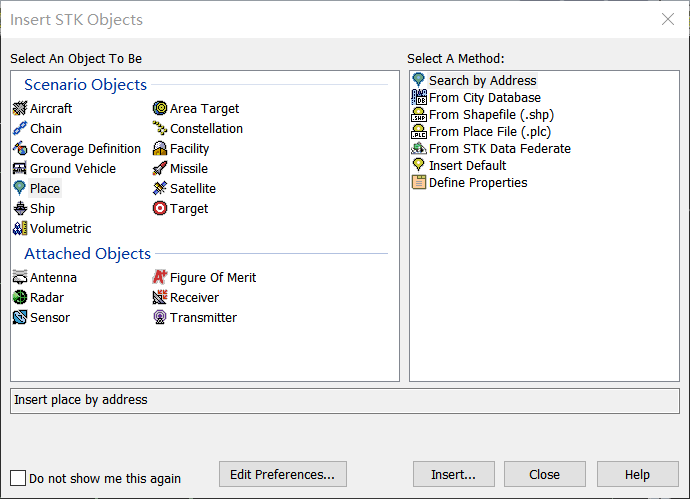


Figure 3-1-3 Insert a place

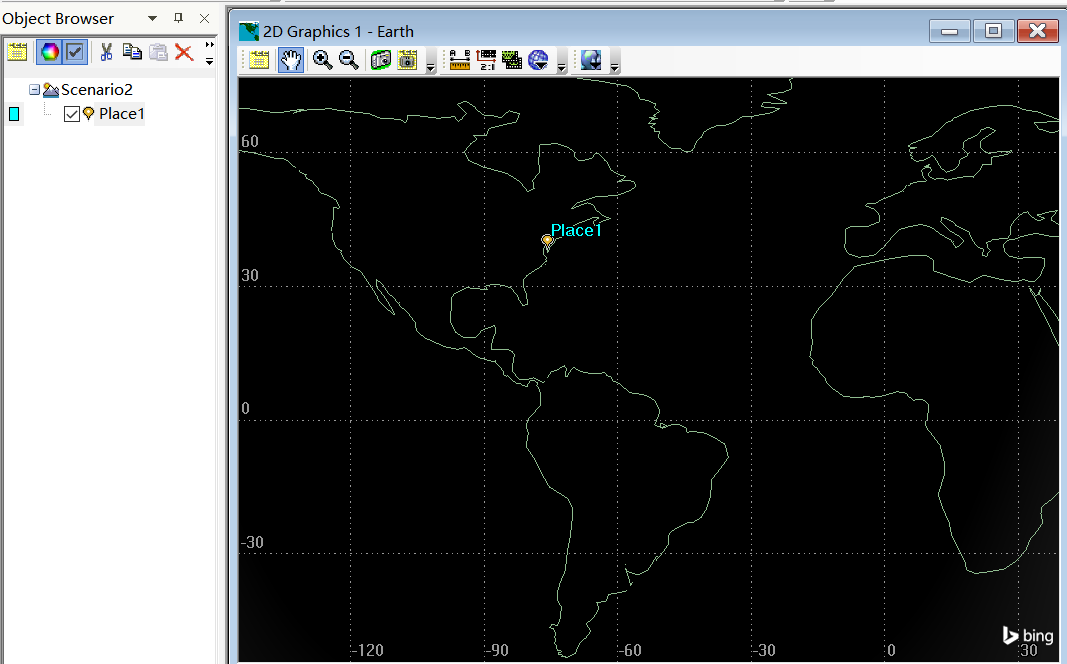
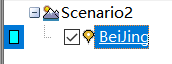


Figure 3-1-4 Insert place successfully

1. **Change the parameter of the place.** For example,change the place to Beijing. First, rename the place1,.Second, double click Beijing to change its attribute. In this project, just change the latitude and longitude of the place to 39.9 and 116.467 respectively. After that the ideal place is obtained. Other city’s related parameters are displayed in table 3-1-1.

**Table 3-1-1: parameters of some cities[[3]](#endnote-2)**

|  |  |  |
| --- | --- | --- |
| Place name | Latitude | Longitude |
| Beijing | 39.9 | 116.467 |
| Shanghai | 31.2333 | 121.468 |
| Chengdu | 30.67 | 104.07 |
| Tianjin | 39.15 | 117.186 |
| Chongqing | 29.5333 | 106.533 |

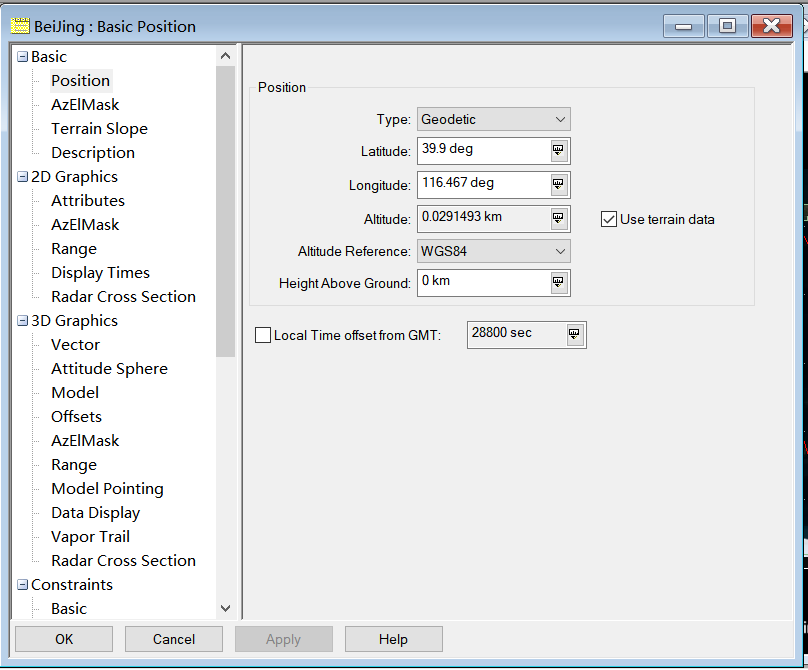


Figure 3-1-5: Set city’s parameters

1. **Insert a satellite.** Step is similar to step 2.First click the insert and find the icon ,double click it, figure 3-1-6 shows the page after click it. Some parameters can be changed in this page, but not enough. In next step, how to change parameter will be introduced in detail. Finally, click ‘OK’, a satellite is inserted successfully.

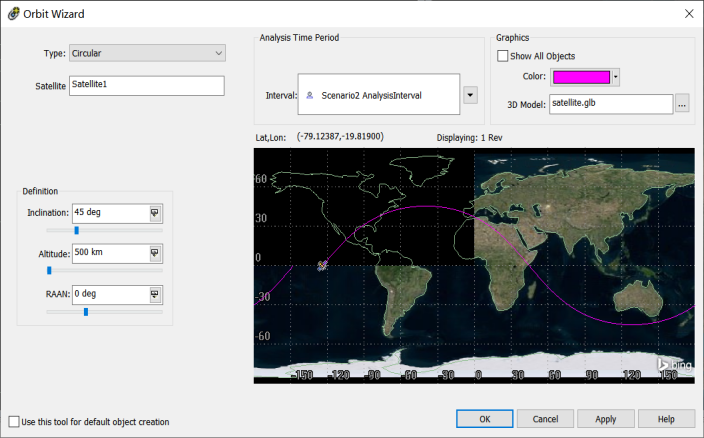


Figure 3-1-6 Insert satellite

1. Change the orbital elements of the satellite. Orbital elements include Semimajor axis,eeccentricity,inclination, argument of perigee, RAAN and mean anomaly. Some real satellite’s data have prepared in table3-1-2.Then double click satellite icon  and change the parameter of satellite according to the data in table1.Figure 3-1-7 shows the interface after changing the six data. Then a satellite is obtained.

**Table 3-1-2:orbital elements of some satellite[[4]](#endnote-3)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Satellite Name | Semimajor axis/km | Eccentricity | Inclination | Argument of perigee | RAAN | Mean Anomaly |
| HRV | 7193.59 | 0.002344 | 98.726 | 107.085 | 77.105 | 184.659 |
| MTI | 6891.832242 | 0.002011 | 97.325 | 333.934 | 237.611 | 112.260 |
| Orb-View | 6826.167581 | 0.001063 | 97.326 | 76.8 | 81.466 | 136.42 |
| IKONOS | 7046.72 | 0.000919 | 98.138 | 178.688 | 80.162 | 244.539 |
| ALI | 7069.32 | 0.000753 | 98.257 | 195.388 | 72.149 | 231.877 |

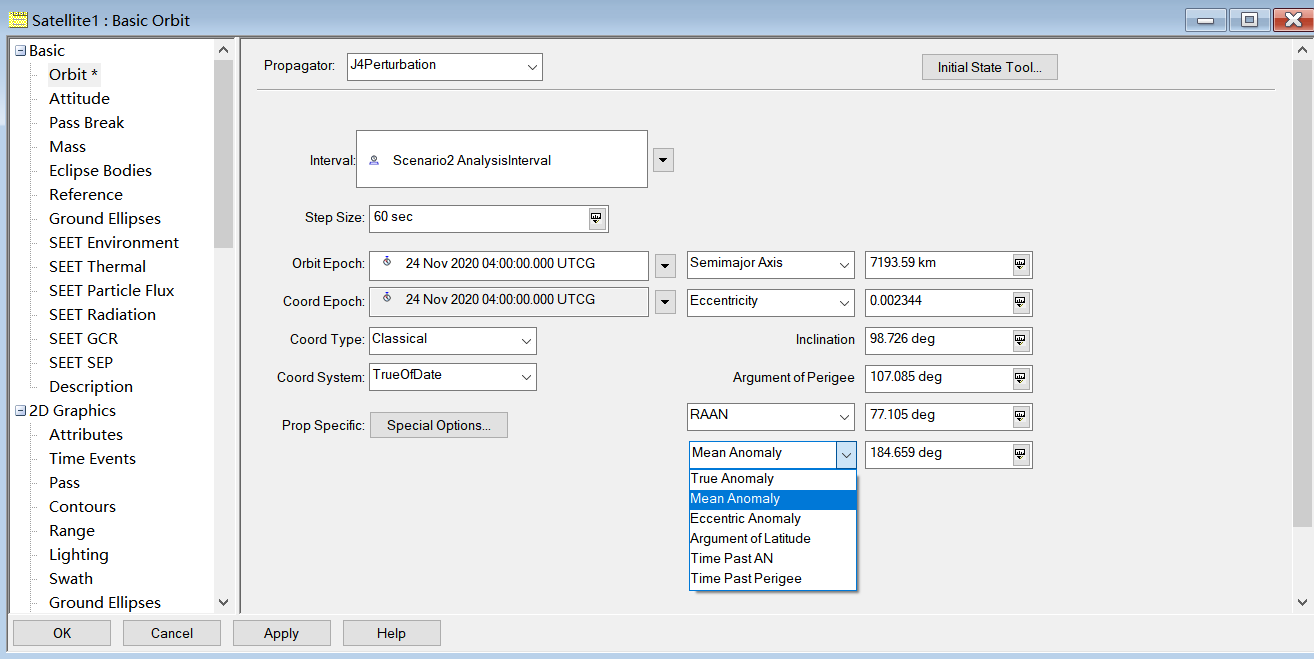


Figure 3-1-7 Satellite1 Setting

1. **Animation presentation**.Figure8 shows the timeline view of this scenario. From 24 Nov 2020 04:00:00 to 25 Nov 2020 04:00:00.A button in the timeline can be adjusted to check satellite’s state in a time point. To show the animation presentation, just click the start button in the top bar. There are other buttons to adjust the animation. 

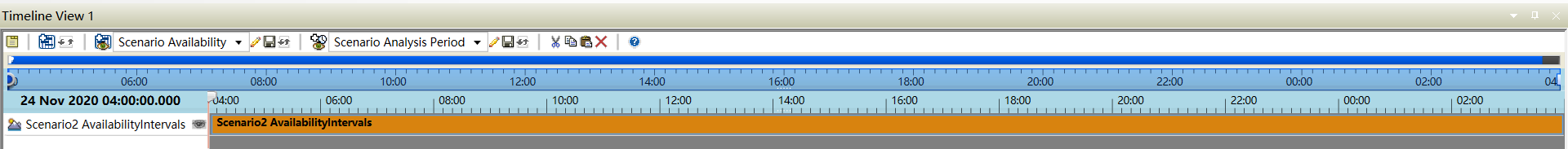


Figure 3-1-8 Timeline View

1. **Analyze, generate and export reports.** First, click the ‘analysis’ button in the top tool bar. Then click ‘Access’, a interface like figure 3-1-9 display. Because the aim is analyze the access for satellite, satellite1 should be select using ‘Select Object’. Then click Beijing let the software which place should be included this time’s analysis. Next, click compute to analyze. Then click the ‘Access’ button in the Reports area. A report will display in your interface like figure 3-1-10 shows. In this report, the total time period, the time of satellite access, and each time windows’ time period are displayed successfully. IN this step, the scientific data are obtained successfully. To export the report, just click ’save’ button and save the file in your computer.

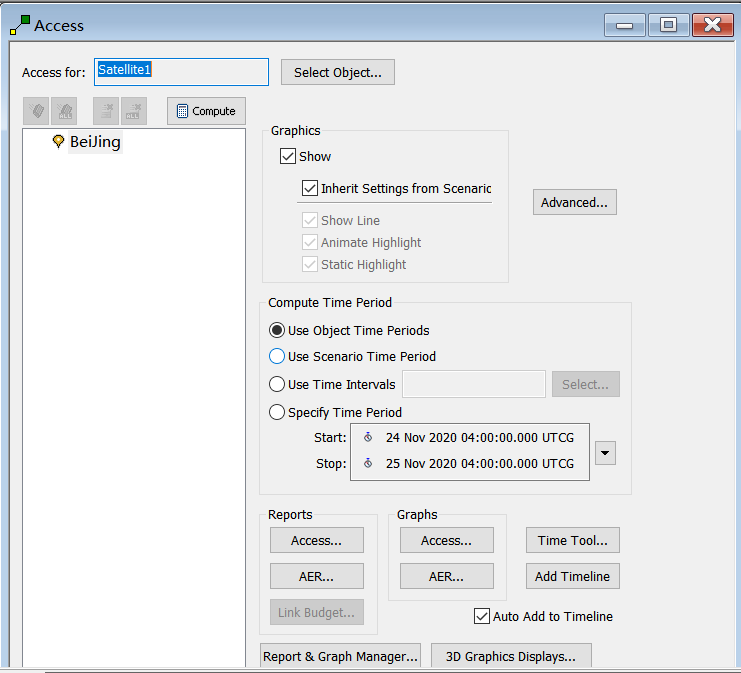


Figure 3-1-9 Access analysis

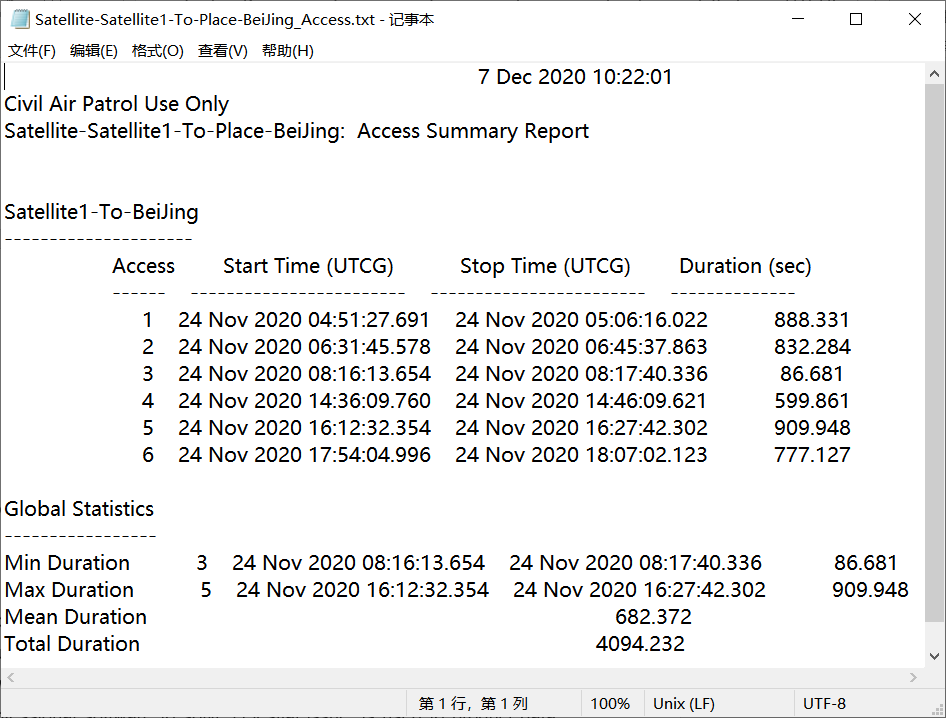


Figure 3-1-10 Access analysis report

**3.2 数据处理**

3.2.1时间窗口的安排

(1).地点选择：

一共选择五个城市： 北京 成都 广州 上海 重庆

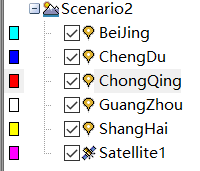


Figure 3-2-1 STK场景设置截图

(2)时间段选择(Time span):

开始时间： 24 Nov 2020 04:00:00.000 UTCG

结束时间： 24 Nov 2020 06:00:00.000 UTCG

(3)单个卫星对于每一个地点的可见窗口：

1. 北京 24 Nov 2020 04:51:27.691 24 Nov 2020 05:06:16.022 888

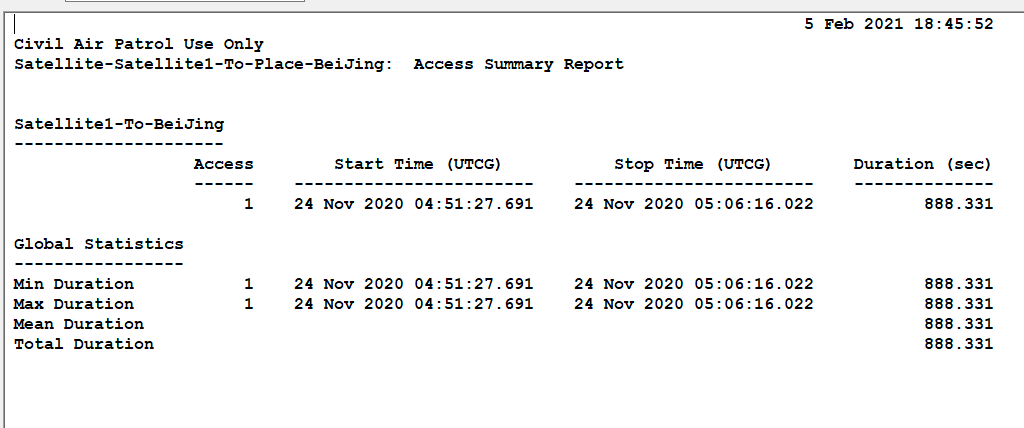


Figure 3-2-2 北京的相关时间窗口

1. 成都 24 Nov 2020 04:55:34.080 24 Nov 2020 05:08:02.459 748

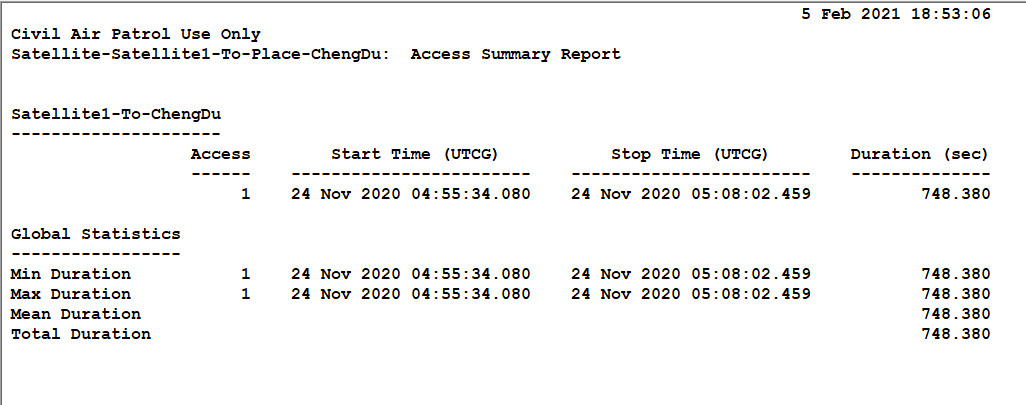


Figure 3-2-3 成都的相关时间窗口

1. 重庆 24 Nov 2020 04:55:27.417 24 Nov 2020 05:08:42.592 795

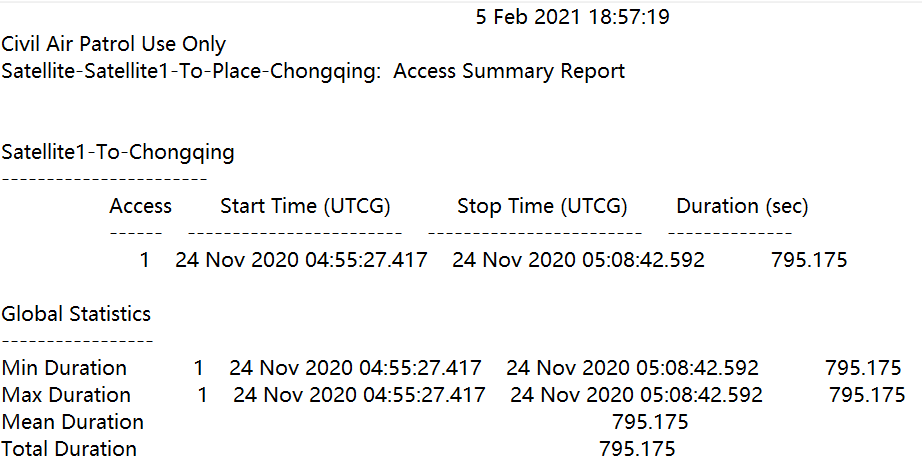


Figure 3-2-4 重庆的相关时间窗口

1. 上海 24 Nov 2020 04:53:26.693 24 Nov 2020 05:08:39.398 913

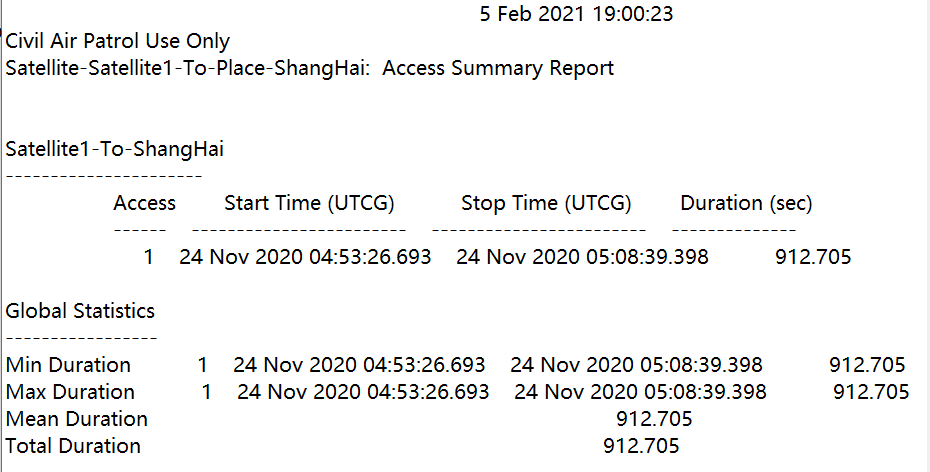


Figure 3-2-5 上海的相关时间窗口

1. 广州 24 Nov 2020 04:56:21.117 24 Nov 2020 05:11:04.047 883

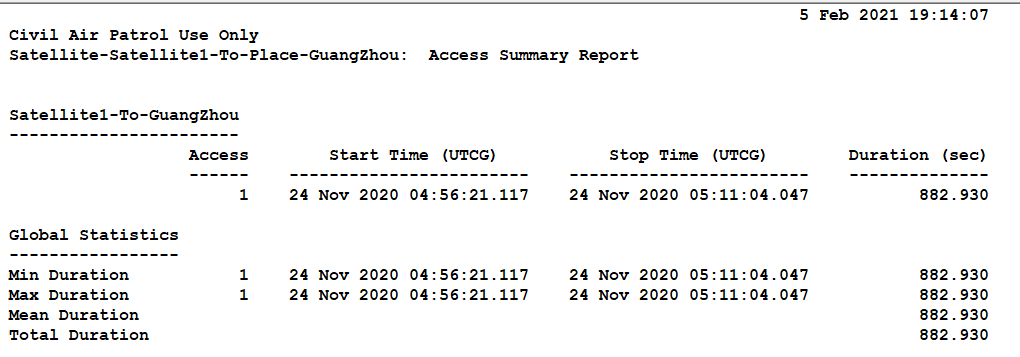


Figure 3-2-6 广州的相关时间窗口

Table 3-2-1 选择时间段内各个地点对应的时间窗口

|  |  |  |  |
| --- | --- | --- | --- |
| 地点 | 起始时间 | 结束时间 | 时长 |
| 北京 | 24 Nov 2020 04:51:27.691 | 24 Nov 2020 05:06:16.022 | 888 |
| 成都 | 24 Nov 2020 04:55:34.080 | 24 Nov 2020 05:08:02.459 | 748 |
| 重庆 | 24 Nov 2020 04:55:27.417 | 24 Nov 2020 05:08:42.592 | 795 |
| 上海 | 24 Nov 2020 04:53:26.693 | 24 Nov 2020 05:08:39.398 | 913 |
| 广州 | 24 Nov 2020 04:56:21.117 | 24 Nov 2020 05:11:04.047 | 883 |

从上表可以得出，总时间窗口的时长约为1200s.

24 Nov 2020 05:11:04.047-24 Nov 2020 04:51:27.6911200(s)

3.2.2任务的设定与冲突测算

(1)为了保证数据是大量而且有效的以更好的适应与验证调度网络的优势，下表给出了本项目关于任务的相关参数设定。

Table 3-2-2 本项目关于任务的相关参数设定

|  |  |
| --- | --- |
| 参数 | 值 |
| 任务的总个数(n) | 500 |
| 每个任务的奖励的取值范围() | 1-10 |
| 每个任务的容量消耗() | 5 |
| 卫星可接受的最大容量() | 1750 |
| 时间间隔(T) | 24 Nov 2020 05:11:04.047  24 Nov 2020 04:51:27.691(1200s) |
| 每个任务的执行时间的设定范围 | 10-15(s) |
| 每次随机产生任务的平均冲突占比 | 0.2 |

(2)产生任务样例：

在有关任务处理的程序设计中，通过调控总时间，任务的个数以及每个任务执行时间的范围来调控冲突任务占比，在当前的参数设置下，在反复调用产生任务的程序100次之后，计算每次冲突任务的平均占比，约为0.2。

冲突的定义：即前一个任务执行结束的时间超过了后一个任务的开始执行时间。下图为一次产生任务程序中500个任务中冲突任务样例,每个任务的最后两个参数分别代表开始和终止时间:

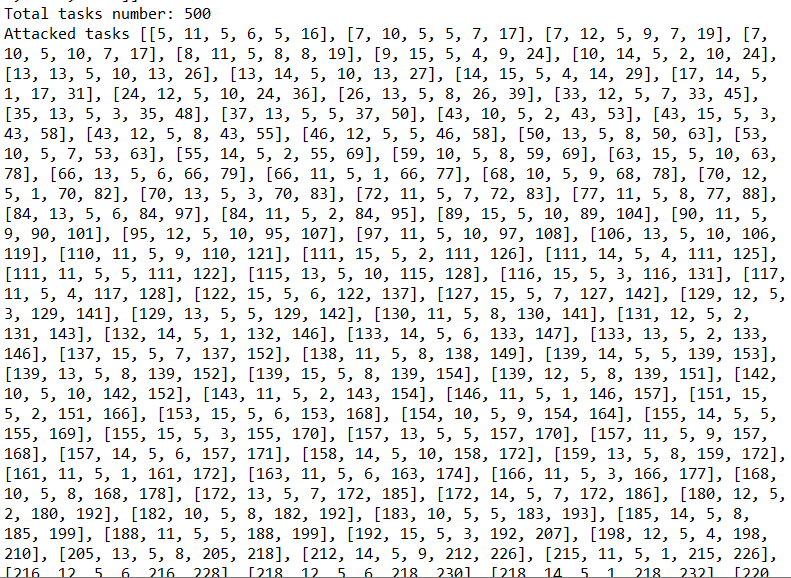


Figure 3-2-7 任务产生样例

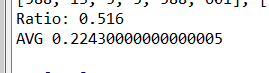


Figure 3-2-8 一百次生成任务之后的平均冲突占比

Ratio 为本次产生任务的过程中冲突任务占比，而AVG表示运行产生任务程序100次之后，整个冲突任务的平均占比。

**3.3 问题建模**

3.3.1问题建模

1.1变量定义

(1)调度时间段

该问题是在一个规定的时间段内完成的，根据上一部分对STK的使用，我们已经获取了科学的数据。为了方面操作，我们首先对数据进行处理，然后可以获得模拟一个规定调度时间段为 =[0,T]，假设这个时间段内涉及到的所有要素都在这个时间段内，且不受时间段外的影响。

(2)观测需求

在卫星在线调度的过程中，由于任务是动态到达，所以定义任务的到达时刻为，其中，为本时间段内第i个任务的到达时刻，n为到达任务的总数目。由于每一个观测任务都需消耗卫星一定的资源，所以定义为第i个任务的容量消耗。同时由于每一个任务都有自己的收益，定义其为为第i个任务的收益。用 来分别表示第i个任务的开始观测时间和结束观测时间。定义为第i个任务的最短观测时间。

(3)卫星

在本项目的环境下，卫星通过消耗一定资源来完成观测需求，同时获得相应收益。定义该卫星的最大容量为：。同时，考虑到同一个卫星在执行两个相邻任务时需要间隔时间，所以定义两个相邻任务之间最短的间隔时间为 。

(4)时间窗口的处理

（由于英文论文中提到的时间窗口是通过STK获得的），并未像中文论文中又定义一个可观测的时间段与开始和结束观测时间段取交集，而在本项目中我们通过STK找到了卫星一天内观测北京的可观测时间段，我们选取了第一个可观测时间段作为T，即为一个大的时间窗口，意味着这个时间段内所有到达的任务都是可观测的。

1.2约束条件的说明

1. 卫星的约束

定义t时刻第i个任务是否被观测：

变量表示：当前任务是否被观测：

0

=

1 且i任务被接受

一个卫星在同一时刻只能观测一个目标即：



1. 连续两个任务的约束

对于两个任务a,b,如果它们开始和结束观测的时间分别为和，且，则

1. 卫星的容量约束

卫星观测任务的总容量不能超过卫星的最大容量：



1. 空余时间间隔

卫星的空闲时间段,假定在t=时刻，卫星的任务负载如图一所示，图中未接受任务的时间段为空闲时间段，记为[,]。且应当满足以下条件:



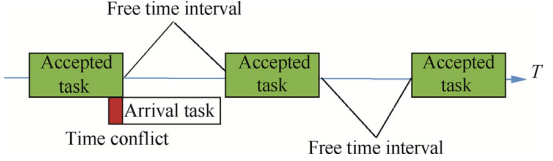


图3-3-1 可以接受观测任务的空闲时间段

1.3 DQN框架中涉及到的相关变量以及数据结构的声明

(1)任务

本项目中的任务是由程序自动产生的，都以一个列表的形式存起来，定义为Task。其中，每一个任务又是一个列表,包含6个变量分别是任务的到达时间，任务的执行时间，任务的容量，任务的收益，以及任务开始观测的时间和结束观测的时间。其数据结构为：

Task=[[,,,],...,[,,,]]

图3-3-2展示了一个任务输出的实例：

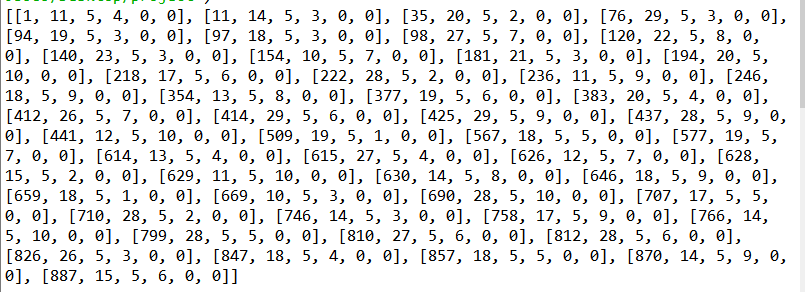


图3-3-2 程序生成的五十个任务

1. 状态

**状态：**状态由四个变量组成分别是当前任务的到达时间，当前任务的收益，剩余卫星容量占比，以及剩余空余时间容量占比。在更新函数中，这些数据都会经过归一化处理之后放到一个列表里面，即一个列表有四个变量，分别是这四个数据归一化后的数据。而这个也将作为DQN神经网络的输入。

变量表示：

当前任务到达时间占比：

当前任务收益占比：

当前状态卫星容量占比：

当前状态剩余空闲时间占比：

当前状态的数据结构表示：

： [,,,]

1. 奖励

第i个任务的奖励由表示，在在线调度的过程中，如果卫星选择接受了这个任务 这个变量就会加上当前任务的收益，反之，则不会。卫星的总奖励用W表示。

(m为接受的任务总数）

1. 环境

类比与背包问题，我声明了一个名为accepted\_tasks\_list的列表，类似于一个空背包，如果接受了一个任务就把任务的序号存到这个里面，在最终执行结束后就可以指导在这个时间窗口内卫星一共接受了哪个任务。比如：

accepted\_tasks\_list=[0,1,2](表示任务1，2，3被卫星接受）

1. 动作和决策

表示对第i个任务的决策，表示采用的决策策略。在本项目中，动作即用1和0来表示：0代表接受当前任务，用1来代表不接受当前任务。

0 如果任务i被接受

=

1 如果任务i不被接受

1. 结束条件

本问题有三个截止条件：

1. 如果容量超过了最大限定容量。用Store(t)来表示t时刻来表示卫星剩余的容量，则终止条件可以表示为：

Store(t)=0

2.如果对于一个任务没有空余时间间隔来观测它。

=Ø

3.时间到达最终调度时间。

t=T

1.4应用DQN框架的基本处理思路

(1)基本流程

图3-3-3展示了强化学习智能体于环境交互的过程，在DQN中，类比与q-learning，DQN采用神经网络来代替Q表，在DQN的网络架构中，首先通过神经网络获取当前状态可以产生得所有动作的Q值，然后根据e-贪婪算法确定动作，之后，环境文件会根据当前动作对状态进行更新，同时会返回下一步的状态，奖励，完成状态等等，这是一个step。之后在DQN框架中中会将这些更新之后变量作为经验添加到经验池中并完成神经网络参数的更新然后进行下一步的循环。

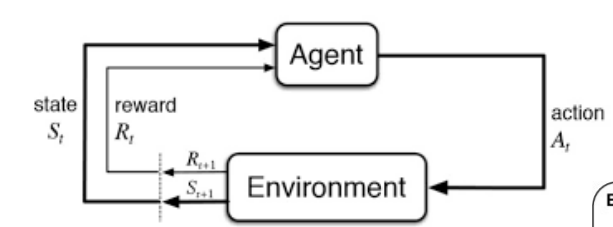


图3-3-3 智能体与环境的交互过程

(2)DQN网络的输入和输出

DQN的输入：当前的状态： [,,,] 作为DQN网络的输入。

DQN的输出： 根据当前状态采取不同动作所获得的Q值，在本项目中即为与，即接受任务所得到的Q值和不接受任务所获得的Q值。

(3)更新神经网络的参数

在DQN中，神经网络输出的到的Q值都会被用来对神经网络的参数进行更新，更新的基本步骤为：

1.结合神经网络的输出和经验回放来更新Q值：



2.计算损失函数的值：



3.通过反向传播和梯度下降法来更新神经网络的连接参数。

1.5 应用A3C框架的基本处理思路。

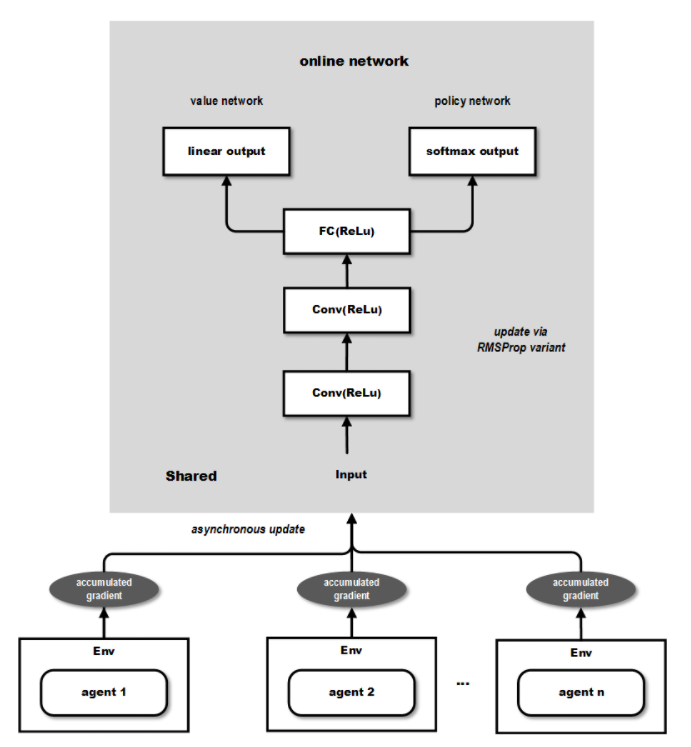


图3-3-4 A3C框架基本处理流程

1. 基本流程

A3C框架应用了多线程的方法，同时在多个环境里面分别和环境进行交互学习，每一个线程吧学习成果汇总起来并整理到一个公共的地方，定期从公共的地方把学习成果拿出来并指导后边的交互，图四展示了A3C框架的基本处理流程，首先存在一个公共的神经网络模型，这个神经网络包括Actor网络和Critic网络两部分的功能。下面有n个agent线程，每个线程里有和公共的神经网络一样的网络结构，每个线程会独立的和环境进行交互得到经验数据，这些线程之间互不干扰，独立运行。

每个线程和环境交互到一定量的数据后，就计算在自己线程里的神经网络损失函数的梯度，但是这些梯度却并不更新自己线程里的神经网络，而是去更新公共的神经网络。也就是n个线程会独立的使用累积的梯度分别更新公共部分的神经网络模型参数。每隔一段时间，线程会将自己的神经网络的参数更新为公共神经网络的参数，进而指导后面的环境交互。

1. A3C网络的输入和输出。

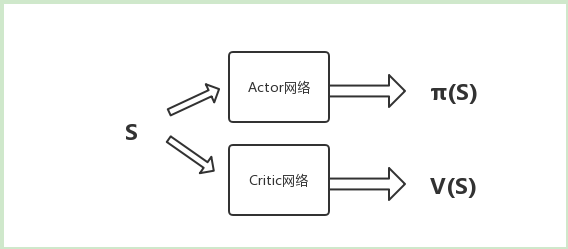


图3-3-5 A3C的输入和输出

**输入：**在本项目中，通DQN的输入一样，A3C网络的输入为当前的状态： [,,,] 作为A3C网络的输入。

**输出：** A3C 的输出一共包括两部分，一个对总期望收益的估计值 用表示， 表示 Critic网络的参数，另一个是，包括在当前策略下接受当前任务的概率和不接受当前任务的概率。或者说在当前状态下采取接受或不接受的动作的概率遵循有参数确定的策略。

在本项目的A3C神经网络设计中，Actor网络和Critic网络分别共享神经网络的前端。

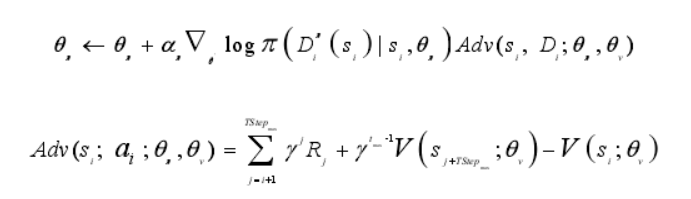
1. 参数更新

本项目中采用蒙特卡洛策略更新方法,更新公式如下：

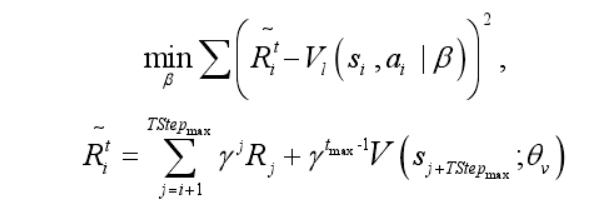


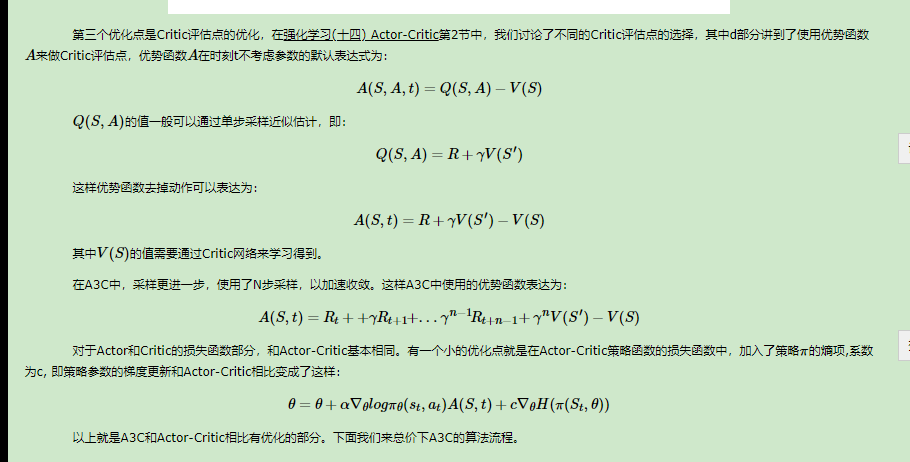


对于策略网络(Actor)更新方法如下：



对于值函数的更新方式如下：





DDPG(D（Deep）+D（Deterministic ）+ PG(Policy Gradient)):(感觉当前对这个框架理解还不太够，等后边完成之后会具体补充）

1. 基本流程：

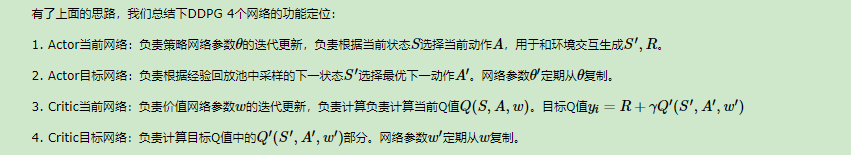


表3-3-1 本项目中所涉及到的相关变量及其表示

|  |  |
| --- | --- |
| 第i个任务的序号 | i |
| 第i个任务的到达时间 |  |
| 第i个任务的收益 |  |
| 第i个任务的开始观测时间和结束观测时间 |  |
| 第i个任务的存储消耗： |  |
| 第i个状态表示 |  |
| 调度结束时刻 |  |
| 本环境中第i个任务的最短执行时间 |  |
| 在时间t 时的空余时间间隔 |  |
| t时刻未接受任务的第k个空闲时段 | [,] |
| 最短的转移时间 |  |
| 卫星的最大容量 |  |
| t时刻时卫星剩余的容量 |  |
| 第k个开始和结束的时间间隔 |  |

1. 效果分析：

为了验证深度强化学习是解决该卫星调度问题的一种有效的解决方案，本项目采用了DQN 框架对数据进行训练，表3-3-2展示了DQN网络训练中涉及到的相关参数及其取值的设定。

|  |  |
| --- | --- |
| 参数 | 值 |
| 任务的总个数(n) | 50 |
| 每个任务的奖励的取值范围() | 1-10 |
| 每个任务的容量消耗() | 5 |
| 卫星可接受的最大容量() | 175 |
| 时间间隔(T) | 24 Nov 2020 04:51:27  24 Nov 2020 05:06:16 (888s) |
| 学习率() | 0.0068 |
| 时间折扣率() | 0.99 |
| 批量大小 | 80 |
| 训练次数 | 100 |

图3-3-2： DQN网络训练涉及到的相关参数

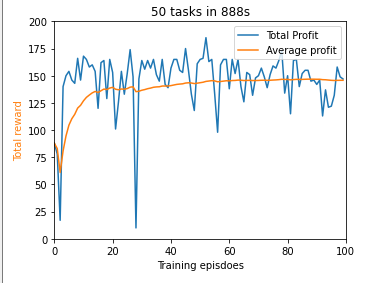
考虑到评估网络训练好坏的两个重要因素为任务的总收益和接受任务的总个数，所以在每次训练之后，相关的数据都会被分别记录到不同的折线图上，其中一个折线图主要是用来展示总收益在整个训练过程中的变化。另一个折线图则主要用来展示接受任务总个数在整个训练过程中的变化，同时为了更好的展示训练的结果，每一次训练的到的总收益和总任务之比，即平均收益也会被记录到一个折线图中。

**3.4 实验结果展示（当前暂时还是用的50个任务的数据与毕设结果比较，后边两天等新的数据跑出来之后我会把这部分的内容更新）。**

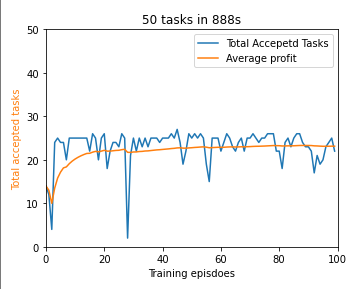
下面是在不同条件下用DQN框架训练的得到的相关结果。(后续新的数据产生之后，会换上截图和陈述比较）

(1)100训练次数，888秒内50个任务

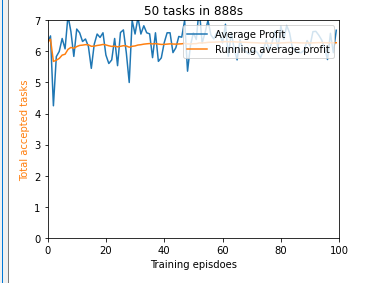
总收益



接受任务的总个数



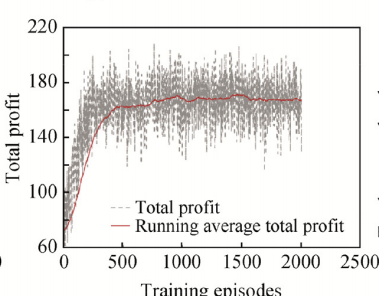
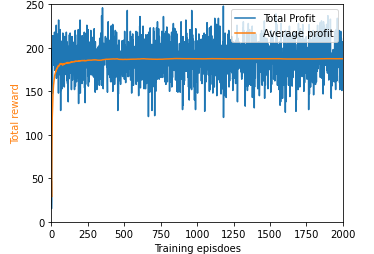
平均收益



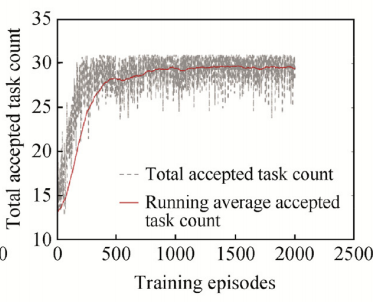
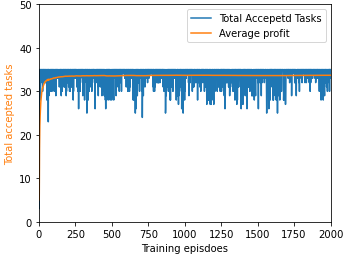
在训练一百次之后可以看到任务的总收益稳定在140以上但仍有一定的上升趋势，接受任务的总数稳定在20以上，而平均收益稳定在6以上。

(2)1200秒内50个任务

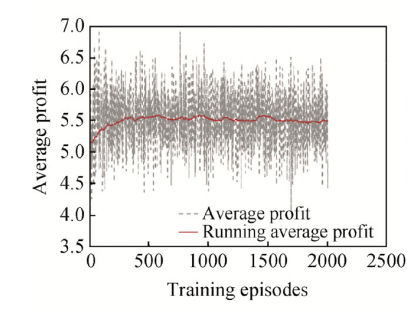
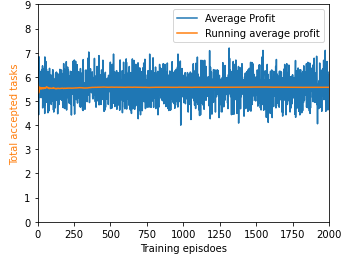
总收益



接受任务的总个数

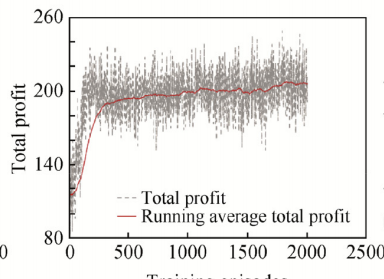
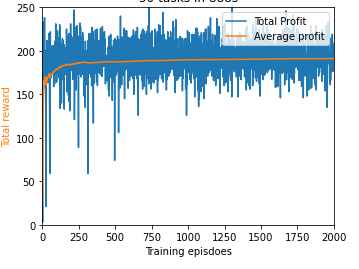


平均收益

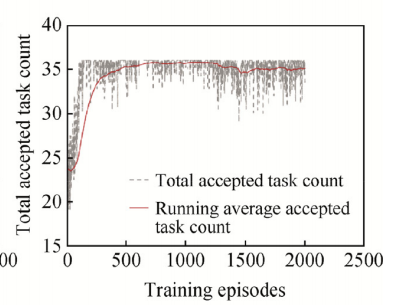
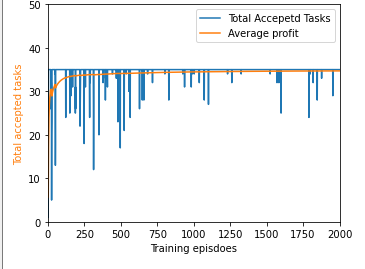


(3)3600秒内50个任务

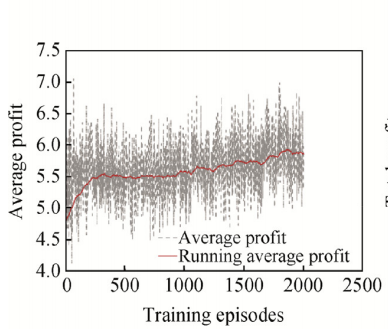
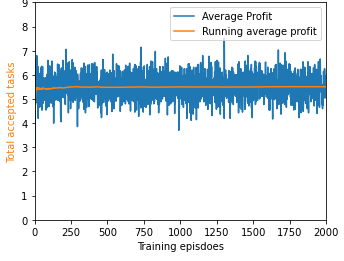
总收益



接受任务的总个数



平均收益



**3.5 调度算法比较与评估**

3.5.1：数据的选择与算法评判标准

为了凸显神经网络在卫星调度问题中的优势，如何选择数据就变得十分重要，本项目尝试了以下几种情况，每一种随机产生的任务数据都会有一定的冲突比例，以保证数据的准确性和对比的可信度。下面给了本项目在不同情况下采用DQN框架的输出结果。

**任务冲突占比：**对产生任务的程序进行处理，首先通过程序产生多组任务，每一组任务都会统计冲突任务的个数，之后除以该组任务的总数即为该组任务的冲突占比，最后求取所有组任务的平均冲突占比，用该数据来衡量数据的可用性。

不同算法的评估标准：

1. Average Profit(AP)：记录所有循环循环得到的收益之和P，运行结束后除以总循环的个数N。

计算公式为： 

1. Average total accepted tasks(AT)：记录所有循环结束之后接受任务的总数之和M，运行结束后除以总循环的个数N。

计算公式为： 

1. Running average profit(RAP)：对于每一次循环的的平均收益的计算公式为每一次循环接受之后的收益P除以对应的接受任务个数AT即，之后再求取所有循环结束之后的平均RAP值：

 （i表示第i次循环）

1. Average Response Time: (ART):记录所有循环结束之后的响应时间之和time，运行结束后除以总循环的个数N

计算公式为：

3.5.3不同情景下的数据记录

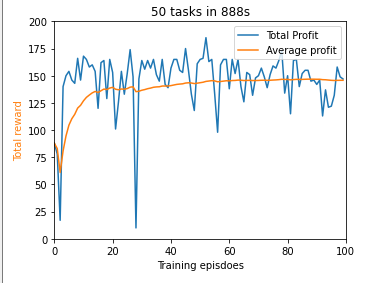
50 个任务888秒：

1. 参数设定

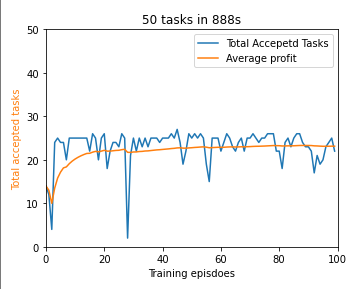
|  |  |
| --- | --- |
| 学习率() | 0.0068 |
| 时间折扣率() | 0.99 |
| 批量大小 | 80 |
| 训练次数 | 100 |
| 任务总数 | 50 |
| 任务平均冲突比 | 1% |
| 时间窗口 | 888 |
| 执行时间范围 | 10-15 |
| 最大容量限制 | 175 |

1. 结果展示

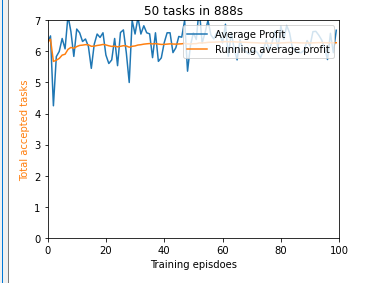
总收益



接受任务的总个数



平均收益



数据结果

|  |  |  |  |
| --- | --- | --- | --- |
| Average Profit | Average total accepted tasks | Running average profit | Average Response Time |
| 145.52 | 25.98 | 5.618 | 0.179 |

在当前的参数设定下，平均收益稳定在146左右，而平均接受任务稳定在26左右，收益与接受任务数量之比大概为5.6，平均响应时间的为0.179。

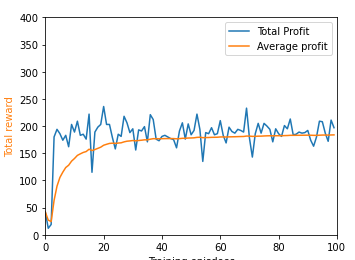
50 个任务1200秒：

1. 参数设定

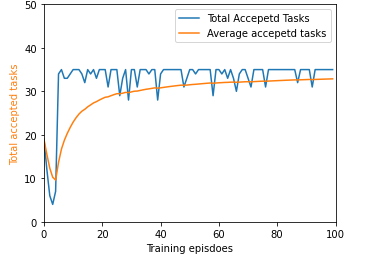
|  |  |
| --- | --- |
| 学习率() | 0.0068 |
| 时间折扣率() | 0.99 |
| 批量大小 | 80 |
| 训练次数 | 100 |
| 任务总数 | 50 |
| 任务平均冲突比 | 1% |
| 时间窗口 | 888 |
| 执行时间范围 | 10-15 |
| 最大容量限制 | 175 |

1. 结果展示

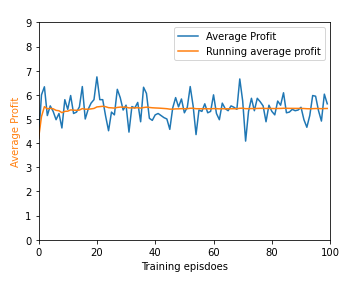
总收益



接受任务的总个数



平均收益



|  |  |  |  |
| --- | --- | --- | --- |
| Average Profit | Average total accepted tasks | Running average profit | Average Response Time |
| 181.57 | 32.77 | 5.525 | 0.171 |

选择50个任务在888s内执行时：在训练一百次之后可以看到任务的总收益稳定在140以上但仍有一定的上升趋势，接受任务的总数稳定在20以上，而平均收益稳定在6以上。当时间提升到1200秒可以明显看到任务数量已经增加到几乎每次35个达到了临界值，而且总收益相比于前者也有所提高。

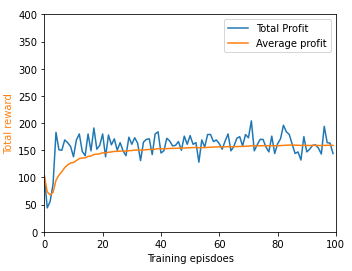
100个任务1200秒

1. 参数设定

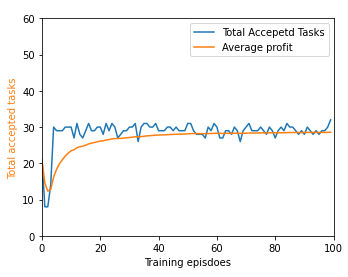
|  |  |
| --- | --- |
| 学习率() | 0.0028 |
| 时间折扣率() | 0.99 |
| 批量大小 | 80 |
| 训练次数 | 100 |
| 任务总数 | 100 |
| 任务平均冲突比 | 10% |
| 时间窗口 | 1200 |
| 执行时间范围 | 20-40 |
| 最大容量限制 | 250 |

1. 结果展示

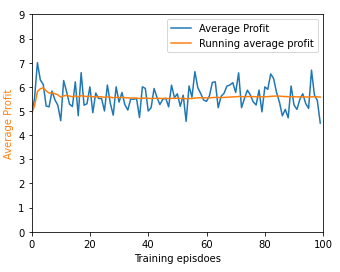
总收益



接受任务的总个数



平均收益



|  |  |  |  |
| --- | --- | --- | --- |
| Average Profit | Average total accepted tasks | Running average profit | Average Response Time |
| 158.26 | 28.45 | 5.554 | 0.38 |

在当前的参数设定下，平均收益稳定在158左右，而平均接受任务稳定在28左右，收益与接受任务数量之比大概为5.5，平均响应时间的为0.38。

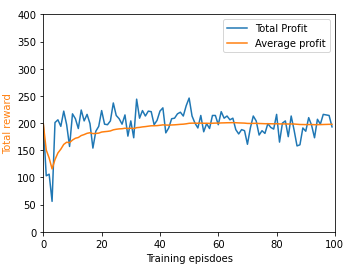
200个任务1200秒

(1)参数设定

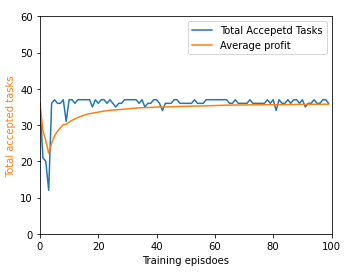
|  |  |
| --- | --- |
| 学习率() | 0.0008 |
| 时间折扣率() | 0.99 |
| 批量大小 | 150 |
| 训练次数 | 100 |
| 任务总数 | 200 |
| 任务平均冲突比 | 18% |
| 时间窗口 | 1200 |
| 执行时间范围 | 20-25 |
| 最大容量限制 | 300 |

1. 结果展示

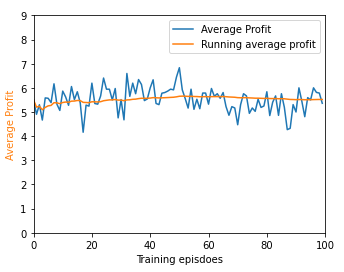
总收益



接受任务的总个数



平均收益



|  |  |  |  |
| --- | --- | --- | --- |
| Average Profit | Average total accepted tasks | Running average profit | Average Response Time |
| 197.42 | 36.29 | 5.4 | 0.938 |

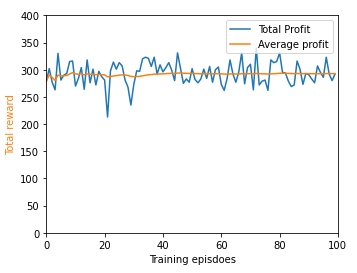
在当前的参数设定下，平均收益稳定在197左右，而平均接受任务稳定在36左右，收益与接受任务数量之比大概为5.4，平均响应时间的为0.938。

500个任务1200秒

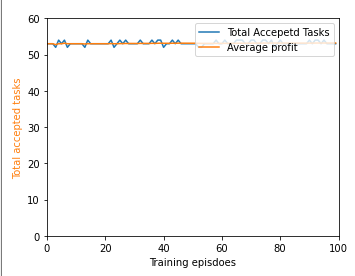
(1)参数设定

|  |  |
| --- | --- |
| 学习率() | 0.0068 |
| 时间折扣率() | 0.99 |
| 批量大小 | 200 |
| 训练次数 | 100 |
| 任务总数 | 500 |
| 任务平均冲突比 | 20% |
| 时间窗口 | 1200 |
| 执行时间范围 | 10-15 |
| 最大容量限制 | 300 |

(2)结果展示



接受任务的总个数



平均收益



|  |  |  |  |
| --- | --- | --- | --- |
| Average Profit | Average total accepted tasks | Running average profit | Average Response Time |
| 293.57 | 53.19 | 5.518 | 2.044 |

在当前的参数设定下，平均收益稳定在293左右，而平均接受任务稳定在53左右，收益与接受任务数量之比大概为5.518，平均响应时间的为2.044。

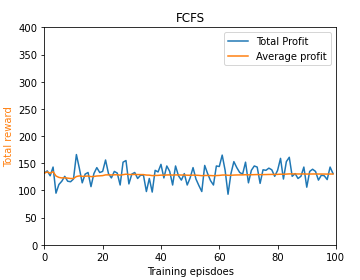
通过比较多组不同任务和时间的设定，本项目选择使用1200s, 100个任务作为之后进行不同调度算法比较的背景数据，选择该数据背景的原因有两个：一，相对于五十个任务的背景环境数据量增多，更加适合网络框架来跑。二，从时间成本上来看，1200s一百个任务的相比于200,500个任务的响应时间要少。

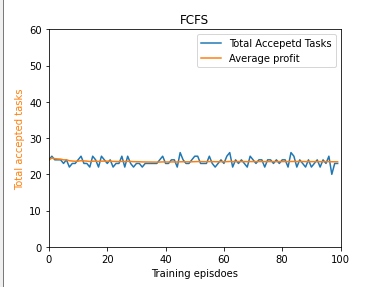
三调度算法比较(T=1200s,n=100）

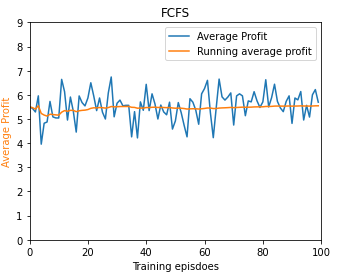
根据上一个部分的实验结果，本项目采用100个任务，1200s的环境背景来实验不同调度算法在该数据环境的实验结果，主要选择比较的调度算法有：FCFS, RP,GA 和DITH，为了尽可能保证数据的一致性，本项目采用相同的产生任务的程序随机产生100组任务然后分别在不同的调度算法中运行，最后对每个评估标准取平均值，然后再进行比较。

1. FCFS调度算法

定义：FCFS，即first come first service ,严格按照任务到达的时间来执行，到达时间在前的任务先被执行，到达时间在后的任务在前一个任务完成之后依次执行,在本项目用考虑到任务是动态到达的，所以要在原本FCFS的基础上加入对于时间冲突的相关处理，下图显示了FCFS的相关运行结果。



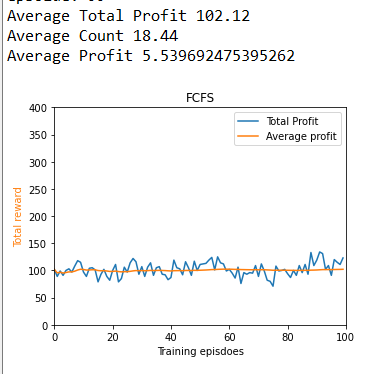


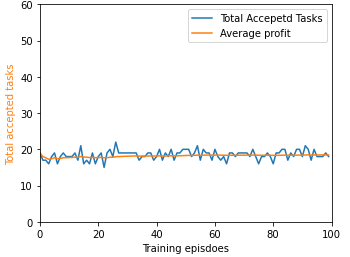


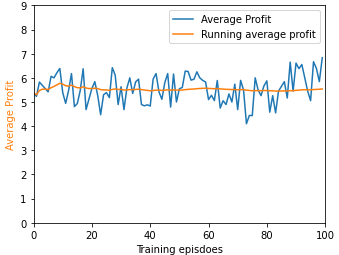
|  |  |  |  |
| --- | --- | --- | --- |
| Average Profit | Average total accepted tasks | Running average profit | Average Response Time |
| 130.18 | 23.46 | 5.5467 | 0.0188 |

1. 随机调度算法（Random Pick)

定义：RP， 即 random pick, 即根据任务的动态到达，随机选择接受当前任务还是拒绝当前任务，同时可以控制当前任务接受和不接受的概率：（下图展示了在接受和不接受的概率均为0.5）下图显示了RP的相关运行结果。







|  |  |  |  |
| --- | --- | --- | --- |
| Average Profit | Average total accepted tasks | Running average profit | Average Response Time |
| 102.12 | 18.44 | 5.55397 | 0.010 |

1. GA 遗传调度算法

定义： GA，即遗传调度算法，类比于达尔文的进化理论，通过将一个问题转化为‘数字化’编码的形式，再对其进行处理.GA算法的基本步骤如下：首先随机初始化一个种群，之后通过解码程序得到相应种群的价值，然后通过‘优胜劣汰’重新组合种群，之后对编码进行变异然后反复重复循环，最终得出最优解。

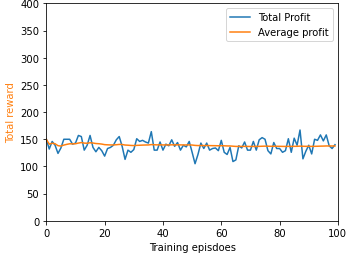
下面本次GA算法的相关参数设定以及实验结果

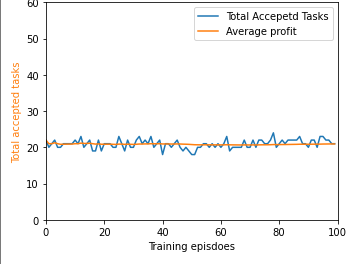
GA算法相关参数设定

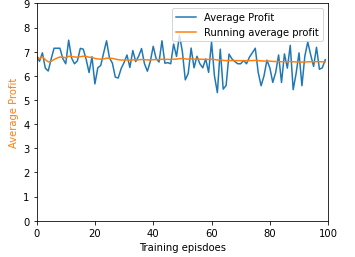
|  |  |
| --- | --- |
| 任务总数 | 50 |
| 任务平均冲突比 | 1% |
| 时间窗口 | 888 |
| 执行时间范围 | 10-15 |
| 迭代次数 | 1000 |
| 交叉概率 | 0.25 |
| 变异概率 | 0.02 |
| 种群大小 | 10 |
| 迭代次数 | 200 |

GA算法实验结果

|  |  |  |  |
| --- | --- | --- | --- |
| Average Profit | Average total accepted tasks | Running average profit | Average Response Time |
| 137.65 | 20.93 | 6.582 | 0.60 |







1. DITH调度算法 (Dynamic Insert Task Heuristic)

定义：

(5)不同调度算法的性能比较

下表给出了不同调度算法的实验结果：

不同调度算法的性能比较

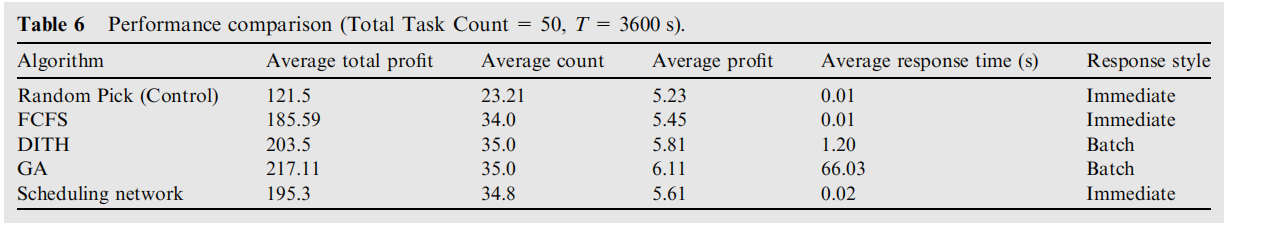
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Average Profit | Average total accepted tasks | Running average profit | Average Response Time |
| RP | 102.12 | 18.44 | 5.55397 | 0.010 |
| FCFS | 130.18 | 23.46 | 5.5467 | 0.0188 |
| GA | 137.65 | 20.93 | 6.582 | 0.60 |
| DQN | 158.26 | 28.45 | 5.554 | 0.38 |
| DITH |  |  |  |  |

通过上表总结的数据可以发现，DQN网络调度算法的性能最好，不仅体现在AP上，平均接受任务和运行中的平均收益都相对较好，响应时间也在可接受的范围之内。其次是GA，虽然GA的运行时间相对较长，但是总会得到相对高的平均收益。RP由于是采用概率随机选择任务的所以各方面的性能相对较差。

**下一步的计划：**

0.分析实验结果

1. 记录任务冲突的数据：任务数量，存在冲突的任务数量，冲突比(STK产生数据）
2. FCFS原理，python类似代码，改成输入输出，同时查看其它的调度算法。 四个指标计算公式



1. A3C和DDPG。
2. 在网上看A3C和DDPG的框架调试。
3. 扩展到多卫星。输入要改 产生数据
4. 增大任务量

初步计划从本周一开始两周之内把DQN相关的结果跑出来并记入到报告里面，然后后边一周再去看A3C和DDPG

# References

1. [1] Li Z, Li X. Current status and prospect of imaging satellite task dynamic scheduling methods. 8th international conference on intelligent human-machine systems and cybernetics; 2016 Sep 11–12; Hangzhou, China. Piscataway, IEEE Press; 2016. p. 436–9. [↑](#endnote-ref-0)
2. [2] Zhao,H. 2020. Deep reinforcement Learning (A3C) network. [Online]. [Accessed 26 March 2020]. Available from: <https://blog.csdn.net/weixin_43283397/article/details/105120623> [↑](#endnote-ref-1)
3. [3] Wang M,Dai G,Song Z, Peng L.2016.Research on tasks Planning and Scheduling algorithm of imaging Satellite. p.93 [↑](#endnote-ref-2)
4. [4] Wang M,Dai G,Song Z, Peng L.2016.Research on tasks Planning and Scheduling algorithm of imaging Satellite. p.94 [↑](#endnote-ref-3)