МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2**

**по дисциплине**  
 **«МНОГОАГЕНТНОЕ МОДЕЛИРОВАНИЕ»**

Выполнил студент группы 45/2                                       Т. Э. Айрапетов

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Курс    4

Отчет принял доктор физико-математических наук, профессор                                                                                       А.И. Миков

Краснодар

2024 г.

**Задание**: В системе есть n агентов. У каждого агента есть целевой набор патентов (может быть представлен как строка, глобальный идентификатор и тд.) At, который ему нужно собрать для реализации некоторой глобальной задачи. Его целью является сбор всех элементов своего целевого набора. В начале моделирования каждому агенту выдается непустое множество патентов As. Они могут быть как из целевого набора, так и не из него. В процессе моделирования агенты общаются между собой.

Целью коммуникации является обмен патентами. При коммуникации с другим агентом, текущий агент спрашивает есть ли у другого один из нужных ему патентов. Если нет, то раунд коммуникации завершается. Если у другого агента есть нужный патент, то он готов его обменять на один из нужных для себя. Если у текущего агента есть один из подходящих для второго патентов, обмен завершается. Если агент собрал свой целевой набор, то он может безвозмездно делиться своими патентами. Цикл моделирования заканчивается, когда у всех агентов собран целевой набор. В результате необходимо вывести список агентов с размером их целевого набора, количеством итераций и раундов коммуникаций, которые ему потребовались для его сбора.

Ремарки

Раундом коммуникации считается общение агента с агентом. Например, агент пообщался с агентом 2 и агентом 3. Тогда у агента 1 будет два раунда, а у второго и третьего по одному.

Для генерации начальных условий агента (целевой набор и множество патентов) можно сначала сгенерировать каждому агенту свой набор, затем объединить получившиеся множества в одно, и раздать его в случайном порядке. По итогам раздачи размеры множеств As каждого агента должны быть примерно одинаковыми.

**Решение**.

Для решения задания опишем класс Agent, представляющий собой агента, содержащий поля id (идентификатор), target (целевое множество патентов), count (кол-во целевых патентов), patents (множество выданных патентов), communications (кол-во коммуникаций), iters (количество итераций). Опишем также функцию generate\_target, создающую для каждого агента его целевой набор.

В основном теле программы наборы всех агентов объединяются в одно множество и распределяются почти поровну (остаток уходит к последнему агенту). Далее создается стек из агентов, которым необходимо заполнить их целевой набор, и в цикле, пока этот стек не пуст, из него берется очередной агент, который ищет среди остальных агентов нужные себе патенты. Согласно условию задания, если агенты могут обменяться патентами, либо второй агент уже собрал свой набор и теперь просто раздает оставшиеся патенты, текущий агент получает необходимый патент и, возможно, отдает ненужный. Если агент заполняет целевой набор (в данной реализации опустошает множество target), он выходит из стека агентов.

Стоит отметить, что возможны также ситуации, когда целевые наборы всех агентов не заполнены, но они не могут дальше обменяться или получить необходимые патенты. Это может произойти, например, если агентов в стеке больше 2, но попарно они не могут произвести обмен, так как не обладают интересующим для “оппонента” патентом. Одна из исключительных ситуаций проиллюстрирована на рисунке 1.

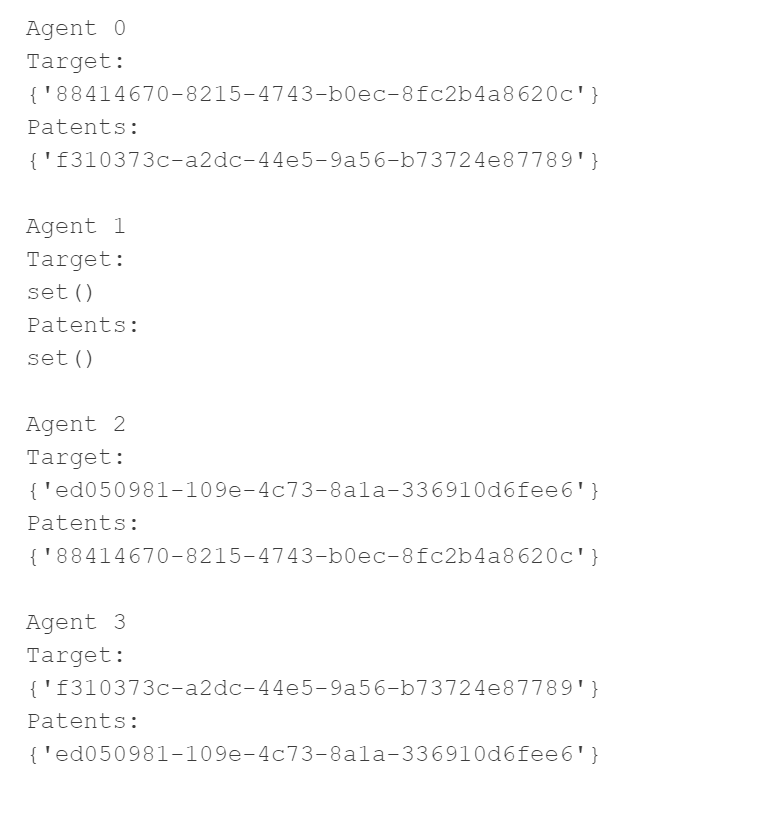


Рисунок 1 - Пример исключительного случая

В таких исключительных ситуациях возникает бесконечный цикл. Это поведение можно обработать, проверяя на равенство нулю количество обменов на одном прогоне по стеку.

Текст программы на языке Python:

*import* uuid

*import* random

n = 4

a, b = 3, 5

def generate\_target():

k = random.randint(a,b)

*return* set(str(uuid.uuid4()) *for* i *in* range(k)), k

class Agent:

def \_\_init\_\_(*self*, *id*):

*self*.id = *id*

*self*.target, *self*.count = generate\_target()

*self*.patents = None

*self*.communications = 0

*self*.iters = 0

agents = [Agent(i) *for* i *in* range(n)]

*# Создаем общее множество и распределяем по агентам*

t = set()

*for* i *in* agents:

t |= i.target

m, c = len(t)//n, len(t)%n

t = list(t)

*for* i *in* range(n):

*# выдача патентов*

agents[i].patents = set(t[i\*m: i\*m+m+c\*(i==n-1)])

temp = agents[i].target & agents[i].patents

*# сразу убираем целевые патенты из обеих множеств*

agents[i].target -= temp

agents[i].patents -= temp

print(f"Agent {agents[i].id}")

print("Target:")

print(agents[i].target)

print("Patents:")

print(agents[i].patents)

print()

stack = agents.copy()

i = 0

swaps = 0

*while* stack:

tek = stack[i]

*if* not tek.target:

stack.pop(i)

*continue*

tek.iters += 1

j = 0

*while* j < n and tek.target:

*if* tek.id != agents[j].id:

tek.communications += 1

agents[j].communications += 1

t1 = tek.target & agents[j].patents

*if* t1:

t1 = {t1.pop()}

*# если можно поменяться*

*if* not agents[j].target:

swaps += 1

tek.target -= t1

agents[j].patents -= t1

*else*:

t2 = agents[j].target & tek.patents

*# если можно просто взять*

*if* t2:

swaps += 1

t2 = {t2.pop()}

tek.target -= t1

tek.patents -= t2

agents[j].target -= t2

agents[j].patents -= t1

*if* not tek.target:

stack.pop(i)

i -= 1

j += 1

i+=1

*if* i == len(stack):

*if* not swaps:

*break*

i = 0

swaps = 0

*if* stack:

print("!!Во время выполнения не все собрали набор патентов!!\n")

*for* i *in* range(n):

print(f"Agent {agents[i].id}, Target size: {agents[i].count}")

print(f"Iters: {agents[i].iters}, Communications: {agents[i].communications}")

print()

Результаты тестов:

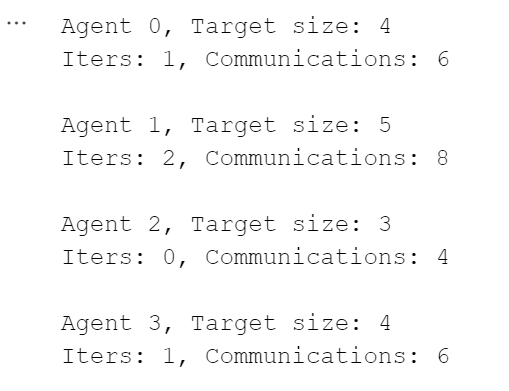


Рисунок 2 - Пример успешного завершения

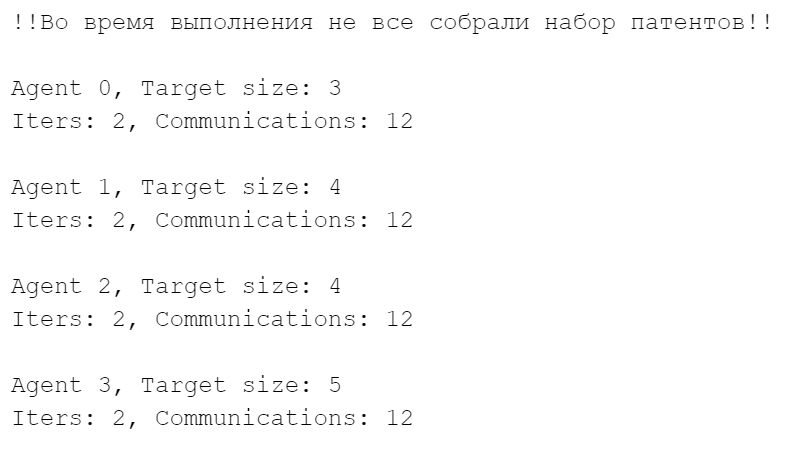


Рисунок 3 - Пример вывода при возникновении исключительного случая

**Вывод**:

На тестовых запусках программы и при анализе задания было выявлено, что некоторые задачи могут не иметь решения. Большую важность в успехе решения задачи играет её постановка и определение исходных данных. Так, при достаточно малом количестве агентов (до 4), решение может быть получено в большей части запусков. Однако при увеличении числа агентов мы почти никогда не сможем “договорить” наших агентов. Для достижения успеха в решении такой задачи можно добавить механизм случайной передачи некоторого патента от одного агента другому (чтобы сдвинуть положение с мертвой точки).