Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчёт по расчетному заданию №1**

Дисциплина: Системный анализ и принятие решений

Выполнил студент гр. 5130901/10xxx \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.О. Фамилия

(подпись)

Принял преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Г. Сиднев

(подпись)

“ ” 2024 г.

Санкт-Петербург

2024

# Исходные данные

* Вариант: {{variant}}
* Число исполнителей: {{performers\_n}}
* Решающее правило: {{rule}}

Граф:

{{graph\_img}}

# Задание 1

*Определение наиболее ранних моментов начала работ с использованием метода математического программирования.*

Введем обозначения

* *–* время начала работы
* *-* продолжительность работы
* – время окончания работы
* – номер последнего узла
* – время окончания всех работ
* – множество обратного соответствия, включающее все соседние вершины, из которых можно попасть в вершину

Тогда время окончания работы равно

Составим модель задачи линейного программирования вида:

В нашем случае ограничения задачи будут выглядеть так:

{{default\_math\_constraints}}

Решив эту задачу с помощью программных средств решения ЗЛП, мы получили следующие значения переменных:

{{default\_math\_result\_table}}

Таким образом нам стали известны:

* Минимальное время начала каждой работы
* Суммарное время выполнения работ ()

# Задание 2

*Считать, что вместо длительностей работ заданы трудоемкости. Длительность равна трудоемкость/интенсивность выполнения работы. Определить наиболее ранние моменты начала работ и назначенные работам интенсивности их выполнения при условии, что суммарная интенсивность не превышает 75% общего числа выполняемых работ.*

Введем обозначения

* *–* время начала работы
* – интенсивность работы
* – трудоемкость работы
* – время окончания работы
* – номер последнего узла
* – время окончания всех работ
* – общее число выполняемых работ
* – множество обратного соответствия, включающее все соседние вершины, из которых можно попасть в вершину

Изменим модель задачи следующим образом:

В нашем случае ограничения будут выглядеть так:

{{intensive\_math\_constraints}}

Решив эту задачу с помощью программных средств решения ЗНП, мы получили следующие значения переменных

Моменты начала работ:

{{intensive\_math\_result\_t\_table}}

Интенсивности:

{{intensive\_math\_result\_m\_table}}

# Задание 3

*Самостоятельно распределить работы между заданным числом исполнителей и сформулировать задачу математического программирования с бинарными индикаторными переменными. Определить число бинарных переменных и дополнительных ограничений в этой задаче и дать формулировку части ограничений с бинарными переменными.*

Распределим работы между {{performers\_n}} исполнителями:

{{binary\_performers\_tasks\_allocation}}

Введем некоторую постоянную

Тогда каждой паре работ , назначенных на исполнителя можно поставить в соответствие 3 ограничения:

Эти ограничения обеспечивают выполнение условия невозможности наложения процессов выполнения работ и во времени

Пусть количество работ, назначенных на исполнителя равно

Число дополнительных ограничений задачи равно :

{{binary\_constraints\_num}}

Число бинарных переменных :

{{binary\_variables\_num}}

# Задание 3.1

*Изменить формулировку задачи так, чтобы число бинарных переменных не превышало 10. Решить полученную задачу. Определить мощность множества бинарных переменных задачи и дать содержательную интерпретацию полученному решению.*

Изменим задачу так, чтобы число бинарных переменных не превышало 10. Для этого пусть на исполнителя 1 назначены следующие 3 задачи, остальные задачи не закреплены за исполнителями.

{{changed\_binary\_performers\_tasks\_allocation}}

Число бинарных переменных:

{{changed\_binary\_variables\_num}}

Число ограничений:

{{changed\_binary\_constraints\_num}}

Дополнительные ограничения задачи:

{{changed\_binary\_constraints}}

Решив эту задачу с помощью программных средств решения ЗЛП в целых числах, мы получили следующие значения переменных

{{changed\_binary\_result\_t\_table}}

И следующие значения переменных

{{changed\_binary\_result\_Y\_table}}

Таким образом порядок исполнения работ первым исполнителем выглядит следующим образом:

{{changed\_binary\_tasks\_order}}

# Задание 4

*Найти характеристики , и расписания выполнения комплекса работ с использованием метода динамического программирования. Привести соответствующие уравнения Беллмана. Определить критические пути на графе.*

- наиболее ранний момент наступления события

- наиболее поздний момент наступления события

*–* длительность работы

- резерв времени выполнения работы

Уравнения Беллмана поиска на каждом шаге выглядят следующим образом:

где, - множество обратного соответствия, включающее все соседние вершины, из которых можно попасть в вершину

Процесс вычисления производится от начального узла к конечному (:

{{dynamical\_min\_t\_calculations}}

Уравнения Беллмана поиска на каждом шаге выглядят следующим образом:

где, - множество прямого соответствия, включающее все соседние вершины, в которые можно попасть из вершины

Процесс вычисления производится от конечного узла к начальному (, где – номер конечного узла):

{{dynamical\_max\_t\_calculations}}

Полные резервы вычисляются по формуле :

{{full\_reserves\_calculations}}

Критические пути на графе – пути где все ребра имеют нулевой полный резерв. В нашем случае критические пути:

{{critical\_paths}}

# Задание 5

*Найти те же характеристики , и расписания выполнения комплекса работ с использованием математического программирования.*

Для поиска наиболее ранних моментов сформулируем ЗЛП:

В нашем случае получаем следующие ограничения задачи:

{{math\_min\_t\_constraints}}

Решив эту задачу с помощью программных средств решения ЗЛП в целых числах, мы получили следующие значения переменных

{{math\_min\_t\_values\_table}}

Они аналогичны тем же значениям, что были получены с помощью динамического программирования.

Для поиска наиболее поздних моментов сформулируем ЗЛП:

В нашем случае получаем следующие ограничения задачи:

{{math\_max\_t\_constraints}}

Решив эту задачу с помощью программных средств решения ЗЛП в целых числах, мы получили следующие значения переменных

{{math\_max\_t\_values\_table}}

Они аналогичны тем же значениям, что были получены с помощью динамического программирования.

Полные резервы вычисляются по той же формуле, что и в динамическом программировании

# Задание 6

*Определить помимо полных резервов времени работ резервы времени, относящиеся к событиям сетевого графа, а именно: .*

Определим независимые резервы 1-го порядка по формуле :

{{ir1\_calculations}}

Определим свободные резервы по формуле :

{{fr\_calculations}}

Определим независимые резервы 2-го порядка по формуле :

{{ir2\_calculations}}

# Задание 7

*Рассмотреть вероятностную постановку задачи анализа расписания. Считать СКО времен выполнения работ равными 5% от их длительностей. Предполагая неизменным критический путь (оценить справедливость этого предположения) найти вероятность того, что время выполнения комплекса работ не превысит найденного для детерминированной задачи в п.1 на 10%.*

Математическое ожидание длительности критического пути:

{{expected\_value\_calculation}}

Дисперсия суммы критического пути:

{{dispersion\_calculation}}

Посчитаем вероятность, что время выполнения комплекса работ не превысит детерминированное значение на {{overtime\_limit\_percent}}:

* {{overtime\_limit}}

*{{overtime\_probability\_formula}}*

Видим, что вероятность того, что время выполнения комплекса работа не превысит найденного для детерминированной задачи на {{overtime\_limit\_percent}} равна {{overtime\_probability\_percent}}.

Т.к. этот процент достаточно велик, то наше предположение о неизменности критического пути было верным.

# Задание 8

*Представить пошаговую процедуру имитационного моделирования расписания по схеме событий с учетом числа исполнителей и решающего правила ранжирования работ из числа возможных. По результатам моделирования построить диаграмму Гантта.*

Правило выбора работ:

* Число исполнителей: {{performers\_n}}
* Решающее правило: {{rule}}

Параметры:

* – текущее общее время выполнения
* – список пройденных событий
* – список выполненных работ на момент времени Т
* – список выполняемых на момент времени Т работ
* – список доступных на выполнение работ на момент времени Т
* – список длительностей доступных на выполнение работ
* – список резервов доступных на выполнение работ
* – список уровней доступных на выполнение работ
* – список работ, начатых в момент времени T
* – список времен освобождения ресурсов

{{schedule\_problem\_calculation\_table}}

Итоговое время работы: {{scheduling\_problem\_time}}

{{chart\_img}}

Диаграмма Ганта