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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.С. Сабонис

(подпись)

“ ” 2024 г.

Санкт-Петербург

2024

# Исходные данные

Вариант: {{variant}}

Граф:

{{graph\_img}}

* Число ресурсов : {{performers\_n}}
* Правила выбора работ:
  + {{rule\_1}}
  + {{rule\_2}}
  + {{rule\_3}}
* Коэффициент уменьшения общего времени выполнения при интенсификации : {{intensity\_limit}}
* СКО длительности работ при вероятностной постановке задачи : {{standard\_deviation\_percent}}
* Ограничение относительного превышения общего времени выполнения при вероятностной постановке задачи: {{overtime\_limit\_percent}}

# Часть 1. Задача сетевого планирования, метод динамического программирования

## Определить наиболее ранние и наиболее поздние моменты наступления событий;

- 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}}

## Определить резервы времени: полные резервы, написать матрицу резервов

Полные резервы вычисляются по формуле :

{{full\_reserves\_calculations}}

Матрица резервов:

{{reserves\_matrix}}

## Определить резервы времени: независимые резервы 1 рода, свободные резервы, независимые резервы 2 рода

Определим независимые резервы 1-го рода по формуле :

{{ir1\_calculations}}

Определим свободные резервы по формуле :

{{fr\_calculations}}

Определим независимые резервы 2-го рода по формуле :

{{ir2\_calculations}}

## Найти критический путь (пути);

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

{{critical\_paths}}

## Определить минимально возможное время (T) выполнения всего комплекса работ.

Минимально возможное время выполнения всего комплекса работ: {{moments\_T\_calculation}}

# Часть 2. Задача сетевого планирования, метод математического программирования

Найти наиболее ранние моменты начала работ и минимально возможное время выполнения (T) всех работ методом математического программирования.

## Составить задачу линейного 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}}

## Решить ЗЛП

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

{{default\_math\_result\_table}}

# Часть 3. Составление расписания

Найти распределение работ по ресурсам по следующим трем правилам выбора работ для выполнения. В каждом случае количество ресурсов равно {{performers\_n}}

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

## Правило №1. {{rule\_1}}

{{schedule\_problem\_calculation\_table\_1}}

Итоговое время работы: {{scheduling\_problem\_time\_1}}

{{chart\_img\_1}}

Диаграмма выполнения работ

Времена простоя каждого ресурса:

{{resources\_downtimes\_1}}

## Правило №2. {{rule\_2}}

{{schedule\_problem\_calculation\_table\_2}}

Итоговое время работы: {{scheduling\_problem\_time\_2}}

{{chart\_img\_2}}

Диаграмма выполнения работ

Времена простоя каждого ресурса:

{{resources\_downtimes\_2}}

## Правило №3. {{rule\_3}}

{{schedule\_problem\_calculation\_table\_3}}

Итоговое время работы: {{scheduling\_problem\_time\_3}}

{{chart\_img\_3}}

Диаграмма выполнения работ

Времена простоя каждого ресурса:

{{resources\_downtimes\_3}}

Наилучшим является расписание по правилу «{{best\_rule}}» с общим временем равным {{best\_time}}

# Часть 4. Интенсификация, метод математического программирования

Рассмотреть задачу из Части 2, считать, что вместо длительностей работ заданы их трудоёмкости, а длительности равны отношениям трудоемкостей к интенсивностям выполнения работ. Найти наиболее ранние моменты начала работ и их интенсивности, при условии, что комплекс работ должен завершиться не позднее, чем за

## Определить время выполнения комплекса работ T\*w, составить задачу математического программирования,

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

T – время выполнения всех работ при расчете модели без интенсивностей.

Текущая же модель должна завершиться не раньше чем за {{Tw\_value}}

Изменим модель задачи следующим образом:

В нашем случае ограничения будут выглядеть так:

{{intensive\_math\_constraints}}

## Решить ЗНП

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}}

# Часть 5. Составление расписания при заданном распределении, метод математического программирования

Предположить, что работы распределены по ресурсам. Построить расписание

## Задать, какая работа должна выполняться каким ресурсом

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

{{binary\_performers\_tasks\_allocation}}

## Определить количество бинарных переменных и количество дополнительных ограничений, сформулировать задачу математического программирования с бинарными переменными

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}}

## Упростить задачу и, если требуется, задать другое распределение работ по ресурсам, чтобы число бинарных переменных не превышало 10

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}}

## Решить ЗЛП

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}}

# Часть 6. Вероятностная постановка задачи

Считать СКО времён выполнения работ равными {{standard\_deviation\_percent}}

## Найти вероятность того, что время выполнения комплекса работ не превысит T более, чем на {{overtime\_limit\_percent}}

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, видим, что наше допущение о неизменности критического пути было верным.