Министерство образования Республики Беларусь

Учреждение высшего образования

«Белорусский государственный университет

информатики  и радиоэлектроники»

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Лабораторная работа № 2 по дисциплине

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СИСТЕМ

на тему

**«АНАЛИЗ И ОПТИМИЗАЦИЯ РЕШЕНИЙ В ДЕТЕРМИНИРОВАННЫХ ЗАДАЧАХ НА ОСНОВЕ МЕТОДА**

**МОНТЕ-КАРЛО»**

Вариант 1

Выполнили студент группы 820603:

Кулинкович И. В.

Горбач В. С.

Пелецкий А.Е.

Агинский В.Д.

Проверил:

Cевернев А.М.

Минск 2020

## 1 ЦЕЛЬ РАБОТЫ

**1.1** Изучить принципы и примеры применения метода Монте-Карло для оптимизации решений в детерминированных задачах.

**1.2** Разработать алгоритм для решения задачи на основе метода Монте-Карло. Выполнить три испытания разработанного алгоритма.

## 2 ПОСТАНОВКА ЗАДАЧИ

Предприятие имеет 20 станков типа Ст1 и 30 станков – типа Ст2. На каждом из них можно выпускать детали трёх видов. Производительность станков (деталей в день) приведена в таблице 2.1.

Таблица 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Станок | Деталь | | |
| Д1 | Д2 | Д3 |
| Ст1 | 20 | 35 | 15 |
| Ст2 | 15 | 30 | 45 |

Каждый станок настраивается на выпуск детали только какого-либо одного вида. В течение каждого рабочего дня предприятие должно выпускать не менее 150 деталей Д1 и не менее 100 деталей Д2. Прибыль от продажи одной детали Д1 составляет 6 ден. ед., от продажи детали Д2 – 4 ден. ед., от продажи Д3 – 8 ден. ед.

Требуется составить план использования станков, обеспечивающий получение максимальной прибыли от выпуска деталей.

## 3 ХОД ВЫПОЛНЕНИЯ

Разработаем алгоритмы для решения задачи на основе метода Монте-Карло согласно варианту задания. Код программы на языке *JavaScript* приведен в приложении А.

Результаты трёх испытаний разработанного алгоритма приведены в таблицах 3.1 – 3.3.

Таблица 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| Станок | Детали | | |
| Д1 | Д2 | Д3 |
| Ст1 | 9 | 8 | 3 |
| Ст2 | 0 | 0 | 30 |
| Прибыль | 13360 | | |

Таблица 3.2

|  |  |  |  |
| --- | --- | --- | --- |
| Станок | Детали | | |
| Д1 | Д2 | Д3 |
| Ст1 | 8 | 11 | 1 |
| Ст2 | 0 | 0 | 30 |
| Прибыль | 13420 | | |

Таблица 3.3

|  |  |  |  |
| --- | --- | --- | --- |
| Станок | Детали | | |
| Д1 | Д2 | Д3 |
| Ст1 | 8 | 9 | 3 |
| Ст2 | 0 | 0 | 30 |
| Прибыль | 13380 | | |

**4 ВЫВОДЫ**

Таким образом, в ходе выполнения данной лабораторной работы мы изучили алгоритмы для решения задачи на основе метода Монте-Карло. Разработали собственные алгоритмы, необходимые для решения задачи оптимизации согласно полученному варианту задания. Составили план использования станков, обеспечивающий предприятию максимальную прибыль от выпуска деталей. Для этого мы провели три испытания разработанного алгоритма, после чего выбрали наилучший вариант.

# Приложение А

(справочное)

Получение количества станков, работающих над каждым видом деталей в зависимости от имеющегося количества станков

const config1 = [20, 35, 15];

const config2 = [15, 30, 45];

const distributionOfMachines = count => {

  let rnd1 = Math.random();

  let rnd2 = (1 - rnd1) \* Math.random();

  const machines1 = Math.floor(count \* rnd1);

  const machines2 = Math.floor(count \* rnd2);

  const machines3 = count - machines2 - machines1;

  return [machines1, machines2, machines3];

}

const parts = (machines, config) => {

  return machines.map((machine, i) => machine \* config[i]);

}

const machines = (parts, config) => {

  return parts.map((part, i) => part / config[i]);

}

const process = (mass = { profit: 0 }) => {

  mass.machine1 = parts(distributionOfMachines(20), config1);

  mass.machine2 = parts(distributionOfMachines(30), config2);

  if ((mass.machine1[0] + mass.machine2[0] >= 150) && (mass.machine1[1] + mass.machine2[1] >= 100)) {

    mass.profit = (mass.machine1[0] + mass.machine2[0]) \* 6

      + (mass.machine1[1] + mass.machine2[1]) \* 4

      + (mass.machine1[2] + mass.machine2[2]) \* 8;

    return mass

  } else {

    return process()

  }

}

(() => {

  let mass = process()

  for (let i = 0; i < 100000; i++) {

    const option = process()

    if (option.profit > mass.profit) {

      mass = { ...option }

    }

  }

  mass.machine1 = [...machines(mass.machine1, config1)]

  mass.machine2 = [...machines(mass.machine2, config2)]

  console.log(mass)

})()