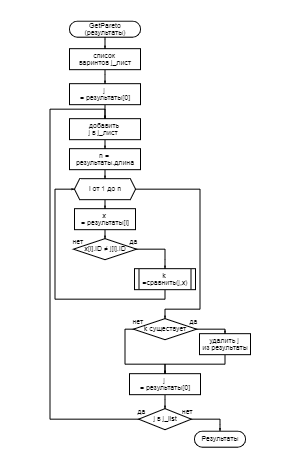
1. 1. студента группы ИТ – 42  
      Курбатовой Софьи Андреевны

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнение: |  | Защита |  |

* + 1. Оптимизация системы массового обслуживания

**Цель работы**: произвести оптимизацию системы массового обслуживания, пользуясь разработанной имитационной моделью.

* + - * 1. Содержание работы
      1. Система обработки информации содержит мультиплексный канал и N ЭВМ. Сигналы поступают на вход канала через t1(мкс).
      2. В канале они предварительно обрабатываются в течение t2 (мкс). Затем они поступают на обработку в ту ЭВМ, где наименьшая очередь. Емкости входных накопителей в каждой ЭВМ - E. Время обработки сигнала в каждой из ЭВМ - t3 (мкс).
      3. Смоделировать процесс обработки 1000 сигналов.
      4. **Данные для детерминированной модели СМО:** N=3, t1=10, t2=10 , t3=33, Е=4.
      5. **Данные для стохастической модели СМО:** интервал t1 распределен по показательному закону с параметром λ1=0,1, интервалы t2, t3 распределены нормально с параметрами m2=10, m3=33, σ2=1,5, σ3=3; вследствие возмущающих воздействий емкости входных накопителей каждой из ЭВМ непрерывно меняются, поэтому величина E является стационарным случайным процессом с нормальным законом распределения и интервалом разброса [2... 6] (сигналы, находившиеся в накопителе до изменения его емкости и не вмещающиеся в него после изменения его емкости, уничтожаются).
      6. **Варьируемые параметры:** N.
      7. Показатели работы: производительность системы, стоимость обработки, вероятность переполнения накопителей.
      8. 1. Провести анализ исходной задачи векторной оптимизации и составить перечень показателей (критериев оптимизации) и варьируемых параметров задачи. Уточнить или составить алгоритмы вычисления критериев оптимизации, включить их в состав цифровой имитационной модели.
      9. 2. Варьируя параметры системы, провести ряд экспериментов (несколько десятков) с имитационной моделью СМО и получить исходные данные для решения оптимизационной задачи (таблицу альтернатив).
      10. 3. Составить блок-схему и разработать процедуру, решающую задачу Парето-оптимизации методом квадрантов.
      11. 4. Получить таблицу Парето-оптимальных параметров модели исследуемой системы.
          1. Ход работы
      12. 1. Показатели работы системы: вероятность потери сигналов, скорость обработки сигналов:
* X1 – t1 – время между поступлением сигналов, 0.05 до 0.15
* X2 – t2 – время обработки сигналов в канале
* X3 – t3 – время обработки сигнала в ЭВМ
* X4 – E – емкость накопителя очереди
  + - 1. Варьируемые параметры t1, t2,t3, N:
* параметр показательного распределения времени t1 λ – [1,2];
* математические ожидания m1 – [1,2] и m2 [2,3]
* емкость ЭВМ – E [2,3]
  + - 1. Необходимо учесть, что показатель «Скорость обработки сигналов» должен быть наибольшим, а «Количество потерянных сигналов» наименьшим.
      2. 2.
      3. 

Блок-схема алгоритма

* + - 1. Был создан новый класс для хранения результатов одного эксперимента с методом для сравнения двух объектов этого класса CompareTo (листинг 1):
      2. Листинг 1. Класс вывода данных эксперимента

public class ResultLineExtend:IComparable<ResultLineExtend>

{

[Name("ID (exp)")]

public int ParamID { get; set; }

[Name("X1 (lambda)")]

public double X1 { get; set; }

[Name("X2 (m1)")]

public double X2 { get; set; }

[Name("X3 (m2)")]

public double X3 { get; set; }

[Name("X4 (E)")]

public int X4 { get; set; }

[Name("t1")]

public double T1 { get; set; }

[Name("t2")]

public double T2 { get; set; }

[Name("t3")]

public double T3 { get; set; }

//Показатели работы моделируемой системы

[Name("L% (LostSignalChance)")]

public double L { get; set; }

[Name("W% (WaitSignalChance)")]

public double W { get; set; }

[Name("S (SpeedSignalProcessing)")]

public double S { get; set; }

public ResultLineExtend() { }

public ResultLineExtend(int PID,

double x1, double x2, double x3, double x4, double t1,

double t2, double t3, double l, double w, double s)

{

ParamID = PID;

L = l; W = w; S = s;

X1 = x1; X2 = x2; X3 = x3; X4 = (int)x4;

T1 = t1; T2 = t2; T3 = t3;

}

public int CompareTo(ResultLineExtend resultLineExtend)

{

var ResLine = new int[2];

ResLine[0] = L == resultLineExtend.L? 0 :

L < resultLineExtend.L ? 1 : -1;

ResLine[1] = S == resultLineExtend.S

? 0

: S > resultLineExtend.S ? 1 : -1;

return ResLine.All(x => x == 0)

? 0

: ResLine.All(x => x == 1 || x == 0)

? 1

: ResLine.All(x => x == -1 || x == 0) ? -1 : 0;

}

}

* + - 1. Далее была создана функция, решающая задачу Парето-оптимизации методом квадрантов, суть которого состоит в следующем: если для варианта j найдется вариант k, лучший чем j пo всем s критериям, то вариант j исключается из таблицы альтернатив:
      2. Листинг 2. Функция метода квандрантов

static void MethodPareto(List<ResultLineExtend> ResultExtend)

{

var j\_list= new List<int>();

var j = ResultExtend.FirstOrDefault(); //выберем вариант j

do

{

j\_list.Add(j.ParamID); //добавим вариант в список

var k = ResultExtend.FirstOrDefault(x => j.ParamID != x.ParamID && j.CompareTo(x) == -1);//для варианта k != j

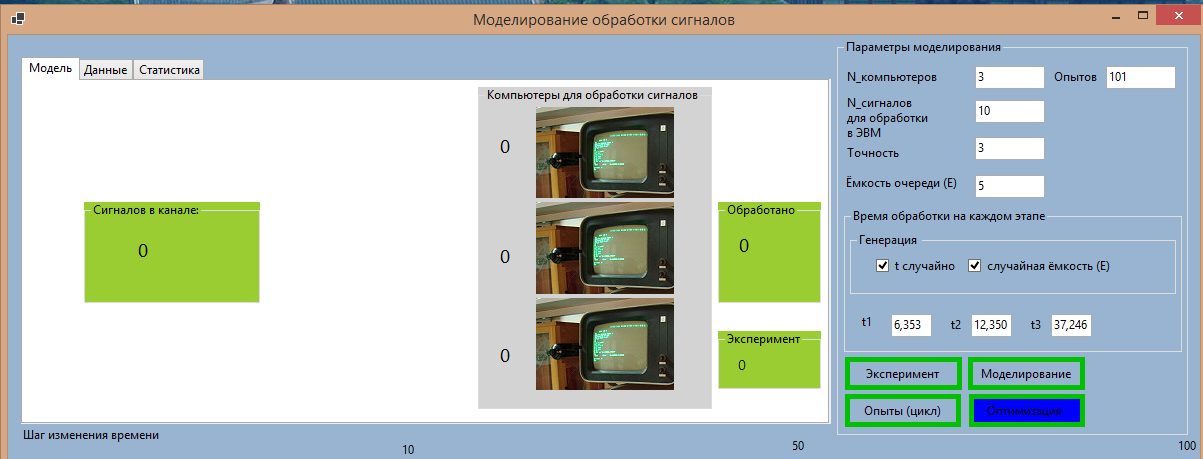
if (k != null) {

ResultExtend.Remove(j); }

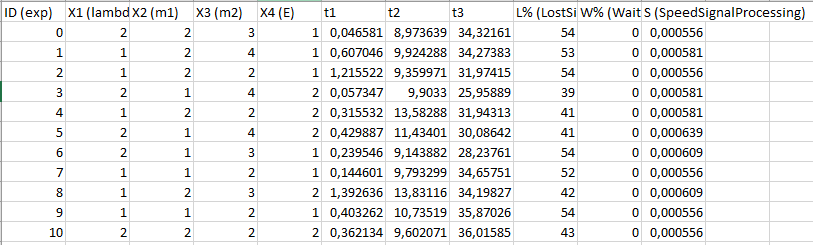
j = ResultExtend.FirstOrDefault(x => !j\_list.Contains(x.ParamID));

}

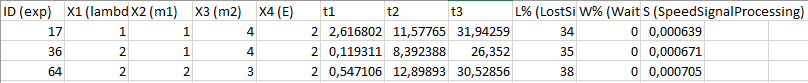
while (j != null); //пока не просмотрены все варианты

* + - 1. }
      2. 3. Тестирование:
      3. 

Интерфейс моделирования

* + - 1. 

Результаты экспериментов

* + - 1. 

Результаты после оптимизации

* + - 1. **Вывод:** Таким образом в ходе выполнения лабораторной работы была произведена оптимизация системы массового обслуживания, с использованием разработанной имитационной модели.