*Московский Технический Университет*

*Связи и Информатики*

Кафедра теории вероятностей и прикладной математики

**Математическое моделирование**

Лабораторная работа №3

**«Построение имитационной модели»**

Выполнила

Студентка

группы БПМ1301

Голяева А.В.

Проверила

Смирнова Н.И.

Москва 2016

**2. Задание.**

Для выбранного в лабораторной работе 1 объекта моделирования на основе описательной математической модели, построенной в лабораторной работе 2 , построить имитационную модель.

Содержание отчета:

1. Титульный лист
2. Задание
3. Подробное описание объекта моделирования
4. Цель моделирования
5. Набор экспериментов , которые планируется выполнить
6. Результаты
7. Заключение по результатам проведенных экспериментов.

***3.*** *«Задача о распределении средств поражения».*

 **Задача:** В условиях военного времени,Планируются боевые действия самолетами по обороняющимся целям. Самолеты находятся на боевом вооружении в течение всего срока эксплуатации. При повреждении самолета, его ремонтируют и снова отправляют в бой. Списание самолета возможно только при полном его уничтожении. Дальность полета самолета зависит от количества топлива. Управляет самолетов 2 пилота (главный пилот и стажер).Все пилоты проходят жесткий контроль.

Цели эшелонированы по глубине территории на четырех параллельных рубежах. Перед тем как выйти на данный рубеж, самолеты проходят зону действия огневых средств этого рубежа, где подвергаются обстрелу со стороны последних. Огневые средства каждого рубежа могут вести огонь не только по средствам поражения, направляющимся непосредственно на цели данного рубежа, но и по тем самолетам, которые проходят через зону действия, направляясь на следующие рубежи.

Вероятность поражения одного самолета, пролетающего зону действий орудий i-го рубежа -

vi=1-e-alphaiNi(ср) i=1,2,3,4

 Ni(ср)-среднее число орудий, сохранивших боеспособность на I м рубеже ,alphai –эффективность стрельбы орудий по самолетам.

Среднее число орудий i-го рубежа, пораженных j-й волной вылетов самолетов, направленных на цели этого рубежа, определяется формулой Qi=Ni(1-e vj/Ni\*Pi )

 **Ni** число орудий на I м рубеже. **Vj-** среднее число самолетов в j волне вылетов, сохранивших боевые свойства до I го рубежа **Pi-** средняя вероятность поражения одного орудия I го рубежа атакующим его самоелетом.

Имеется N орудий на всех рубежах:   N=[Σ](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%B3%D0%BC%D0%B0_%28%D0%B1%D1%83%D0%BA%D0%B2%D0%B0%29) Ni  vj/Ni\*Pi

Ni число орудий на I м рубеже. На первом рубеже – 10 орудий, на втором – 12 , на третьем – 15, на четвертом – 10, следовательно, N=47.

Волны вылетов распределены по времени так, чтобы к моменту подлета следующей волны предшествующая ей волна самолетов успела выполнить боевое задание .

**Пусть:** P1=0.4 P2=0.5 P3=0.4 P4=1.0

 Alpha 1=0.05 Alpha 2=0.04 Alpha 3=0.04 Alpha 4=0.05 **4.Цель**: имеется z0=80 самолетов, которые необходимо распределить на четыре волны вылетов так, чтобы сделать максимальное среднее число поражений на всех рубежах. f(x)=  [Σ](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%B3%D0%BC%D0%B0_%28%D0%B1%D1%83%D0%BA%D0%B2%D0%B0%29)fi(x) fi(x)среднее число поражений на i-м рубеже

**5. Моделирующий алгоритм**. Для решения данной задачи будем пользоваться общей схемой решения задач методом динамического программирования.

**Динамическое программирование** в теории управления  — способ решения сложных задач путём разбиения их на более простые подзадачи. Он применим к задачам с оптимальной подструктурой , выглядящим как набор перекрывающихся подзадач, сложность которых чуть меньше исходной. В этом случае время вычислений, можно значительно сократить.

Ключевая идея в динамическом программировании достаточно проста. Как правило, чтобы решить поставленную задачу, требуется решить отдельные части задачи (подзадачи), после чего объединить решения подзадач в одно общее решение. Часто многие из этих подзадач одинаковы. Подход динамического программирования состоит в том, чтобы решить каждую подзадачу только один раз, сократив тем самым количество вычислений. Это особенно полезно в случаях, когда число повторяющихся подзадач экспоненциально велико.

В общем случае мы можем решить задачу, в которой присутствует оптимальная подструктура, проделывая следующие три шага.

1. Разбиение задачи на подзадачи меньшего размера.
2. Нахождение оптимального решения подзадач рекурсивно, проделывая такой же трехшаговый алгоритм.
3. Использование полученного решения подзадач для конструирования решения исходной задачи.

 Разобьем нашу задачу на 4 пункта и проведем оптимизацию этих пунктов.

Оптимизация на 4 шаге

Оптимизация на 3 шаге

Оптимизация на 1 шаге

Оптимизация на 1 шаге

**6. Набор экспериментов, которые планируется выполнить.** В ходе обработки программы проведем несколько экспериментов ( введем различные входные данные), что бы увидеть , что программа работает правильно и данные полученные во второй лабораторной работе совпадают с результатами программы. Так же введем некорректные данные, что бы понять , что программа распознает неверные начальные условия .

**7.Программа и результаты работы программы.**

N1:=10 ;N2:=12; N3:=15;N4:=10; z0=80:

Optimal:=proc (z0,N1 ;N2; N3;N4; )

with (LinearAlgebra):

P1:=0.4; P2:=0.5;P3:=0.4; P4:=1.0; alpha1:=0.05 ;alpha2:=0.04 ;alpha3:=0.04 ;alpha4:=0.05 ;

 x4 :=z3:

Nc4:=N4:

v4:=1-exp (-alpha4\*N4):

Q(x[4]):=N4(1- exp(v[4]/N4 \*P4):

f4(z3):=Q(x[4]):

for i from 3 by -1 to 1 do

 for j from 4 by -1 to 1 do

V[i]:=x[i]\*exp(-alpha[i]\*N[i]):

Q(x[i]):=N[i](1-exp(-V{i]/N[i]\*P[i]):

f[i,j]:=Q(x[i])+f[i+1](z[i]):

n[i]:=N[i]-Q(x[i]):

 od:

od :

for i from 3 by -1 to 1 do

z[i]:=(z[i-1] -x[i])exp(-alpha[i]n[i]):

od

print z[i]:

print Q(x[i]):

print x[i]:

end proc:

Результат работы программы.



Получаем z1= 37, и на первом рубеже поражено Q1(x1)=5,6 орудий. При z1=37 по значениям, приведенным на пред предыдущем графике получили оптимальное уравнение на втором шаге x2=23 т.е. дальше следует направить14 самолетов. Пользуясь формулами аналогичными предыдущим , получим :

(в первую волну вылетов войдут 34 самолета n1=34, ко второму рубежу подойдут 37 самолетов из них полетят дальше только 14, а так как всего самолетов осталось 46 , то во вторую волну следует включить n2=46(14/23)=28 самолетов, n3=0,т.к. x3=0 и в четвертую волну-оставшиеся 18 самолетов.)

Проведем второй эксперимент с 85 самолетами на начальном этапе. 

(в первую волну вылетов войдут 39 самолета n1=39, ко второму рубежу подойдут 42 самолетов из них полетят дальше только 14, а так как всего самолетов осталось 46 , то во вторую волну следует включить n2=46(14/23)=28 самолетов, n3=0,т.к. x3=0 и в четвертую волну-оставшиеся 18 самолетов.)

При вводе всех начальных данных отрицательными, программа выдаёт ошибку.



**8.Заключение. Вывод:**

Динамическое программирование - это область математического программирования, включающая совокупность приемов и средств для нахождения оптимального решения, а также оптимизации каждого шага в системе и выработке стратегии управления, то есть процесс управления можно представить, как многошаговый процесс. Динамическое программирование, используя поэтапное планирование, позволяет не только упростить решение задачи, но и решить те из них, которым нельзя применить методы математического анализа. Упрощение решения достигается за счет значительного уменьшения количества исследуемых вариантов, так как вместо того, чтобы один раз решать сложную многовариантную задачу, метод поэтапного планирования предполагает многократное решение относительно простых задач. Планируя поэтапный процесс, исходят из интересов всего процесса в целом, т.е. при принятии решения на отдельном этапе всегда необходимо иметь в виду конечную цель.

Анализируя полученные результаты мы видим, что эффективнее осуществлять планирование на подступах к каждому рубежу, т.е. каждый раз решать задачу, аналогично рассмотренной.

Продолжа решение, мы можем убедиться, что это задача является вырожденной задачей динамического программирования - планировать нужно каждый шаг отдельно, распределяя самолёты перед рубежом так, чтобы получить максимальное число самолетов, преодолевающих данный рубеж.

Данная модель является адекватной.

Построив имитационную модель, мы убедились, что полученные результаты совпадают с результатами из второй лабораторной работы. Полученные данные являются однозначно верными. Так же мы поняли, что программа распознает некорректные данные.