

МИНОБРНАУКИ РОССИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования «Национальный
исследовательский университет «МИЭТ»

Кафедра: ВМ-1

Дисциплина: Теория игр и исследование операций

01.03.04 «Прикладная математика» (выпускающая кафедра ВМ-1)

**Курсовая работа на тему:
«Модель управления складскими запасами»**

Выполнил:

студент группы ПМ-41

Алимагадов Курбан Алимагадович

Преподаватель:

доцент кафедры ВМ-1, к.ф.-м.н.

Лисовец Юрий Павлович

Дата:

Содержание

1.Постановка задачи и исходные данные.....	3
2.Алгоритм управления складом.....	4
3.Результаты работы программы и экспорт данных в Excel.....	8
4.Вывод	11

1. Постановка задачи и исходные данные

Построить математическую модель управления складом на основе анализа спроса на товар. Исходными данными являются значения проданного товара из файла D_Z_2019_all_MIET.xls в каждый день работы. Анализируя эти данные, нужно определить даты пополнения склада и количество подвозимого в эти дни товара.

В рамках решения задачи будем использовать следующие обозначения и переменные:

warehouse_vol – объём склада;

warehouse – текущее состояние склада;

vv – вектор, содержащий значения объёмов продаж в каждый день работы;

mean_vv – среднее количество товара, проданное за время depth, (средняя скорость продаж/спрос);

std_vv – среднеквадратическое отклонение спроса от mean_vv;

depth – количество дней, данные за которые используются для расчёта mean_vv и std_vv;

price – цена за продажу одной единицы товара;

S – стоимость хранения одной единицы товара на складе в течение месяца;

k – стоимость одного подвоза очередного заказа;

order – размер очередного заказа, количество товара, подвозимого на склад;

orders – вектор, содержащий сведения о днях, в которые были сделаны заказы на подвоз товара;

orders_size – вектор, содержащий размеры всех заказов;

warehouse_l – количество товара на складе в момент последнего пополнения;

$flag$ – флаг, помогающий зафиксировать $warehouse_l$ при каждом пополнении склада;

τ – оценка времени, за которое склад должен опустеть до очередного, пополнения;

vw – вектор, содержащий значения количества товара на складе в каждый день работы;

$Income$ – вектор доходов;

$Costs$ – вектор расходов;

$Itog$ – вектор прибыли.

2. Алгоритм управления складом

Для управления складом, нам необходимо знать среднее значение спроса $mean_vv$ и его среднеквадратическое отклонение std_vv , рассчитанные на основе данных за $depth$ последних дней работы склада. Если с момента начала анализа прошло меньше $depth$ дней, расчёт проводится на основе данных за то количество дней, которое успело пройти.

В очередной i -й день работы вычисляется текущее значение $mean_vv$ и std_vv . Затем находится размер очередной продажи

$$sales[i] = \min(warehouse, vv[i]). \quad (1)$$

Если при этом значение флага $flag$ установлено в 1, количество товара на складе в этот день фиксируется в $warehouse_l$:

$$warehouse_l = warehouse - sales[i] + order, \quad (2)$$

а значение $flag$ устанавливается в 0 (изначально на 1-й итерации $flag = 0$). После этого рассчитывается количество товара в i -й день с учётом заказа и продажи:

$$warehouse = warehouse - sales[i] + order. \quad (3)$$

Воспользуемся правилом 3-х сигм: если количество товара на складе меньше, чем $mean_vv + 3 * std_vv$, тогда происходит формирование очередного заказа. Размер заказа $order$ формируется как

$$order = \min(\text{floor}(\tau(mean_vv + 3std_vv)), warehouse_vol - warehouse), \quad (4)$$

где $\text{floor}()$ – функция MATLAB, округляющая число до ближайшего целого. То есть, мы заказываем такое количество товара, которое предположительно удастся реализовать за время τ , если для такого количества товара места на складе нет, то мы заказываем $\text{warehouse_vol} - \text{warehouse}$ единиц товара. Для этого сначала необходимо вычислить оценку τ времени, в течение которого склад снова опустеет:

$$\tau = \frac{\text{warehouse_l} - \text{warehouse}}{\text{mean_vw}}. \quad (5)$$

Оформление очередного заказа в i -й день фиксируется в векторе orders ($\text{orders}[i] = 1$), а также значение флага flag устанавливается в 1. Размер заказа в i -й день записывается в вектор orders_size ($\text{orders_size}[i] = \text{order}$). После всех операций с содержимым склада в i -й день значение количества товаров записывается в вектор vw ($\text{vw}[i] = \text{warehouse}$).

Суммарный доход рассчитывается как

$$\text{Income} = \text{cumsum}(\text{price} * \text{sales}), \quad (6)$$

где $\text{cumsum}()$ – функция MATLAB, рассчитывающая для входного массива суммы с накоплением.

Расходы на содержание товаров на складе рассчитывается как сумма произведений максимальных объёмов товара на складе за месяц на S . Суммарные расходы рассчитываются как

$$\text{Costs} = \text{cumsum}([0, \text{orders}] * (-k)) - \text{cumsum}([0, vS]), \quad (7)$$

где vS – вектор, содержащий значения затрат на хранение товара на складе, а добавления к массивам 0 необходимо, чтобы получить в векторе Costs начальное значение затрат, то есть до того, когда началась работа расходов не было.

Прибыль рассчитывается как разность между доходами и расходами (так как мы посчитали расходы со знаком минус, мы будем брать сумму).

$$\text{Itog} = [0, \text{Income}] + \text{Costs}, \quad (8)$$

Код программы на языке MATLAB:

```
S = 4;
k = 700;
price = 20;
warehouse_vol = 1000;
vv = xlsread('D_Z_2019_all_MIET.xls', ['Продажи']);
vv = vv(243:494,6);
N = length(vv);
depth = 20;
warehouse = 100;
orders = zeros(1,N);
order = 0;
vw(1) = warehouse;
flag = 0;
warehouse_l = warehouse;
for i = 1:N
    mean_vv = mean(vv(max([1,i - depth]):i));
    std_vv = sqrt(var(vv(max([1,i - depth]):i)));
    sales(i) = min(warehouse, vv(i));
    if (flag == 1)
        warehouse_l = warehouse - sales(i) + order;
        flag = 0;
    end
    warehouse = warehouse - sales(i) + order;
    if (warehouse < mean_vv + 3*std_vv)
        tau = (warehouse_l - warehouse) / mean_vv;
        order = min(floor(tau*(mean_vv + 3*std_vv)),
warehouse_vol - warehouse);
        orders(i) = 1;
        flag = 1;
    else
        order = 0;
    end
    orders_size(i) = order;
    vw(i + 1) = warehouse;
end
figure
hold on
grid on
plot(vv, '*')
title('Значение спроса в каждый день')
xlabel('День')
ylabel('Спрос')
mas = find(orders == 1) + 1;
figure
hold on
grid on
plot(vw, 'b')
plot(mas,vw(mas), 'b*')
title('Состояния складов в течение ' + string(N) + ' дней')
xlabel('День')
ylabel('Кол-во единиц товара')
```

```

Income = cumsum(price * sales);
figure
hold on
grid on
plot(Income, 'b')
title('График доходов')
xlabel('День')
ylabel('Кол-во рублей')
time = 1:30:N;
vS = zeros(1,N);
for i = 2:length(time)
    vS(time(i)) = S .* max(vw(time(i - 1):time(i)));
end
Costs = cumsum([0,orders].*(-k)) - cumsum([0, vS]);
figure
hold on
grid on
plot(Costs, 'b')
title('График расходов')
xlabel('День')
ylabel('Кол-во рублей')
Itog = [0, Income] + Costs;
figure
hold on
grid on
plot(Itog, 'b')
title('Прибыль')
xlabel('День')
ylabel('Кол-во рублей')
xlswrite('result.xls', [vv, vw(2:end) ', orders', orders_size', Itog(2
:end) '])

```

Значения входных параметров:

$S = 4$;

$k = 700$;

$price = 20$;

$warehouse_vol = 1000$;

$depth = 20$;

$warehouse = 100$;

$order = 0$; (начальное значение)

$vw[1] = 100$;

$flag = 0$; (начальное значение)

$warehouse_1 = 100$; (начальное значение)

3. Результаты работы программы и экспорт данных в Excel

По результатам работы программы были построены следующие графики:

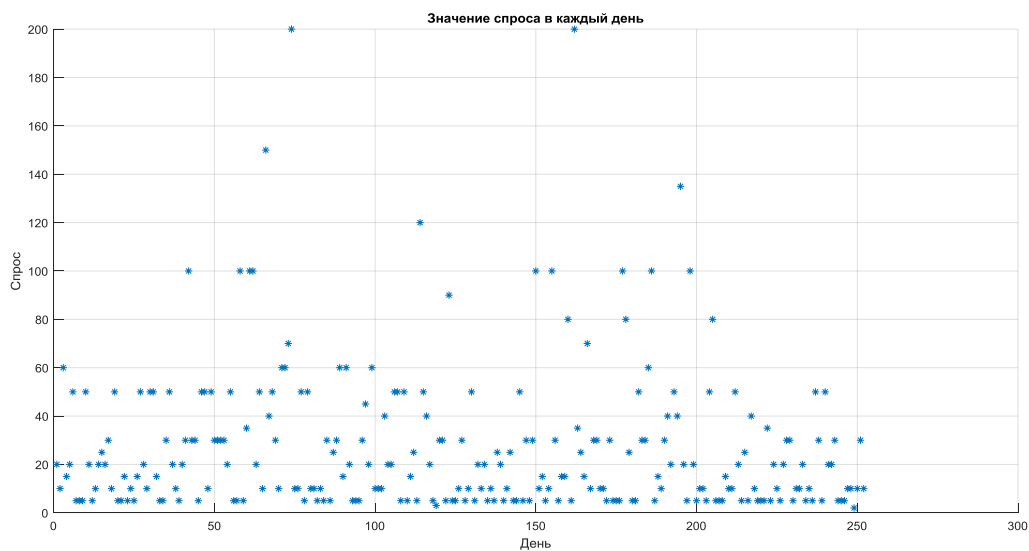


Рис. 1 Значение спроса в каждый день работы

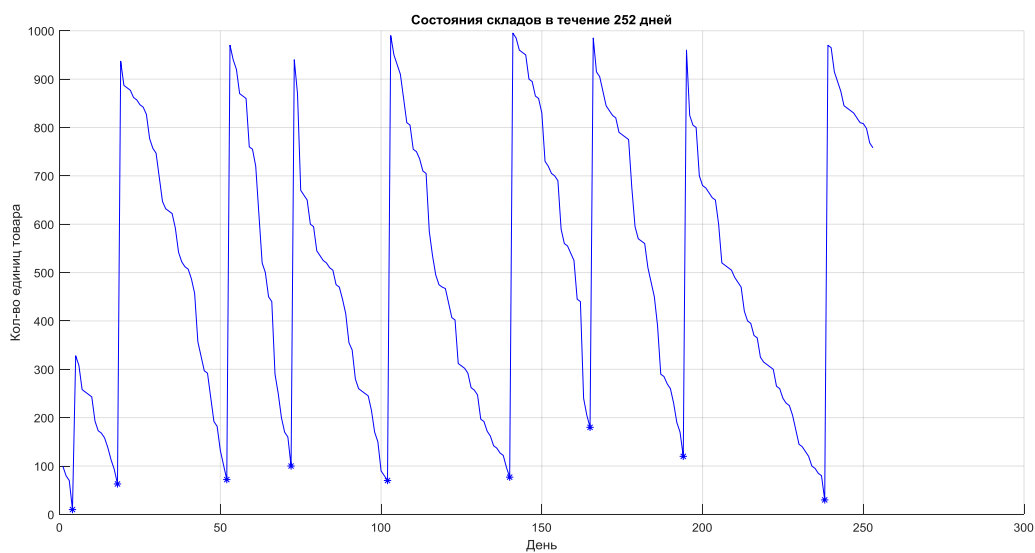


Рис. 2 Состояние склада в каждый день работы

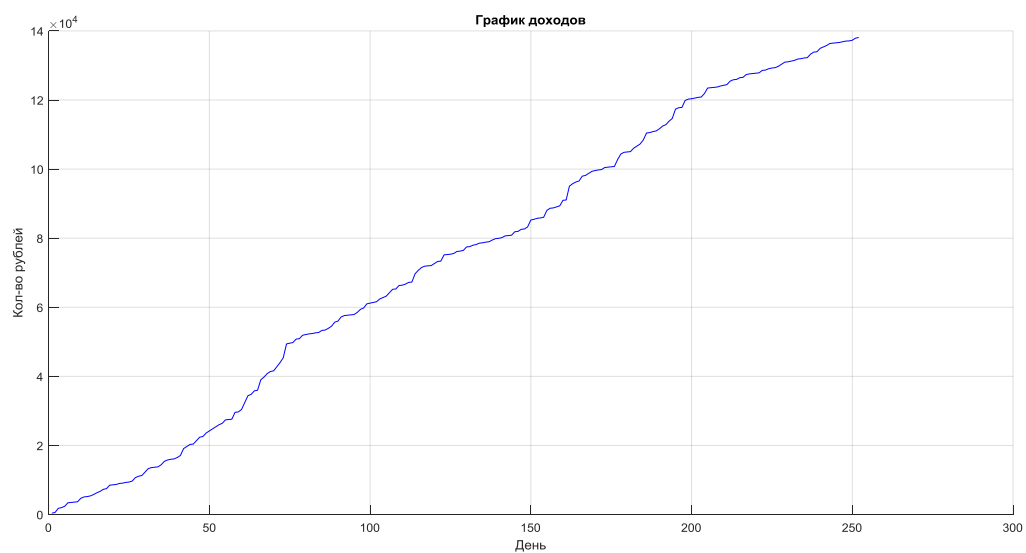


Рис. 3 График доходов

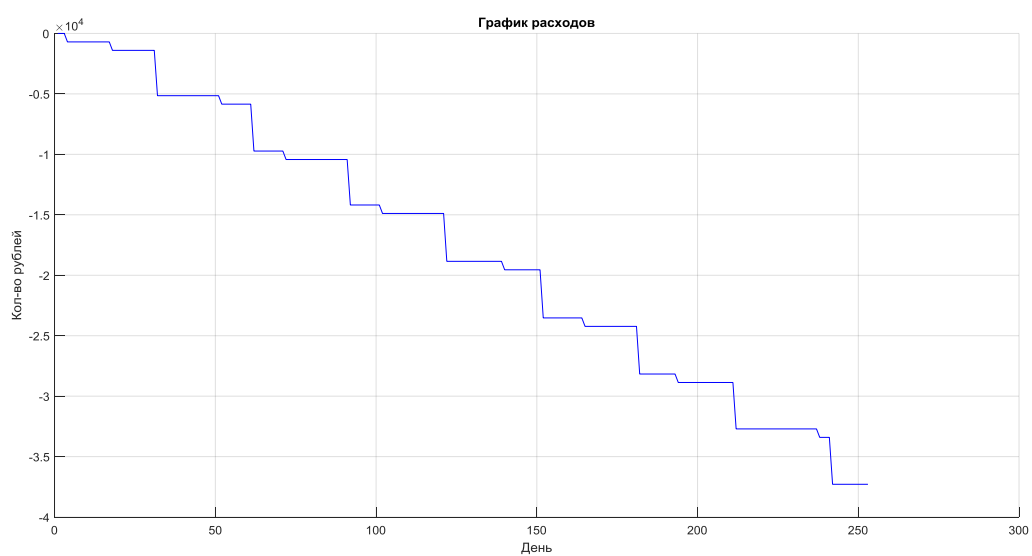


Рис. 4 График расходов

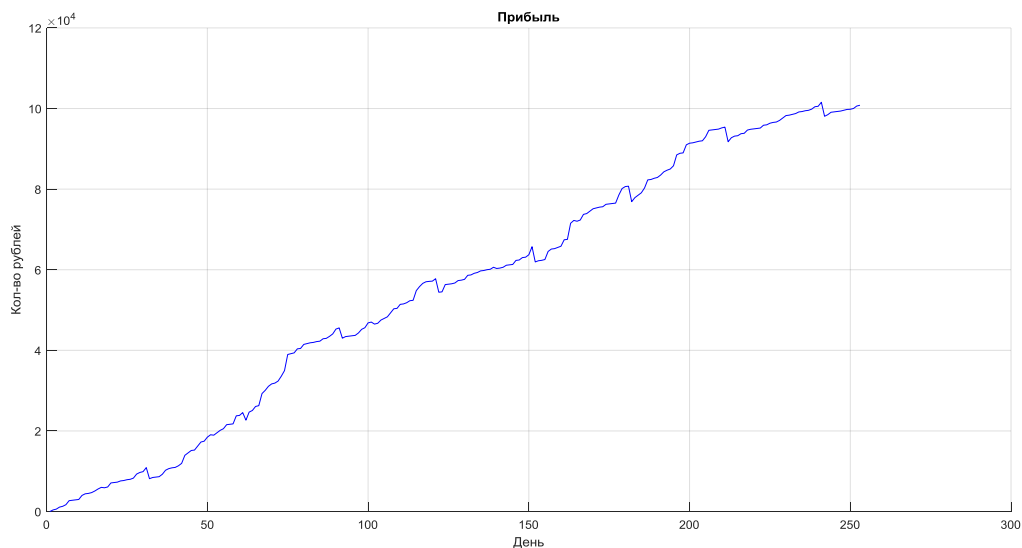


Рис. 5 График прибыли

Как можно видеть по графику прибыли, алгоритм справляется со своей задачей, и прибыль возрастает с течением времени.

Данные, полученные в результате расчётов были экспортированы в файл result.xls. Файл содержит: объём продаж в каждый день работы, данные о состоянии склада, данные о оформлении заказов, объёмы заказов и данные о прибыли за каждый день работы. С ними можно работать непосредственно в Excel, к примеру строить графики.



Рис. 6 График состояния склада, построенный в Excel

4. Вывод

В результате исследования был разработан алгоритм, справляющийся с задачей управления складскими запасами. Эффективность работы алгоритма можно оценить по объёму прибыли, которую он позволяет получить. При заданных входных данных программа, написанная на основе алгоритма, демонстрирует возрастающую с течением времени прибыль. Данные можно экспортировать в Excel для дальнейшей обработки.