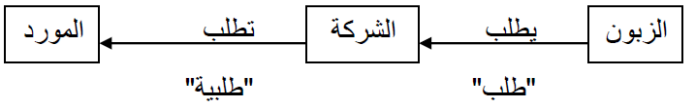
محاكاة نظام مخزن

وصف النظام:

شركة تبيع منتج واحد وتريد تقدير كمية البضاعة التي يجب أن توجد في مخزن الشركة لكل شهر من ال n شهر القادمة.

ترميز:

١. سنطلق على طلب الزبون لبضاعة من الشركة: طلب (D)demand.
٢. سنطلق على على طلب الشركة من المورد: طلبية order.



- الزمن بين طلبات الزبائن عشوائي.
- وكذلك حجم الطلبات D متغير عشوائي يتبع للتوزيع المتظم وليس له علاقة بزمن الطلب.
- في بداية الشهر تتقوم الشركة بمراجعة المخزون لديها. ثم تقرر كم وحدة يجب طلبها من المورد Supplier. إذا طلبت الشركة z وحدة فإن ذلك سيكلفها

$$k + iz$$

حيث k = تكلفة التجهيزات (الترحيل، عمال النقل،... الخ)

i = التكلفة التزايدية لكل وحدة شهرية.

- عندما يتم طلب طلبية فإن زمن وصولها من المورد ايضا متغير عشوائي يسمى lead time أو delivering lag يتوزع بين نصف شهر و شهر.
- لدى الشركة سياسة تخزين (s,S) لحديد حجم الطلبية كالآتي:

$$z = \begin{cases} S - I & I < s \\ 0 & I \geq 0 \end{cases}$$

حيث:

I = البضاعة الموجودة في المخزن في بداية الشهر.

s = أقل كمية بضاعة يجب وجودها في المخزن.

S = أكبر كمية بضاعة يجب وجودها في مخزن الشركة.

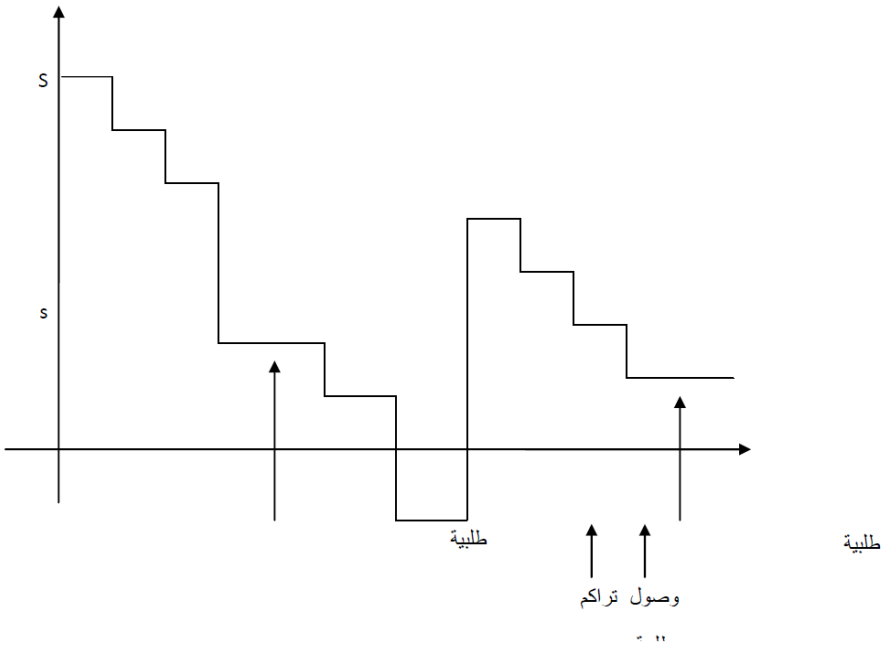
- عندما يطلب الزبون بضاعة فإنه يعطى ما يطلب اذا كانت البضاعة الموجودة في المخزن على الاقل مساوية للطلب (≤) . اذا كان الطلب كبير فسيكون هناك باقي على الشركة توفيه في المرات القادمة، وفي هذه الحالةفإن المخزون الجديد (المخزون القديم ناقص الطلب) عدد سالب ويسمى التراكم (backlog).
- توجد تكاليف أخرى على الشركة (عدا تكلفة الطلبيات) هي:
  ١. تكلفة الملكية holding cost
  ٢. تكلفة التقصير shortage cost

لحساب التكاليف أعلاه يجب أولاً تعريف القيم الثلاثة التالية:

١. $I(t)$ = محتوى المخزن في الزمن t. نلاحظ ان $I(t)$ يمكن ان يكون صفر، سالب أو موجب.

٢. $I^+(t)$ = عدد الوحدات الموجودة فعلياً في المخزن في الزمن t. = $I^+(t)$ $\max\{I(t),0\}$ حيث $I^+(t) \geq 0$.

٣. $I^-(t)$ = الكمية التي لم توفى من الطلب – التراكم – في الزمن t ، $I^-(t) = \max\{-I(t),0\}$ وأيضاً $I^-(t) \geq 0$



حساب تكلفتي الملكية والتقصير:

- في هذا النموذج افترضنا ان تكلفة الملكية للوحدة الواحدة هو h = 1$ للشهر والتي تتضمن الايجار، التأمين، الماء، الكهرباء ...الخ. وهي مطلوبة حتى في حالة $I^+(t) = 0$ وبما ان الغرض من المحاكاة هو مقارنة سياسات الطلبيات فقد تم تجاهل هذه التكلفة – تم اعطاءها ١$ للوحدة – لأنها لاتؤثر أو تتأثر بنوعية السياسة المتبعة. الآن: متوسط عدد الوحدات الموجودة في المخزن لكل شهر ل n شهر القادمة هي:

$$\overline{I^+} = \int_0^n \frac{I^+(t)}{n} dt$$

** وذلك لأنها متغير عشوائي يتبع للتوزيع المنتظم الذي دالته الاحتمالية = $\frac{1}{n-0} = \frac{1}{n}$

اذن تكلفة الملكية للشهر هي:

hcost = $\overline{I^+}$ h .....(1)

- ايضا نفرض أن تكلفة التراكم backlog للوحدة المتراكمة يساوي $\Pi = 5\$$، متوسط عدد الوحدات المتراكمة:

$$\overline{I^-} = \int_0^n \frac{I^-(t)}{n} dt$$

اذن متوسط تكلفة التراكم للشهر هي:

scost = $\overline{I^-}$ Π .....(2)

- بعد هذا يتم حساب التكلفة الكلية والتي تتكون من:

تكلفة الطلبيات + تكلفة الملكية + تكلفة التقصير

للشهر للمقارنة بين السياسات المتبعة. السياسة التي تعطي أقل تكلفة هي الأفضل.

الأحداث في النظام:

١. حدث وصول طلبية من المورد الى الشركة.

٢. حدث طلب زبون لمنتج من الشركة.
٣. نهاية المحاكاة بعد n شهر.
٤. جرد مخزن واحتمال حدوث طلبية في بداية الشهر.

الاحداث ١،٢،٤ تسبب تغير في حالة النظام.

خوارزميات الاحداث أعلاه:

١/ خوارزمية حدث وصول طلبية:

١. البداية.
٢. وصول الطلبية.
٣. اضافة حجم الطلبية الى كمية البضاعة الموجودة في المخزن.
٤. منع خروج اي طلبية.
٥. النهاية.

٢/ خوارزمية حدوث طلب بضاعة من الشركة:

١. البداية.
٢. حدوث الطلب.
٣. حساب حجم الطلب.
٤. انقاص كمية الطلب من كمية البضاعة الموجودة في المخزن.
٥. الترتيب للطلب القادم.
٦. النهاية.

٣/ خوارزمية جرد المخزن:

١. البداية.
٢. هل $I(t) \leq s$ ؟ اذا كانت الاجابة نعم اذهب الى الخطوة رقم (٣). والا اذهب الى الخطوة رقم (٦)،
٣. حساب حجم الطلبية $S - I(t)$.
٤. حساب تكلفة الطلبية.
٥. الترتيب لحدث وصول طلبية.
٦. الترتيب لحدث الجرد التالي.
٧. النهاية.

٤/ الخوارزمية العامة للنظام:

١. البداية.
٢. إدخال قيم s ، S.

٣. ادخال k ،i،I(0)،n.
٤. t = 0.
٥. .حساب قيمة الطلبية من الشركة z بإستخدام بالقانون:

$$z = \begin{cases} S - I & I \langle s \\ 0 & I \geq 0 \end{cases}$$

٦. كمية البضاعة الموجودة في المخزن بعد وصول الطلبية هو:
I(t) = I(t) + z

٧. اذا طلب الزبون بضاعة قدرها D فإن:
I(t) = I(t) + D

$I^{+}(t) = \max\{I(t),0\}$

$I^{-}(t) = \max\{-I(t),0\}$

٨. حساب تكلفة الطلبية للشهر t كالتالي:
DCost =k +iz

٩. حساب تكلفة الملكية للشهر t .
١٠. حساب تكلفة التقصير للشهر t.
١١. t = t+1 .
١٢. تكرار الخطوات ٥ – ١١ حتى حدوث شرط نهاية المحاكاة t = n.
١٣. حساب متوسط التكاليف الثلاثة ل n من الشهور.
١٤. حساب التكلفة الكلية.
١٥. تكرار الخوارزمية لسياسات مختلفة ومقارنتها.
ملحوظة: يتم توليد المتغير D كمتغير عشوائي.

مثال:

اذا كانت الوحدات الموجودة في مخزن الشركة في بداية المحاكاة هي I(0) = 60 ، وانه لاتوجد طلبيات متأخرة بين الشركة والمورد. حاكى نظام جرد الخزن ل ١٢٠ شهر بإستخدام التكلفة التكلية للمقارنة بين السياسات ال ٩ التالية:

	1	2	3	4	5	6	7	8	9
s	20	20	20	20	40	40	40	60	60
S	40	60	100	60	60	80	100	80	100

الحل:

السياسة الاولى $s = 20$, $S = 40$

نفرض ان $k = 20, i = 2,$

عند $t = 0$: $I(0) = 60$

$Z = 0$; $I(t) > s$

$DCost = 20 + 0(2) = 20$

نفرض ان $D = 30$ اذن

$I(0) = 60 - 30 = 30$

$I^{+}(0) = \max\{30,0\} = 30$ ..... $hcost = 30$

$I^{-}(0) = \max\{-30,0\} = 0$ ..... $scost = 0$

عند $t = 1$

$Z = 0$; $I(0) > 0$

$Dcost = 20 + 2(0) = 20$

نفرض ان $D = 20$ اذن

$I(1) = 30 - 20 = 10$

$I^{+}(1) = \max\{10,0\} = 10$ ..... $hcost = 10$

$I^{-}(0) = \max\{-10,0\} = 0$ ..... $scost = 0$

عند $t = 2$

$Z = 40 - 10 = 30$; $I(1) < s$

Dcost = 20 + 2(30) = 20 +60 = 80

I(t) = 10 +30 = 40

نفرض ان D = 50 اذن

I(1) = 40-50 = -10

$I^{+}(1)$ = max{-10,0} = 0 .....hcost = 0

$I^{-}(0)$ = max{-(-10),0} = 10 ..... scost =5(10) =50

عند t = 3

Z = 40-(-10) = 50 ; I(2) < s

Dcost = 20+2(50) = 20+100 = 120

I(t) = -10 +50 = 40

نفرض ان D = 40 اذن

I(1) = 40-40 = 0

$I^{+}(1)$ = max{0,0} = 0 ..... hcost = 0

$I^{-}(0)$ = max{0,0} = 0 ..... scost = 0

** يتم حساب باقي التكاليف بنفس الطريقة.

متوسط تكاليف الطلبيات = $\frac{20+20+80+120}{4} = \frac{240}{4} = 60$

متوسط تكاليف الملكية: $\frac{30+10+0+0}{4} = \frac{50}{4} = 12.5$

متوسط تكاليف التقصير = $\frac{0+0+50+0}{4} = \frac{50}{4} = 12.4$

التكلفة الكلية = ٦٠ + ١٢,٥ + ١٢,٥ = ٦٠ + ٢٥ = ٧٥

```
#include <conio.h>
#include <iostream.h>
void main()
{
int i,is,z,dcost[9],hcost[9],scost[9],s1[9],s2[9],t,k,d;
double hc,sc,dc;

for(t=1;t<=9;t++)
{
cout<<"enter Less the amount of goods must exist in the store s=";
cin>>s1[t];
}
for(t=1;t<=9;t++)
{
cout<<"enter The largest amount of goods should their presence in the store S=";
cin>>s2[t];
}


i=2;
is=60;
k=20;
for(t=0;t<=3;t++)
{
if(is>s1[1])
{
```

```
z=0;
dcost[t]=k+i*z;
cout<<"enter demand d=";
cin>>d;
is=is-d;
if(is>0)
hcost[t]=is;
else
hcost[t]=0;
if((-1*is)>0)
scost[t]=is;
else
scost[t]=0;
}
else {
z=s2[1]-is;
dcost[t]=k+i*z;
is=is+z;
cout<<"enter demand d=";
cin>>d;
is=is-d;
if(is>0)
hcost[t]=is;
else
hcost[t]=0;
if((-1*is)>0)
scost[t]=-1 * is*5;
```

```
else
scost[t]=0;
}
}

for(t=0;t<=3;t++)
{
hc=hc+hcost[t];
cout<<scost[t]<<endl;
sc=sc+scost[t];
dc=dc+dcost[t];
}
dc=dc/4;
sc=sc/4;
hc=hc/4;
 cout<<"the avrg order dcost="<<dc<<endl;
 cout<<"the avrg holding cost hcost="<<hc<<endl;
 cout<<"the avrg shortage cost scost="<<sc<<endl;
 cout<<"the sum all cost ="<<(hc+dc+sc);

 getch();
}
```