

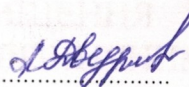
**СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ**  
Навчально-науковий інститут транспорту та будівництва  
Кафедра логістичного управління та безпеки руху на транспорті

**ПОЯСНОВАЛЬНА ЗАПИСКА**  
до кваліфікаційної роботи  
освітньо-кваліфікаційного рівня магістр

галузі знань 27 – «Транспорт»  
спеціальності 273 «Залізничний транспорт». Інтероперабельність і безпека на залізничному транспорті

на тему: «Підвищення рівня інтероперабельності вагонопотоків на станціях стикування колій різної ширини»


Виконав: студент групи ІБЗТ-19зм  
Двуречанська Л.В.

  
.....  
(підпис)

Керівник: доц. Семенов С.О.

  
.....  
(підпис)

Завідувач кафедри: проф. Чернецька-Білецька Н.Б.

  
.....  
(підпис)

Рецензент: Шерест О.М.

.....  
(підпис)

### 3.5 Програми розрахунку та результати моделювання

Далі наведено викладки з програм для розрахунку різних залежностей.

Програма побудови залежності мінімальних витрат  $Z$  від параметра  $c$

ПРОГРАМА МОВОЮ Maple 8

```
>restart: with (plots) :
Warning, the name changecoords has been redefined
>k:=3;
>f:=x->sum(x^i, i=1..k);

$$f := x \rightarrow \sum_{i=1}^k x^i$$

>MZS:=proc (A, B, E, F)
local a, b, e, x1, x2, f1, f2, f;
global y;
a:=A: e:=E: f:=F: b:=B: x1:=evalf (a+(3sqrt(5))*(b-a)/2,
9) :x2:=evalf(a+(-1+sqrt(5))*(b-a)/2,
9) : f1:=evalf(f(x1),
9) : f2:=evalf(f(x2),
9) : while (b-a)>e do if (f1<f2) then b:=x2 : x2:=x1 : f2:=f1 :
x1:=evalf (a+(3sqrt(5))*(b-a)/2,
9) : f1:=f(x1):b-a:=fi:if(f1<f2) then
a:=x1 : x1:=x2 : f1:=f2 : x2:=evalf(a+(-1+sqrt(5))*(b-a)/2,
9) : f2:=f(x2) : b-a : fi:if (f1<f2) then a:=x1 : b:=x2 : b-a:=fi:od:y:=(a+b)/2: end:
>
R:=array (1..500) : ZZ:=array (1..500) : C:=array (1..500):
>m:=1: for c from 0.01 by 0.01 to 1 do z0:=1000: for r from 0.01 by 0.01 to k do
A:=0.01: B:=k-0.01 : E:= 0.0001: F:=x->abs(f(x)-r) :MZS (A,B,E,F):
po:=(1-y)-y^2/((1-y))^2+(k-1)*(1-y)-y^2+y^(k+1)+r*(1-y)):
Mcp:=r*po/((1-y^k)*(1-y)): Z:=po+c*Mcp: if z0>Z then r0:=r z0:=Z Tsp0=Tsp
end if: end
do:=R[m]:=r0: ZZ[m]:=5*z0: C[m]:=c: m:=m+1: end
do:=pR:=plot([c[m1],R[m1],m1=1..m-1], color=blue):
pZZ:=plot([c[m1],R[m1],m1=1..m-1]): display( {pR,pZZ} ).
```

*Програми розрахунку та результати моделювання раціональних параметрів  
для технології перевантаження вантажів*

Програмна реалізація методу фаз:

> **restart; with(linalg):**  
> **n:=3:** кількість фаз (i)  
> **kol:=10:** кількість вагонів

Масив витрат часу на відповідні операції у першій фазі

> **T[1]:=[110,140]; q[1]:=2;**

**Необхідно упорядкувати масиви.**

$$T_1 := [110, 140]$$

$$q_1 := 2$$

Масив витрат грошових ресурсів на відповідні операції в першій фазі

> **Z[1]:=[250,550]:**

Масив витрат часу на відповідні операції в другій фазі

> **T[2]:=[30,50]; q[2]:=2:**

$$T_2 := [40, 30]$$

Масив витрат грошових ресурсів на відповідні операції в другій фазі

> **Z[2]:=[300,550]:**

Масив витрат часу на відповідні операції в третій фазі

> **T[3]:=[6]; q[3]:=1;**

$$T_3 := [6]$$

$$q_3 := 1$$

Масив витрат грошових ресурсів на відповідні операції в третій фазі

> **Z[3]:=[1200]:**

Масив витрат часу на відповідні операції в четвертій фазі

> **T[4]:=[0.32\*60,0.48\*60]; q[1]:=2;**

$$T_4 := [19.20, 28.80]$$

$$q_1 := 2$$

Масив витрат грошових ресурсів на відповідні операції в четвертій фазі

> **Z[4]:=[750,1000]:**

Масив витрат часу на відповідні операції в п'ятій фазі

> **T[5]:=[0.25\*24\*60,0.5\*24\*60,24\*60,1.25\*24\*60,1.5\*24\*60]; q[1]:=5;**

$T_5 := [360.00, 720.0, 1440, 1800.00, 2160.0]$

$q_1 := 5$

Масив витрат грошових ресурсів на відповідні операції в п'ятій фазі

> **Z[5]:=[200,250,300,350,400]:**

>

Масив витрат часу на відповідні операції в шостій фазі

> **T[6]:=[30]; q[1]:=1;**

$T_6 := [30]$

$q_1 := 1$

Масив витрат грошових ресурсів на відповідні операції в шостій фазі

> **Z[6]:=[400]:**

>

Масив витрат часу на відповідні операції в сьомій фазі

> **T[7]:=[90]; q[1]:=1;**

$T_7 := [90]$

$q_1 := 1$

Масив витрат грошових ресурсів на відповідні операції в сьомій фазі

> **Z[7]:=[300]:**

> **X1 := array(1..1000):Y1 := array(1..1000):Z1 := array(1..1000):**

> **j1:=[]:k:=1:**

> **V:=[]:**

> **for lambda from 0.01 by 0.1 to 61**

```

> do
>   z:=0:t:=0:
>   for i from 1 to n
>     do
>       p:=infinity;
>       for j from 1 to q[i]
>         do
>           a:=Z[i][j]+lambda*T[i][j];
>           if a<p
>             then
>               p:=a;
>               j0:=j;
>             fi;
>         od;
>       j1[i]:=j0;
>       V:=[seq(w[i,j1[i]],i=1..n)];
>       z:=z+Z[i][j0];
>       t:=t+T[i][j0];
>     od;
>   if z0<>z then
>     X1[k]:=z:Масив грошових витрат на реалізацію селектора при
конкретному «мю»
>     Y1[k]:=t:Масив витрат часу на реалізацію селектора при
конкретному «мю»

>     Z1[k]:=lambda:Відповідний масив «мю».
>     z0:=z:
>     t0:=t:
>     lambda0:=lambda:
>     print(V,lambda0,z0,t0):

```

```

> k:=k+1:
> fi;
> od:
>

```

**Безліч незрівнянних за Парето селекторів при конкретному «лямбда»:**

$[w_{1,1}, w_{2,1}, w_{3,1}], 0.01, 1470, 156$

$[w_{1,1}, w_{2,2}, w_{3,1}], 28.01, 1750, 146$

*Побудова графіків*

```

> with(plots):
> p1:=plot([Z1[m],Y1[m],m=1..k-1],style=line,color=blue,thickness=3):
> p2:=plot([Z1[m],X1[m],m=1..k-1],style=line,thickness=3):
> display({p1,p2},title=`Функції залежності часу обробки та витрат від
«лямбда»`);

```

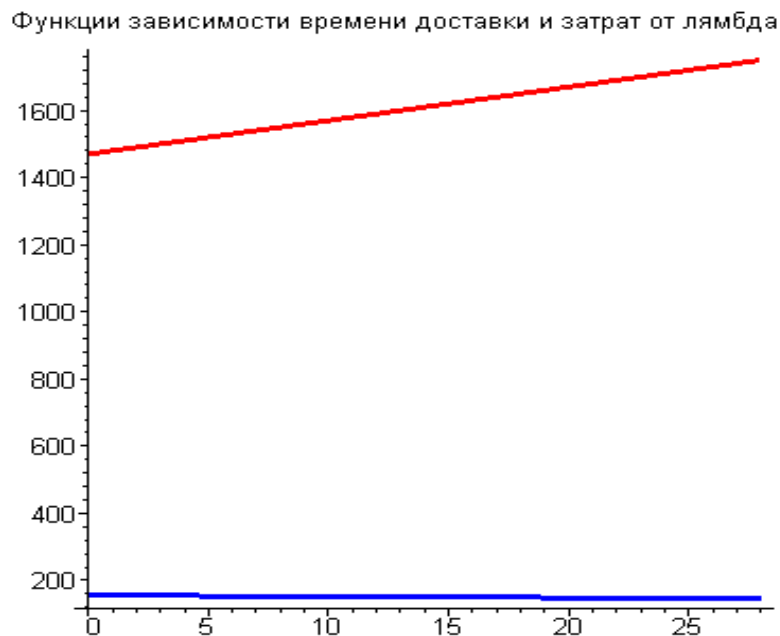


Рис. 3.15 Графік функції залежності часу обробки та грошових витрат на перевантажування вагонів від  $\lambda$

```
> p1:=plot([Y1[m],X1[m],m=1..k-
1],style=line,color=blue,thickness=3):display({p1},title=`График функции
зависимости затрат от времени`,labels=[Время,Затраты]);
```

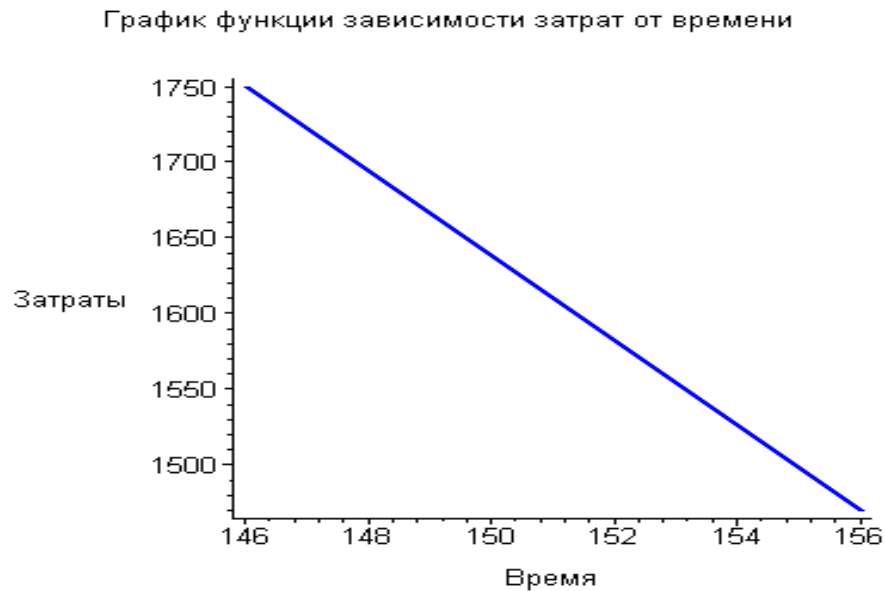


Рис. 3.16 График функції залежності витрат грошей від витрат часу при перевантажуванні вагонів

### Програмна реалізація методу Лагранжа.

```
> restart;
```

#### Процедура знаходження мінімального елемента масиву.

```
> minimum:=proc(C,n)
>   local i;
>   global min_,num_min;
>
>
>   min_:=C[1]; num_min:=1;
>   for i from 2 to n do
>       if C[i]<min_ then min_:=C[i]; num_min:=i;
```

```
> fi;  
> od;  
> end:
```

---

```
> eps:=0.001: точність рішення  
> M:=7: N:=5: M - кількість фаз, N - найбільша кількість операцій у  
всіх фазах.
```

```
>
```

```
TT:=array(1..M,1..N,[[110,140],[40,50],[6],[0.32*60,0.48*60],[0.25*24*60,0.5*  
24*60,24*60,1.25*24*60,1.5*24*60],[30],[90]]); матриця витрат часу
```

$$TT := \begin{bmatrix} 110 & 140 & TT_{1,3} & TT_{1,4} & TT_{1,5} \\ 40 & 30 & TT_{2,3} & TT_{2,4} & TT_{2,5} \\ 6 & TT_{3,2} & TT_{3,3} & TT_{3,4} & TT_{3,5} \\ 19.20 & 28.80 & TT_{4,3} & TT_{4,4} & TT_{4,5} \\ 360.00 & 720.0 & 1440 & 1800.00 & 2160.0 \\ 30 & TT_{6,2} & TT_{6,3} & TT_{6,4} & TT_{6,5} \\ 90 & TT_{7,2} & TT_{7,3} & TT_{7,4} & TT_{7,5} \end{bmatrix}$$

```
> ZZ:=array(1..M,1..N,[[250,550],[  
300,550],[1200],[750,1000],[200,250,300,350,400],[400],[300]]): матриця витрат  
грошових коштів
```

```
> KOL_oper:=array(1..M,[2,2,1,2,5,1,1]); Кількість операцій у кожній фазі.
```

$$KOL\_oper := [2, 2, 1, 2, 5, 1, 1]$$

---

```
> allres:=0;
```

$$allres := 0$$

```
> for z from 1 to M do
```

```
>
```

```
> SIZE_MAS:=KOL_oper[z];
```

```
> T:=array(1..SIZE_MAS);
```



```

> Z:=array(1..SIZE_MAS);
>
> for i from 1 to SIZE_MAS do
> T[i]:=TT[z,i]; od;
>
> for i from 1 to SIZE_MAS do
> Z[i]:=ZZ[z,i]; od;
>
> SIZE_RES:=sum(k,'k=1..SIZE_MAS-1'):
> mas:=array(1..SIZE_MAS):
> res:=array(1..SIZE_RES,1..3):
> Res_mu:=array(1..SIZE_RES,1..3):
> p_mu:=0:
> Res:=array(1..SIZE_RES+1,1..3):
> for i from 1 to SIZE_MAS do mas[i]=(Z[i]+mu*T[i]); od:
> p:=0:
> for i from 1 to SIZE_MAS-1 do
>                                     обчислення усіх можливих точок
перетину рівнянь
>   for j from i+1 to SIZE_MAS do
>     p:=p+1;
>     res[p,1]:=solve(mas[i]=mas[j],mu);
>     res[p,2]:=i;
>     res[p,3]:=j;
>   od;
>   SIZE_RES:=SIZE_RES-1;
>
> od:
> #print(res,p);
> flag:=true:
> while flag do
>                                     сортування точок перетину за зростанням X

```

```

> flag := false;
> for i from 1 to p-1 do
>
> if res[i,1]>res[i+1,1] then
>     flag:=true:
>     Temp[1]:=res[i,1];
>     Temp[2]:=res[i,2];
>     Temp[3]:=res[i,3];
>     res[i,1]:=res[i+1,1];
>     res[i,2]:=res[i+1,2];
>     res[i,3]:=res[i+1,3];
>     res[i+1,1]:=Temp[1];
>     res[i+1,2]:=Temp[2];
>     res[i+1,3]:=Temp[3];
> fi; od; od:
> #print(res);
>
> for i from 1 to p do      вибір точок перетинів лише з додатними X, та
розрахунок їх кількості в p_mu
> if res[i,1]>=0 then
> p_mu:=p_mu+1:
> Res_mu[p_mu,1]:=res[i,1];
> Res_mu[p_mu,2]:=res[i,2];
> Res_mu[p_mu,3]:=res[i,3];
> fi; od:
>> #print(Res_mu):
>> if p_mu>0 then      перевірка: чи є перетини з додатними X у
виконанні основного алгоритму.
>
> m_psi:=array(1..SIZE_MAS):

```

```

>
> kol_res:=0; line:=0; begX:=0;
>
> for j from 1 to p_mu do      перебирання усіх точок перетину
> mu:=Res_mu[j,1]:
>
> for i from 1 to SIZE_MAS do  значення усіх прямих в точці перетину
> m_psi[i]:=mas[i]: od:
>
> minimum(m_psi,SIZE_MAS); print(m_psi,min_,mu);  пошук мінімуму
>
> for i from 1 to SIZE_MAS do
> #print(i,num_min,m_psi[i],min_);
> if (i<>num_min) and (m_psi[i]-min_ <=eps) then умова того,що точка
перетину є переходом на другу грань
>
> майже дорівнює
> kol_res:=kol_res+1;      лічильник кількості граней, що утворюють
рішення
> if kol_res=1 then      алгоритм розпочинається з першого рівняння
> Res[kol_res,1]:=1;
> Res[kol_res,2]:=0;
> Res[kol_res,3]:=mu;
>
> else
> Res[kol_res,1]:=last_line;
> Res[kol_res,2]:=endX;
> Res[kol_res,3]:=mu;
> fi;
>

```

> endX:=mu;

запам'ятовуємо значення X

> if i>num\_min then last\_line:=i; else last\_line:=num\_min; fi;

запам'ятовуємо номер грані

> fi; od; od;

>

> if kol\_res<>0 then

> kol\_res:=kol\_res+1;            випадок останнього ребра, від останнього X  
до нескінченності,

> Res[kol\_res,1]:=last\_line;    побудова рішення для і фази

> Res[kol\_res,2]:=endX;

> Res[kol\_res,3]:=infinity;

>

> else

> kol\_res:=kol\_res+1;

> Res[kol\_res,1]:=1;

> Res[kol\_res,2]:=0;

> Res[kol\_res,3]:=infinity;

> fi;

>

> else            випадок, коли немає жодного перетину з додатним X,  
(відповідь - перша пряма)

> kol\_res:=1;

> Res[kol\_res,1]:=1;

> Res[kol\_res,2]:=0;

> Res[kol\_res,3]:=infinity;

> fi;

> FSIZE[z]:=kol\_res; allres:=allres+kol\_res;

> FRES[z]:=copy(Res);

> unassign('mu');

> od;

[1140, 1140], 1140, 28

> print(FRES,FSIZE);

table([1 =  $\begin{bmatrix} 1 & 0 & \infty \\ w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix}$ , 2 =  $\begin{bmatrix} 1 & 0 & 28 \\ 2 & 28 & \infty \end{bmatrix}$ , 3 = [1 0  $\infty$ ], 4 =  $\begin{bmatrix} 1 & 0 & \infty \\ w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix}$ ,

5 =  $\begin{bmatrix} 1 & 0 & \infty \\ w_{2,1} & w_{2,2} & w_{2,3} \\ w_{3,1} & w_{3,2} & w_{3,3} \\ w_{4,1} & w_{4,2} & w_{4,3} \\ w_{5,1} & w_{5,2} & w_{5,3} \\ w_{6,1} & w_{6,2} & w_{6,3} \\ w_{7,1} & w_{7,2} & w_{7,3} \\ w_{8,1} & w_{8,2} & w_{8,3} \\ w_{9,1} & w_{9,2} & w_{9,3} \\ w_{10,1} & w_{10,2} & w_{10,3} \\ w_{11,1} & w_{11,2} & w_{11,3} \end{bmatrix}$ , 6 = [1 0  $\infty$ ],

7 = [1 0  $\infty$ ]

), table([1 = 1, 2 = 2, 3 = 1, 4 = 1, 5 = 1, 6 = 1, 7 = 1])

> pt:=1: лічильник «МЮ»

> for k from 1 to M do по усіх фазах

>

> for i from 1 to FSIZE[k] do по усіх операціях у фазах

>

> if i<>1 then

> pt:=pt+1;

> mm[pt]:=FRES[k][i,2]; складаємо масив усіх «МЮ»

> fi;

> od;

> od;

>

> mm[1]:=0: pt:=pt+1: mm[pt]:=infinity:

> print(mm,pt);

table([1 = 0, 2 = 28, 3 =  $\infty$ ]), 3

```

> flag:=true:
> while flag do                сортування масиву «мю» за зростанням X
>   flag := false;
>   for i from 1 to pt-1 do
>     if mm[i]>mm[i+1] then
>       flag:=true:
>       temp:=mm[i];
>       mm[i]:=mm[i+1];
>       mm[i+1]:=temp;
>   fi; od; od:
> for i from 1 to pt-1 do      по усіх «мю»
>
>   for j from 1 to M do      по фазах
>     for k from 1 to FSIZE[j] do перебираємо усе рішення у фазі
>       if (mm[i]>=FRES[j][k,2]) and (mm[i+1]<=FRES[j][k,3]) then
>         result[i][j]:=FRES[j][k,1];
>       fi;
>     od;
>   od;   V:=[seq(w[x,result[i][x]],x=1..M)];
>   print(V);
> od:

```

**Безліч незрівнянних за Парето селекторів:**

$$[w_{1,1}, w_{2,1}, w_{3,1}, w_{4,1}, w_{5,1}, w_{6,1}, w_{7,1}]$$

$$[w_{1,1}, w_{2,2}, w_{3,1}, w_{4,1}, w_{5,1}, w_{6,1}, w_{7,1}]$$

```

> print(result);
      table([1 = table([1 = 1, 2 = 1, 3 = 1, 4 = 1, 5 = 1, 6 = 1, 7 = 1]),
            2 = table([1 = 1, 2 = 2, 3 = 1, 4 = 1, 5 = 1, 6 = 1, 7 = 1])
            ])
>
> plotmu:=array(1..100): plotZ:=array(1..100): plotT:=array(1..100):
> for i from 1 to pt-1 do

```

```
> plotmu[i]:=mm[i];
> od:
>
> for i from 1 to pt-1 do
>
> z:=0; t:=0;
>
> for j from 1 to M do
> z:=z+ZZ[j,result[i][j]];
> t:=t+TT[j,result[i][j]];
> od;
> plotZ[i]:=z;
> plotT[i]:=t;
> od:
>
```

### Побудова графіків

```
> with(plots):
```

Warning, the name changecoords has been redefined

```
> p1:=plot([plotmu[m],plotT[m],m=1..pt-1],color=blue,thickness=3):
```

```
> p2:=plot([plotmu[m],plotZ[m],m=1..pt-1],thickness=3):
```

```
> display({p1,p2},title=` Функції залежності часу обробки та грошових  
витрат від «лямбда»`);
```

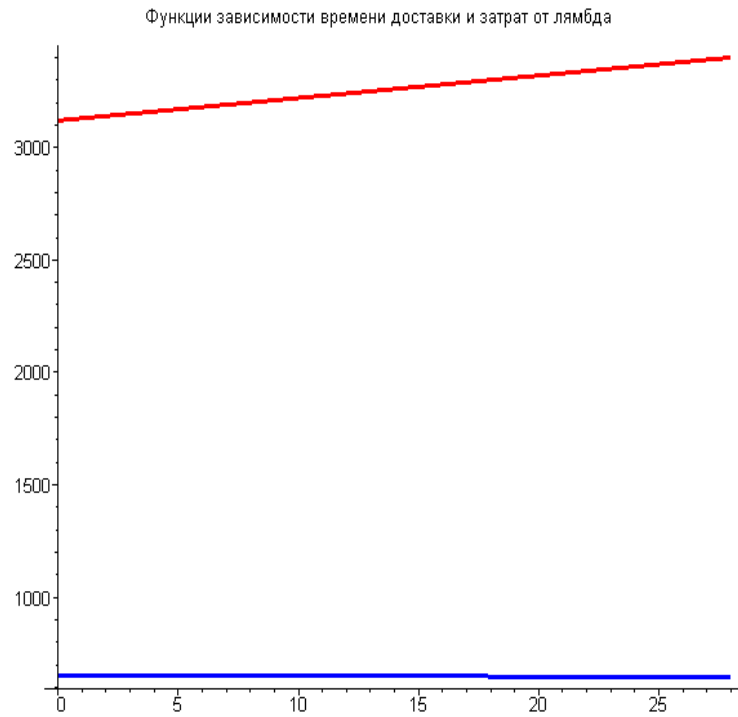


Рис. 3.17 Графік функції залежності часу обробки вагонів та грошових витрат при перевантажуванні від  $\lambda$

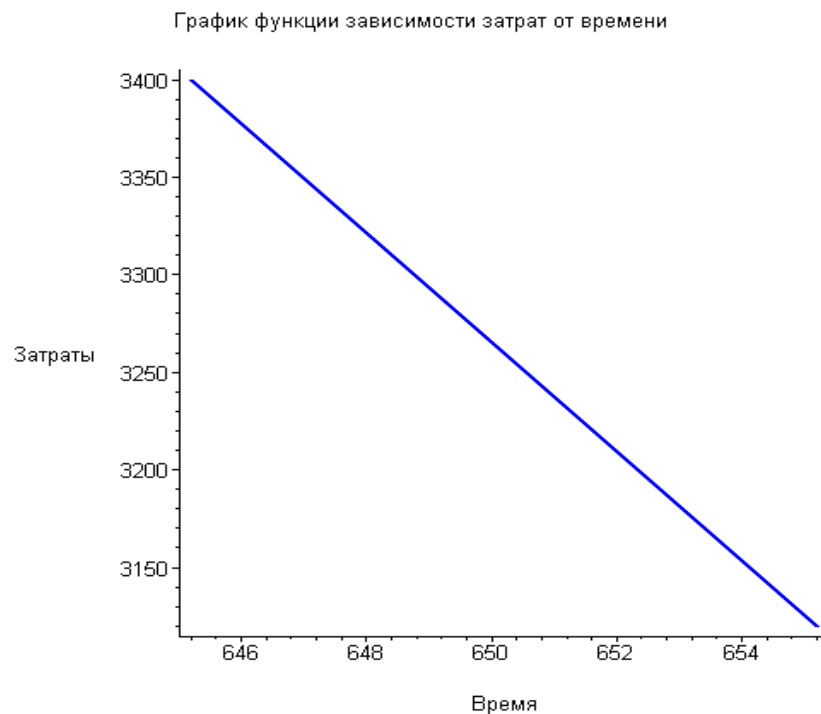


Рис. 3.18 Графік функції залежності грошових витрат від часу обробки вагонів при їх перевантажуванні



```
> p:=plot([plotT[m],plotZ[m],m=1..pt-1],color=blue,thickness=3):  
> display({p},title=`Графік функції залежності грошових витрат від  
часу`,labels=[Время,Затраты]).
```

*Програми розрахунку та результати моделювання раціональних параметрів  
для технології перестановки візків вантажних вагонів*

Текст програми.

```
> restart: with(linalg):
```

Warning, the protected names norm and trace have been redefined and unprotected

```
> NV:=4; кількість вагонів
```

```
NV:=4
```

коефіцієнти витрат на перестановку состава при перевищенні кількості вагонів пропускної здатності перестановочного пункту

```
> if (NV<=5) then koef:=1; koef2:=1; fi;
```

```
> if (NV>5) and (NV<=10) then koef:=2; koef2:=1; fi:
```

```
> if (NV>10) and (NV<16) then koef:=3; koef2:=2; fi:
```

```
> print(koef,koef2);
```

```
1, 1
```

```
> n:=16: число фаз(i)
```

Масив витрат часу на відповідні операції в першій фазі

```
> T[1]:=[28]: q[1]:=1:
```

Масив витрат коштів на відповідні операції в першій фазі

```
> Z[1]:=[110]:
```

Масив витрат часу на відповідні операції в другій фазі

```
> T[2]:=[20,50]: q[2]:=2:
```

Масив витрат коштів на відповідні операції в другій фазі

> Z[2]:=[120,80]:

Масив витрат часу на відповідні операції в третій фазі

> T[3]:=[5]: q[3]:=1:

Масив витрат коштів на відповідні операції в третій фазі

> Z[3]:=[195]:

Масив витрат часу на відповідні операції в четвертій фазі

> T[4]:=[20\*koef,40\*koef2,60]: q[4]:=3:

Масив витрат коштів на відповідні операції в четвертій фазі

> Z[4]:=[70\*koef,70\*koef2,70]:

Масив витрат часу на відповідні операції в п'ятій фазі

> T[5]:=[20\*koef,40\*koef2,60]: q[5]:=3:

Масив витрат коштів на відповідні операції в п'ятій фазі

> Z[5]:=[30\*koef,44\*koef2,56]:

Масив витрат часу на відповідні операції в шостій фазі

> T[6]:=[10\*koef,20\*koef2,30]: q[6]:=3:

Масив витрат коштів на відповідні операції в шостій фазі

> Z[6]:=[80\*koef,90\*koef2,100]:

Масив витрат часу на відповідні операції в сьомій фазі

> T[7]:=[10\*koef,15\*koef2,20]: q[7]:=3:

Масив витрат коштів на відповідні операції в сьомій фазі

> Z[7]:=[20\*koef,25\*koef2,30]:

Масив витрат часу на відповідні операції у восьмій фазі

> T[8]:=[35\*koef,70\*koef2,100]: q[8]:=3:

Масив витрат коштів на відповідні операції у восьмій фазі

> Z[8]:=[20\*koef,25\*koef2,30]:

Масив витрат часу на відповідні операції в дев'ятій фазі

> T[9]:=[5\*koef,10\*koef2,15]: q[9]:=3:

Масив витрат коштів на відповідні операції в дев'ятій фазі

> Z[9]:=[80\*koef,120\*koef2,160]:

Масив витрат часу на відповідні операції в десятій фазі

> T[10]:=[8\*coef,16\*coef2,24,12\*coef,12\*coef2,12]: q[10]:=6:

Масив витрат коштів на відповідні операції в десятій фазі

> Z[10]:=[40\*coef,52\*coef2,64,70\*coef,75\*coef2,80]:

Масив витрат часу на відповідні операції в одинадцятій фазі

> T[11]:=[10\*coef,20\*coef2,30,10\*coef,10\*coef2,10]: q[11]:=6:

Масив витрат коштів на відповідні операції в одинадцятій фазі

> Z[11]:=[30\*coef,45\*coef2,60,70\*coef,70\*coef2,70]:

Масив витрат часу на відповідні операції у дванадцятій фазі

> T[12]:=[55\*coef,100\*coef2,145]: q[12]:=3:

Масив витрат коштів на відповідні операції у дванадцятій фазі

> Z[12]:=[170\*coef,210\*coef2,250]:

Масив витрат часу на відповідні операції в тринадцятій фазі

> T[13]:=[5]: q[13]:=1:

Масив витрат коштів на відповідні операції в тринадцятій фазі

> Z[13]:=[20]:

Масив витрат часу на відповідні операції в чотирнадцятій фазі

> T[14]:=[19\*coef,38\*coef2,57]: q[14]:=3:

Масив витрат коштів на відповідні операції в чотирнадцятій фазі

> Z[14]:=[15\*coef,20\*coef2,25]:

Масив витрат часу на відповідні операції в п'ятнадцятій фазі

> T[15]:=[5\*coef,10\*coef2,15]: q[15]:=3:

Масив витрат коштів на відповідні операції в п'ятнадцятій фазі

> Z[15]:=[140\*coef,155\*coef2,170]:

Масив витрат часу на відповідні операції в шістнадцятій фазі

> T[16]:=[90]: q[16]:=1:

Масив витрат коштів на відповідні операції в шістнадцятій фазі

> Z[16]:=[120]:

---

Матриці обмежень:

> M[1]:=array([1,1]);

$$M_1 := [1, 1]$$

> M[2]:=array([[1],[1]]);

$$M_2 := \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

> M[3]:=array([1,1,1]);

$$M_3 := [1, 1, 1]$$

> M[4]:=array([[1,0,0],[0,1,0],[0,0,1]]);

$$M_4 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

> M[5]:=array([[1,0,0],[0,1,0],[0,0,1]]):

> M[7]:=copy(M[5]):

> M[8]:=copy(M[5]):

> M[9]:=array([[1,0,0,1,0,0],[0,1,0,0,1,0],[0,0,1,0,0,1]]):

$$M_9 := \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

>

M[10]:=array([[1,0,0,1,0,0],[0,1,0,0,1,0],[0,0,1,0,0,1],[1,0,0,1,0,0],[0,1,0,0,1,0],[0,1,0,0,1,0]]);

$$M_{10} := \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

> M[11]:=array([[1,0,0],[0,1,0],[0,0,1],[1,0,0],[0,1,0],[0,0,1]]);

$$M_{11} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

> M[12]:=array([[1],[1],[1]]);

$$M_{12} := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

> M[13]:=array([1,1,1]);

$$M_{13} := [1, 1, 1]$$

> M[14]:=copy(M[5]);

$$M_{14} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

> M[15]:=array([[1],[1],[1]]);

$$M_{15} := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

=====  
 Дозволені варіанти селекторів виходячи з обмежень матриць M[i]:

> G[1]:=array([1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]):

> G[2]:=array([1,1,1,2,2,2,2,2,2,2,2,2,1,2,2,1]):

> G[3]:=array([1,1,1,3,3,3,3,3,3,3,3,3,1,3,3,1]):

> G[4]:=array([1,1,1,1,1,1,1,1,1,4,4,1,1,1,1,1]):

> G[5]:=array([1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1]):

> G[6]:=array([1,1,1,2,2,2,2,2,2,5,5,2,1,2,2,1]):

> G[7]:=array([1,1,1,3,3,3,3,3,3,6,6,3,1,3,3,1]):

> G[8]:=array([1,2,1,1,1,1,1,1,1,4,4,1,1,1,1,1]):

> G[9]:=array([1,2,1,2,2,2,2,2,2,2,2,2,1,2,2,1]):

> G[10]:=array([1,2,1,3,3,3,3,3,3,3,3,3,1,3,3,1]):

> G[11]:=array([1,2,1,2,2,2,2,2,2,5,5,2,1,2,2,1]):

> G[12]:=array([1,2,1,3,3,3,3,3,3,6,6,3,1,3,3,1]): s:=12: кількість селекторів



```
> kol_res:=kol_res+1;
> mres[kol_res,1]:=lambda;
> mres[kol_res,2]:=p;
> fi; fi; print(lambda,p);
> od:
[ 1263.45, 1426.27, 1587.04, 1333.49, 1223.75, 1474.13, 1612.72, 1293.79, 1386.57,
  1547.34, 1434.43, 1573.02 ]
0.01, 5
[ 1332.45, 1531.67, 1727.84, 1403.29, 1298.75, 1576.73, 1747.12, 1369.59, 1497.97,
  1694.14, 1543.03, 1713.42 ]
0.21, 5
[ 1401.45, 1637.07, 1868.64, 1473.09, 1373.75, 1679.33, 1881.52, 1445.39, 1609.37,
  1840.94, 1651.63, 1853.82 ]
0.41, 5
[ 1470.45, 1742.47, 2009.44, 1542.89, 1448.75, 1781.93, 2015.92, 1521.19, 1720.77,
  1987.74, 1760.23, 1994.22 ]
0.61, 5
[ 1539.45, 1847.87, 2150.24, 1612.69, 1523.75, 1884.53, 2150.32, 1596.99, 1832.17,
  2134.54, 1868.83, 2134.62 ]
0.81, 5
[ 1608.45, 1953.27, 2291.04, 1682.49, 1598.75, 1987.13, 2284.72, 1672.79, 1943.57,
  2281.34, 1977.43, 2275.02 ]
1.01, 5
[ 1677.45, 2058.67, 2431.84, 1752.29, 1673.75, 2089.73, 2419.12, 1748.59, 2054.97,
  2428.14, 2086.03, 2415.42 ]
1.21, 5
[ 1746.45, 2164.07, 2572.64, 1822.09, 1748.75, 2192.33, 2553.52, 1824.39, 2166.37,
  2574.94, 2194.63, 2555.82 ]
1.41, 1
[ 1815.45, 2269.47, 2713.44, 1891.89, 1823.75, 2294.93, 2687.92, 1900.19, 2277.77,
  2721.74, 2303.23, 2696.22 ]
1.61, 1
[ 1884.45, 2374.87, 2854.24, 1961.69, 1898.75, 2397.53, 2822.32, 1975.99, 2389.17,
  2868.54, 2411.83, 2836.62 ]
```

1.81, 1

[ 1953.45, 2480.27, 2995.04, 2031.49, 1973.75, 2500.13, 2956.72, 2051.79, 2500.57, 3015.34, 2520.43, 2977.02 ]

2.01, 1

[ 2022.45, 2585.67, 3135.84, 2101.29, 2048.75, 2602.73, 3091.12, 2127.59, 2611.97, 3162.14, 2629.03, 3117.42 ]

2.21, 1

[ 2091.45, 2691.07, 3276.64, 2171.09, 2123.75, 2705.33, 3225.52, 2203.39, 2723.37, 3308.94, 2737.63, 3257.82 ]

2.41, 1

[ 2160.45, 2796.47, 3417.44, 2240.89, 2198.75, 2807.93, 3359.92, 2279.19, 2834.77, 3455.74, 2846.23, 3398.22 ]

2.61, 1

[ 4092.45, 5747.67, 7359.84, 4195.29, 4298.75, 5680.73, 7123.12, 4401.59, 5953.97, 7566.14, 5887.03, 7329.42 ]

.....

8.21, 1

[ 4161.45, 5853.07, 7500.64, 4265.09, 4373.75, 5783.33, 7257.52, 4477.39, 6065.37, 7712.94, 5995.63, 7469.82 ]

8.41, 1

[ 4230.45, 5958.47, 7641.44, 4334.89, 4448.75, 5885.93, 7391.92, 4553.19, 6176.77, 7859.74, 6104.23, 7610.22 ]

8.61, 1

[ 4299.45, 6063.87, 7782.24, 4404.69, 4523.75, 5988.53, 7526.32, 4628.99, 6288.17, 8006.54, 6212.83, 7750.62 ]

8.81, 1

[ 4368.45, 6169.27, 7923.04, 4474.49, 4598.75, 6091.13, 7660.72, 4704.79, 6399.57, 8153.34, 6321.43, 7891.02 ]

9.01, 1

[ 4437.45, 6274.67, 8063.84, 4544.29, 4673.75, 6193.73, 7795.12, 4780.59, 6510.97, 8300.14, 6430.03, 8031.42 ]

9.21, 1



```
[ 4506.45 , 6380.07 , 8204.64 , 4614.09 , 4748.75 , 6296.33 , 7929.52 , 4856.39 , 6622.37 ,  
8446.94 , 6538.63 , 8171.82 ]
```

```
9.41, 1
```

```
[ 4575.45 , 6485.47 , 8345.44 , 4683.89 , 4823.75 , 6398.93 , 8063.92 , 4932.19 , 6733.77 ,  
8593.74 , 6647.23 , 8312.22 ]
```

```
9.61, 1
```

```
[ 4644.45 , 6590.87 , 8486.24 , 4753.69 , 4898.75 , 6501.53 , 8198.32 , 5007.99 , 6845.17 ,  
8740.54 , 6755.83 , 8452.62 ]
```

```
9.81, 1
```

```
> print(mres);
```

```
[ 0.01      5  
1.41      1  
mres3,1  mres3,2  
mres4,1  mres4,2  
mres5,1  mres5,2  
mres6,1  mres6,2  
mres7,1  mres7,2  
mres8,1  mres8,2  
mres9,1  mres9,2  
mres10,1 mres10,2 ]
```

```
> XX:=array(1..20,[]): YY:=array(1..20,[]):
```

```
> ZZ:=array(1..20,[]):
```

масиви результатів для графіків

```
> X:=0: Y:=0:
```

```
> for i from 1 to kol_res do
```

```
> ZZ[i]:=mres[i,1];
```

нараховуємо результати для побудови графіка розв'язку

```
> for j from 1 to n do
```

```
> X:=X+Z[j][ G[mres[i,2]][j] ]; масив витрат
```

```
> Y:=Y+T[j][ G[mres[i,2]][j] ]; масив часу
```

```
> od;
```

```
>
```

```
> XX[i]:=X; YY[i]:=Y;
```

```
> X:=0: Y:=0:
```

> od:

> print(XX,YY,ZZ);

```
[1220, 1260, XX3, XX4, XX5, XX6, XX7, XX8, XX9, XX10, XX11, XX12, XX13, XX14, XX15,  
XX16, XX17, XX18, XX19, XX20], [375, 345, YY3, YY4, YY5, YY6, YY7, YY8, YY9, YY10,  
YY11, YY12, YY13, YY14, YY15, YY16, YY17, YY18, YY19, YY20], [0.01, 1.41, ZZ3, ZZ4, ZZ5,  
ZZ6, ZZ7, ZZ8, ZZ9, ZZ10, ZZ11, ZZ12, ZZ13, ZZ14, ZZ15, ZZ16, ZZ17, ZZ18, ZZ19, ZZ20]
```

> for i from 1 to kol\_res do

> V:=[seq(w[j, G[mres[i,2]][j]],j=1..n)];

> print(V) od:

```
[w1,1, w2,2, w3,1, w4,1, w5,1, w6,1, w7,1, w8,1, w9,1, w10,1, w11,1, w12,1, w13,1, w14,1, w15,1,  
w16,1]
```

```
[w1,1, w2,1, w3,1, w4,1, w5,1, w6,1, w7,1, w8,1, w9,1, w10,1, w11,1, w12,1, w13,1, w14,1, w15,1,  
w16,1]
```

> with(plots):

Warning, the name changecoords has been redefined

> p1:=plot([ZZ[m],YY[m],m=1..kol\_res],style=line,color=blue,thickness=3):

> p2:=plot([ZZ[m],XX[m],m=1..kol\_res],style=line,thickness=3):

> display({p1,p2},title=`Функції залежності часу доставки й витрат від  
лямбда`,labels=[Лямбда,Час\_й\_Витрати]);

Функции зависимости времени доставки и затрат от лямбда

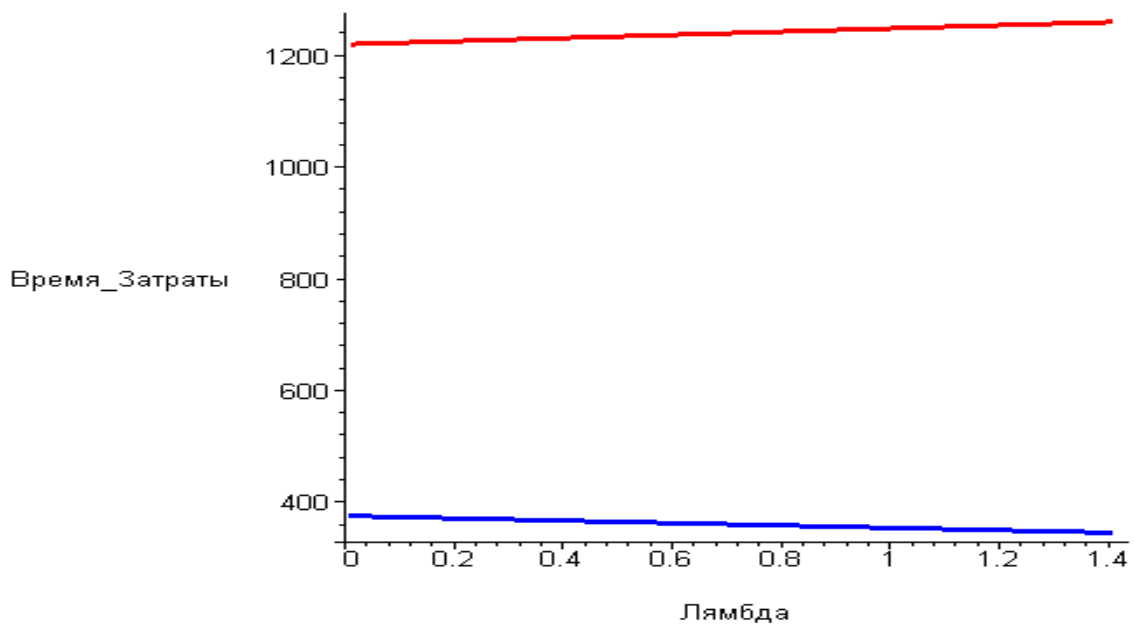


Рис. 3.19 Графік функції залежності часу обробки вагонів та грошових витрат при зміні візків від  $\lambda$

>

```
p3:=plot([YY[m],XX[m],m=1..kol_res],style=line,color=blue,thickness=3):display  
({p3},title='Графік функції залежності витрат від часу',labels=[Час,Витрати]);
```

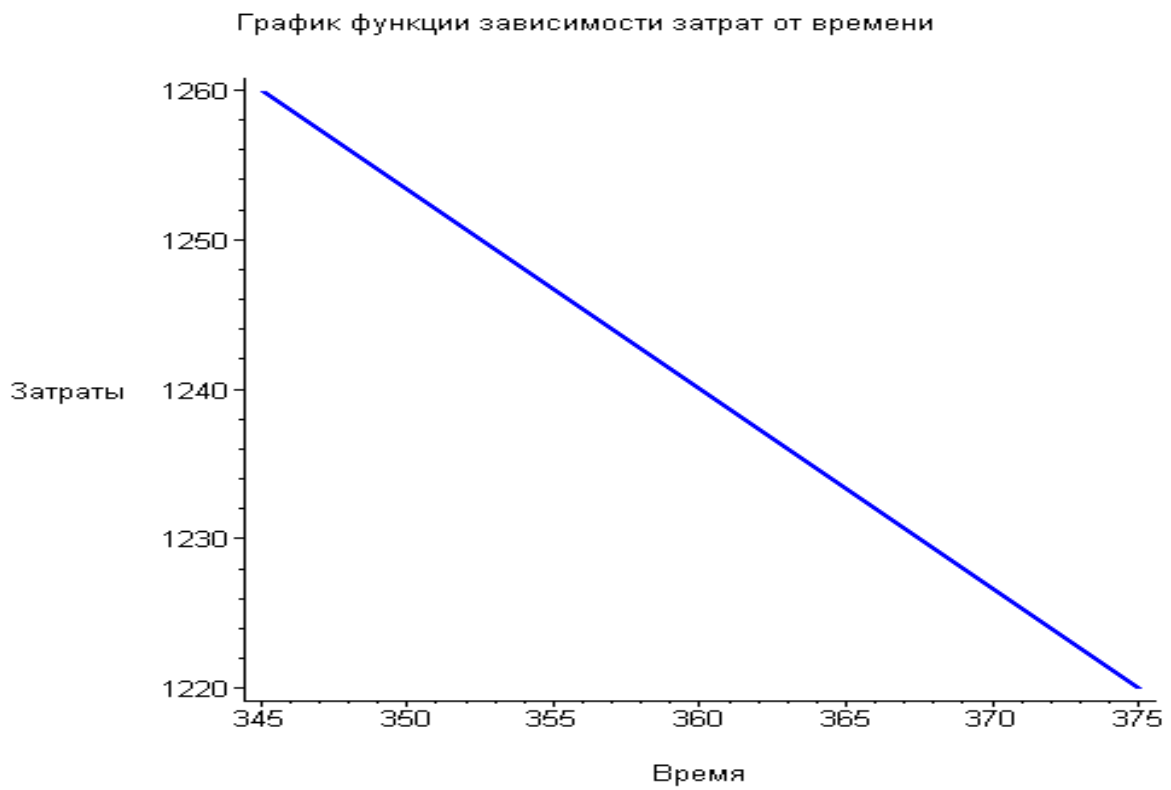


Рис. 3.20 Залежність грошових витрат від часу зміни візків при пропускній здатності пункту перестановки 5 вагонів за технологічний цикл

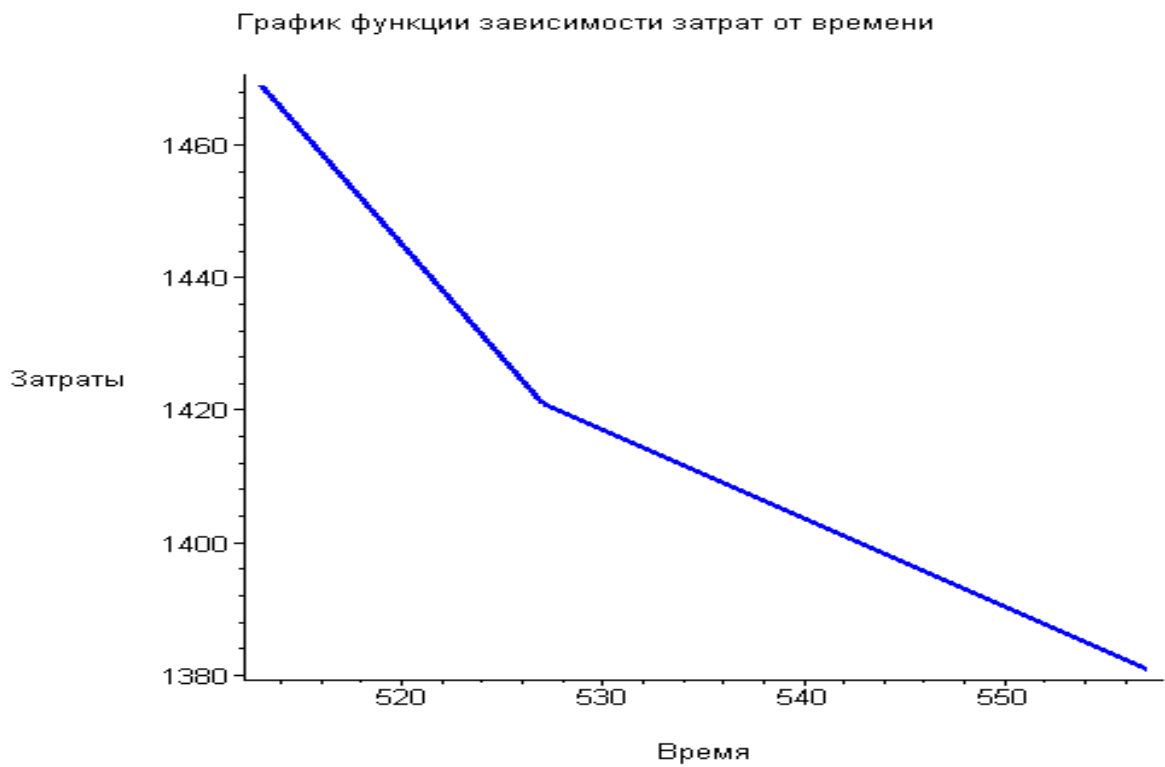


Рис. 3.21 Залежність грошових витрат від часу зміни візків при пропускній здатності пункту перестановки 10 вагонів за технологічний цикл

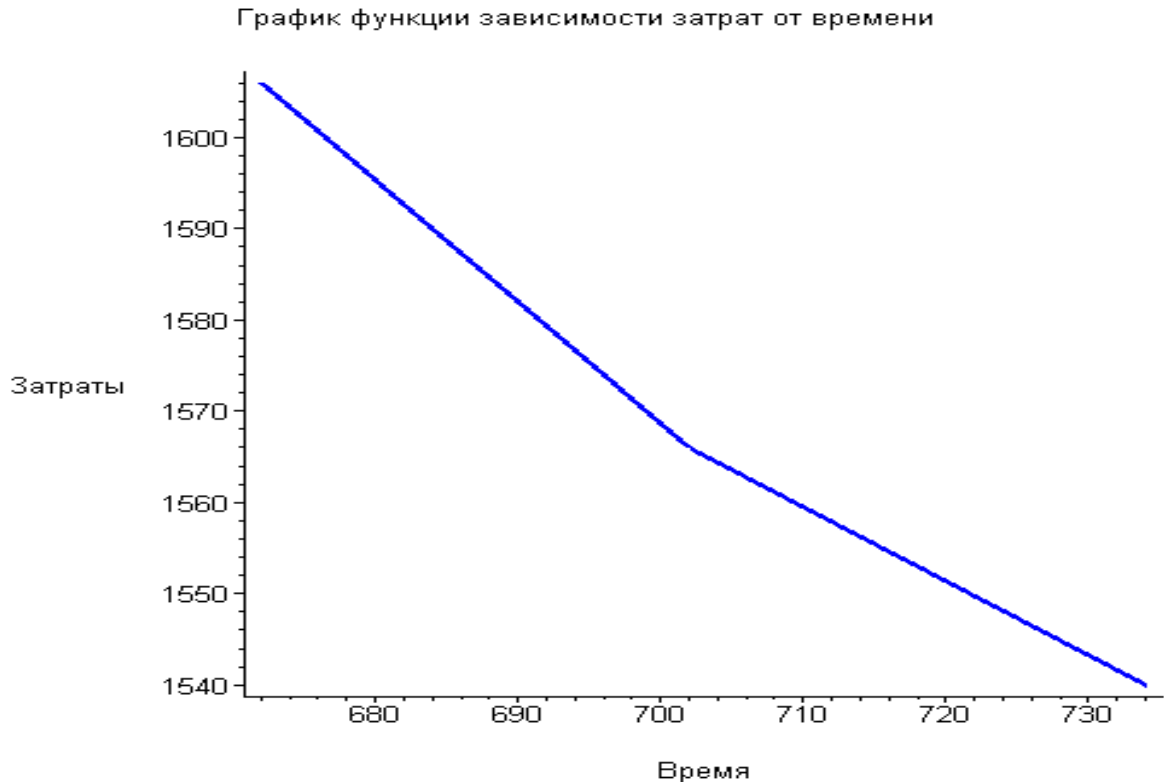


Рис. 3.22 Залежність грошових витрат від часу зміни візків при пропускній здатності пункту перестановки 15 вагонів за технологічний цикл

Програма побудови графіків залежності витрат коштів і витрат часу від кількості вагонів:

```
> restart: with(plots):
```

Warning, the name changecoords has been redefined

```
> X1:=array(1..3,[1220,1381,1540]); масив витрат грошей
```

```
X1 :=[ 1220, 1381, 1540 ]
```

```
> Y1:=array(1..3,[375,557,734]); масив витрат часу
```

```
Y1 :=[ 375, 557, 734 ]
```

```
> Z:=array(1..3,[5,10,15]);
```

```
Z :=[ 5, 10, 15 ]
```

```
> p1:=plot([Z[i],X1[i],i=1..3],color=blue): p2:=plot([Z[i],Y1[i],i=1..3],color=blue):
```

```
> X2:=array(1..3,[1260,1469,1606]); масив витрат грошей
```

```
X2 :=[ 1260, 1469, 1606 ]
```

```

> Y2:=array(1..3,[345,513,672]);    масив витрат часу
Y2 := [345, 513, 672]
> p3:=plot([Z[i],X2[i],i=1..3]): p4:=plot([Z[i],Y2[i],i=1..3]):
> display({p2,p4},title=`Залежність витрат часу від кількості
вагонів`,labels=[Вагони,Час]);
> display({p1,p3},title=`Залежність витрат грошей від кількості
вагонів`,labels=[Вагони,Гроші]);

```

*Програми розрахунку та результати моделювання раціональних параметрів  
для технології роботи системи SUW – 2000*

### Метод фаз

```

> restart:

```

```

t := [.5, .83]

```

```

Z := [98.71, 163.85]

```

```

n := 2

```

```

> t:=[0.5,0.83];Z:=[98.71,163.85];n:=2;

```

```

> for j from 1 to n-1 do for i from 1 to n-1 do

```

```

if t[i]>t[i+1] then rab:=t[i+1]; t[i+1]:=t[i]; t[i]:=rab; fi; od;od; print(t);
[.5, .83]

```

```

> for j from 1 to n-1 do for i from 1 to n-1 do

```

```

if Z[i]>Z[i+1] then rab:=Z[i+1]; Z[i+1]:=Z[i]; Z[i]:=rab; fi; od;od; print(Z);

```

```

[98.71, 163.85]

```

```

>

```

```

> a:=t;b:=Z;

```

```

a := [.5, .83]

```

```

b := [98.71, 163.85]

```

```

> for i from 1 to n do phi[i]:=1; od:

```

```

> for i from 1 to n-1 do

```

```

(phi[i] = 1)

```

```

for j from (i+1) to n do

```

```

a[j])/(b[i]-b[j]);

```

```

if

```

```

then

```

```

mu[i,j]:=(a[i]-

```

```

if (mu[i,j] < 0)

```

```

then phi[j] := 0;
od;
j := i+1;
while ((mu[i,j] < 0) and (j<=n))do
:= j+1;
od;
m := j; if (mu[i,j] > 0)
mum[i] := mu[i,j];
for j from m to n-1 do
> 0)
then
if (mu[i,j+1] < mum[i])
mum[i] := mu[i,j+1];
fi;
od;
fi;
fi;
od;
print (phi); print (mu); print(mum);

```

table([1 = 1, 2 = 1])

table([(1, 2) = .005066011667])

table([1 = .005066011667])

*Побудова залежності витрат енергії від витрат часу у  
фазі з континуумом операцій*

> restart;

> with(plots):

> A:=0.00027;B:=0.045;C:=1.5;e:=7;

A := .00027

B := .045

C := 1.5

e := 7

> v:=s->s\*(1-s)+e;

$v := s \rightarrow s(1-s) + e$

➤ plot(v(s),s=0..1);

➤

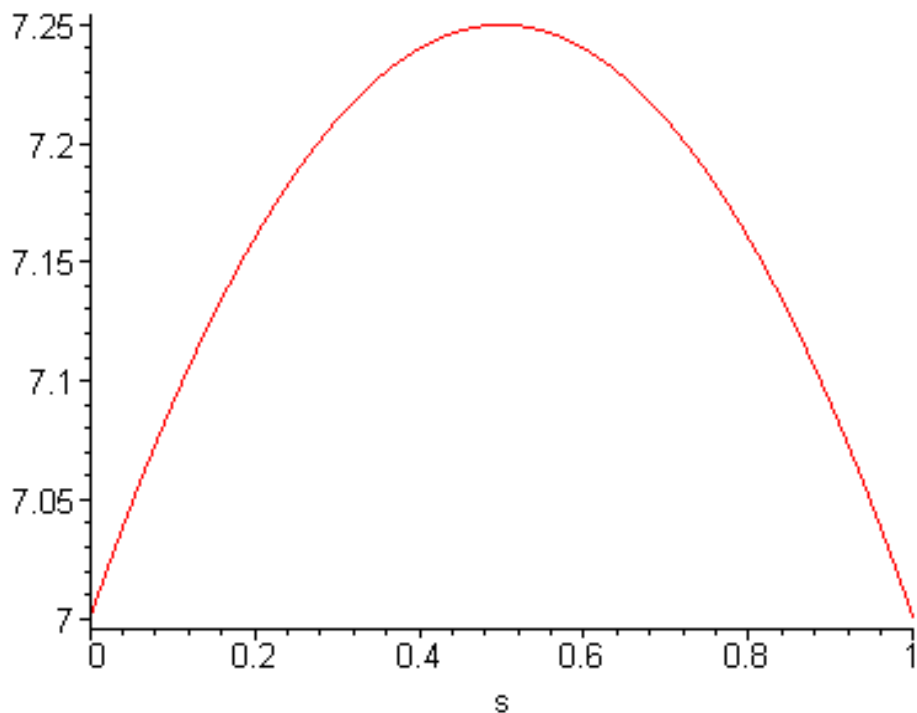


Рис. 3.23 Крива енергетичних витрат при русі поїзда через колієперевідний пристрій SUW-2000

```

> tau:=[solve(v(s)=7.1,s)];
            $\tau := [.1127016654, .8872983346]$ 

> print(tau[1]);
           .1127016654

> v0:=v(tau[1]);
            $v0 := 7.100000000$ 

> t:=int(1/v(s),s=0..tau[1])+(tau[2]-tau[1])/v0+int(1/v(s),s=tau[2]..1);
            $t := .1410611648$ 

> Ar:=int(A*v(s)^2+B*v(s)+C,s=0..tau[1])+(tau[2]-
tau[1])*(A*v0^2+B*v0+C)+int(A*v(s)^2+B*v(s)+C,s=tau[2]..1);
            $Ar := 1.832583824$ 

> tmin:=evalf(int(1/v(s),s=0..1),7);
            $tmin := .1395500$ 

> Amax:=int(A*v(s)^2+B*v(s)+C,s=0..1);

```



*Amax := 1.836369000*

**> d:=0:for i from 7 by 0.05 to 7.25 do tau:=[solve(v(s)=i,s)]; d:=d+1: end do;print(d);**

*τ := [0, 1]*

*d := 1*

*τ := [.05278640450, .9472135955]*

*d := 2*

*τ := [.1127016654, .8872983346]*

*d := 3*

*τ := [.1837722340, .8162277660]*

*d := 4*

*τ := [.2763932022, .7236067978]*

*d := 5*

*τ := [.5000000000, .5000000000]*

*d := 6*

**> v1:=7;**

*v1 := 7*

**> vk:=7.25;**

*vk := 7.25*

**> shag:=0.05;j:=1;**

*shag := .05*

*j := 1*

**> Ar:=array (1..6);**

*Ar := array(1 .. 6, [ ])*

**> t:=array (1..6);**

*t := array(1 .. 6, [ ])*

**> while v1<=vk do tau:=[solve(v(s)=v1,s)]; v0:=v(tau[1]):**

**t[j]:=int(1/v(s),s=0..tau[1])+(tau[2]-tau[1])/v0+int(1/v(s),s=tau[2]..1);**

**Ar[j]:=int(A\*v(s)^2+B\*v(s)+C,s=0..tau[1])+(tau[2]-**

**tau[1])\*(A\*v0^2+B\*v0+C)+int(A\*v(s)^2+B\*v(s)+C,s=tau[2]..1); j := j+1; v1**

**:= v1 + shag; od;print(Ar);print(t);**

$$\tau := [0, 1]$$

$$v0 := 7$$

$$t_1 := \frac{1}{7}$$

$$Ar_1 := 1.82823$$

$$j := 2$$

$$v1 := 7.05$$

$$\tau := [.05278640450, .9472135955]$$

$$v0 := 7.050000000$$

$$t_2 := .1418963331$$

$$Ar_2 := 1.830543273$$

$$j := 3$$

$$v1 := 7.10$$

$$\tau := [.1127016654, .8872983346]$$

$$v0 := 7.100000000$$

$$t_3 := .1410611648$$

$$Ar_3 := 1.832583824$$

$$j := 4$$

$$v1 := 7.15$$

$$\tau := [.1837722340, .8162277660]$$

$$v0 := 7.150000000$$

$$t_4 := .1403657200$$

$$Ar_4 := 1.834307929$$

$$j := 5$$

$$v1 := 7.20$$

$$\tau := [.2763932022, .7236067978]$$

$$v0 := 7.200000000$$

$$t_5 := .1398360526$$

$$Ar_5 := 1.835640060$$

$$j := 6$$

$vI := 7.25$

$\tau := [.5000000000, .5000000000]$

$v0 := 7.2500000000$

$t_6 := .1395500806$

$Ar_6 := 1.836369000$

$j := 7$

$vI := 7.30$

$[1.82823, 1.830543273, 1.832583824, 1.834307929, 1.835640060, 1.836369000]$

$\left[ \frac{1}{7}, .1418963331, .1410611648, .1403657200, .1398360526, .1395500806 \right]$

**> Xt:=** $[1/7, .1418963331, .1410611648, .1403657200, .1398360526,$   
**.1395500806];YA:=** $[1.82823, 1.830543273, 1.832583824, 1.834307929,$   
**1.835640060, 1.836369000];**

$Xt := \left[ \frac{1}{7}, .1418963331, .1410611648, .1403657200, .1398360526, .1395500806 \right]$

$YA := [1.82823, 1.830543273, 1.832583824, 1.834307929, 1.835640060, 1.836369000]$

**> pare:=(YA,Xt)->** $[YA,Xt];$

$pare := (YA, Xt) \rightarrow [YA, Xt]$

**> CoorXY:=zip(pare,YA,Xt,2);**

$CoorXY := \left[ \left[ 1.82823, \frac{1}{7} \right], [1.830543273, .1418963331], [1.832583824, .1410611648], \right.$   
 $[1.834307929, .1403657200], [1.835640060, .1398360526],$   
 $\left. [1.836369000, .1395500806] \right]$

**➤ pointplot(CoorXY);**

**➤**

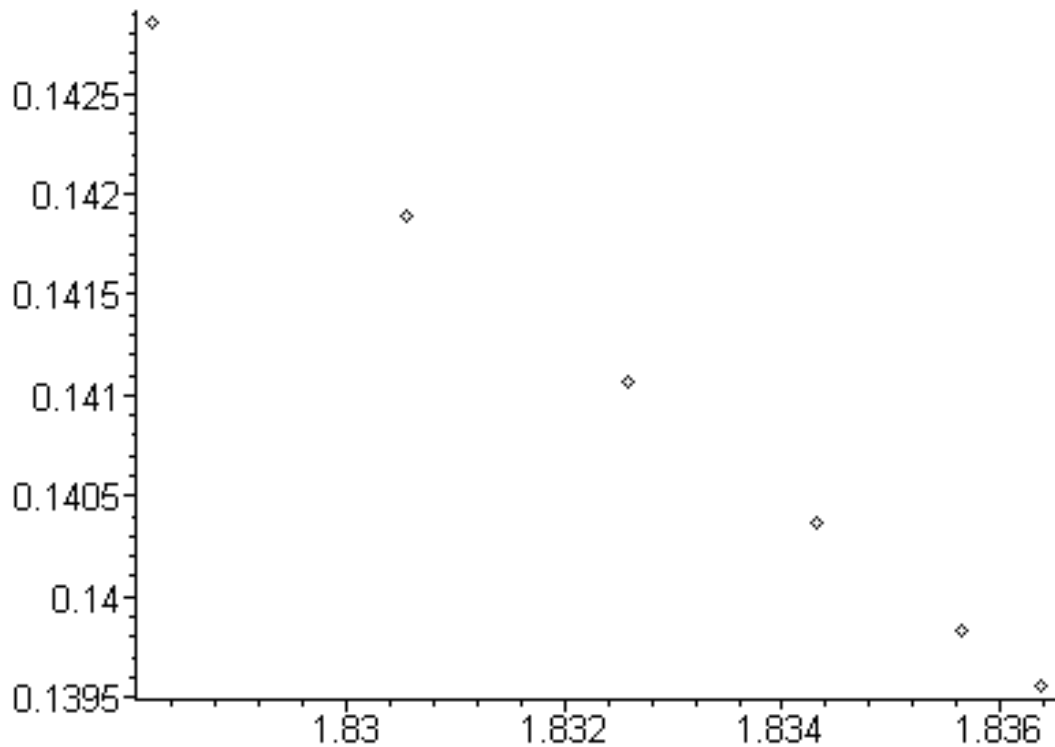


Рис. 3.24 Залежність витрат грошей від витрат часу під час переведення маневрового составу через колійний стенд SUW-2000

### Метод фаз (фаза з континуумом операцій)

> restart;

> t:=[1/7, .1418963331, .1410611648, .1403657200, .1398360526,  
.1395500806];Z:=[1.82823, 1.830543273, 1.832583824, 1.834307929,  
1.835640060, 1.836369000];n:=6;

$t := \left[ \frac{1}{7}, .1418963331, .1410611648, .1403657200, .1398360526, .1395500806 \right]$

$Z := [1.82823, 1.830543273, 1.832583824, 1.834307929, 1.835640060, 1.836369000]$

$n := 6$

> for j from 1 to n-1 do for i from 1 to n-1 do

if t[i]>t[i+1] then rab:=t[i+1]; t[i+1]:=t[i]; t[i]:=rab; fi; od;od; print(t);

$\left[ .1395500806, .1398360526, .1403657200, .1410611648, .1418963331, \frac{1}{7} \right]$

```

for j from 1 to n-1 do for i from 1 to n-1 do
if Z[i]>Z[i+1] then rab:=Z[i+1]; Z[i+1]:=Z[i]; Z[i]:=rab; fi; od;od;
print(Z);

```

```

[1.82823, 1.830543273, 1.832583824, 1.834307929, 1.835640060, 1.836369000]

```

```

>

```

```

> a:=t;b:=Z;

```

```

a := [ .1395500806, .1398360526, .1403657200, .1410611648, .1418963331,  $\frac{1}{7}$  ]

```

```

b := [1.82823, 1.830543273, 1.832583824, 1.834307929, 1.835640060, 1.836369000]

```

```

> for i from 1 to n do phi[i]:=1; od:

```

```

> for i from 1 to n-1 do

```

```

if

```

```

(phi[i] = 1)

```

```

then

```

```

for j from (i+1) to n do

```

```

mu[i,j]:=(a[i]-

```

```

a[j])/(b[i]-b[j]);

```

```

if (mu[i,j] < 0)

```

```

then phi[j] := 0;

```

```

fi;

```

```

od;

```

```

j := i+1;

```

```

while ((mu[i,j] < 0) and (j<=n))do

```

```

j

```

```

:= j+1;

```

```

od;

```

```

m := j; if (mu[i,j] > 0)

```

```

then

```

```

mum[i] := mu[i,j];

```

```

for j from m to n-1 do

```

```

if (mu[i,j+1]

```

```

> 0)

```

**then**

**if (mu[i,j+1] < mum[i])**

**mum[i] := mu[i,j+1];**

**fi; fi; od; print (phi); print (mu); print(mum);**

**table([1 = 1, 2 = 1, 3 = 1, 4 = 1, 5 = 1, 6 = 1])**

**table([(5, 6) = 1.318091750, (2, 5) = .4042312343, (2, 4) = .3254247400,**

**(2, 3) = .2595707728, (3, 6) = .6582052988, (2, 6) = .5185773896,**

**(1, 6) = .4063229266, (1, 2) = .1236222443, (1, 3) = .1873386246,**

**(4, 5) = .6269415696, (1, 4) = .2486182711, (1, 5) = .3166307020,**

**(4, 6) = .8713809956, (3, 5) = .5008163964,**

**(3, 4) = .4033656883**

**])**

**table([1 = .1236222443, 2 = .2595707728, 3 = .4033656883, 4 = .6269415696,**

**5 = 1.318091750**

**])**

**then**

**fi; fi; od;**