XXXIV OLIMPIADA WIEDZY TECHNICZNEJ

Zawody II stopnia

Rozwiązania zadań dla grupy elektryczno-elektronicznej

Rozwiązanie zadania 1

Ad.a)
Na podstawie tabeli podanej w zadaniu dla poszczególnych zmiennych $y1$, $y2$, $y3$, $y4$ tworzymy tablice Karnaugha i dokonujemy minimalizacji.

Dla zmiennej $y1$

x4x3 \ x2x1	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

Dla zmiennej $y2$

x4x3 \ x2x1	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

Dla zmiennej $y3$

x4x3 \ x2x1	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

Dla zmiennej $y4$

x4x3 \ x2x1	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

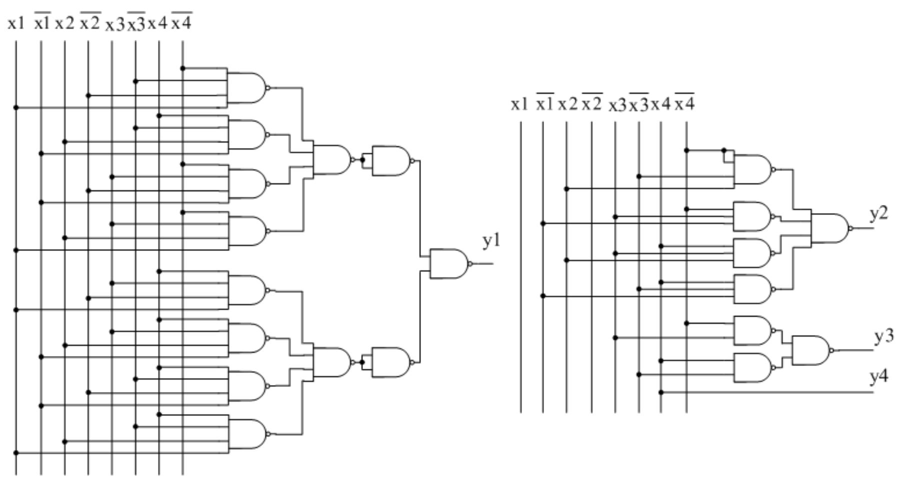
Patronem medialnym Olimpiady Wiedzy Technicznej jest „Przegląd Techniczny”

Po minimalizacji funkcje zmiennych $y1$, $y2$, $y3$, $y4$ mają postać:

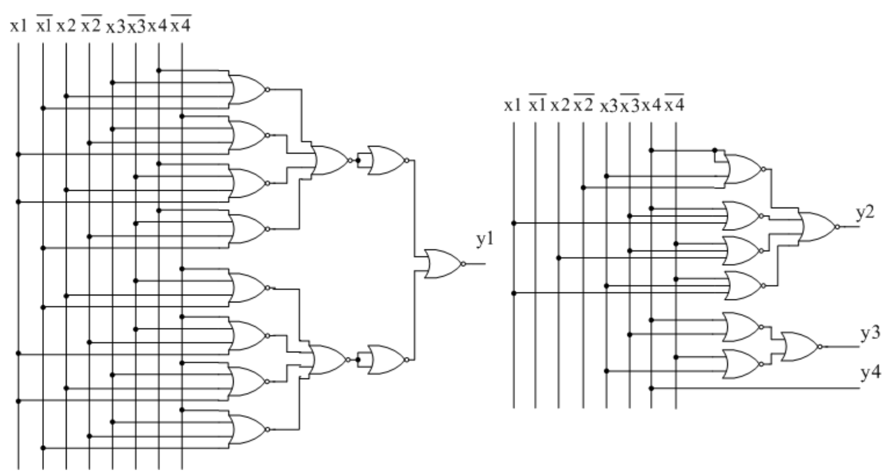
$$\begin{aligned}
y1 &= \overline{x}_4\overline{x}_3\overline{x}_2x_1 + \overline{x}_4\overline{x}_3x_2\overline{x}_1 + \overline{x}_4x_3\overline{x}_2\overline{x}_1 + \overline{x}_4x_3x_2x_1 + \\
&+ x_4x_3\overline{x}_2x_1 + x_4x_3x_2\overline{x}_1 + x_4\overline{x}_3\overline{x}_2\overline{x}_1 + x_4\overline{x}_3x_2x_1\ , \\
y2 &= \overline{x}_4\overline{x}_3x_2 + \overline{x}_4x_3\overline{x}_1 + x_4x_3x_2 + x_4\overline{x}_3\overline{x}_1\ , \\
y3 &= \overline{x}_4x_3 + x_4\overline{x}_3\ , \\
y4 &= x_4\ .
\end{aligned}$$

Przykłady realizacji:

1. Bramki NAND

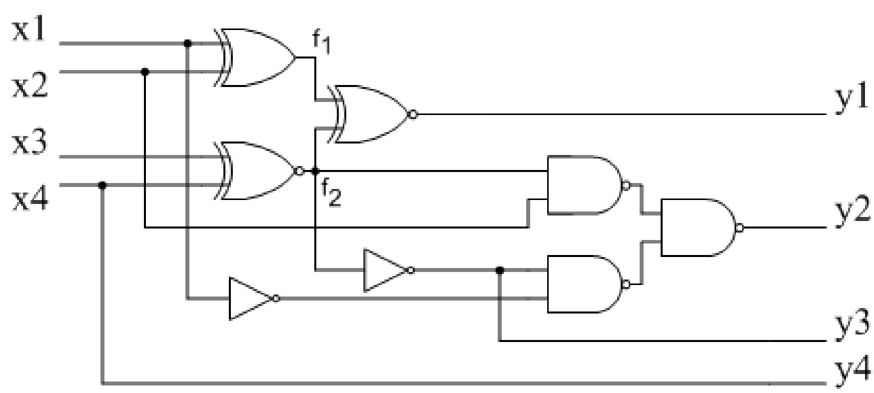


2. Bramki NOR



3. Układy złożone

$$
\begin{aligned}
y1 &= \overline{x_4 x_3}\left(\overline{x_2}x_1 + x_2\overline{x_1}\right) + \overline{x_4}x_3\left(\overline{x_2 x_1} + x_2 x_1\right) + \\
&+ x_4 x_3\left(\overline{x_2}x_1 + x_2\overline{x_1}\right) + x_4\overline{x_3}\left(\overline{x_2 x_1} + x_2 x_1\right) = \\
&= \left(\overline{x_2}x_1 + x_2\overline{x_1}\right)\left(\overline{x_4 x_3} + x_4 x_3\right) + \left(\overline{x_2 x_1} + x_2 x_1\right)\left(\overline{x_4}x_3 + x_4\overline{x_3}\right) = \\
&= f_1 f_2 + \overline{f_1 f_2}\,, \\
y2 &= x_2\left(\overline{x_4 x_3} + x_4 x_3\right) + \overline{x_1}\left(\overline{x_4}x_3 + x_4\overline{x_3}\right) = x_2 f_2 + \overline{x_1}\,\overline{f_2}\,, \\
y3 &= \overline{x_4}x_3 + x_4\overline{x_3} = \overline{f_2}\,, \\
y4 &= x_4\,.
\end{aligned}
$$



Ad.b)
Kod Graya jest kodem niewagowym, w którym sąsiednie stany (słowa) różnią się wartością tylko jednego bitu. Kody tego typu stosuje się np. w układach cyfrowych czujników położenia.

Ad.c)
Przykłady innych kodów Graya

0	0	0	0		0	0	0	0
1	0	0	0		1	0	0	0
1	1	0	0		1	0	1	0
0	1	0	0		0	0	1	0
0	1	1	0		0	1	1	0
1	1	1	0		1	1	1	0
1	0	1	0		1	1	1	1
0	0	1	0		0	1	1	1
0	0	1	1		0	0	1	1
1	0	1	1		1	0	1	1
1	1	1	1		1	0	0	1
0	1	1	1		0	0	0	1
0	1	0	1		0	1	0	1
1	1	0	1		1	1	0	1
1	0	0	1		1	1	0	0
0	0	0	1		0	1	0	0

Rozwiązanie zadania 2

Ad.1.

Tabela 1

	Stan ruchu wirników		Stan ruchu wałka napędowego silnika				
Lp.	silnik **A**	silnik **B**	**STOP**	**OL**	**OP**	**PL**	**PP**
1	**AL**	**BL**		**X**		**X**	
2	**AP**	**BP**			**X**		**X**
3	**AS**	**BS**	**X**				
4	**AL**	**BP**			**X**		
5	**AP**	**BS**					**X**
6	**AS**	**BL**		**X**		**X**	
7	**AL**	**BS**				**X**	
8	**AP**	**BL**		**X**			
9	**AS**	**BP**			**X**		**X**

Ad.2. Jeżeli silnik B w jednym skoku obraca się o kąt $\alpha = 3°$ to jeden pełny obrót wałka napędowego jest realizowany w ciągu $k = 120$ skoków silnika. Podczas jednego obrotu wałka

napędowego posuw wałka jest równy skokowi gwint. Stąd dla jednego skoku wirnika modułowego silnika B:

$$\Delta l = \frac{P}{\frac{2\pi}{\alpha}} = \frac{P}{k} = \frac{0,001}{120} = 8,333 \cdot 10^{-6}\mathrm{m}.$$

Ad.3. Jeżeli pominie się straty moc w układzie wirującym jest równa mocy w układzie posuwistym:

$$P_B = P_{\mathrm{WAŁKA}},$$

a zatem

$$M_A \cdot \omega = F \cdot v\,,$$

a ponieważ

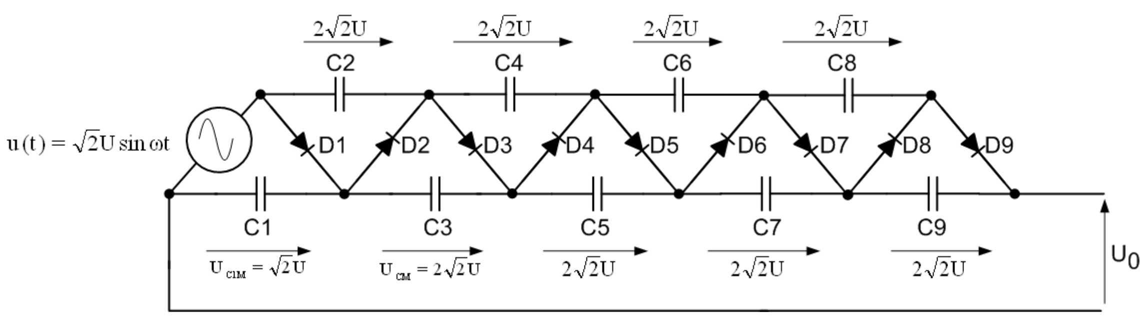
$$\omega = \frac{2\pi\, n}{60}\,, \qquad v = \frac{n\, P}{60}\,,$$

to

$$F = M_A \frac{2\pi\, n}{n\, P} = 1 \cdot \frac{20 \cdot \pi}{10 \cdot 0,001} = 6283,2\ \mathrm{N}.$$

Rozwiązanie zadania 3

Przy przyjętych w zadaniu założeniach w stanie ustalonym pracy układu rozkład stałych napięć na kondensatorach będzie jak na rysunku 2.



Rys.2. Rozkład napięć na zaciskach kondensatorów w powielaczu napięcia stałego w stanie pracy ustalonej.

Ad.1. Wartość stałego napięcie wyjściowego U_0 na zaciskach powielacza jest zatem równa:

$$U_0 = 9\sqrt{2}\,U\,.$$

Po przekształceniu tej zależności wartość skuteczna napięcia na zaciskach wyjściowych generatora zasilającego powielacz napięcia jest równa:

$$U = \frac{U_0}{9\sqrt{2}} = \frac{6000}{9\sqrt{2}} = 471,4 \text{ V}.$$

Ad.2. Maksymalne napięcie robocze na kondensatorach jest równe:
dla kondensatora C_1

$$U_{C1M} = \sqrt{2}\,U = \sqrt{2}\cdot 471,4 = 666,7 \text{ V}.$$

a dla pozostałych kondensatorów

$$U_{CM} = 2\sqrt{2}\,U = 2\sqrt{2}\cdot 471,4 = 1333,3 \text{ V}.$$

Ad.3. Całkowita energia zgromadzona w polu elektrycznym kondensatorów jest równa sumie energii zgromadzonej w poszczególnych kondensatorach:

$$W_C = W_{C1} + W_{C2} + W_{C3} + W_{C4} + W_{C5} + W_{C6} + W_{C7} + W_{C8} + W_{C9},$$

$$W_{C1} = \frac{2\,C\,U^2}{2} = C\,U^2 = 470\cdot 10^{-9}\cdot 471,4^2 = 0,104 \text{ J},$$

$$W_{Cn} = \frac{4\cdot 2\cdot C\,U^2}{2} = 4\,C\,U^2 = 4\,W_{C1} = 4\cdot 0,104 = 0,416 \text{ J, gdzie } n = 2,3,4,5,6,7,8,9.$$

Całkowita energia kondensatorów jest zatem równa:

$$W_C = W_{C1} + k\cdot W_{Cn} = 0,104 + 8\cdot 0,416 = 3,432 \text{ J, gdzie } k = 8\,.$$

Ad.4. Napięcia wsteczne diod jest równe:

$$U_R = \sqrt{2}U = \sqrt{2}\cdot 471,4 = 666,7 \text{ V}.$$

Rozwiązanie zadania z optymalizacji

Zestaw $A - B$

	Produkt A	Produkt B	Max. czas pracy
–	h	h	h
Obrabiarka I	2	2	16
Obrabiarka II	4	1	16
Obrabiarka III	4	2	20
Zysk zł/szt.	6000	4000	

Oznaczając liczbę produktów przez X z powyższych danych wynikają następujące relacje:

$$\begin{aligned} 2 \cdot X_A + 2 \cdot X_B &\leq 16 \\ 4 \cdot X_A + 1 \cdot X_B &\leq 16 \\ 4 \cdot X_A + 2 \cdot X_B &\leq 20 \end{aligned}$$

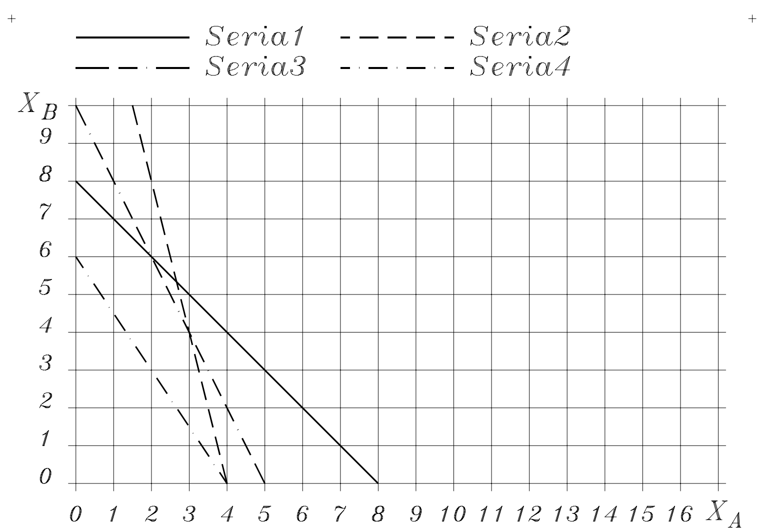
dodatkowo, w każdym przypadku spełnione być muszą nierówności:

$$X_A \geq 0; \quad X_B \geq 0; \quad X_C \geq 0.$$

Natomiast zysk wyniesie:

$$Z_{A-B} = 6000 \cdot X_A + 4000 \cdot X_B \, .$$

Układ nierówności można przedstawić na wykresie. Rozwiązaniem jest każdy punkt leżący wewnątrz obszaru ograniczonego osiami układu oraz liniami wynikającymi z zastąpienia w nierównościach znaku nierówności znakiem równości. Wybór najlepszej pary wartości produkcji wynika z maksymalizacji wyrażenia na zysk.



Kolejne granice nierówności przedstawiają linie przerywane oraz osie układu współrzędnych, natomiast linia $Seria4$ obrazuje jedynie pochylenie linii dla określonego zysku. Różne wartości zysku obrazują linie do niej równoległe, przy czym większemu zyskowi odpowiadają linie przesunięte w prawo. Stąd wynika, że maksymalnemu zyskowi przy spełnieniu podanych nierówności odpowiada punkt przecięcia $Serii1$ i $Serii3$. I stąd rozwiązując układ równań:

$$\begin{aligned} 2 \cdot X_A + 2 \cdot X_B &= 16 \\ 4 \cdot X_A + 1 \cdot X_B &= 16 \end{aligned}$$

otrzymuje się: $X_A = 2$; $X_B = 6$,
oraz zysk: $2 \cdot 6000 + 6 \cdot 4000 = 36000$ zł.

<u>Zestaw $A - C$</u>

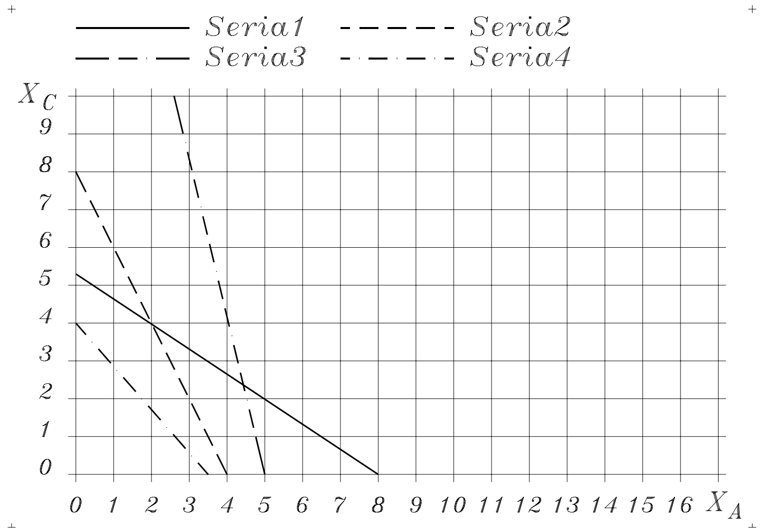
	Produkt A	Produkt C	Max. czas pracy
–	h	h	h
Obrabiarka I	2	3	16
Obrabiarka II	4	2	16
Obrabiarka III	4	1	20
Zysk zł/szt.	6000	5000	

Postępując jak poprzednio:

$$\begin{aligned} 2 \cdot X_A + 3 \cdot X_C &\leq 16 \\ 4 \cdot X_A + 2 \cdot X_C &\leq 16 \\ 4 \cdot X_A + 1 \cdot X_C &\leq 20 \end{aligned}$$

Natomiast zysk wyniesie:

$$Z_{A-C} = 6000 \cdot X_A + 5000 \cdot X_C \,.$$



Maksymalny zysk zapewni produkcja odpowiadająca przecięciu $Serii1$ i $Serii2$. Rozwiązać należy układ:

$$
\begin{aligned}
2 \cdot X_A + 3 \cdot X_C &= 16 \\
4 \cdot X_A + 2 \cdot X_C &= 16
\end{aligned}
$$

otrzymuje się: $X_A = 2$; $X_B = 4$,
oraz zysk: $2 \cdot 6000 + 4 \cdot 5000 = 32000$ zł.

Zestaw $B - C$

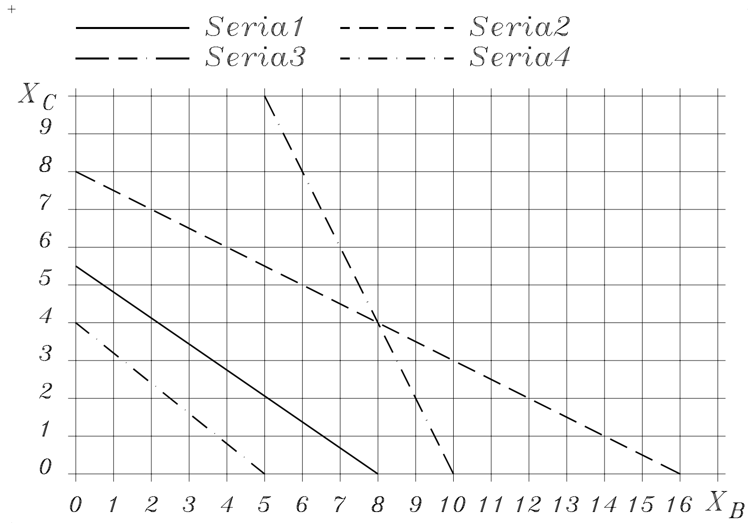
	Produkt B	Produkt C	Max. czas pracy
–	h	h	h
Obrabiarka I	2	3	16
Obrabiarka II	1	2	16
Obrabiarka III	2	1	20
Zysk zł/szt.	4000	5000	

Postępując jak poprzednio:

$$
\begin{aligned}
2 \cdot X_B + 3 \cdot X_C &\leq 16 \\
1 \cdot X_B + 2 \cdot X_C &\leq 16 \\
2 \cdot X_B + 1 \cdot X_C &\leq 20
\end{aligned}
$$

Uzyskany zysk wyniesie:

$$Z_{B-C} = 4000 \cdot X_B + 5000 \cdot X_C \, .$$



Maksymalny zysk zapewni produkcja odpowiadająca najniższemu punktowi *Serii*1 i stąd otrzymuje się: $X_B = 8$. Okazje się, że przy zestawie $B - C$ najkorzystniej jest produkować wyłącznie B przy zysku: $8 \cdot 4000 = 32000$ zł

Z przeprowadzonej analizy wynika, że najkorzystniejsze dla Zakładu jest podjęcie się produkcji $A + C$, co zapewni zysk 36000 zł na dobę.

Rozwiązanie zadania z zastosowania informatyki

Algorytm

Wylosować n pozycji samolotów, pomiędzy promieniami R_w a R_z każdorazowo sprawdzając, czy nowo wylosowana pozycja nie leży zbyt blisko wcześniej wylosowanych. Jeżeli to będzie miało miejsce powtórzyć dane losowanie. Losowanie prowadzić we współrzędnych biegunowych. Położenia przeliczyć na współrzędne kartezjańskie.

Wyznaczyć położenia na okręgu o promieniu R_w punktów styczności prostych prowadzonych z każdej wylosowanej pozycji. Określić położenie tych punktów we współrzędnych biegunowych.

Obliczyć odległość pomiędzy punktem określającym położenie początkowe i punktem styczności oraz określić czas przelotu pomiędzy tymi punktami dla każdego samolotu.

Określić czas przebywania samolotu na okręgu od punktu styczności do punktu o kącie biegunowym 0 tzw. punkt A (jest to punkt, z którego samoloty mogą odchodzić do lądowania).

Obliczyć łączny czas dolotu do punktu A dla każdego samolotu. Sprawdzić, czy samoloty na okręgu znajdują się w bezpiecznej odległości. Można korygować czas dolotu samolotu do okręgu poprzez korektę jego prędkości na pierwszym etapie lotu. Zmianie podlega w tej sytuacji całkowity czas dolotu.

W kolejności wynikającej z czasu dolotu do punktu A „przeprowadzać” lądowanie. Jeżeli łączny czas lotu następnego samolotu po lądującym jest mniejszy niż Δt samolot musi wykonać następne pełne okrążenie tj. należy do całkowitego czasu dolotu do punktu A dodać czas pełnego okrążenia.

Wydrukować kolejno numer samolotu, łączny czas lotu i lądowania oraz informację czy i ile razy nastąpiła korekta prędkości.

Część obliczeniowa

Kolejno wyznaczamy:

Współrzędne kartezjańskie punktu początkowego:

$$x_{0i} = R_{0i} \cos\varphi_{0i} \qquad ; \qquad y_{0i} = R_{0i} \sin\varphi_{0i} ;$$

Kąt φ_{si}:

$$\varphi_{si} = \arccos\left(\frac{R_w}{R_{0i}}\right) + \varphi_{0i} .$$

Jeżeli $\varphi_{si} > 2\,\pi$ wtedy $\varphi_{si} = \varphi_{si} - 2\,\pi$.

Odległość punktów: początkowego i styczności

$$d = \sqrt{R_{0i}^2 - R_w^2}$$

Czas lotu do punktu styczności:

$$t_1 = \frac{d}{v} .$$

Prędkość kątowa na okręgu:

$$\omega = \frac{v}{R_w} .$$

Całkowity czas lotu do punktu A:

$$t_2 = \frac{2\,\pi - \varphi_{si}}{\omega} + t_1 .$$

Całkowity czas lotu jaki upłynął od „startu" do wylądowania:

$$t_c = t_2 + t_{\mathrm{lad}} .$$

Szczegóły „przesuwania" samolotów przedstawiono w programie.

Przykładowy program

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>
#define Nmax  20
void Pozycja_poczatkowa(int,double[][9]);
void Punkt_stycznosci(int,double[][9]);
void Czas_przelotu_do_okregu(int,double[][9]);
void Sort (int, double[][9]);
void Czas_ladowania(int, double[][9]);
void Kontrola_odleglosci (int, double [][9]);
void Wydruk (int, double[][9]);
double Odleglosc(double,double,double,double);
double pi=3.1415926,dmin=100,tlad=300;
double Rw=5000,Rz=10000,v=120,dt=30,delta=0.05;
void main()
{
 int n;
 double tab[Nmax][9];
 printf("Podaj liczbe samolotow\n");
 scanf("%d",&n);
 Pozycja_poczatkowa(n,tab);
 Punkt_stycznosci(n,tab);
 Czas_przelotu_do_okregu(n,tab);
 Czas_ladowania(n,tab);
 Wydruk(n,tab);
}

/* Losowanie położenia N samolotow we wspolrzednych biegunowych:
numer losowania  - tab[i][0]; promień r0_i - tab[i][1];
kat biegunowy fi0_i - tab[i][2]
obliczenie wspolrzednych kartezjanskich polozenia:
x0_i - tab[i][3]; y0_i - tab[i][4].
Kontrola odleglosci losowanych polozen */

void Pozycja_poczatkowa(int n,double tab[Nmax][9])
{
 int i,j=0;
 double od,d1=10000;
 srand((unsigned)time(NULL));
 for(i=0;i<n;i++)
 {
  j++;
```

```
  tab[i][0]=(double)i+1;
  tab[i][1]=Rw+(Rz-Rw)*rand()/(float)RAND_MAX;
  tab[i][2]=2*pi*rand()/(float)RAND_MAX;
  tab[i][3]=tab[i][1]*cos(tab[i][2]);
  tab[i][4]=tab[i][1]*sin(tab[i][2]);
  if(i>0)
  {
   for(int j=0;j<i;j++)
   {
    od=Odleglosc(tab[i][3],tab[i][4],tab[j][3],tab[j][4]);
    if(od<d1) d1=od;
   }
  if(d1<dmin) i--;
  }
 }
}

/* Obliczanie odleglosci */

double Odleglosc(double x1,double y1,double x2,double y2)
{
 double d;
 d=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
 return d;
}

/* Wyznaczenie wspolrzednych biegunowych punktow stycznosci
z okregiem o promieniu Rw: kat biegunowy fis_i - tab[i][5].
Przygotowanie wstepnych wartosci w kolumnie 10 -- zapisane
w niej zostaną ewentualne przyszle redukcje predkosci */

void Punkt_stycznosci(int n,double tab[Nmax][9])
{
 int i;
 double fi;
 for(i=0;i<n;i++)
 {
 fi=pi/2-asin(Rw/tab[i][1])+tab[i][2];
 if(fi>2*pi) fi=fi-2*pi;
 tab[i][5]=fi;
 tab[i][8]=0;
 }
}
```

```
/* Obliczenie czasu przelotu na odcinku od początkowego
polozenia do punktu stycznosci t1_i - tab[i][6]  */

void Czas_przelotu_do_okregu(int n,double tab[Nmax][9])
{
 int i;
 double od;
 for(i=0;i<n;i++)
 {
  od=sqrt(tab[i][1]*tab[i][1]-Rw*Rw);
  tab[i][6]=od/v;
 }
}
/* Sortowanie wierszy tablicy tab [i][] wg. kolumny 7
zawierajacej calkowity czas lotu */

void Sort (int n, double tab[Nmax][9])
{
int i,j,k;
double r;
for (i=0;i<n-1;++i)
   for (j=0; j<n-i-1;++j)
       if (tab[j][7]>tab[j+1][7])
 {
 for(k=0;k<9;k++)
  {
  r=tab[j][k];
  tab[j][k]=tab[j+1][k];
  tab[j+1][k]=r;
  }
 }
}

/* Obliczenie calkowitego czasu lotu tc_i - tab[i][7],
bedacego suma czasu lotu do punktu stycznosci oraz czasu
lotu po okregu do punktu o wspolrzednej katowej fi =0.
Jezeli calkowite czasy lotu dwu kolejnych samolotow roznia
się o mniej niz delta t do czasu drugiego dodaje się
dodatkowy czas pelnego okrazenia. Zmiany w tab[i][7] */

void Czas_ladowania(int n, double tab[Nmax][9])
{
```

```
 int i,k;
 double omega,cz_okr;
 omega=v/Rw;
 cz_okr=2*pi/omega;
  for (i=0;i<n;i++)
  tab[i][7]=(2*pi-tab[i][5])/omega+tab[i][6];
 Sort(n,tab);
 Kontrola_odleglosci(n,tab);
 do
 {
  k=0;
  for(i=1;i<n;i++)
   {
    if((tab[i][7]-tab[i-1][7])<dt)
     {
      k=1;
      tab[i][7]=tab[i][7]+cz_okr;
      Sort(n,tab);
     }
   }
 }
 while(k==1);
}

/* Kontrola odleglosci pomiedzy samolotami w locie po okregu.
Gdyby dwa samoloty mialy by być w odleglosci mniejszej
niż dmin drugiemu zmniejsza się predkosc lotu o delta %
w ruchu do punktu stycznosci. Zmiana w tab[i][6],
tab[i][7] i tab[i][8]. Kazda zmiana powoduje
dodanie 1 do tab[i][8]  */

void Kontrola_odleglosci (int n, double tab[Nmax][9])
{
 int i;
 double czas_min;
 czas_min=dmin/v;
  for(i=1;i<n;i++)
   {
    if((tab[i][7]-tab[i-1][7])<czas_min)
     {
      tab[i][7]=tab[i][7]-tab[i][6];
      tab[i][8]=tab[i][8]+1;
      tab[i][6]=tab[i][6]/(1-delta);
```

```
      tab[i][7]=tab[i][7]+tab[i][6];
      Sort(n,tab);
      i--;
     }
   }
}

/* Przy wydruku dodaje sie do calkowitego czasu
lotu (tab[i][7]) czas ladowania tl */

void Wydruk (int n, double tab[Nmax][9])
{
 int i;
 for(i=0;i<n;i++)
 {
  printf("%5.0f %8.0f \t",tab[i][0],tab[i][7]+tlad);
  if(tab[i][8]!=0) printf("zredukowano predkosc %3.0f razy",tab[i][8]);
  printf("\n");

 }
}
```