Zastosowania funkcji `fmincon` (funkcji minimalizacji z ograniczeniami), w implementacji zadań optymalizacji z ograniczeniami, ciąg dalszy

Tą dalszą część omówień zagadnienia uzupełniono poglądowymi rysunkami oraz nieco poszerzono o bardziej złożone przypadki składania brył (tj. opakowań) przez pojedyncze wycięcie i złożenie figury płaskiej z prostokątnego arkusza (blachy, tektury, tworzywa sztucznego). Poniżej przedstawiono w jednym bloku zawartość dwóch funkcji (i jednocześnie dwóch niezależnych plików systemowych!):

```
%%A.Bernat 27.IV.11, All rights reserved (tested with version: 2010a)%
%%wyw. funk. celu: fc_ogrodzenie, wraz z funkcją ograniczeń (ogr_ogrodzenie) jest następujące:
%%------------------------------------------------------------------------------------------%
%options = optimset('Algorithm', 'interior-point');
%x = fmincon(@(x) fc_ogrodzenie(x), [1 2],[],[],[],[],[],[],@(x) ogr_ogrodzenie(x),options);
%disp(['Pole obszaru OPTYMALNIE ogrodzonego: ' num2str(x(1)*x(2))...
%    ', przy długościach boków: ' num2str(x(1))  ', ' num2str(x(2))]);
%%------------------------------------------------------------------------------------------%
function pole = fc_ogrodzenie(x)
pole = - x(1)*x(2);
disp(['Szukana wartość max. pola obszaru ogrodzonego: ' num2str(-pole)...
    ', przy dł. boków: ' num2str(x(1))  ', ' num2str(x(2))]);
%---------koniec funkcji celu--------------------------------%
%------------------------------------------------------------%
%
%--------początek definicji funkcji ograniczeń----------------%
%uwaga! poniższy kod funkcji ograniczeń powinien być zlokalizowany
%w oddzielnym pliku systemowym o nazwie: ogr_ogrodzenie.m,
%natomiast tutaj - zlokalizowany jedynie z celach porządkowych.
function [ogr_nierown,ogr_ronw] = ogr_ogrodzenie(x)
ogr_nierown(1) = 2*(x(1)+x(2))-100;
ogr_nierown(2) = - x(1);
ogr_nierown(3) = - x(2);
ogr_ronw = [];
%---------koniec funkcji ograniczeń---------------------------%
```

Pierwszy blok kodu z nagłówkiem funkcyjnym dotyczy implementacji funkcji celu w klasycznym zadaniu maksymalizacji wielkości pola powierzchni ogrodzonego ogrodzeniem, o długości określonej ustalonym parametrem równym 100. Drugi blok kodu z nagłówkiem funkcyjnym (tutaj wymagany jest oddzielny plik systemowy!) dotyczy implementacji funkcji ograniczeń – zbioru zarówno ograniczeń nierównościowych, jak i równościowych (o ile takie zostaną zdefiniowane oraz zachodzić będzie konieczność przytaczania ich definicji).

Wywołanie funkcji celu realizowane jest poprzez funkcję nadrzędną wywołań `fmincon` o liście argumentów wejściowych i wyjściowych, jak i wielu czynności i zmiennych pomocniczych, przytoczonych w nagłówku komentarza – pliku szczegółowej pomocy.

Wywołanie tekstu szczegółowej pomocy:

```
help fc_ogrodzenie
```

skutkuje pojawieniem się tekstu komentarza do funkcji celu `fc_ogrodzenie`:

```
>> help fc_ogrodzenie
 A.Bernat 27.IV.11, All rights reserved (tested with version: 2010a)%
 %wyw. funk. celu: fc_ogrodzenie, wraz z funkcją ograniczeń (ogr_ogrodzenie) jest następujące:
 %-----------------------------------------------------------------------------------------%
 options = optimset('Algorithm', 'interior-point');
 x = fmincon(@(x) fc_ogrodzenie(x), [1 2],[],[],[],[],[],[],@(x) ogr_ogrodzenie(x),options);
 disp(['Pole obszaru OPTYMALNIE ogrodzonego: ' num2str(x(1)*x(2))...
     ', przy długościach boków: ' num2str(x(1))  ', ' num2str(x(2))]);
 %-----------------------------------------------------------------------------------------%
```

Zaznaczenie wycinka komentarza pomiędzy dwoma liniami przerywanymi (liniami *de facto* opatrzonymi jako komentarze dodatkowymi znakami %) oraz wciśnięcie klawisza funkcyjnego **F9**, skutkuje wywołaniem funkcji `fmincon`, wraz z szeregiem poleceń pomocniczych…

Wykonanie w powyższy sposób szeregu poleceń pomocniczych, skutkuje szeregiem wywołań funkcji celu wraz z funkcją ograniczeń oraz w rezultacie znalezieniem optymalnych długości dwóch boków ograniczonej powierzchni pola, lecz z udziałem tylko i wyłącznie ograniczeń nierównościowych!

```
Szukana wartość max. pola obszaru ogrodzonego: 2, przy dł. boków: 1, 2
Szukana wartość max. pola obszaru ogrodzonego: 2, przy dł. boków: 1, 2
Szukana wartość max. pola obszaru ogrodzonego: 2, przy dł. boków: 1, 2
Szukana wartość max. pola obszaru ogrodzonego: 8.6483, przy dł. boków: 2.896, 2.9863
Szukana wartość max. pola obszaru ogrodzonego: 8.6483, przy dł. boków: 2.896, 2.9863
Szukana wartość max. pola obszaru ogrodzonego: 8.6483, przy dł. boków: 2.896, 2.9863
Szukana wartość max. pola obszaru ogrodzonego: 470.44, przy dł. boków: 23.4206, 20.0866
Szukana wartość max. pola obszaru ogrodzonego: 470.4401, przy dł. boków: 23.4206, 20.0866
Szukana wartość max. pola obszaru ogrodzonego: 470.4401, przy dł. boków: 23.4206, 20.0866
Szukana wartość max. pola obszaru ogrodzonego: 620.5506, przy dł. boków: 26.8911, 23.0764
Szukana wartość max. pola obszaru ogrodzonego: 620.5506, przy dł. boków: 26.8911, 23.0764
Szukana wartość max. pola obszaru ogrodzonego: 620.5506, przy dł. boków: 26.8911, 23.0764
Szukana wartość max. pola obszaru ogrodzonego: 621.5345, przy dł. boków: 26.8604, 23.1394
Szukana wartość max. pola obszaru ogrodzonego: 621.5345, przy dł. boków: 26.8604, 23.1394
Szukana wartość max. pola obszaru ogrodzonego: 621.5345, przy dł. boków: 26.8604, 23.1394
Szukana wartość max. pola obszaru ogrodzonego: 624.9307, przy dł. boków: 25.2217, 24.7775
Szukana wartość max. pola obszaru ogrodzonego: 624.9307, przy dł. boków: 25.2217, 24.7775
Szukana wartość max. pola obszaru ogrodzonego: 624.9307, przy dł. boków: 25.2217, 24.7775
Szukana wartość max. pola obszaru ogrodzonego: 624.98, przy dł. boków: 24.9996, 24.9996
Szukana wartość max. pola obszaru ogrodzonego: 624.98, przy dł. boków: 24.9996, 24.9996
Szukana wartość max. pola obszaru ogrodzonego: 624.98, przy dł. boków: 24.9996, 24.9996
Szukana wartość max. pola obszaru ogrodzonego: 624.9998, przy dł. boków: 25, 25
```

```
Szukana wartość max. pola obszaru ogrodzonego: 624.9998, przy dł. boków: 25, 25
Szukana wartość max. pola obszaru ogrodzonego: 624.9998, przy dł. boków: 25, 25
Szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 25, 25
Szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 25, 25
Szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 25, 25

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

Pole obszaru OPTYMALNIE ogrodzonego: 625, przy długościach boków: 25, 25
```

Rozwiązaniem szukanym jest oczywiście szczególny przypadek obrysu prostokątnego, ogradzanego wycinka pola, którym jest kwadrat, w rozważanym przypadku kwadrat o boku równym 25 jednostek długości(metrów, stóp, łokci, jardów, kilometrów?).
Pomimo prostoty w analitycznej postaci funkcji celu oraz postaci analitycznej wszystkich ograniczeń nierównościowych, to najważniejsze ograniczenie, nakładane na długość ogrodzenia względem odgradzanego od otoczenia pola powierzchni (ogrodu, poletka upraw) nie musi w praktyce implementacyjnej zawsze prowadzić do poprawnych rezultatów.

$$\begin{aligned}
&f_{\max} \to x(1)\cdot x(2) &&\Leftrightarrow\quad f_{\min} \to -x(1)\cdot x(2)\\
&x(1)\ge 0 &&\Leftrightarrow\quad -x(1)\le 0\\
&x(2)\ge 0 &&\Leftrightarrow\quad -x(2)\le 0\\
&100-2\cdot(x(1)+x(2))\ge 0 &&\Leftrightarrow\quad 2\cdot(x(1)+x(2))-100\le 0
\end{aligned} \qquad (1)$$

Powyżej w zależności (1) przedstawiono po lewej oryginalną postać trzech ograniczeń nierównościowych. Natomiast nieco bardziej z prawej, przedstawiono postać przyjmowaną w konwencji implementacyjnej środowiska Matlab, z dwustronnym przemnażaniem zależności nierównościowych przez minus jeden.
Wygodniejszą oraz pewniejszą w działaniu wersję ograniczeń nakładanych na powyższe zadanie optymalizacyjne, prowadzących ponadto do rozwiązania w mniejszej liczbie kroków, przedstawiono w zależnościach analitycznych poniżej:

$$\begin{aligned}
&f_{\max} \to x(1)\cdot x(2) &&\Leftrightarrow\quad f_{\min} \to -x(1)\cdot x(2)\\
&x(1)\ge 0 &&\Leftrightarrow\quad -x(1)\le 0\\
&x(2)\ge 0 &&\Leftrightarrow\quad -x(2)\le 0\\
&100-2\cdot(x(1)+x(2)) = 0 &&
\end{aligned} \qquad (2)$$

W implementacji funkcji fc_ogrodzenie_ceq (ang. *ceq – c vector of equalities* – wektor ograniczeń równościowych) wykorzystano dodatkowo zmienną permanentną (ze słowem kluczowym w inicjalizacji `persistent`), celem dogodniejszego ustalania liczby kroków:

```
%%A.Bernat 27.IV.11, All rights reserved (tested with version: 2010a)%
%%wyw. funk. celu: fc_ogrodzenie, wraz z funkcją ograniczeń (ogr_ogrodzenie) jest następujące:
%%----------------------------------------------------------------------------------------%
%munlock(fc_ogrodzenie_ceq);clear call_number;
%options = optimset('Algorithm', 'interior-point');
%x = fmincon(@(x) fc_ogrodzenie_ceq(x), [1 2],[],[],[],[],[],[],@(x) ogr_ogrodzenie_ceq(x),options);
%disp(['Pole obszaru OPTYMALNIE ogrodzonego: ' num2str(x(1)*x(2))...
%    ', przy długościach boków: ' num2str(x(1))  ', ' num2str(x(2))]);
%%----------------------------------------------------------------------------------------%
function pole = fc_ogrodzenie_ceq(x)
persistent call_number;%<=
if isempty(call_number)%<=inicjalizacja wewn. licznika wywołań
    call_number = 0;
end;
pole = - x(1)*x(2);
call_number=call_number+1;
disp(['Iter.no: ' num2str(call_number) ', szukana wartość max. pola obszaru ogrodzonego: ' num2str(-pole)...
    ', przy dł. boków: ' num2str(x(1))  ', ' num2str(x(2))]);
%---------koniec funkcji celu---------------------------------%
%-------------------------------------------------------------%
%
%--------początek definicji funkcji ograniczeń----------------%
%uwaga! poniższy kod funkcji ograniczeń powinien być zlokalizowany
%w oddzielnym pliku systemowym o nazwie: ogr_ogrodzenie.m,
%natomiast tutaj - zlokalizowany jedynie z celach porządkowych.
function [ogr_nierown,ogr_ronw] = ogr_ogrodzenie_ceq(x)
%ogr_nierown(1) = 2*(x(1)+x(2))-100; %<=poprzednie ograniczenie nierównościowe!
ogr_nierown(1) = - x(1);
ogr_nierown(2) = - x(2);
ogr_ronw = 100-2*(x(1)+x(2));
%---------koniec funkcji ograniczeń---------------------------%
```

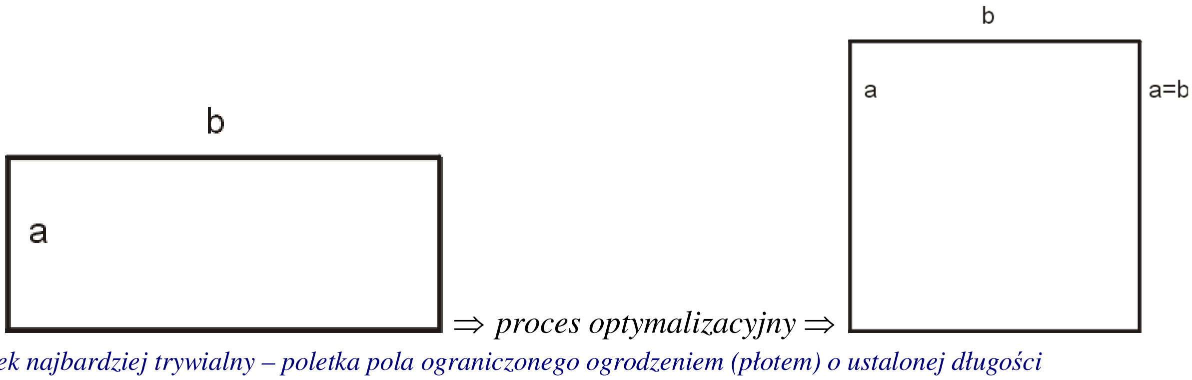
Wywołanie po raz pierwszy szeregu poleceń pomocniczych oraz samej funkcji `fmincon` (przytoczonej w nagłówku komentarza funkcji):

```
help fc_ogrodzenie_ceq
 A.Bernat 27.IV.11, All rights reserved (tested with version: 2010a)%
 %wyw. funk. celu: fc_ogrodzenie, wraz z funkcją ograniczeń (ogr_ogrodzenie) jest następujące:
 %-------------------------------------------------------------------------------------------%
 munlock(fc_ogrodzenie_ceq);clear call_number;
 options = optimset('Algorithm', 'interior-point');
 x = fmincon(@(x) fc_ogrodzenie_ceq(x), [1 2],[],[],[],[],[],[],@(x) ogr_ogrodzenie_ceq(x),options);
 disp(['Pole obszaru OPTYMALNIE ogrodzonego: ' num2str(x(1)*x(2))...
     ', przy długościach boków: ' num2str(x(1))  ', ' num2str(x(2))]);
 %-------------------------------------------------------------------------------------------%
```

skutkuje mniejszą liczbą iteracyjnego wywołania funkcji celu i funkcji ograniczeń, w poszukiwaniu rozwiązania optymalnego:

```
Iter.no: 1, szukana wartość max. pola obszaru ogrodzonego: 2, przy dł. boków: 1, 2
Iter.no: 2, szukana wartość max. pola obszaru ogrodzonego: 2, przy dł. boków: 1, 2
Iter.no: 3, szukana wartość max. pola obszaru ogrodzonego: 2, przy dł. boków: 1, 2
Iter.no: 4, szukana wartość max. pola obszaru ogrodzonego: 624.8473, przy dł. boków: 24.6092, 25.3908
Iter.no: 5, szukana wartość max. pola obszaru ogrodzonego: 624.8473, przy dł. boków: 24.6092, 25.3908
Iter.no: 6, szukana wartość max. pola obszaru ogrodzonego: 624.8473, przy dł. boków: 24.6092, 25.3908
Iter.no: 7, szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 24.9995, 25.0005
Iter.no: 8, szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 24.9995, 25.0005
Iter.no: 9, szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 24.9995, 25.0005
Iter.no: 10, szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 25, 25
Iter.no: 11, szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 25, 25
Iter.no: 12, szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 25, 25
Iter.no: 13, szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 25, 25
Iter.no: 14, szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 25, 25
Iter.no: 15, szukana wartość max. pola obszaru ogrodzonego: 625, przy dł. boków: 25, 25

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

Pole obszaru OPTYMALNIE ogrodzonego: 625, przy długościach boków: 25, 25
```

Natomiast sam proces poszukiwania rozwiązania optymalnego charakteryzuje się nieco innym trybem wykonania. Z uwagi na sposób zdefiniowania podstawowego ograniczenia, nałożonego na długość ogradzanego pola, jako **ograniczenia równościowego**(!), sam proces znajdywania rozwiązania musi się charakteryzować niezwykle dużą '*sztywnością*', jako, że przestrzeń rozwiązań dopuszczalnych faktycznie dotyczy ustalonego na 100 jednostek długości ogrodzenia.
Stąd, obrazowo ujmując zagadnienie, w tym rozważanym przypadku może zmieniać się tylko kształt ogradzanego wycinka pola, przy niezmiennej długości ogrodzenia (lub zmiennego w niezmiernie wąskim zakresie rzędu np. tysięcznych lub milionowych części długości zadanej jako parametr stały liczbowy równy 100).
Natomiast w poprzednim rozważanym przypadku, obrazowo ujmując proces znajdywania rozwiązania optymalnego, wektor startowy przestrzeni stanów `x(1)` i `x(2)` (odpowiednio: długości boków `a` i `b` prostokąta) mógł być taki w wartościach początkowych, że wymagał jednocześnie powiększania długości ogrodzenia, do wartości dopuszczalnej, określonej ostatnią nierównością w zależności (1).
Chociaż jednocześnie, większa nieco złożoność której z postaci analitycznej, czy to funkcji celu, czy to funkcji ograniczeń mogłaby potencjalnie 'uniemożliwić' wzrost długości ogrodzenia do stanu faktycznego, zgodnego z wartością stałej parametru równego 100.



Rys.1 Przypadek najbardziej trywialny – poletka pola ograniczonego ogrodzeniem (płotem) o ustalonej długości

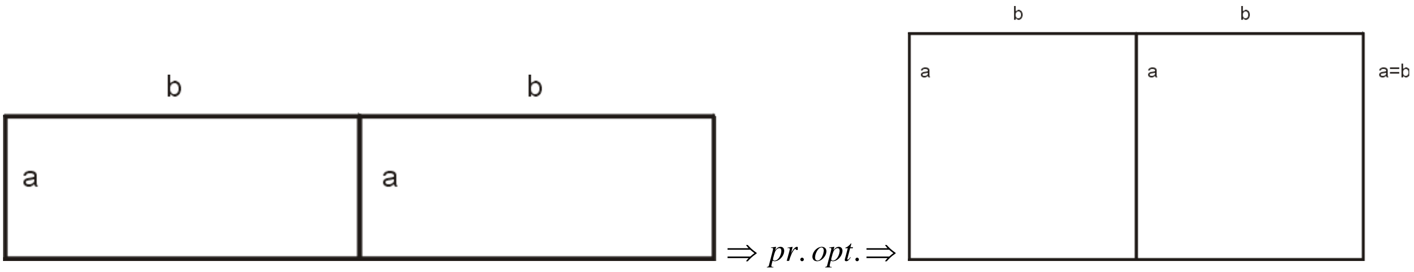
Pozostałe przypadki implementacyjne w archiwum zip w ogólności dotyczą:

- przekazywania parametru/parametrów do funkcji celu/ograniczeń, gdy nie jest/są on/one zdefiniowane jako stałe wewnętrzne tych funkcji,
- przylegania do siebie wzajemnie dwóch, czterech lub nawet sześciu poletek pól, z wewnętrznymi ogrodzeniami tych podpól, globalnie jednakże : opisanych na spójnym jednym obrysie prostokątnym obszaru pola.

Przykładowo dla zbioru ograniczeń nierównościowych (dodatniość długości dwóch boków podstawowych a i b), jak i równościowych (utrzymywania w procesie szukania rozwiązania długości ogrodzenia zgodnie ze stanem faktycznym) oraz schematu, w którym do siebie przylegają jednym bokiem dwa poletka podpól otrzymujemy następującą implantację:

```
%%A.Bernat 27.IV.11, All rights reserved (tested with version: 2010a)%
%%wyw. funk. celu: fc_2przyleg_ogrodzenia, wraz z funkcją ograniczeń (ogr_2przyleg_ogrodzenia_z_param) jest następujące:
%%------------------------------------------------------------------------------------------%
```

```
%%munlock(fc_2przyleg_ogrodzenia_z_param_ceq);clear call_number;
%options = optimset('Algorithm', 'interior-point');
%dlg = input('Podaj długość ogrodzenia dwóch przylegających do siebie obszarów: ');
%x = fmincon(@(x) fc_2przyleg_ogrodzenia_z_param_ceq(x,dlg), [1 0.5],[],[],[],[],[],[],@(x)
ogr_2przyleg_ogrodzenia_z_param_ceq(x,dlg),options);
%disp(['Pole obszaru OPTYMALNIE ogrodzonego: ' num2str(2*x(1)*x(2))...
%    ', przy długościach boków obsz. podst.: ' num2str(x(1))  ', ' num2str(x(2))...
%    ' oraz przy zadanej dł. ogrodzenia: ' num2str(dlg)]);
%%-----------------------------------------------------------------------------------------%
function pole = fc_2przyleg_ogrodzenia_z_param_ceq(x,dlug)
persistent call_number;
if isempty(call_number)
    call_number=0;
end;
pole = - 2*x(1)*x(2);
call_number = call_number + 1;
disp(['Iter.no: ' num2str(call_number) ', szukana wart. max. p. obsz. ogrodz.: ' num2str(-pole)...
    ', przy dł. boków: ' num2str(x(1))  ', ' num2str(x(2))...
    ' oraz zadanej dł. ogrodz: ' num2str(dlug)]);
%---------koniec funkcji celu---------------------------------%
%-----------------------------------------------------------%
%
%--------początek definicji funkcji ograniczeń----------------%
%uwaga! poniższy kod funkcji ograniczeń powinien być zlokalizowany
%w oddzielnym pliku systemowym o nazwie: ogr_2przyleg_ogrodzenia.m,
%natomiast tutaj - zlokalizowany jedynie z celach porządkowych
function [ogr_nierown,ogr_ronw] = ogr_2przyleg_ogrodzenia_z_param_ceq(x,dlug)
%ogr_nierown(1) = 4*x(1)+3*x(2)- dlug;%<=poprzednie ograniczenie nierównościowe!%
ogr_nierown(1) = - x(1);
ogr_nierown(2) = - x(2);
ogr_ronw = 4*x(1)+3*x(2) - dlug;
%---------koniec funkcji ograniczeń---------------------------%
```



Rys.2 Przypadek 2 poletek pól, ograniczonych częściowo współdzielonym ogrodzeniem (płotem) o ustalonej długości

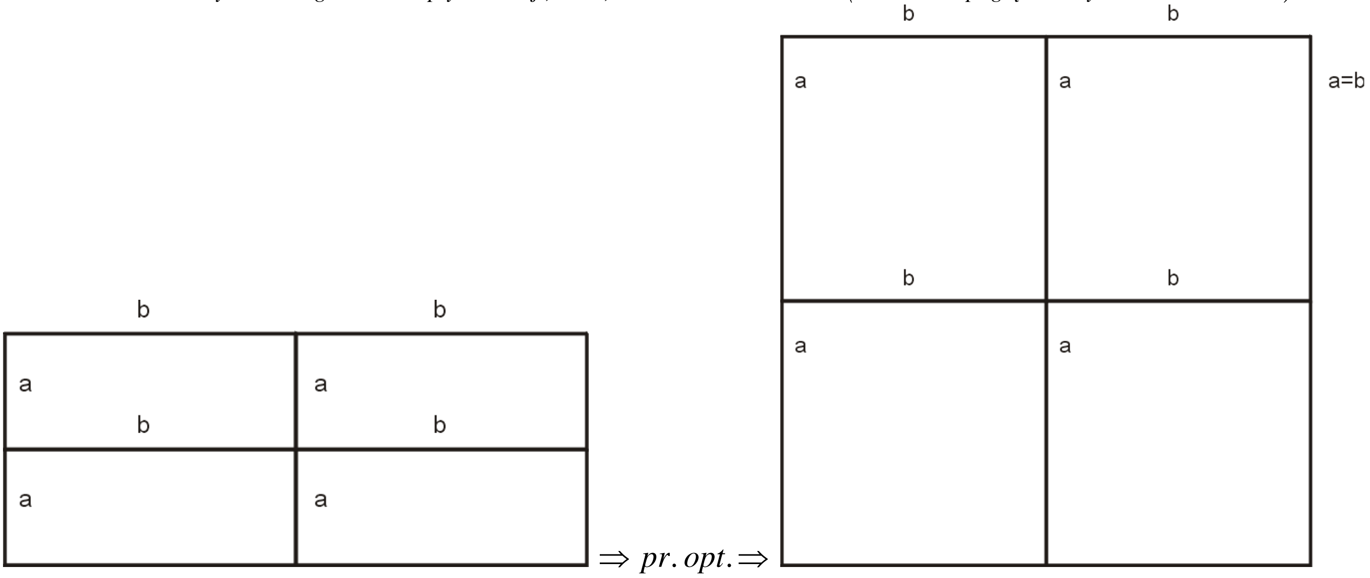
W wyniku uruchomień powyższej implementacji:

```
Podaj dlugość ogrodzenia dwóch przylegających do siebie obszarów: 100
Iter.no: 1, szukana wart. max. p. obsz. ogrodz.: 1, przy dł. boków: 1, 0.5 oraz zadanej dł. ogrodz: 100
Iter.no: 2, szukana wart. max. p. obsz. ogrodz.: 1, przy dł. boków: 1, 0.5 oraz zadanej dł. ogrodz: 100
Iter.no: 3, szukana wart. max. p. obsz. ogrodz.: 1, przy dł. boków: 1, 0.5 oraz zadanej dł. ogrodz: 100
Iter.no: 4, szukana wart. max. p. obsz. ogrodz.: 380.9104, przy dł. boków: 16.1618, 11.7843 oraz zadanej dł. ogrodz: 100
Iter.no: 5, szukana wart. max. p. obsz. ogrodz.: 380.9105, przy dł. boków: 16.1618, 11.7843 oraz zadanej dł. ogrodz: 100
Iter.no: 6, szukana wart. max. p. obsz. ogrodz.: 380.9105, przy dł. boków: 16.1618, 11.7843 oraz zadanej dł. ogrodz: 100
Iter.no: 7, szukana wart. max. p. obsz. ogrodz.: 398.9474, przy dł. boków: 9.9223, 20.1037 oraz zadanej dł. ogrodz: 100
Iter.no: 8, szukana wart. max. p. obsz. ogrodz.: 398.9474, przy dł. boków: 9.9223, 20.1037 oraz zadanej dł. ogrodz: 100
Iter.no: 9, szukana wart. max. p. obsz. ogrodz.: 398.9474, przy dł. boków: 9.9223, 20.1037 oraz zadanej dł. ogrodz: 100
Iter.no: 10, szukana wart. max. p. obsz. ogrodz.: 416.6667, przy dł. boków: 12.4985, 16.6687 oraz zadanej dł. ogrodz: 100
Iter.no: 11, szukana wart. max. p. obsz. ogrodz.: 416.6667, przy dł. boków: 12.4985, 16.6687 oraz zadanej dł. ogrodz: 100
Iter.no: 12, szukana wart. max. p. obsz. ogrodz.: 416.6667, przy dł. boków: 12.4985, 16.6687 oraz zadanej dł. ogrodz: 100
Iter.no: 13, szukana wart. max. p. obsz. ogrodz.: 416.6667, przy dł. boków: 12.5, 16.6667 oraz zadanej dł. ogrodz: 100
Iter.no: 14, szukana wart. max. p. obsz. ogrodz.: 416.6667, przy dł. boków: 12.5, 16.6667 oraz zadanej dł. ogrodz: 100
Iter.no: 15, szukana wart. max. p. obsz. ogrodz.: 416.6667, przy dł. boków: 12.5, 16.6667 oraz zadanej dł. ogrodz: 100

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

Pole obsz. OPTYMALNIE ogrodzonego: 416.6667, przy dł. boków obsz. podst.: 12.5, 16.6667 oraz przy zadanej dł. ogrodzenia: 100
```

otrzymano niższą od poprzedniej wartość pola (416.67 zamiast 625 jednostek powierzchni) przy długości boków podstawowego poletka pola równych odpowiednio 12.5 oraz 16.67 (zamiast 25 – jako długości boku pola o obrysie kwadratowym).

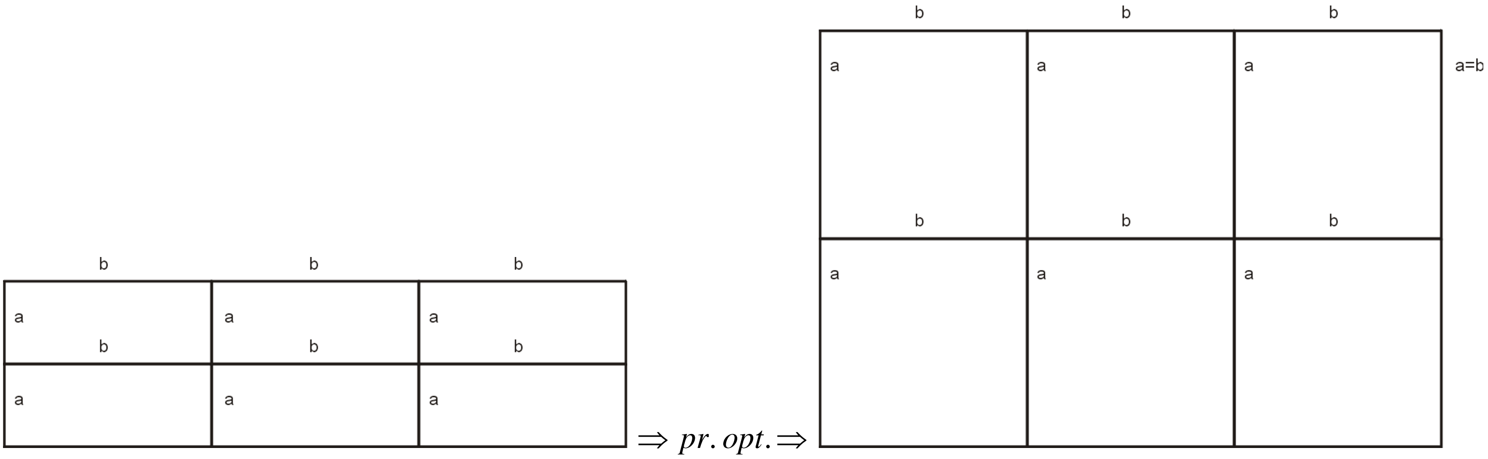
Uruchomienia dalszych zadań sukcesywnie, w otrzymywanych wynikach, wskazują na zmniejszanie:

- pola powierzchni całkowitej zestawu 4, 6 podpól poletek,
- zmniejszanie również długości boków a i b poletka podstawowego.



Rys.3 Przypadek 4 poletek pól, ograniczonych częściowo współdzielonym ogrodzeniem (płotem) o ustalonej długości

W wyniku uruchomień powyższej implementacji:



Rys.4 Przypadek 6 poletek pól, ograniczonych częściowo współdzielonym ogrodzeniem (płotem) o ustalonej długości

W wyniku uruchomień powyższej implementacji:

Przy czym zadania wyglądające na wprost proporcjonalne (przypadek 4 poletek pól, przylegających do siebie w zwartym prostokącie, o dwóch wierszach i dwóch kolumnach elementów podstawowych) okazują się wcale nieproporcjonalne do zadania podstawowego. Innymi słowy z postacią funkcji celu, jak w przypadku 4 poletek pól (podpól) oraz adekwatną postacią funkcji ograniczeń (wyglądającą prawie tak samo jednakże 4-krotnie przemnożonej – stąd proporcjonalnie) otrzymuje się inne wyniki niż w przypadku pierwszym, pierwotnie rozważanym na początku tego dokumentu. Stąd wniosek: wewnętrzne ogrodzenia poszczególnych poletek pól, wprowadzają takie wzajemne modyfikacje postaci funkcji celu i ograniczeń, że nie możliwe odtąd staje się z tą samą długością ogrodzenia ogrodzić tej samej optymalnej wartości powierzchni pola (w jej obrysie globalnym – podstawowym).

Poniżej przedstawiono wyniki dla 4 poletek pól wzajemnie stykających się na obrysie prostokąta:

```
Podaj długość ogrodzenia czterech przylegających do siebie obszarów: 100
Iter.no: 1, szukana wart. max. p. obsz. ogrodz.: 2, przy dł. boków: 1, 0.5 oraz zadanej dł. ogrodz: 100
Iter.no: 2, szukana wart. max. p. obsz. ogrodz.: 2, przy dł. boków: 1, 0.5 oraz zadanej dł. ogrodz: 100
Iter.no: 3, szukana wart. max. p. obsz. ogrodz.: 2, przy dł. boków: 1, 0.5 oraz zadanej dł. ogrodz: 100
Iter.no: 4, szukana wart. max. p. obsz. ogrodz.: 276.7778, przy dł. boków: 7.8333, 8.8333 oraz zadanej dł. ogrodz: 100
Iter.no: 5, szukana wart. max. p. obsz. ogrodz.: 276.7778, przy dł. boków: 7.8333, 8.8333 oraz zadanej dł. ogrodz: 100
Iter.no: 6, szukana wart. max. p. obsz. ogrodz.: 276.7778, przy dł. boków: 7.8333, 8.8333 oraz zadanej dł. ogrodz: 100
Iter.no: 7, szukana wart. max. p. obsz. ogrodz.: 271.0179, przy dł. boków: 9.6333, 7.0333 oraz zadanej dł. ogrodz: 100
Iter.no: 8, szukana wart. max. p. obsz. ogrodz.: 277.1378, przy dł. boków: 8.7333, 7.9333 oraz zadanej dł. ogrodz: 100
Iter.no: 9, szukana wart. max. p. obsz. ogrodz.: 277.1378, przy dł. boków: 8.7333, 7.9333 oraz zadanej dł. ogrodz: 100
Iter.no: 10, szukana wart. max. p. obsz. ogrodz.: 277.1378, przy dł. boków: 8.7333, 7.9333 oraz zadanej dł. ogrodz: 100
Iter.no: 11, szukana wart. max. p. obsz. ogrodz.: 277.7778, przy dł. boków: 8.3339, 8.3328 oraz zadanej dł. ogrodz: 100
Iter.no: 12, szukana wart. max. p. obsz. ogrodz.: 277.7778, przy dł. boków: 8.3339, 8.3328 oraz zadanej dł. ogrodz: 100
Iter.no: 13, szukana wart. max. p. obsz. ogrodz.: 277.7778, przy dł. boków: 8.3339, 8.3328 oraz zadanej dł. ogrodz: 100
Iter.no: 14, szukana wart. max. p. obsz. ogrodz.: 277.7778, przy dł. boków: 8.3333, 8.3333 oraz zadanej dł. ogrodz: 100
Iter.no: 15, szukana wart. max. p. obsz. ogrodz.: 277.7778, przy dł. boków: 8.3333, 8.3333 oraz zadanej dł. ogrodz: 100
Iter.no: 16, szukana wart. max. p. obsz. ogrodz.: 277.7778, przy dł. boków: 8.3333, 8.3333 oraz zadanej dł. ogrodz: 100
Iter.no: 17, szukana wart. max. p. obsz. ogrodz.: 277.7778, przy dł. boków: 8.3333, 8.3333 oraz zadanej dł. ogrodz: 100
Iter.no: 18, szukana wart. max. p. obsz. ogrodz.: 277.7778, przy dł. boków: 8.3333, 8.3333 oraz zadanej dł. ogrodz: 100
Iter.no: 19, szukana wart. max. p. obsz. ogrodz.: 277.7778, przy dł. boków: 8.3333, 8.3333 oraz zadanej dł. ogrodz: 100

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
```

```
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

Pole obsz. OPTYMALNIE ogrodzonego: 277.7778, przy dł. boków obsz. podst.: 8.3333, 8.3333 oraz przy zad. dł. ogrodzenia: 100
```

Ponadto przedstawiono wyniki dla 6 poletek pól stykających się na obrysie prostokąta o dwóch wierszach i trzech elementach podpól:

```
Podaj dlugość ogrodzenia sześciu przylegających do siebie obszarów: 100
Iter.no:1, szukana wart. max. p. obsz. ogrodz.: 3, przy dł. boków: 1, 0.5 oraz zadanej dł. ogrodz: 100
Iter.no:2, szukana wart. max. p. obsz. ogrodz.: 3, przy dł. boków: 1, 0.5 oraz zadanej dł. ogrodz: 100
Iter.no:3, szukana wart. max. p. obsz. ogrodz.: 3, przy dł. boków: 1, 0.5 oraz zadanej dł. ogrodz: 100
Iter.no:4, szukana wart. max. p. obsz. ogrodz.: 206.0298, przy dł. boków: 4.9714, 6.9072 oraz zadanej dł. ogrodz: 100
Iter.no:5, szukana wart. max. p. obsz. ogrodz.: 206.0298, przy dł. boków: 4.9714, 6.9072 oraz zadanej dł. ogrodz: 100
Iter.no:6, szukana wart. max. p. obsz. ogrodz.: 206.0298, przy dł. boków: 4.9714, 6.9072 oraz zadanej dł. ogrodz: 100
Iter.no:7, szukana wart. max. p. obsz. ogrodz.: 193.0025, przy dł. boków: 7.0626, 4.5546 oraz zadanej dł. ogrodz: 100
Iter.no:8, szukana wart. max. p. obsz. ogrodz.: 206.8961, przy dł. boków: 6.017, 5.7309 oraz zadanej dł. ogrodz: 100
Iter.no:9, szukana wart. max. p. obsz. ogrodz.: 206.8961, przy dł. boków: 6.017, 5.7309 oraz zadanej dł. ogrodz: 100
Iter.no:10, szukana wart. max. p. obsz. ogrodz.: 206.8961, przy dł. boków: 6.017, 5.7309 oraz zadanej dł. ogrodz: 100
Iter.no:11, szukana wart. max. p. obsz. ogrodz.: 208.3333, przy dł. boków: 5.5564, 6.2491 oraz zadanej dł. ogrodz: 100
Iter.no:12, szukana wart. max. p. obsz. ogrodz.: 208.3333, przy dł. boków: 5.5564, 6.2491 oraz zadanej dł. ogrodz: 100
Iter.no:13, szukana wart. max. p. obsz. ogrodz.: 208.3333, przy dł. boków: 5.5564, 6.2491 oraz zadanej dł. ogrodz: 100
Iter.no:14, szukana wart. max. p. obsz. ogrodz.: 208.3333, przy dł. boków: 5.5556, 6.25 oraz zadanej dł. ogrodz: 100
Iter.no:15, szukana wart. max. p. obsz. ogrodz.: 208.3333, przy dł. boków: 5.5556, 6.25 oraz zadanej dł. ogrodz: 100
Iter.no:16, szukana wart. max. p. obsz. ogrodz.: 208.3333, przy dł. boków: 5.5556, 6.25 oraz zadanej dł. ogrodz: 100
Iter.no:17, szukana wart. max. p. obsz. ogrodz.: 208.3333, przy dł. boków: 5.5556, 6.25 oraz zadanej dł. ogrodz: 100
Iter.no:18, szukana wart. max. p. obsz. ogrodz.: 208.3333, przy dł. boków: 5.5556, 6.25 oraz zadanej dł. ogrodz: 100
Iter.no:19, szukana wart. max. p. obsz. ogrodz.: 208.3333, przy dł. boków: 5.5556, 6.25 oraz zadanej dł. ogrodz: 100

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

Pole obsz. OPTYMALNIE ogrodzonego: 208.3333, przy dł. boków obsz. podst.: 5.5556, 6.25 oraz przy zadanej dł. ogrodzenia: 100
```

Uwaga:

Na rysunkach 1-4 umieszczonych powyżej starano się zachować j**edynie przybliżone** proporcje (względne) długości boków: **a** względem **b**. Jednakże nie przedstawiono (z uwagi na konieczność zachowania czytelności opisu tych boków na tych rysunkach) rzeczywistych proporcji długości ogrodzenia, to znaczy, odpowiednio:

- tej, zadanej jako parametr zadania optymalizacyjnego przed procesem optymalizacyjnym

oraz

- tej, otrzymanej w wyniku przeprowadzenia optymalizacji.

Stąd, rysunki powyższe należy traktować jako poglądowe przedstawienie sytuacji, w której kształt ogradzanych poletek również ulega pewnej optymalizacji. Mianowicie, w zgodzie ze stanem faktycznym, uzyskanym w wyniku procesu optymalizacyjnego, kształt poletek **odpowiadający obrysowi kwadratu** uzyskujemy jedynie dla przypadku jednego oraz czterech przylegających do siebie poletek (rys. 1 i 3), podczas gdy kształt poletek jedynie **co najwyżej zbliżony** do obrysu kwadratu uzyskujemy dla rozważanego przypadku dwóch i sześciu poletek (rys. 2 i 4).

Przykłady zastosowania funkcji `fmincon` w tworzeniu brył (tj. opakowań) poprzez jedno wycięcie i złożenie figur płaskich

Poniżej przedstawiono podstawowe odmiany zadania określającego:

- maksymalizację objętości bryły, otrzymanej przez złożenie:
- figury otrzymanej przez;
- jedno wycięcie z arkusza (w miarę cienkiego arkusza, wykonanego z blachy lub tektury lub tworzywa sztucznego) o prostokątnym obrysie.

$$f_{\max} \to V_{obj} \qquad \Leftrightarrow \qquad f_{\min} \to -V_{obj}$$

$$\begin{cases} zmn\_dec(1) \ge 0 & \Leftrightarrow \quad -zmn\_dec(1) \le 0 \\ zmn\_dec(2) \ge 0 & \Leftrightarrow \quad -zmn\_dec(2) \le 0 \\ \ldots & \\ zmn\_dec(n) \ge 0 & \Leftrightarrow \quad -zmn\_dec(n) \le 0 \end{cases} \tag{2}$$

$$S_{arkusza} - (S_{figury} + S_{skrawków}) = 0 \qquad \Leftrightarrow \qquad S_{figury} = f_1\{zmn\_dec(1)..zmn\_dec(n)\}$$

$$oraz: \qquad S_{skrawków} = f_2\{zmn\_dec(1)..zmn\_dec(n)\}$$

Nie każda z odmian przedstawionych tutaj kilku podstawowych zadań optymalizacyjnych będzie funkcjonować, dając poprawne wyniki. To znaczy na przykład: w odmianie z ograniczeniami stricte nierównościowymi, albo tylko równościowymi, lub też mieszanymi, albo też z nieznacznymi dodatkowymi ograniczeniami, o nieco innymi charakterze, narzucanymi jedynie na kształt, a nie na rozmiar arkusza!.

Oto poniżej przytoczono przypadek implementacji zadania, w którym z prostokątnego arkusza wycina się trzy przyległe do siebie (w założeniu) figury płaskie:

- po środku prostokąt, będący ścianą boczną walca o wysokości **h** oraz długości podstawy równej **l = 2πr** oraz
- dwóch przyległych do prostokąta kół, każde o promieniu równym **r**

```
%%A.Bernat 30.IV.11, All rights reserved (tested with version: 2010a)%
%%wyw. funk. celu: fc_cylinder_volume_ceq, wraz z funkcją ograniczeń (ogr_cylinder_area_ceq):
%%-------------------------------------------------------------------------------------------%
%%munlock(fc_cylinder_volume_ceq);clear call_number;
%options = optimset('Algorithm', 'interior-point');
%pow_arkusza = input('Określ pow. kwadratu arkusza, do uform. walca (przez 1 wycięcie figury i jej złożenie): ');
%x = fmincon(@(x) fc_cylinder_volume_ceq(x,pow_arkusza), [1 1],[],[],[],[],[],[],...
%@(x) ogr_cylinder_area_ceq(x,pow_arkusza),options);
%disp(['OPTYMALNA objętość opakowania z kwadratowego ark.: '  num2str((1/3)*x(1)^2*sin(pi/3)*x(2))...
%    ', przy promieniu podst. walca: ' num2str(x(1))  ', przy wysokości walca: ' num2str(x(2))...
%    ' ,przy zadanej pow. arkusza: ' num2str(pow_arkusza)]);
%%-------------------------------------------------------------------------------------------%
function obj = fc_cylinder_volume_ceq(x,pow_arkusza)
persistent call_number;
if isempty(call_number)
    call_number = 0;
end;
PI_const=3.141592689;
obj = -PI_const*x(1)^2*x(2);
call_number = call_number +1;
disp(['Iter.no: ' num2str(call_number) ', szuk. objętość opakowania: ' num2str(-obj)...
    ', przy promieniu podst. walca: ' num2str(x(1))  ', oraz wysokości walca: ' num2str(x(2))...
    ' oraz zadanej pow. arkusza opak.: ' num2str(pow_arkusza)]);
%---------koniec funkcji celu---------------------------------%
%-------------------------------------------------------------%
%
%--------początek definicji funkcji ograniczeń----------------%
%uwaga! poniższy kod funkcji ograniczeń powinien być zlokalizowany
%w oddzielnym pliku systemowym o nazwie: ogr_cylinder_area_ceq.m,
%natomiast tutaj - zlokalizowany jedynie z celach porządkowych
function [ogr_nierown,ogr_ronw] = ogr_cylinder_area_ceq(x,pow_arkusza)
PI_const=3.141592689;
ogr_nierown(1) = - x(1);
ogr_nierown(2) = - x(2);
ogr_ronw(1) = pow_arkusza - (2*x(1)+x(2)+2*x(1))*(2*PI_const*x(1));
%---------koniec funkcji ograniczeń---------------------------%
```

Ograniczenia w postaci:

$$f_{\max} \to V_{obj} \qquad \Leftrightarrow \qquad f_{\min} \to -V_{obj} = -\pi \cdot r^2 \cdot h$$

$$\begin{cases} x(1) \ge 0 & \Leftrightarrow \quad -x(1) \le 0 \quad \Leftrightarrow \quad -r \le 0 \\ x(2) \ge 0 & \Leftrightarrow \quad -x(2) \le 0 \quad \Leftrightarrow \quad -h \le 0 \end{cases} , \tag{3}$$

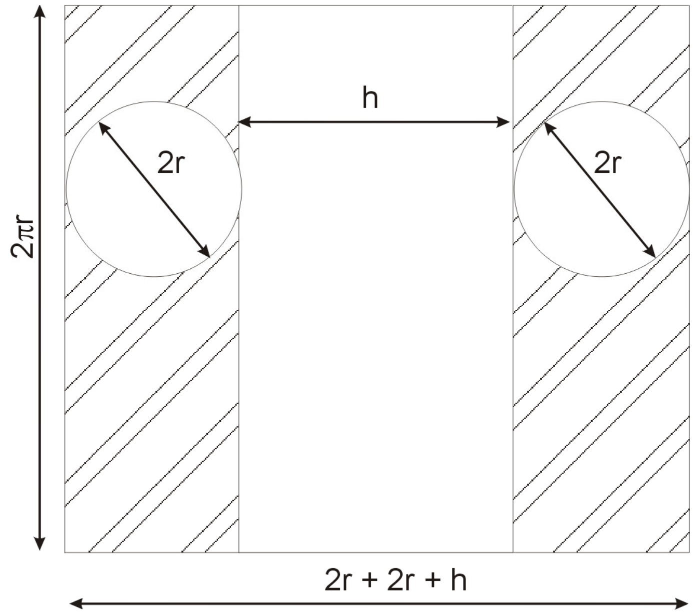
$$S_{arkusza} - (S_{figury} + S_{skrawków}) = 0 \quad \Leftrightarrow \quad S_{figury} + S_{skrawków} = (2 \cdot x(1) + x(2) + 2 \cdot x(1)) \cdot (2\pi \cdot x(1))$$

$$\Leftrightarrow \quad S_{figury} + S_{skrawków} = (2 \cdot r + h + 2 \cdot r) \cdot (2\pi \cdot r)$$

są nieco odmienne w ostatnim ograniczeniu równościowym, względem całkowitej powierzchni walca wycinanej z arkusza o wymiarach **2πr** na **(4r+h)**:

$$S_{figury} = (2 \cdot \pi \cdot x(1)\text{^}2) + 2 \cdot \pi \cdot x(1) \cdot x(2) \Leftrightarrow S_{figury} = 2 \cdot \pi \cdot r^2 + 2 \cdot \pi \cdot r \cdot h\,, \qquad (4)$$

ponieważ nie uwzględnienie powierzchni skrawków (ścinków – odpadów) może prowadzić do mylnego rozwiązania, lub nawet przy dość specyficznie zdefiniowanych ograniczeniach równościowych (jedynie w oparciu o wymiary bryły tworzonej, z pominięciem jawnej deklaracji wymiarów arkusza) do braku istotnego rozwiązania.



Rys.5 Powierzchnia boczna walca oraz dwie powierzchnie jego podstawy w jednokrotnym rozcięciu arkusza o obrysie prostokątnym

Definicja: przez czynność jednokrotnego rozcięcia arkusza (z dość cienkiej blachy, tektury lub na przykład tworzywa sztucznego) rozumiana jest tutaj taka operacja, w wyniku której, w praktyce powstaje pewien nierozłączny zbiór figur płaskich wypukłych (na przykład prostokąta oraz dwóch kół do niego przylegających), z których bryła tworzona jest przez odpowiednie zagięcie poszczególnych podczęści-figur płaskich oraz ich zespolenie ze sobą krawędziowo.

Podanie bardziej szczegółowej definicji ścisłej w sensie topologicznym, jak i przydatnej w praktyce, graniczy z ryzykiem popełnienia jakieś nieścisłości (gafy słowno-definicyjnej). W praktyce bowiem koła przyległe do prostokąta, nigdy nie będą przylegać punktowo, a jedynie na pewnej stosunkowo niewielkiej długości, a arkusz rozcięty w ten sposób, nie będzie się charakteryzował krawędziami o nieskończenie małej grubości. Natomiast same krawędziowe zespalanie zagiętych w przestrzeni 3D figur wypukłych wymagać będzie użycia czynnika zespalającego lub zmieniającego tylko lokalnie właściwości adhezyjne ścian tworzonej figury…

Poniżej podano wynik realizacji procesu optymalizacji w którym jednakże, ograniczenia nierównościowe i równościowe dotyczą tylko specyficznej geometrii tworzonego walca i jego dwóch podstawowych parametrów wymiarowych (tutaj traktowanych jako zmienne decyzyjne!) oraz zadawanej z góry powierzchni arkusza. Innymi słowy arkusz może posiadać dowolną proporcję jednego jego boku do drugiego. Wywołanie tekstu szczegółowej pomocy:

>> help fc_cylinder_volume_ceq

powoduje wyświetlenie tekstu wraz z szeregiem poleceń/wywołań pomocniczych służących uruchomieniu zadania optymalizacji:

```
A.Bernat 30.IV.11, All rights reserved (tested with version: 2010a)%
 %wyw. funk. celu: fc_cylinder_volume_ceq, wraz z funkcją ograniczeń (ogr_cylinder_area_ceq):
 %-------------------------------------------------------------------------------------------%
 %munlock(fc_cylinder_volume_ceq);clear call_number;
 options = optimset('Algorithm', 'interior-point');
 pow_arkusza = input('Określ pow. kwadratu arkusza, do uform. walca (przez 1 wycięcie figury i jej złożenie): ');
 x = fmincon(@(x) fc_cylinder_volume_ceq(x,pow_arkusza), [1 1],[],[],[],[],[],[],...
 @(x) ogr_cylinder_area_ceq(x,pow_arkusza),options);
 disp(['OPTYMALNA objętość opakowania z kwadratowego ark.: '  num2str((1/3)*x(1)^2*sin(pi/3)*x(2))...
     ', przy promieniu podst. walca: ' num2str(x(1))  ', przy wysokości walca: ' num2str(x(2))...
     ' ,przy zadanej pow. arkusza: ' num2str(pow_arkusza)]);
 %-------------------------------------------------------------------------------------------%
```

Wykonanie powyższego zadania:

```
Określ pow. kwadratu arkusza, do uform. walca (przez 1 wycięcie figury i jej złożenie): 100
Iter.no: 1, szuk. objętość opakowania: 3.1416, przy promieniu podst. walca: 1, oraz wysokości walca: 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 2, szuk. objętość opakowania: 3.1416, przy promieniu podst. walca: 1, oraz wysokości walca: 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 3, szuk. objętość opakowania: 3.1416, przy promieniu podst. walca: 1, oraz wysokości walca: 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 4, szuk. objętość opakowania: 40.6035, przy promieniu podst. walca: 1.9439, oraz wysokości walca: 3.4203 oraz zadanej pow. arkusza opak.: 100
Iter.no: 5, szuk. objętość opakowania: 40.6035, przy promieniu podst. walca: 1.9439, oraz wysokości walca: 3.4203 oraz zadanej pow. arkusza opak.: 100
Iter.no: 6, szuk. objętość opakowania: 40.6035, przy promieniu podst. walca: 1.9439, oraz wysokości walca: 3.4203 oraz zadanej pow. arkusza opak.: 100
Iter.no: 7, szuk. objętość opakowania: 35.5393, przy promieniu podst. walca: 1.2963, oraz wysokości walca: 6.732 oraz zadanej pow. arkusza opak.: 100
Iter.no: 8, szuk. objętość opakowania: 35.5393, przy promieniu podst. walca: 1.2963, oraz wysokości walca: 6.732 oraz zadanej pow. arkusza opak.: 100
Iter.no: 9, szuk. objętość opakowania: 35.5393, przy promieniu podst. walca: 1.2963, oraz wysokości walca: 6.732 oraz zadanej pow. arkusza opak.: 100
Iter.no: 10, szuk. objętość opakowania: 37.7868, przy promieniu podst. walca: 1.2556, oraz wysokości walca: 7.6289 oraz zadanej pow. arkusza opak.: 100
Iter.no: 11, szuk. objętość opakowania: 37.7868, przy promieniu podst. walca: 1.2556, oraz wysokości walca: 7.6289 oraz zadanej pow. arkusza opak.: 100
```

```
Iter.no: 12, szuk. objętość opakowania: 37.7868, przy promieniu podst. walca: 1.2556, oraz wysokości walca: 7.6289 oraz zadanej pow. arkusza opak.: 100
Iter.no: 13, szuk. objętość opakowania: 38.1218, przy promieniu podst. walca: 1.1767, oraz wysokości walca: 8.7636 oraz zadanej pow. arkusza opak.: 100
Iter.no: 14, szuk. objętość opakowania: 38.3518, przy promieniu podst. walca: 1.1783, oraz wysokości walca: 8.7925 oraz zadanej pow. arkusza opak.: 100
Iter.no: 15, szuk. objętość opakowania: 38.3519, przy promieniu podst. walca: 1.1783, oraz wysokości walca: 8.7925 oraz zadanej pow. arkusza opak.: 100
Iter.no: 16, szuk. objętość opakowania: 38.3518, przy promieniu podst. walca: 1.1783, oraz wysokości walca: 8.7925 oraz zadanej pow. arkusza opak.: 100
Iter.no: 17, szuk. objętość opakowania: 38.3652, przy promieniu podst. walca: 1.1551, oraz wysokości walca: 9.1529 oraz zadanej pow. arkusza opak.: 100
Iter.no: 18, szuk. objętość opakowania: 38.3876, przy promieniu podst. walca: 1.1553, oraz wysokości walca: 9.1549 oraz zadanej pow. arkusza opak.: 100
Iter.no: 19, szuk. objętość opakowania: 38.3876, przy promieniu podst. walca: 1.1553, oraz wysokości walca: 9.1549 oraz zadanej pow. arkusza opak.: 100
Iter.no: 20, szuk. objętość opakowania: 38.3876, przy promieniu podst. walca: 1.1553, oraz wysokości walca: 9.1549 oraz zadanej pow. arkusza opak.: 100
Iter.no: 21, szuk. objętość opakowania: 38.3877, przy promieniu podst. walca: 1.1518, oraz wysokości walca: 9.2107 oraz zadanej pow. arkusza opak.: 100
Iter.no: 22, szuk. objętość opakowania: 38.3882, przy promieniu podst. walca: 1.1518, oraz wysokości walca: 9.2107 oraz zadanej pow. arkusza opak.: 100
Iter.no: 23, szuk. objętość opakowania: 38.3882, przy promieniu podst. walca: 1.1518, oraz wysokości walca: 9.2107 oraz zadanej pow. arkusza opak.: 100
Iter.no: 24, szuk. objętość opakowania: 38.3882, przy promieniu podst. walca: 1.1518, oraz wysokości walca: 9.2107 oraz zadanej pow. arkusza opak.: 100
Iter.no: 25, szuk. objętość opakowania: 38.3882, przy promieniu podst. walca: 1.1516, oraz wysokości walca: 9.2132 oraz zadanej pow. arkusza opak.: 100
Iter.no: 26, szuk. objętość opakowania: 38.3882, przy promieniu podst. walca: 1.1516, oraz wysokości walca: 9.2132 oraz zadanej pow. arkusza opak.: 100
Iter.no: 27, szuk. objętość opakowania: 38.3882, przy promieniu podst. walca: 1.1516, oraz wysokości walca: 9.2132 oraz zadanej pow. arkusza opak.: 100

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

OPTYMALNA objętość opakowania z kwadratowego ark.: 3.5274, przy promieniu podst. walca: 1.1516, przy wysokości walca: 9.2132 ,przy zadanej pow. arkusza: 100
```

pozwoliło określić wartość promienia podstawy **r** na 1.1516 oraz wysokość walca **h** na 9.2132, co zgodnie z ostatnim ograniczeniem równościowym określa optymalną długość boku pierwszego i drugiego arkusza na odpowiednio: 7.2357 oraz 13.8196.

W praktyce produkcji opakowań dysponujemy zazwyczaj dyskretnym zbiorem znormalizowanych (lokalnie lub ogólnokrajową regulacją prawną) wymiarów arkuszy. Na przykład, w kategorii blach nierdzewnych lub ocynkowanych o uniwersalnym rozmiarze, można posłużyć w praktyce implementacyjnej zadań, ustaleniami Polskiej Normy Materiałowej: PN-H-92203. Norma ta dotyczy jednakże blach o grubościach rzędu paru milimetrów lub więcej oraz powierzchni arkuszy liczonych w metrach kwadratowych.

W rozważanym przypadku, przy punkcie odniesienia, jaki został przyjęty w określeniu znaczeń zmiennych decyzyjnych (określają one: promień podstawy i wysokość walca o poszukiwanej możliwie największej objętości, zamiast: długość boku pierwszego i drugiego prostokątnego arkusza tworzywa!) mamy do czynienia z dyskretnym zbiorem raczej warunków ograniczających (!) niż z dyskretnym ograniczeniem na wartości przyjmowane przez wybrane zmienne decyzyjne.

Równie problematyczne wydaje się przyjęcie podobnego punktu widzenia w określaniu zmiennych decyzyjnych dla przypadków:

- prostopadłościanu wycinanego z prostokątnego arkusza,
- ostrosłupa przy podstawie kwadratowej wycinanego z prostokątnego arkusza,
- graniastosłupa przy podstawie w postaci równobocznego trójkąta wycinanego z arkusza.

Zwłaszcza, że narzucenie dodatkowych ograniczeń, nałożonych na przykład na proporcje arkusza z którego ma być wycięta figura, celem złożenia jej ścian do postaci bryły jest utrudnione. Jednym słowem nie każde intuicyjne wprowadzenie dodatkowych ograniczeń, skutkuje takim wykonaniem zadania optymalizacyjnego, by uzyskane wyniki mogły być uznane za prawidłowe.

Poniżej podano analogiczne do powyższej implementacje zadań optymalizacyjnych odpowiednio dla tychże trzech figur, w kolejności jak powyżej wymienionej:

```
%%A.Bernat 30.IV.11, All rights reserved (tested with version: 2010a)%
%%wyw. funk. celu: fc_box_volume_ceq, wraz z funkcją ograniczeń (ogr_box_area_ceq):
%%-----------------------------------------------------------------------------------------%
%%munlock(fc_box_volume_ceq);clear call_number;
%options = optimset('Algorithm', 'interior-point');
%pow_arkusza = input('Określ pow. arkusza, do uformowania prostopadłościanu (przez 1 wycięcie figury i jej złożenie): ');
%x = fmincon(@(x) fc_box_volume_ceq(x,pow_arkusza), [1 1 1],[],[],[],[],[],[],@(x) ogr_box_area_ceq(x,pow_arkusza),options);
%disp(['OPTYMALNA objętość opakowania: ' num2str(x(1)*x(2)*x(3))...
%    ', przy dł. boków: ' num2str(x(1))  ', ' num2str(x(2)) ', ' num2str(x(3))...
%    ' ,przy zadanej pow. arkusza: ' num2str(pow_arkusza)]);
%%-----------------------------------------------------------------------------------------%
function obj = fc_box_volume_ceq(x,pow_arkusza)
persistent call_number;
if isempty(call_number)
    call_number = 0;
end;
obj = - x(1)*x(2)*x(3);
call_number = call_number +1;
disp(['Iter.no: ' num2str(call_number) ', szuk. wart. objętości opakowania: ' num2str(-obj)...
    ', przy dł. boków: ' num2str(x(1))  ', ' num2str(x(2)) ', ' num2str(x(3))...
    ' oraz zadanej pow. arkusza opak.: ' num2str(pow_arkusza)]);
%---------koniec funkcji celu-----------------------------------%
%--------------------------------------------------------------%
%
%--------początek definicji funkcji ograniczeń----------------%
%uwaga! poniższy kod funkcji ograniczeń powinien być zlokalizowany
%w oddzielnym pliku systemowym o nazwie: ogr_box_area_ceq.m,
%natomiast tutaj - zlokalizowany jedynie z celach porządkowych
function [ogr_nierown,ogr_ronw] = ogr_box_area_ceq(x,pow_arkusza)
ogr_nierown(1) = - x(1);
ogr_nierown(2) = - x(2);
ogr_nierown(3) = - x(3);
ogr_ronw = pow_arkusza - 2*(x(1)+x(3))*(x(2)+2*x(3));
%---------koniec funkcji ograniczeń---------------------------%
```

Wywołanie tekstu szczegółowej pomocy:

```
>> help fc_box_volume_ceq
```

skutkuje wyświetleniem ciągu poleceń/wywołań zadania optymalizacyjnego dla prostopadłościanu:

```
A.Bernat 30.IV.11, All rights reserved (tested with version: 2010a)%
 %wyw. funk. celu: fc_box_volume_ceq, wraz z funkcją ograniczeń (ogr_box_area_ceq):
 %------------------------------------------------------------------------------------------%
 %munlock(fc_box_volume_ceq);clear call_number;
 options = optimset('Algorithm', 'interior-point');
 pow_arkusza = input('Określ pow. arkusza, do uformowania prostopadłościanu (przez 1 wycięcie figury i jej złożenie): ');
 x = fmincon(@(x) fc_box_volume_ceq(x,pow_arkusza), [1 1 1],[],[],[],[],[],[],@(x) ogr_box_area_ceq(x,pow_arkusza),options);
 disp(['OPTYMALNA objętość opakowania: ' num2str(x(1)*x(2)*x(3))...
     ', przy dł. boków: ' num2str(x(1))  ', ' num2str(x(2)) ', ' num2str(x(3))...
     ' ,przy zadanej pow. arkusza: ' num2str(pow_arkusza)]);
 %------------------------------------------------------------------------------------------%
```

W wyniku zaznaczenia tego ciągu poleceń (pomiędzy liniami przerywanymi) oraz wciśnięcia F9 otrzymujemy następujące wyniki:

```
Określ pow. arkusza, do uformowania prostopadłościanu (przez 1 wycięcie figury i jej złożenie): 100
Iter.no: 1, szuk. wart. objętości opakowania: 1, przy dł. boków: 1, 1, 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 2, szuk. wart. objętości opakowania: 1, przy dł. boków: 1, 1, 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 3, szuk. wart. objętości opakowania: 1, przy dł. boków: 1, 1, 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 4, szuk. wart. objętości opakowania: 1, przy dł. boków: 1, 1, 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 5, szuk. wart. objętości opakowania: 66.7307, przy dł. boków: 3.8157, 3.2104, 5.4475 oraz zadanej pow. arkusza opak.: 100
Iter.no: 6, szuk. wart. objętości opakowania: 0.0026088, przy dł. boków: 0.62915, 0.82931, 0.005 oraz zadanej pow. arkusza opak.: 100
Iter.no: 7, szuk. wart. objętości opakowania: 16.3411, przy dł. boków: 2.4078, 2.1052, 3.2237 oraz zadanej pow. arkusza opak.: 100
Iter.no: 8, szuk. wart. objętości opakowania: 16.3411, przy dł. boków: 2.4078, 2.1052, 3.2237 oraz zadanej pow. arkusza opak.: 100
Iter.no: 9, szuk. wart. objętości opakowania: 16.3411, przy dł. boków: 2.4078, 2.1052, 3.2237 oraz zadanej pow. arkusza opak.: 100
Iter.no: 10, szuk. wart. objętości opakowania: 16.3411, przy dł. boków: 2.4078, 2.1052, 3.2237 oraz zadanej pow. arkusza opak.: 100
Iter.no: 11, szuk. wart. objętości opakowania: 35.4257, przy dł. boków: 4.6649, 5.3773, 1.4122 oraz zadanej pow. arkusza opak.: 100
Iter.no: 12, szuk. wart. objętości opakowania: 35.4257, przy dł. boków: 4.6649, 5.3773, 1.4122 oraz zadanej pow. arkusza opak.: 100
Iter.no: 13, szuk. wart. objętości opakowania: 35.4257, przy dł. boków: 4.6649, 5.3773, 1.4122 oraz zadanej pow. arkusza opak.: 100
Iter.no: 14, szuk. wart. objętości opakowania: 35.4257, przy dł. boków: 4.6649, 5.3773, 1.4122 oraz zadanej pow. arkusza opak.: 100
Iter.no: 15, szuk. wart. objętości opakowania: 35.3353, przy dł. boków: 3.3709, 5.5263, 1.8968 oraz zadanej pow. arkusza opak.: 100
Iter.no: 16, szuk. wart. objętości opakowania: 36.2424, przy dł. boków: 4.0179, 5.4518, 1.6545 oraz zadanej pow. arkusza opak.: 100
Iter.no: 17, szuk. wart. objętości opakowania: 36.2424, przy dł. boków: 4.0179, 5.4518, 1.6545 oraz zadanej pow. arkusza opak.: 100
Iter.no: 18, szuk. wart. objętości opakowania: 36.2424, przy dł. boków: 4.0179, 5.4518, 1.6545 oraz zadanej pow. arkusza opak.: 100
Iter.no: 19, szuk. wart. objętości opakowania: 36.2424, przy dł. boków: 4.0179, 5.4518, 1.6545 oraz zadanej pow. arkusza opak.: 100
Iter.no: 20, szuk. wart. objętości opakowania: 36.5898, przy dł. boków: 3.531, 6.1376, 1.6884 oraz zadanej pow. arkusza opak.: 100
Iter.no: 21, szuk. wart. objętości opakowania: 36.5898, przy dł. boków: 3.531, 6.1376, 1.6884 oraz zadanej pow. arkusza opak.: 100
Iter.no: 22, szuk. wart. objętości opakowania: 36.5898, przy dł. boków: 3.531, 6.1376, 1.6884 oraz zadanej pow. arkusza opak.: 100
Iter.no: 23, szuk. wart. objętości opakowania: 36.5898, przy dł. boków: 3.531, 6.1376, 1.6884 oraz zadanej pow. arkusza opak.: 100
Iter.no: 24, szuk. wart. objętości opakowania: 36.9184, przy dł. boków: 3.3144, 6.6158, 1.6836 oraz zadanej pow. arkusza opak.: 100
Iter.no: 25, szuk. wart. objętości opakowania: 36.9184, przy dł. boków: 3.3144, 6.6158, 1.6836 oraz zadanej pow. arkusza opak.: 100
Iter.no: 26, szuk. wart. objętości opakowania: 36.9184, przy dł. boków: 3.3144, 6.6158, 1.6836 oraz zadanej pow. arkusza opak.: 100
Iter.no: 27, szuk. wart. objętości opakowania: 36.9184, przy dł. boków: 3.3144, 6.6158, 1.6836 oraz zadanej pow. arkusza opak.: 100
Iter.no: 28, szuk. wart. objętości opakowania: 37.0371, przy dł. boków: 3.3364, 6.6604, 1.6667 oraz zadanej pow. arkusza opak.: 100
Iter.no: 29, szuk. wart. objętości opakowania: 37.0371, przy dł. boków: 3.3364, 6.6604, 1.6667 oraz zadanej pow. arkusza opak.: 100
Iter.no: 30, szuk. wart. objętości opakowania: 37.0371, przy dł. boków: 3.3364, 6.6604, 1.6667 oraz zadanej pow. arkusza opak.: 100
Iter.no: 31, szuk. wart. objętości opakowania: 37.0371, przy dł. boków: 3.3364, 6.6604, 1.6667 oraz zadanej pow. arkusza opak.: 100
Iter.no: 32, szuk. wart. objętości opakowania: 37.037, przy dł. boków: 3.3334, 6.6664, 1.6667 oraz zadanej pow. arkusza opak.: 100
Iter.no: 33, szuk. wart. objętości opakowania: 37.037, przy dł. boków: 3.3334, 6.6664, 1.6667 oraz zadanej pow. arkusza opak.: 100
Iter.no: 34, szuk. wart. objętości opakowania: 37.037, przy dł. boków: 3.3334, 6.6664, 1.6667 oraz zadanej pow. arkusza opak.: 100
Iter.no: 35, szuk. wart. objętości opakowania: 37.037, przy dł. boków: 3.3334, 6.6664, 1.6667 oraz zadanej pow. arkusza opak.: 100
Iter.no: 36, szuk. wart. objętości opakowania: 37.037, przy dł. boków: 3.3333, 6.6667, 1.6667 oraz zadanej pow. arkusza opak.: 100
Iter.no: 37, szuk. wart. objętości opakowania: 37.037, przy dł. boków: 3.3333, 6.6667, 1.6667 oraz zadanej pow. arkusza opak.: 100
Iter.no: 38, szuk. wart. objętości opakowania: 37.037, przy dł. boków: 3.3333, 6.6667, 1.6667 oraz zadanej pow. arkusza opak.: 100
Iter.no: 39, szuk. wart. objętości opakowania: 37.037, przy dł. boków: 3.3333, 6.6667, 1.6667 oraz zadanej pow. arkusza opak.: 100

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

OPTYMALNA objętość opakowania: 37.037, przy dł. boków: 3.3333, 6.6667, 1.6667 ,przy zadanej pow. arkusza: 100
```

Optymalne wymiary prostopadłościanu o bokach a, b, c, w narzuceniu ograniczeń:

$$f_{\max} \to V_{obj} \quad \Leftrightarrow \quad f_{\min} \to -V_{obj} = -a \cdot b \cdot c$$

$$\left\{\begin{matrix} x(1) \ge 0 & \Leftrightarrow & -x(1) \le 0 & \Leftrightarrow & -a \le 0 \\ x(2) \ge 0 & \Leftrightarrow & -x(2) \le 0 & \Leftrightarrow & -b \le 0 \\ x(3) \ge 0 & \Leftrightarrow & -x(3) \le 0 & \Leftrightarrow & -c \le 0 \end{matrix}\right\}, \tag{5}$$

$$S_{arkusza} - (S_{figury} + S_{skrawków}) = 0 \;\Leftrightarrow\; S_{figury} + S_{skrawków} = (2 \cdot x(1) + x(3)) \cdot (x(2) + 2 \cdot x(3))$$

$$\Leftrightarrow\; S_{figury} + S_{skrawków} = (2 \cdot a + c) \cdot (b + 2 \cdot c)$$

gdzie powierzchnię arkusza zadano jako równą 100, dają w rezultacie zaskakujące wartości, równe odpowiednio: 3.333, 6.667 oraz 1.667, wymykające się intuicyjnej kontroli wyników.

Z kolei, dla implementacji zadania znajdowania ostrosłupa (o podstawie kwadratowej o boku **`a`** oraz o wysokości **`H`** oraz wysokości **`h`** 4 trójkątów przylegających do podstawy), o maksymalnej objętości:

```
%%A.Bernat 30.IV.11, All rights reserved (tested with version: 2010a)%
```

```
%%wyw. funk. celu: fc_pyramid_volume_ceq, wraz z funkcją ograniczeń (ogr_pyramid_area_ceq):
%%-----------------------------------------------------------------------------------------%
%%munlock(fc_pyramid_volume_ceq);clear call_number;
%options = optimset('Algorithm', 'interior-point');
%pow_arkusza = input('Określ pow. kwadratu arkusza, do uform. ostrosłupa (przez 1 wycięcie figury i jej złożenie): ');
%x = fmincon(@(x) fc_pyramid_volume_ceq(x,pow_arkusza), [1 1],[],[],[],[],[],[],...
%@(x) ogr_pyramid_area_ceq(x,pow_arkusza),options);
%disp(['OPTYMALNA objętość opakowania z kwadratowego ark.: '  num2str((1/3)*(x(1)^2)*sqrt((x(2)^2)-0.25*x(1)^2))...
%    ', przy dł. boku podstawy: ' num2str(x(1))  ', przy wysokości 4 trójkątów: ' num2str(x(2))...
%    ' ,przy zadanej pow. arkusza: ' num2str(pow_arkusza)]);
%%-----------------------------------------------------------------------------------------%
function obj = fc_pyramd_volume_ceq(x,pow_arkusza)
persistent call_number;
if isempty(call_number)
    call_number = 0;
end;
obj = - (1/3) *(sqrt(3)/2) *(x(1)^2)*sqrt((x(2)^2)-0.25*x(1)^2);
call_number = call_number +1;
disp(['Iter.no: ' num2str(call_number) ', szuk. objętość opakowania: ' num2str(-obj)...
    ', przy dł. boku podstawy: ' num2str(x(1))  ', oraz wysokości 4 trójkątów: ' num2str(x(2))...
    ' oraz zadanej pow. arkusza opak.: ' num2str(pow_arkusza)]);
%---------koniec funkcji celu--------------------------------%
%----------------------------------------------------------%
%
%--------początek definicji funkcji ograniczeń----------------%
%uwaga! poniższy kod funkcji ograniczeń powinien być zlokalizowany
%w oddzielnym pliku systemowym o nazwie: ogr_pyramid_area_ceq.m,
%natomiast tutaj - zlokalizowany jedynie z celach porządkowych
function [ogr_nierown,ogr_ronw] = ogr_pyramid_area_ceq(x,pow_arkusza)
ogr_nierown(1) = - x(1);
ogr_nierown(2) = - x(2);
ogr_ronw(1) = pow_arkusza - (x(1)+2*x(2))^2;
%---------koniec funkcji ograniczeń---------------------------%
```

uruchomienie zadania o postaci ograniczeń i funkcji celu jak poniżej, a implementacji w kodzie jak powyżej:

$$f_{\max} \to V_{obj} \quad \Leftrightarrow \quad f_{\min} \to -V_{obj} = -\frac{1}{3}\cdot\frac{\sqrt{3}}{2}\cdot a^2\cdot\sqrt{h^2-0{,}25\cdot a^2}$$

$$\begin{cases} x(1)\ge 0 & \Leftrightarrow \quad -x(1)\le 0 \quad \Leftrightarrow \quad -a\le 0 \\ x(2)\ge 0 & \Leftrightarrow \quad -x(2)\le 0 \quad \Leftrightarrow \quad -h\le 0 \end{cases} , \qquad (6)$$

$$S_{arkusza} - (S_{figury} + S_{skrawków}) = 0 \quad \Leftrightarrow \quad S_{figury} + S_{skrawków} = (x(1) + 2\cdot x(2))^{\wedge 2}$$

$$\Leftrightarrow \quad S_{figury} + S_{skrawków} = (a + 2\cdot h)^2$$

daje następujące wyniki:

```
Określ pow. kwadratu arkusza, do uform. ostrosłupa (przez 1 wycięcie figury i jej złożenie): 100
Iter.no: 1, szuk. objętość opakowania: 0.28868, przy dł. boku podstawy: 1, oraz wysokości 4 trójkątów: 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 2, szuk. objętość opakowania: 0.28868, przy dł. boku podstawy: 1, oraz wysokości 4 trójkątów: 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 3, szuk. objętość opakowania: 0.28868, przy dł. boku podstawy: 1, oraz wysokości 4 trójkątów: 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 4, szuk. objętość opakowania: 42.4455, przy dł. boku podstawy: 4.4243, oraz wysokości 4 trójkątów: 6.8712 oraz zadanej pow. arkusza opak.: 100
Iter.no: 5, szuk. objętość opakowania: 9.0587, przy dł. boku podstawy: 2.7121, oraz wysokości 4 trójkątów: 3.9356 oraz zadanej pow. arkusza opak.: 100
Iter.no: 6, szuk. objętość opakowania: 9.0587, przy dł. boku podstawy: 2.7121, oraz wysokości 4 trójkątów: 3.9356 oraz zadanej pow. arkusza opak.: 100
Iter.no: 7, szuk. objętość opakowania: 9.0587, przy dł. boku podstawy: 2.7121, oraz wysokości 4 trójkątów: 3.9356 oraz zadanej pow. arkusza opak.: 100
Iter.no: 8, szuk. objętość opakowania: 6.9095, przy dł. boku podstawy: 4.8534, oraz wysokości 4 trójkątów: 2.5813 oraz zadanej pow. arkusza opak.: 100
Iter.no: 9, szuk. objętość opakowania: 6.9095, przy dł. boku podstawy: 4.8534, oraz wysokości 4 trójkątów: 2.5813 oraz zadanej pow. arkusza opak.: 100
Iter.no: 10, szuk. objętość opakowania: 6.9095, przy dł. boku podstawy: 4.8534, oraz wysokości 4 trójkątów: 2.5813 oraz zadanej pow. arkusza opak.: 100
Iter.no: 11, szuk. objętość opakowania: 9.2752, przy dł. boku podstawy: 2.947, oraz wysokości 4 trójkątów: 3.5265 oraz zadanej pow. arkusza opak.: 100
Iter.no: 12, szuk. objętość opakowania: 9.2752, przy dł. boku podstawy: 2.947, oraz wysokości 4 trójkątów: 3.5265 oraz zadanej pow. arkusza opak.: 100
Iter.no: 13, szuk. objętość opakowania: 9.2752, przy dł. boku podstawy: 2.947, oraz wysokości 4 trójkątów: 3.5265 oraz zadanej pow. arkusza opak.: 100
Iter.no: 14, szuk. objętość opakowania: 10.4971, przy dł. boku podstawy: 3.2742, oraz wysokości 4 trójkątów: 3.3629 oraz zadanej pow. arkusza opak.: 100
Iter.no: 15, szuk. objętość opakowania: 10.4971, przy dł. boku podstawy: 3.2742, oraz wysokości 4 trójkątów: 3.3629 oraz zadanej pow. arkusza opak.: 100
Iter.no: 16, szuk. objętość opakowania: 10.4971, przy dł. boku podstawy: 3.2742, oraz wysokości 4 trójkątów: 3.3629 oraz zadanej pow. arkusza opak.: 100
Iter.no: 17, szuk. objętość opakowania: 9.6774, przy dł. boku podstawy: 4.6346, oraz wysokości 4 trójkątów: 2.6827 oraz zadanej pow. arkusza opak.: 100
Iter.no: 18, szuk. objętość opakowania: 11.9181, przy dł. boku podstawy: 3.9544, oraz wysokości 4 trójkątów: 3.0228 oraz zadanej pow. arkusza opak.: 100
Iter.no: 19, szuk. objętość opakowania: 11.9181, przy dł. boku podstawy: 3.9544, oraz wysokości 4 trójkątów: 3.0228 oraz zadanej pow. arkusza opak.: 100
Iter.no: 20, szuk. objętość opakowania: 11.9181, przy dł. boku podstawy: 3.9544, oraz wysokości 4 trójkątów: 3.0228 oraz zadanej pow. arkusza opak.: 100
Iter.no: 21, szuk. objętość opakowania: 11.9227, przy dł. boku podstawy: 4.0281, oraz wysokości 4 trójkątów: 2.9859 oraz zadanej pow. arkusza opak.: 100
Iter.no: 22, szuk. objętość opakowania: 11.9227, przy dł. boku podstawy: 4.0281, oraz wysokości 4 trójkątów: 2.9859 oraz zadanej pow. arkusza opak.: 100
Iter.no: 23, szuk. objętość opakowania: 11.9227, przy dł. boku podstawy: 4.0281, oraz wysokości 4 trójkątów: 2.9859 oraz zadanej pow. arkusza opak.: 100
Iter.no: 24, szuk. objętość opakowania: 11.9257, przy dł. boku podstawy: 3.9993, oraz wysokości 4 trójkątów: 3.0004 oraz zadanej pow. arkusza opak.: 100
Iter.no: 25, szuk. objętość opakowania: 11.9257, przy dł. boku podstawy: 3.9993, oraz wysokości 4 trójkątów: 3.0004 oraz zadanej pow. arkusza opak.: 100
Iter.no: 26, szuk. objętość opakowania: 11.9257, przy dł. boku podstawy: 3.9993, oraz wysokości 4 trójkątów: 3.0004 oraz zadanej pow. arkusza opak.: 100
Iter.no: 27, szuk. objętość opakowania: 11.9257, przy dł. boku podstawy: 4, oraz wysokości 4 trójkątów: 3 oraz zadanej pow. arkusza opak.: 100
Iter.no: 28, szuk. objętość opakowania: 11.9257, przy dł. boku podstawy: 4, oraz wysokości 4 trójkątów: 3 oraz zadanej pow. arkusza opak.: 100
Iter.no: 29, szuk. objętość opakowania: 11.9257, przy dł. boku podstawy: 4, oraz wysokości 4 trójkątów: 3 oraz zadanej pow. arkusza opak.: 100
Iter.no: 30, szuk. objętość opakowania: 11.9257, przy dł. boku podstawy: 4, oraz wysokości 4 trójkątów: 3 oraz zadanej pow. arkusza opak.: 100
Iter.no: 31, szuk. objętość opakowania: 11.9257, przy dł. boku podstawy: 4, oraz wysokości 4 trójkątów: 3 oraz zadanej pow. arkusza opak.: 100
Iter.no: 32, szuk. objętość opakowania: 11.9257, przy dł. boku podstawy: 4, oraz wysokości 4 trójkątów: 3 oraz zadanej pow. arkusza opak.: 100

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

OPTYMALNA objętość opakowania z kwadratowego ark.: 11.9257, przy dł. boku podstawy: 4, przy wysokości 4 trójkątów: 3 ,przy zadanej pow. arkusza: 100
```

Trzeba przyznać, że długość boku kwadratowej podstawy równa 4 oraz wysokość czterech trójkątów – tworzących cztery ściany ostrosłupa równa 3, dają ładny tandem wartości (na pierwszy rzut oka łatwy do przyjęcia intuicyjnie), oczywiście dla powierzchni arkusza równej 100.

Ostatecznie, dla graniastosłupa, o równobocznym trójkącie w podstawie o boku **a** oraz o jego wysokości **h**, ograniczenia przyjmują nieco bardziej rozwinięty charakter:

$$f_{\max} \to V_{obj} \quad \Leftrightarrow \quad f_{\min} \to -V_{obj} = -\frac{\sqrt{3}}{2} a^2 \cdot h$$

$$\begin{cases} x(1) \ge 0 & \Leftrightarrow \quad -x(1) \le 0 \quad \Leftrightarrow \quad -a \le 0 \\ x(2) \ge 0 & \Leftrightarrow \quad -x(2) \le 0 \quad \Leftrightarrow \quad -h \le 0 \end{cases}$$

$$S_{arkusza} - (S_{figury} + S_{skrawków}) = 0 \quad \Leftrightarrow \qquad , \qquad (7)$$

$$S_{figury} + S_{skrawków} = \left[(x(1) + 2 \cdot \cos(\frac{\pi}{6}) \cdot x(2) \right] \cdot \left[(x(2) + \frac{\sqrt{3}}{2} \cdot x(1) + \sin(\frac{\pi}{6}) \cdot x(2) \right] \Leftrightarrow$$

$$S_{figury} + S_{skrawków} = \left[a + 2 \cdot \cos(\frac{\pi}{6}) \cdot h \right] \cdot \left[(h + \frac{\sqrt{3}}{2} \cdot a + \sin(\frac{\pi}{6}) \cdot h \right]$$

Oto kod źródłowy poniżej:

```
%%A.Bernat 30.IV.11, All rights reserved (tested with version: 2010a)%
%%wyw. funk. celu: fc_prism_volume_ceq, wraz z funkcją ograniczeń (ogr_prism_area_ceq):
%%-----------------------------------------------------------------------------------------%
%%munlock(fc_prism_volume_ceq);clear call_number;
%options = optimset('Algorithm', 'interior-point');
%pow_arkusza = input('Określ pow. kwadratu arkusza, do uform. graniastosłupa (przez 1 wycięcie figury i jej złożenie):
');
%x = fmincon(@(x) fc_prism_volume_ceq(x,pow_arkusza), [1 1],[],[],[],[],[],[],...
%@(x) ogr_prism_area_ceq(x,pow_arkusza),options);
%disp(['OPTYMALNA objętość opakowania z kwadratowego ark.: '  num2str((1/3)*x(1)^2*sin(pi/3)*x(2))...
%    ', przy dł. boku podst. trójk. foremnego: ' num2str(x(1))  ', przy wysokości boków: ' num2str(x(2))...
%    ' ,przy zadanej pow. arkusza: ' num2str(pow_arkusza)]);
%%-----------------------------------------------------------------------------------------%
function obj = fc_prism_volume_ceq(x,pow_arkusza)
persistent call_number;
if isempty(call_number)
    call_number = 0;
end;
obj = - (sqrt(3)/2)*x(1)^2* x(2);
call_number = call_number +1;
disp(['Iter.no: ' num2str(call_number) ', szuk. objętość opakowania: ' num2str(-obj)...
    ', przy dł. boku podstawy: ' num2str(x(1))   ', oraz wysokości 4 trójkątów: ' num2str(x(2))...
    ' oraz zadanej pow. arkusza opak.: ' num2str(pow_arkusza)]);
%---------koniec funkcji celu----------------------------------%
%--------------------------------------------------------------%
%
%--------początek definicji funkcji ograniczeń----------------%
%uwaga! poniższy kod funkcji ograniczeń powinien być zlokalizowany
%w oddzielnym pliku systemowym o nazwie: ogr_prism_area_ceq.m,
%natomiast tutaj - zlokalizowany jedynie z celach porządkowych
function [ogr_nierown,ogr_ronw] = ogr_prism_area_ceq(x,pow_arkusza)
ogr_nierown(1) = - x(1);
ogr_nierown(2) = - x(2);
ogr_ronw(1) = pow_arkusza - (x(1)+2*cos(pi/6)*x(2))*(x(2)+(sqrt(3)/2)*x(1)+sin(pi/6)*x(2));
%---------koniec funkcji ograniczeń----------------------------%
```

oraz wynik uruchomienia zadania:

```
Określ pow. kwadratu arkusza, do uform. graniastosłupa (przez 1 wycięcie figury i jej złożenie): 100
Iter.no: 1, szuk. objętość opakowania: 0.28868, przy dł. boku podstawy: 1, oraz wysokości 4 trójkątów: 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 2, szuk. objętość opakowania: 0.28868, przy dł. boku podstawy: 1, oraz wysokości 4 trójkątów: 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 3, szuk. objętość opakowania: 0.28868, przy dł. boku podstawy: 1, oraz wysokości 4 trójkątów: 1 oraz zadanej pow. arkusza opak.: 100
Iter.no: 4, szuk. objętość opakowania: 114.5068, przy dł. boku podstawy: 6.5669, oraz wysokości 4 trójkątów: 9.1981 oraz zadanej pow. arkusza opak.: 100
Iter.no: 5, szuk. objętość opakowania: 21.0706, przy dł. boku podstawy: 3.7835, oraz wysokości 4 trójkątów: 5.0991 oraz zadanej pow. arkusza opak.: 100
Iter.no: 6, szuk. objętość opakowania: 21.0706, przy dł. boku podstawy: 3.7835, oraz wysokości 4 trójkątów: 5.0991 oraz zadanej pow. arkusza opak.: 100
Iter.no: 7, szuk. objętość opakowania: 21.0706, przy dł. boku podstawy: 3.7835, oraz wysokości 4 trójkątów: 5.0991 oraz zadanej pow. arkusza opak.: 100
Iter.no: 8, szuk. objętość opakowania: 31.7895, przy dł. boku podstawy: 7.0923, oraz wysokości 4 trójkątów: 2.1893 oraz zadanej pow. arkusza opak.: 100
Iter.no: 9, szuk. objętość opakowania: 31.7895, przy dł. boku podstawy: 7.0923, oraz wysokości 4 trójkątów: 2.1893 oraz zadanej pow. arkusza opak.: 100
Iter.no: 10, szuk. objętość opakowania: 31.7895, przy dł. boku podstawy: 7.0923, oraz wysokości 4 trójkątów: 2.1893 oraz zadanej pow. arkusza opak.: 100
Iter.no: 11, szuk. objętość opakowania: 30.6081, przy dł. boku podstawy: 7.3065, oraz wysokości 4 trójkątów: 1.9861 oraz zadanej pow. arkusza opak.: 100
Iter.no: 12, szuk. objętość opakowania: 30.6081, przy dł. boku podstawy: 7.3065, oraz wysokości 4 trójkątów: 1.9861 oraz zadanej pow. arkusza opak.: 100
Iter.no: 13, szuk. objętość opakowania: 30.6081, przy dł. boku podstawy: 7.3065, oraz wysokości 4 trójkątów: 1.9861 oraz zadanej pow. arkusza opak.: 100
Iter.no: 14, szuk. objętość opakowania: 30.6372, przy dł. boku podstawy: 7.1678, oraz wysokości 4 trójkątów: 2.0657 oraz zadanej pow. arkusza opak.: 100
Iter.no: 15, szuk. objętość opakowania: 30.6372, przy dł. boku podstawy: 7.1678, oraz wysokości 4 trójkątów: 2.0657 oraz zadanej pow. arkusza opak.: 100
Iter.no: 16, szuk. objętość opakowania: 30.6372, przy dł. boku podstawy: 7.1678, oraz wysokości 4 trójkątów: 2.0657 oraz zadanej pow. arkusza opak.: 100
Iter.no: 17, szuk. objętość opakowania: 30.6372, przy dł. boku podstawy: 7.1639, oraz wysokości 4 trójkątów: 2.068 oraz zadanej pow. arkusza opak.: 100
Iter.no: 18, szuk. objętość opakowania: 30.6372, przy dł. boku podstawy: 7.1639, oraz wysokości 4 trójkątów: 2.068 oraz zadanej pow. arkusza opak.: 100
Iter.no: 19, szuk. objętość opakowania: 30.6372, przy dł. boku podstawy: 7.1639, oraz wysokości 4 trójkątów: 2.068 oraz zadanej pow. arkusza opak.: 100
Iter.no: 20, szuk. objętość opakowania: 30.6372, przy dł. boku podstawy: 7.1638, oraz wysokości 4 trójkątów: 2.068 oraz zadanej pow. arkusza opak.: 100
Iter.no: 21, szuk. objętość opakowania: 30.6372, przy dł. boku podstawy: 7.1638, oraz wysokości 4 trójkątów: 2.068 oraz zadanej pow. arkusza opak.: 100
Iter.no: 22, szuk. objętość opakowania: 30.6372, przy dł. boku podstawy: 7.1638, oraz wysokości 4 trójkątów: 2.068 oraz zadanej pow. arkusza opak.: 100
```

```
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the function tolerance,
and constraints were satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

OPTYMALNA objętość opakowania z kwadratowego ark.: 30.6372, przy dł. boku podst. trójk. foremnego: 7.1638, przy wysokości boków: 2.068 ,przy zadanej pow. arkusza: 100
```

Wynik otrzymany w tym ostatnim przypadku może nie jest imponujący (objętość opakowania w postaci graniastosłupa o podstawie foremnego trójkąta równa ~30.7, przy długości boku ~7.17 oraz wysokości ~2.1), lecz wskazuje na trzecie miejsce pod względem objętości brył utworzonych przez wycięcie nierozłącznego zbioru ich ścian z arkusza o powierzchni 100. Pierwsze miejsce zajmuje walec z objętością rzędu 38 jednostek, natomiast drugie prostopadłościan z objętością rzędu 37 jednostek.

Przy charakterystycznym wycięciu pól podstawy oraz powierzchni bocznej, to właśnie walec zwyciężył w kategorii najwydajniejszych, jeśli chodzi o maksymalizację objętości- pojemności opakowania z arkusza o zadanej powierzchni, wyprzedzając propozycję opakowania w postaci prostopadłościanu. Tutaj po raz kolejny intuicyjność poszukiwanych rozwiązań zawodzi. Choć tak postrzegana intuicyjność jest w tym miejscu zdefiniowana raczej jako łączny udział doświadczenia i horyzontu myślowego osoby implementującej zadanie w postaci kodu uruchomieniowego.

W zestawie m-skryptów archiwum zip umieszczono zarówno przetestowane uruchomieniowo skrypty (te posiadają odrębne podkatalogi!), jak i szereg skryptów (w katalogu głównym zawartości tego archiwum zip) które być może czekają na bardziej szczegółowe objaśnienia i modyfikacje. Zwłaszcza, że próby narzucania ograniczeń co do proporcji arkusza powierzchni przeznaczonej na rozcięcie, bardziej niż gdziekolwiek pośród omówionych powyżej przykładów wymykają się intuicyjnej definicji.

Na koniec, małe zapytanie: po co to wszystko? (w realizacji implementacji, w objaśnieniach i przytaczaniu kodów).
Po pierwsze istnieje na rynku wydawniczym niezwykle mało publikacji zachęcających do eksperymentowania z implementacjami w środowiskach obliczeniowych, takich jak Matlab, a z drugiej strony zatrzymujących się tylko na problemie geometrycznych prostych zależności figur, może z niewielką domieszką paru pojęć z topologii.
Po drugie 'zaangażowanie' w rozważania optymalizacyjne pojęć znanych z mechaniki klasycznej, oprócz samych definicji i pojęć dotyczących tylko euklidesowej przestrzeni oraz podstaw geometrii figur płaskich i brył, powoduje w najprostszym choćby przypadku, znaczne wykraczanie modelowanych zjawisk poza ramy i możliwości obliczeniowe zestawu funkcji środowiska Matlab. Mam tu na myśli pojęcia masy, pędu, prędkości, biegu czasu. Wszystkie bowiem rozważane tutaj przypadki dotyczyły zagadnień optymalizacji statycznej z ograniczeniami.
Proszę mi wybaczyć brak rysunków 6-8-11, które by przedstawiały odpowiednio: a) rozcięty gotowy do złożenia ostrosłup, następnie b) prostopadłościan (ten miałby w układzie krzyża nierównoramiennego rozlokowane sześć prostokątów) oraz c) graniastosłup o podstawie trójkąta foremnego (ten miałby układ trójkąta foremnego u podstawy z promieniście rozchodzącymi się trzema prostokątami, choć jednocześnie ograniczenia na powierzchnię arkusza – jak obecnie wnioskuję; wcale nie dotyczyłyby jednokrotnego rozcięcia arkusza w tym przypadku; jednym słowem gdzieś albo: zapodziała się górna podstawa tego graniastosłupa, albo: nie mamy do czynienia z jednokrotnym rozcięciem, według definicji gdzieś powyżej w tym dokumencie; a to jedynie motywuje do uważnego rysowania i studiowania rysunków pomocniczych-poglądowych, pomocnych w określaniu postaci funkcji celu i zbioru ograniczeń).
Na rysunki bowiem przyjdzie kolei na zajęciach…
…

Rysunki rzeczywiście pojawiły się na zajęciach (poniżej na ostatniej stronie) oraz nieco później pewne erraty tekstu:

- „..graniastosłup o podstawie w postaci trójkąta **równobocznego..”**, zamiast: **równoramiennego**,
- „..Polska Norma Materiałowa : **PN 92203..”**, zamiast: **PN 92200**,
- różne wtrącenia, niedopatrzenia, przekręcenia wzorów ,na przykład objętość graniastosłupa w funkcji celu:

$$f_{\max} \to V_{obj} \qquad \Leftrightarrow \quad f_{\min} \to -V_{obj} = -\frac{\sqrt{3}}{2} a^2 \cdot h,$$

zamiast:

$$f_{\max} \to V_{obj} \qquad \Leftrightarrow \quad f_{\min} \to -V_{obj} = -\frac{1}{3} a^2 \cdot h,$$

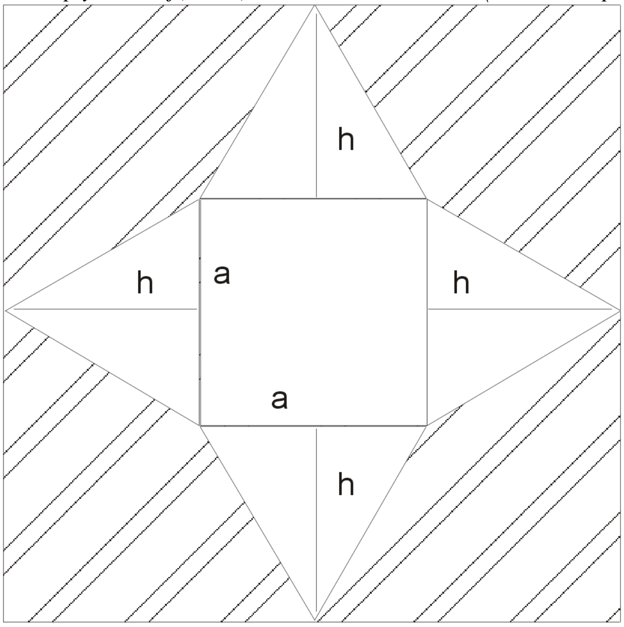
- różne postacie wzorów na ograniczenia równościowe, na przykład powierzchni arkusza w rozcięciu ścian graniastosłupa:

$$S_{figury} + S_{skrawków} = \left[(x(1) + 2\cdot\cos(\frac{\pi}{6})\cdot x(2)) \right] \cdot \left[(x(2) + \frac{\sqrt{3}}{2}\cdot x(1) + \sin(\frac{\pi}{6})\cdot x(2) \right] \Leftrightarrow$$

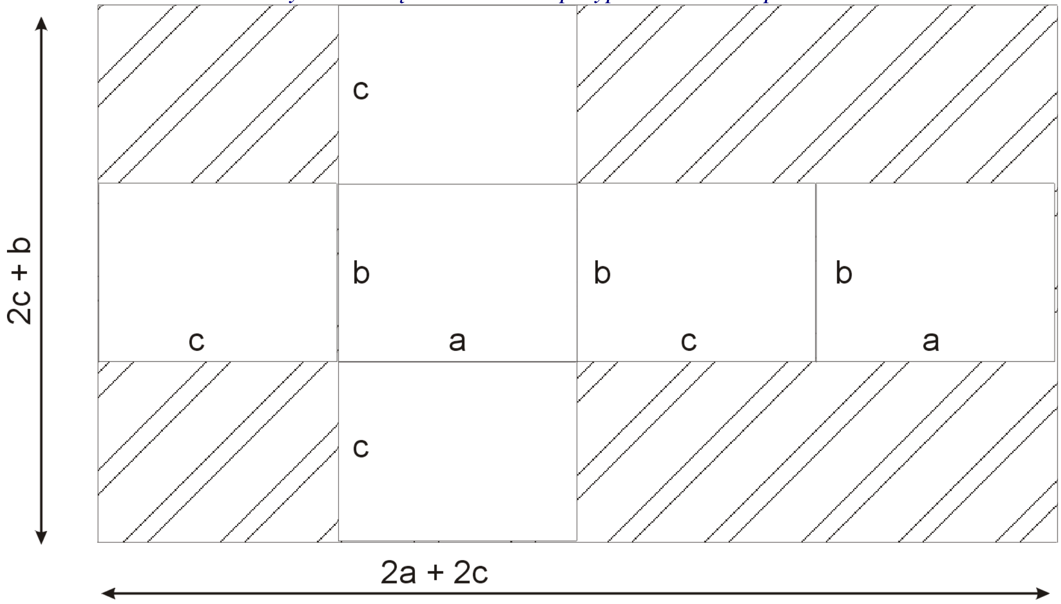
,

$$S_{figury} + S_{skrawków} = \left[a + 2\cdot\cos(\frac{\pi}{6})\cdot h \right] \cdot \left[(h + \frac{\sqrt{3}}{2}\cdot a + \sin(\frac{\pi}{6})\cdot h \right]$$

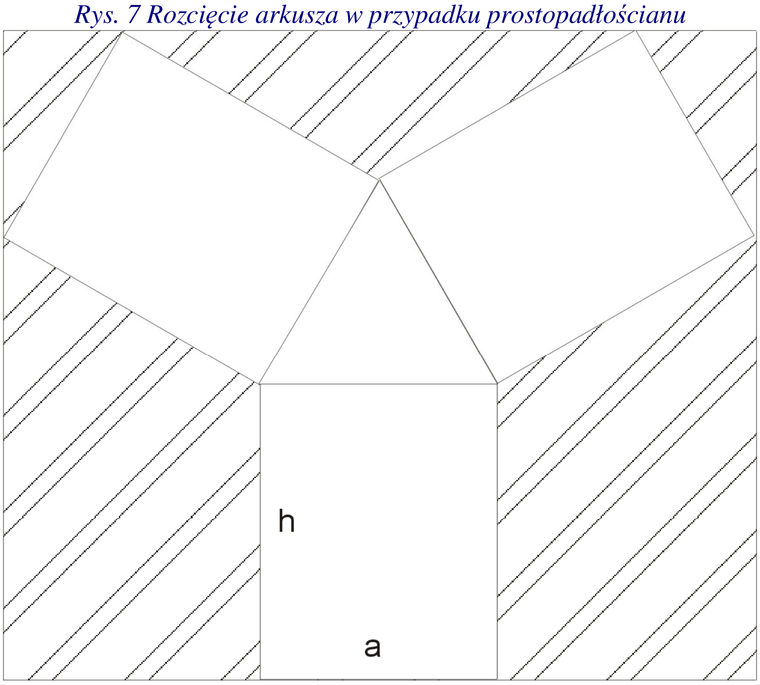
zamiast wersji poprzedniej tej zależności
Poprawiono oczywiście kod źródłowy tam gdzie wymagał on korekty zależności, lecz nie przytoczono wyników nowych uruchomień.



Rys.6 Rozcięcie arkusza w przypadku ostrosłupa



Rys. 7 Rozcięcie arkusza w przypadku prostopadłościanu



Rys.8 Rozcięcie arkusza w przypadku graniastosłupa