

SPRAWOZDANIE - RADIONADAJNIKI

MODELOWANIE MATEMATYCZNE CZĘŚĆ 2

MARTA SOMMER X3

18.05.2013R.

Treść zadania:

Na obszarze $[0,10] \times [0,10]$ mamy dane punkty (plik „radio_nadajniki.txt”), w których mogą znaleźć się nadajniki.

Zasięg nadawania każdego nadajnika:

- i. wynosi dokładnie tyle, co wartość w 4. kolumnie pliku,
- ii. można go ustawić między wartościami 3. i 4. kolumny w pliku.

Nie w każdym potencjalnym punkcie musimy włączyć nadajnik.

Celem jest wybranie, w których z danych punktów powinny stanąć nadajniki oraz (ii) jaki powinny mieć zasięg tak, aby:

- w każdym punkcie o całkowitych współrzędnych z obszaru $[0,10] \times [0,10]$ można było słuchać radia (czyli każdy taki punkt był w zasięgu jakiegoś nadajnika).
- łączny zasięg nadajników był jak najmniejszy.

Poniżej załączam kod do pierwszej części (i) programu:

```
1 # i
2
3 param n := 10;
4 set I := {0..n};
5 set R := {read "radio_nadajniki.txt" as "<ln,2n>" comment "#"};
6 set P := {<i,j> in I*I};
7
8 param zas[R] := read "radio_nadajniki.txt" as "<ln,2n>4n" comment "#";
9 param odleg[<i,j,k,l> in P*R] := sqrt((i-k)^2 + (j-l)^2);
10
11 var x[R] binary;
12
13 minimize fcelu: sum <i,j> in R do zas[i,j]*x[i,j];
14
15 param czywzaz[<i,j,k,l> in P*R] := if odleg[i,j,k,l] <= zas[k,l] then 1 else 0 end;
16
17 subto ogr: forall <i,j> in P do sum <k,l> in R do czywzaz[i,j,k,l]*x[k,l] >= 1;
18
19
```

Opiszę kolejne linijki kodu i postaram się wytłumaczyć, czym różni się zamieszczone powyżej rozwiązanie od kodu, który udało mi się napisać w trakcie laboratorium.

Tak więc po kolei:

(linia 3) param $n := 10$;

Na początek ustalam parametr n . Będzie on wyrażał długość naszego obszaru – w tym przypadku 10.

(linia 4) set $I := \{0..n\}$;

Definiuję zbiór liczb od 0 do n (czyli 10). Będzie on potem potrzebny do utworzenia zbioru wszystkich możliwych punktów o współrzędnych całkowitych na naszym obszarze.

W czasie laboratorium zapomniałam tu o punktach z pierwszą współrzędną zerową i rozpatrywałam zbiór zaczynający się od 1.

(linia 5) set $R := \{\text{read "radio_nadajniki.txt" as "<1n,2n>" comment "#"}\}$;

Definiuję zbiór punktów, w których mogą się znaleźć nadajniki. Jest on opisany za pomocą współrzędnych danych punktów.

*(linia 6) set $P := \{<i,j> \text{ in } I*I\}$;*

Definiuję zbiór wszystkich punktów na naszym obszarze. Notacja również za pomocą współrzędnych punktów.

(linia 8) param $zas[R] := \text{read "radio_nadajniki.txt" as "<1n,2n>4n" comment "#"}\}$;

Tworzę parameter zas jako funkcję punktów, gdzie mogą znaleźć się nadajniki (R), zwracającą zasięg tych punktów.

*(linia 9) param $odleg[<i,j,k,l> \text{ in } P*R] := \text{sqrt}((i-k)^2 + (j-l)^2)$;*

Tworzę parametr $odleg$ jako funkcję iloczynu kartezjańskiego dowolnych punktów z obszaru (P) i punktów, gdzie mogą znaleźć się nadajniki (R), zwracającą odległość między tymi punktami.

(linia 11) var $x[R]$ binary;

Definiuję zmienną binarną x zależną od punktów, w których mogą się znaleźć nadajniki (R). Zwraca ona 1, gdy w danym punkcie nadajnik ma zostać włączony i 0 w przeciwnym przypadku.

*(linia 13) minimize fcelu: sum $<i,j> \text{ in } R$ do $zas[i,j]*x[i,j]$;*

Dążę do zminimalizowania funkcji celu danej jako łączny zasięg nadajników.

*(linia 15) param czywzas[<i,j,k,l> in P*R] := if odleg[i,j,k,l] <= zas[k,l] then 1 else 0 end;*

Tworzę parametr czywzas jako funkcję iloczynu kartezjańskiego dowolnych punktów z obszaru (P) i punktów, gdzie mogą znaleźć się nadajniki (R), zwracającą 1, gdy dany punkt znajduje się w zasięgu danego nadajniki i 0 w przeciwnym przypadku.

W czasie laboratorium nie miałam stworzonego tego parametru - jak się okazało bez niego nie da się potem poprawnie zdefiniować ograniczenia...

*(linia 17) subto ogr: forall <i,j> in P do sum <k,l> in R do czywzas[i,j,k,l]*x[k,l] >= 1;*

Ograniczenie mówiące, że w zasięgu każdego punktu istnieje przynajmniej jeden nadajniki.

W czasie laboratorium miałam źle sformułowane to ograniczenie, gdyż nie miałam stworzonego potrzebnego do tego parametru czywzas.

Po uruchomieniu, program zwraca następujący wynik:

```
C:\Users\Marta\Desktop\MM2\scip-3.0.0.win.x86_64.vc10.opt.spx.mt.exe

time | node | left | LP iter | LP it/n | mem | mdpt | ifrac | lvars | lcons | lcols | lrows
|cuts | confs | strbr | dualbound | primalbound | gap
t 0.0s | 1 | 0 | 0 | 0 | - | 249k | 0 | - | 9 | 5 | 0 | 0
| 0 | 0 | 0 | -- | 15.000000e+001 | Inf
b 0.0s | 1 | 0 | 0 | 0 | - | 254k | 0 | - | 9 | 5 | 9 | 5
| 0 | 0 | 0 | -- | 1.800000e+001 | Inf
* 0.0s | 1 | 0 | 4 | 4 | - | 255k | 0 | - | 9 | 5 | 9 | 5
| 0 | 0 | 0 | 1.300000e+001 | 1.300000e+001 | 0.00%
| 0.0s | 1 | 0 | 4 | 4 | - | 255k | 0 | - | 9 | 5 | 9 | 5
| 0 | 0 | 0 | 1.300000e+001 | 1.300000e+001 | 0.00%

SCIP Status      : problem is solved [optimal solution found]
Solving Time (sec) : 0.01
Solving Nodes    : 1
Primal Bound     : +1.3000000000000000e+001 <4 solutions>
Dual Bound      : +1.3000000000000000e+001
Gap              : 0.00 %

SCIP> dis sol

objective value:      13
x#0#5                 1   <obj:7>
x#8#5                 1   <obj:6>

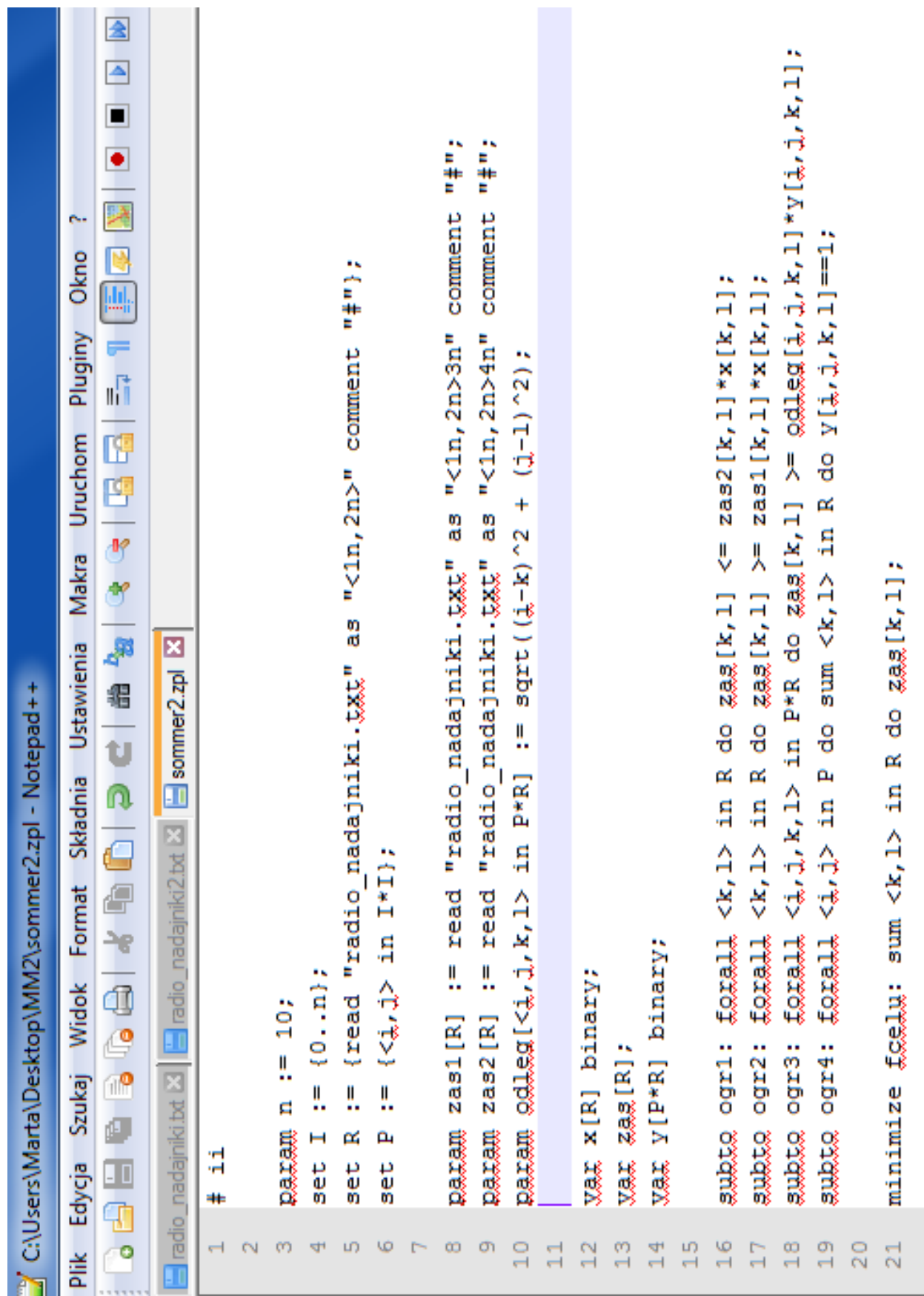
SCIP>
```

Wynik ten interpretujemy następująco:

Nadajniki mają zostać włączone w punktach (0,5) oraz (8,5).

Łączny (minimalny) zasięg nadajników będzie wtedy wynosił 13.

Poniżej załączam kod do drugiej części (ii) programu:



```
1 # ii
2
3 param n := 10;
4 set I := {0..n};
5 set R := {read "radio_nadajniki.txt" as "<ln,2n>" comment "#"};
6 set P := {<i,j> in I*I};
7
8 param zas1[R] := read "radio_nadajniki.txt" as "<ln,2n>3n" comment "#";
9 param zas2[R] := read "radio_nadajniki.txt" as "<ln,2n>4n" comment "#";
10 param odleg[<i,j,k,l> in P*R] := sqrt((i-k)^2 + (j-l)^2);
11
12 var x[R] binary;
13 var zas[R];
14 var y[P*R] binary;
15
16 subto ogr1: forall <k,l> in R do zas[k,l] <= zas2[k,l]*x[k,l];
17 subto ogr2: forall <k,l> in R do zas[k,l] >= zas1[k,l]*x[k,l];
18 subto ogr3: forall <i,j,k,l> in P*R do zas[k,l] >= odleg[i,j,k,l]*y[i,j,k,l];
19 subto ogr4: forall <i,j> in P do sum <k,l> in R do y[i,j,k,l]==1;
20
21 minimize fcelu: sum <k,l> in R do zas[k,l];
```

Opiszę kolejne linijki kodu i niestety nie wytłumaczę, czym różni się zamieszczone powyżej rozwiązanie od kodu, który udało mi się napisać w trakcie laboratorium, gdyż w trakcie laboratorium tej części nawet nie zdążyłam ruszyć...

Tak więc po kolei:

(linia 3) param $n := 10$;

(linia 4) set $I := \{0..n\}$;

(linia 5) set $R := \{\text{read "radio_nadajniki.txt" as "<1n,2n>" comment "#"}\}$;

*(linia 6) set $P := \{\langle i,j \rangle \text{ in } I * I\}$;*

Wszystko analogicznie do części (i) zadania.

(linia 8) param $\text{zas1}[R] := \text{read "radio_nadajniki.txt" as "<1n,2n>3n" comment "#"}\}$;

(linia 9) param $\text{zas2}[R] := \text{read "radio_nadajniki.txt" as "<1n,2n>4n" comment "#"}\}$;

Tworzę parameter zas1 i zas2 jako funkcję punktów, gdzie mogą znaleźć się nadajniki (R), zwracającą zasięg maksymalny (zas1) lub minimalny (zas2) tych punktów.

*(linia 10) param $\text{odleg}[\langle i,j,k,l \rangle \text{ in } P * R] := \text{sqrt}((i-k)^2 + (j-l)^2)$;*

Tworzę parametr odległość - analogicznie jak w części (i).

(linia 12) var $x[R]$ binary;

Definiuję zmienną binarną x zależną od punktów, w których mogą się znaleźć nadajniki (R). Zwraca ona 1, gdy w danym punkcie nadajnik ma zostać włączony i 0 w przeciwnym przypadku.

(linia 13) var $\text{zas}[R]$;

Definiuję zmienną binarną zas zależną od punktów, w których mogą się znaleźć nadajniki (R). Zwraca ona zasięg danego nadajnika.

*(linia 14) var $y[P * R]$ binary;*

Definiuję zmienną binarną y zależną od punktów, w których mogą się znaleźć nadajniki (R) i od dowolnych punktów z obszaru (P). Zwraca ona 1, gdy dany punkt będzie odbierał sygnał z danego nadajnika (gdy nadajnik został wybrany dla danego punktu), a zero w przeciwnym przypadku.

*(linia 16) subto ogr1: forall $\langle k,l \rangle \text{ in } R$ do $\text{zas}[k,l] \leq \text{zas2}[k,l] * x[k,l]$;*

Ograniczenie mówiące, że dla każdego nadajnika jego zasięg ma być mniejszy od maksymalnego zasięgu, jeżeli nadajnik jest włączony, i od 0, gdy nadajnik jest wyłączony.

*(linia 17) subto ogr2: forall $\langle k, l \rangle$ in R do $zas[k, l] \geq zas1[k, l] * x[k, l]$;*

Ograniczenie mówiące, że dla każdego nadajnika jego zasięg ma być większy od minimalnego zasięgu, jeżeli nadajnik jest włączony, i od 0, gdy nadajnik jest wyłączony.

*(linia 18) subto ogr3: forall $\langle i, j, k, l \rangle$ in $P * R$ do $zas[k, l] \geq odleg[i, j, k, l] * y[i, j, k, l]$;*

Ograniczenie mówiące, że dla każdego punktu z obszaru i dla każdego nadajnika zasięg nadajnika ma być większy od odległości tego punktu do nadajnika, jeżeli ten nadajnik został dla tego punktu wybrany, i od 0, gdy nie został.

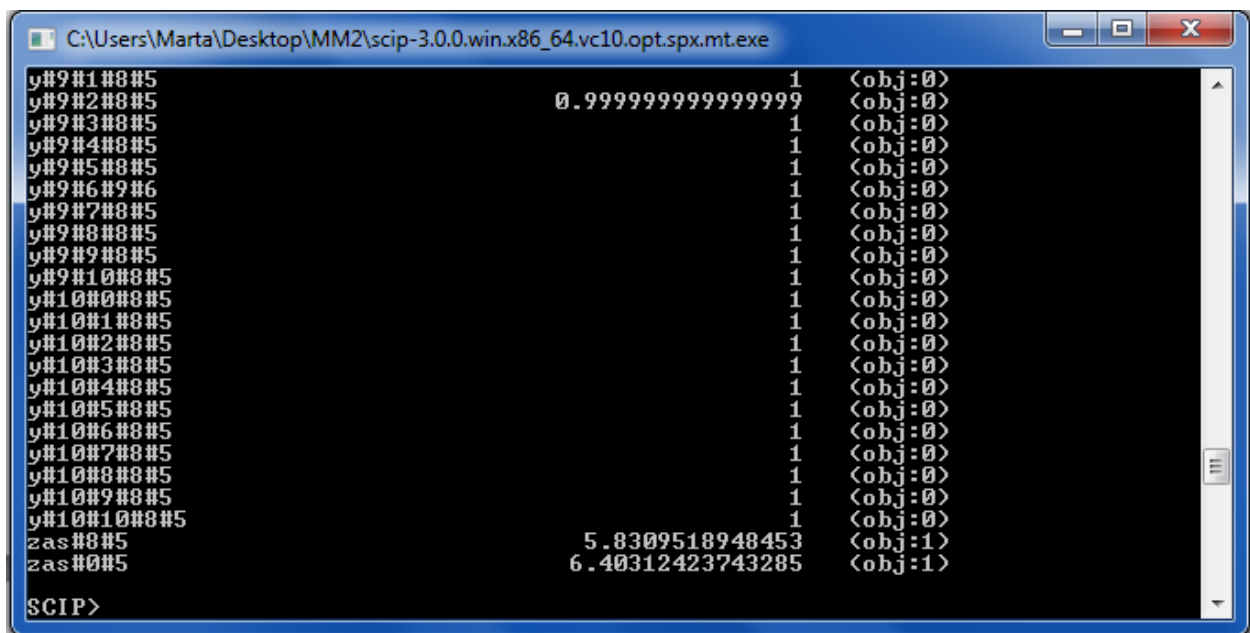
(linia 19) subto ogr4: forall $\langle i, j \rangle$ in P do sum $\langle k, l \rangle$ in R do $y[i, j, k, l] = 1$;

Ograniczenie mówiące, że każdy punkt z obszaru ma przyporządkowany tylko jeden (dokładnie jeden) nadajnik, z którego otrzymuje sygnał.

(linia 21) minimize fcelu: sum $\langle k, l \rangle$ in R do $zas[k, l]$;

Dążę do zminimalizowania funkcji celu danej jako łączny zasięg nadajników.

Po uruchomieniu, program zwraca następujący wynik:



```
C:\Users\Marta\Desktop\MM2\scip-3.0.0.win.x86_64.vc10.opt.spx.mt.exe
y#9#1#8#5      1 (obj:0)
y#9#2#8#5      0.999999999999999 (obj:0)
y#9#3#8#5      1 (obj:0)
y#9#4#8#5      1 (obj:0)
y#9#5#8#5      1 (obj:0)
y#9#6#9#6      1 (obj:0)
y#9#7#8#5      1 (obj:0)
y#9#8#8#5      1 (obj:0)
y#9#9#8#5      1 (obj:0)
y#9#10#8#5     1 (obj:0)
y#10#0#8#5     1 (obj:0)
y#10#1#8#5     1 (obj:0)
y#10#2#8#5     1 (obj:0)
y#10#3#8#5     1 (obj:0)
y#10#4#8#5     1 (obj:0)
y#10#5#8#5     1 (obj:0)
y#10#6#8#5     1 (obj:0)
y#10#7#8#5     1 (obj:0)
y#10#8#8#5     1 (obj:0)
y#10#9#8#5     1 (obj:0)
y#10#10#8#5    1 (obj:0)
zas#8#5        5.8309518948453 (obj:1)
zas#0#5        6.40312423743285 (obj:1)
SCIP>
```

Wynik ten interpretujemy następująco:

Nadajniki mają zostać włączone w punktach (8,5) o zasięgu około 5.83 oraz (0,5) o zasięgu około 6.40.

Łączny (minimalny) zasięg nadajników będzie wtedy wynosił 12.23.