Tomasz Bocheński [261416] tbochens@gmail.com
Marcin Jarząbek [261411] marcin.jarzabek.pl@gmail.com
Prowadzący: mgr inż. Paweł Zawistowski
Algorytmy heurysytczne, EiTI 2016

ROZMIESZCZENIE KAMER BEZPIECZEŃSTWA

OPTYMALNE ROZMIESZCZENIE KAMER MONITORINGU W ZADANYM POMIESZCZENIU

OPIS ZADANIA

Celem projektu jest znalezienie optymalnego rozmieszenia kamer monitoringu w ustalonym pomieszczeniu (rzut z góry) w taki sposób, aby minimalną liczbą kamer móc obserwować dowolny punkt zadanego pomieszczenia (uwzględniając maksymalną dopuszalną odległość od kamery).

REALIZACJA CELU PROJEKTU

Udało się zaimplementować w języku programowania R algorytm symulowanego wyżarzania pozwalający dla zadanych parametrów otrzymać rozkład kamer pokrywających swoim zasięgiem pokój. Algorytm można parametryzować przy pomocy następujących współczynników:

- **radius** - promień określający zasięg kamery,
- **A, B** - współczynniki wagowe funkcji celu,
- **initTemp** - temperatura początkowa, na podstawie której obliczana jest temperatura dla danej iteracji (temperatura jest odwrotnie proporcjonalna do numeru iteracji),
- **rectNumber** - ilość prostokątów, z których składa się losowo generowane pomieszczenie,
- **minLength**, **maxLength** - zakres długości boków pojedyńczych prostokątów, składających się na pomieszczenie,
- **maxIter** - liczba iteracji algorytmu

Wynikiem działania algorytmu jest rekord zawierający m.in. liczbę wykorzystanych kamer oraz ich współrzędne. Rysowany jest również wykres wizualizujący pomieszczenie wraz z rozmieszczonymi kamerami i ich zasięgami (w postaci okręgów).

Przeprowadzono następnie serię eksperymentów (opisaną w dalszej części dokumentacji) potwierdzających założone działanie algorytmu.

PRZEPROWADZONE EKSPERYMENY

Wybrana w projekcie metoda symulowanego wyżarzania jest niedeterministyczna - z tego względu aby otrzymać znaczącą statystycznie pulę wyników algorytm za każdym razem był uruchamiany dla zadanych parametrów 5 razy.

Podczas wykonywania eksperymentów został zbadany - dla różnych, losowych punktów startowych i stopniowo rosnących pomieszczeń (liczba prostokątów składowych w zakresie od 2 do 5) - wpływ wartości A i B, będących wagami funkcji celu, odpowiednio:

A - kara za użycie kolejnej kamery
B - kara za obszar niepokryty zasięgiem kamer

Zadaniem algorytmu była minimalizacja zmiennej quality.

Zaimplementowany algorytm nie wymaga strojenia względem parametrów A i B, ponieważ z założenia to użytkownik wybiera ich wartości, w zależności od zamierzonego celu - może zadecydować czy chce bardzo dokładnie pokryć zasięgiem kamer całe pomieszczenie, kosztem nakładania się pól widzenia kamer, lub też wybrać rozwiązanie "oszczędniejsze", minimalizując liczbę wykorzystanych kamer i godząc się na częściowe niepokrycie powierzchni pokoju.

W związku z tym przeprowadzona została seria eksperymentów, pozwalająca zaobserwować zachowanie algorytmu w zależności od rozwiązania, na które kładzione jest nacisk (minimalizacja niepokrytej powierzchni czy minimalizacja liczby wykorzystanych kamer)

PREZENTACJA WYNIKÓW

W punkcie tym prezentujemy wyniki poszczególnych uruchomień algorytmu. Parametry uruchomieniowe podawane są na samym początku. Następnie prezentujemy tabelkę z wynikami otrzymanymi dla tych samych parametrów, jednak z różnymi punktami początkowymi. Różnice w wynikach są zatem powodowane czynnikami losowymi oraz różnymi punktami początkowymi. Następnie wyznaczamy rozwiązanie najlepsze (o najniższej wartości zmiennej quality), oraz dla tego rozwiązania przedstawione są kolejno:

- wykres liczby kamer i zmiennej quality względem numeru iteracji,
- wizualizacja początkowego, losowego punktu startowego
- wizualizacja punktu zwróconego przez algorytm jako punkt najlepszy.

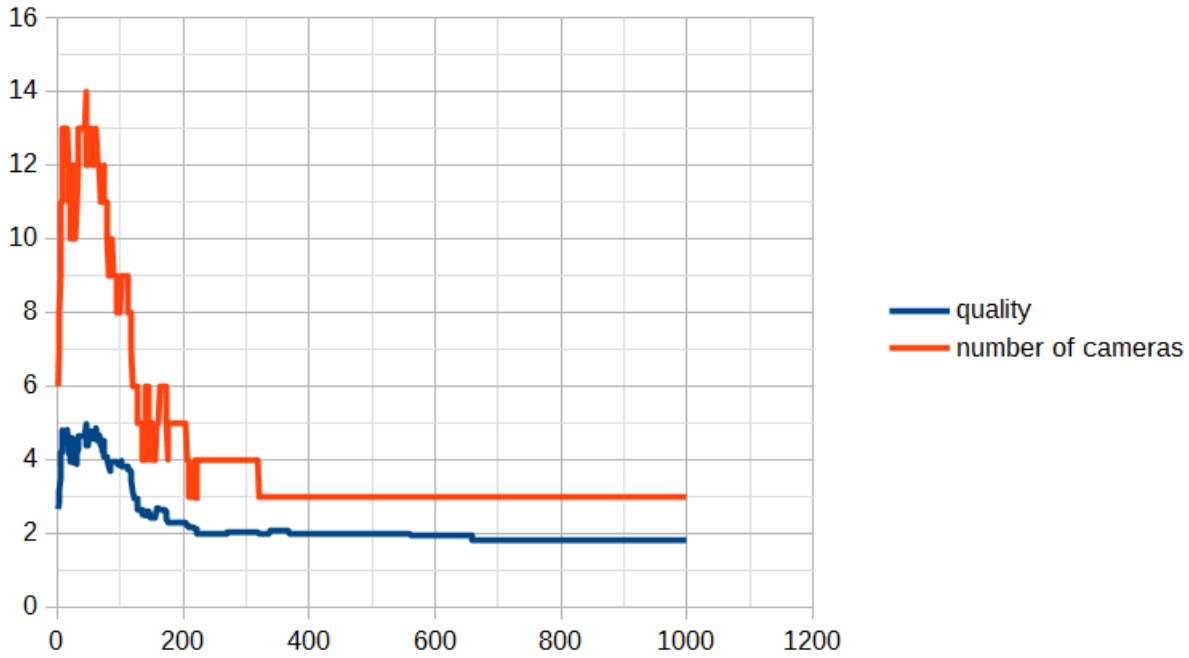
Wyniki eksperymentów (logi algorytmu, wykresy, wizualizacje) znajdują się w katalogu ***tests_results***.

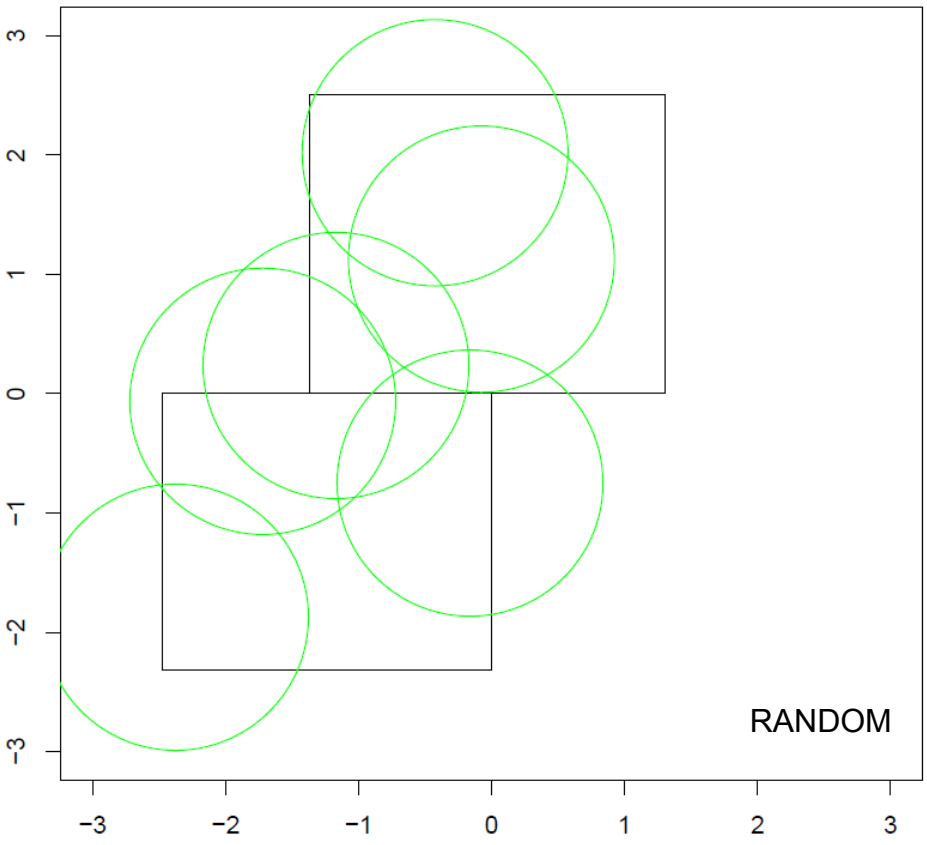
Opis parametrów uruchomieniowych algorytmu znajduje się w pliku ***tests_results/tests_plan.txt***

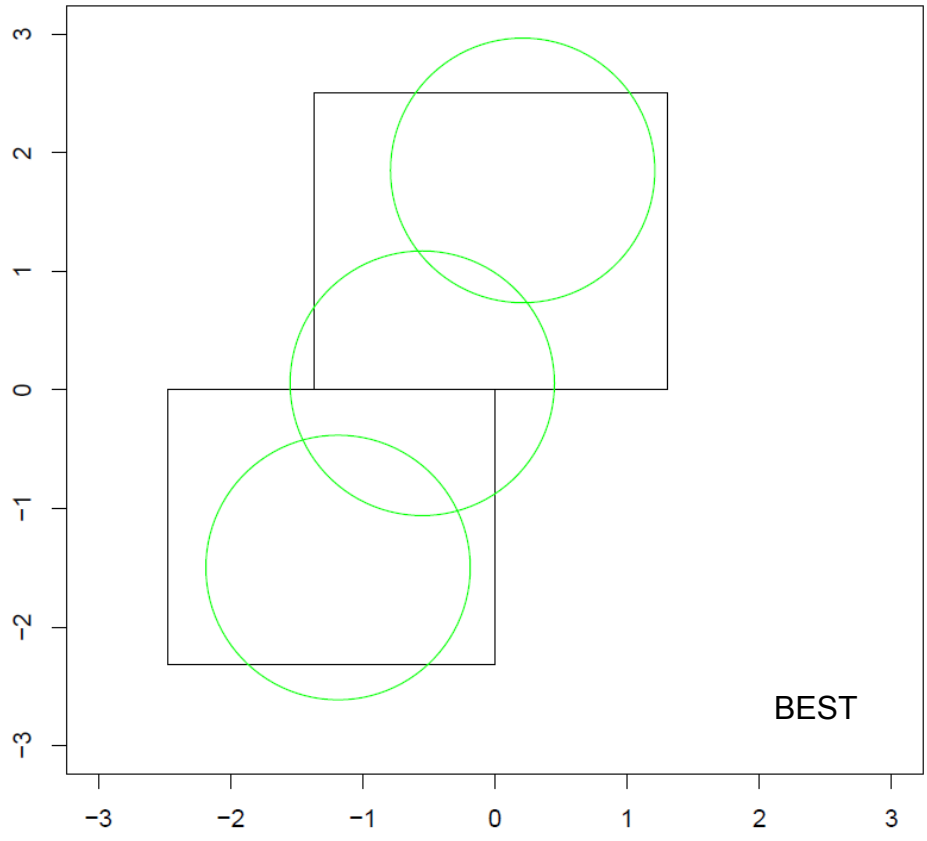
Uruchomienie **run1**: **run(1, 0.35, 0.65, 10, 2, 5, 2, 1000)**

Nr próby	quality	numberOfCameras
1	1.945603	4
2	1.947678	4
3	1.816740	3
4	2.001404	4
5	1.942079	2

Próba z najniższą wartością quality: próba nr 3



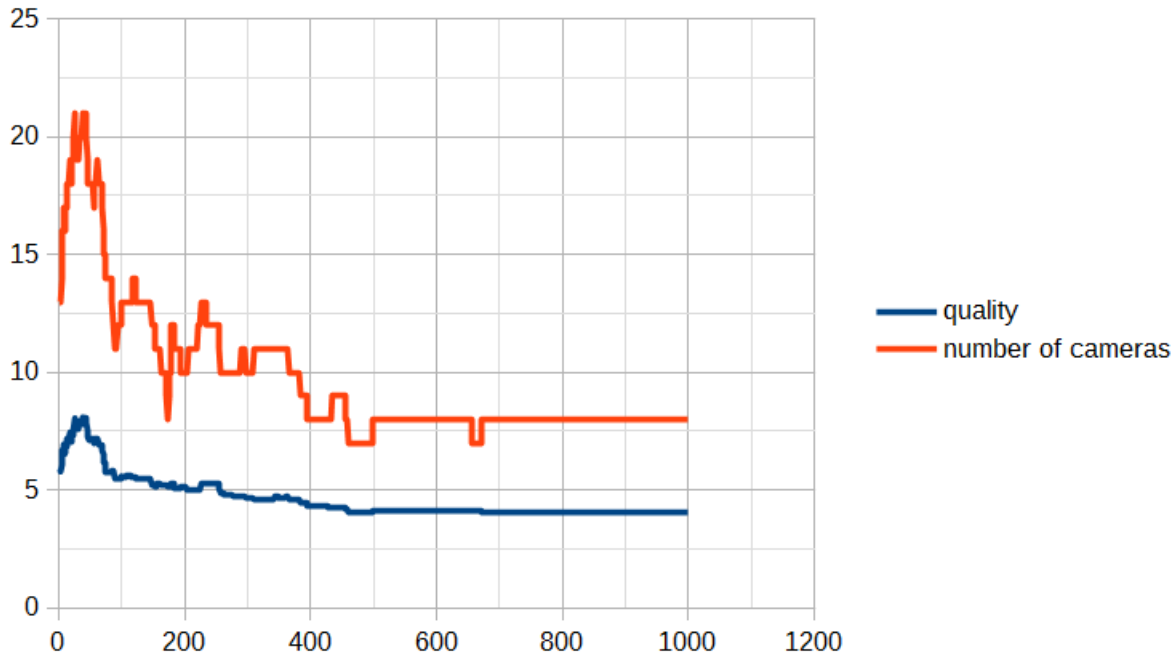

RANDOM

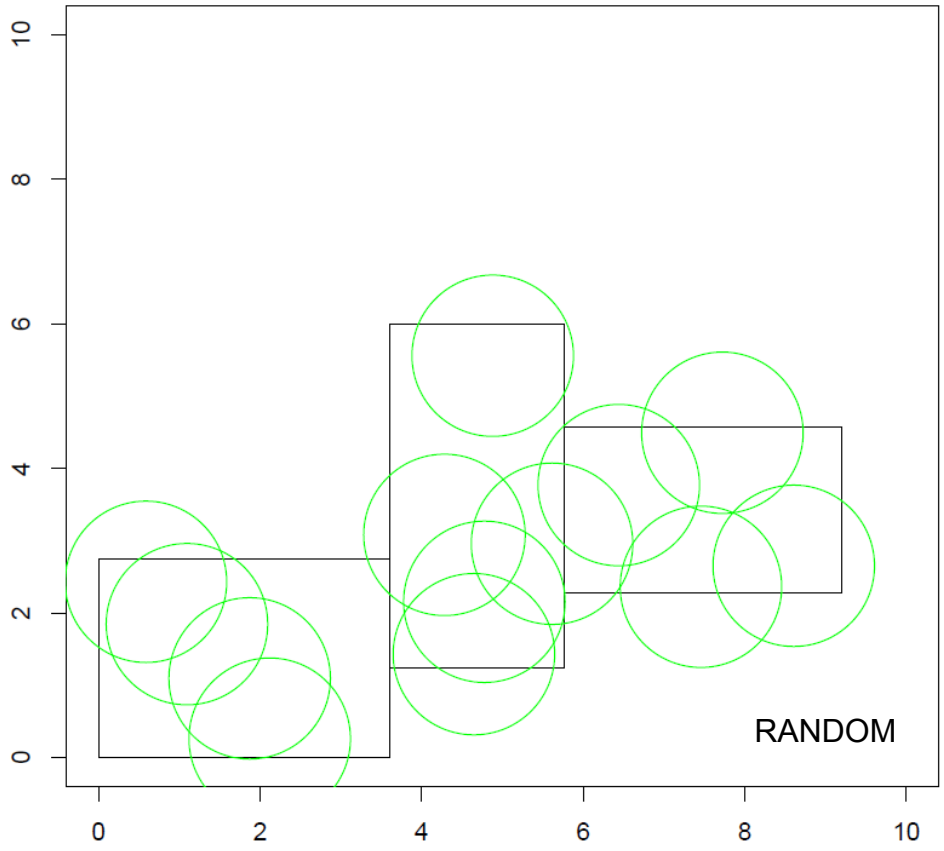

BEST

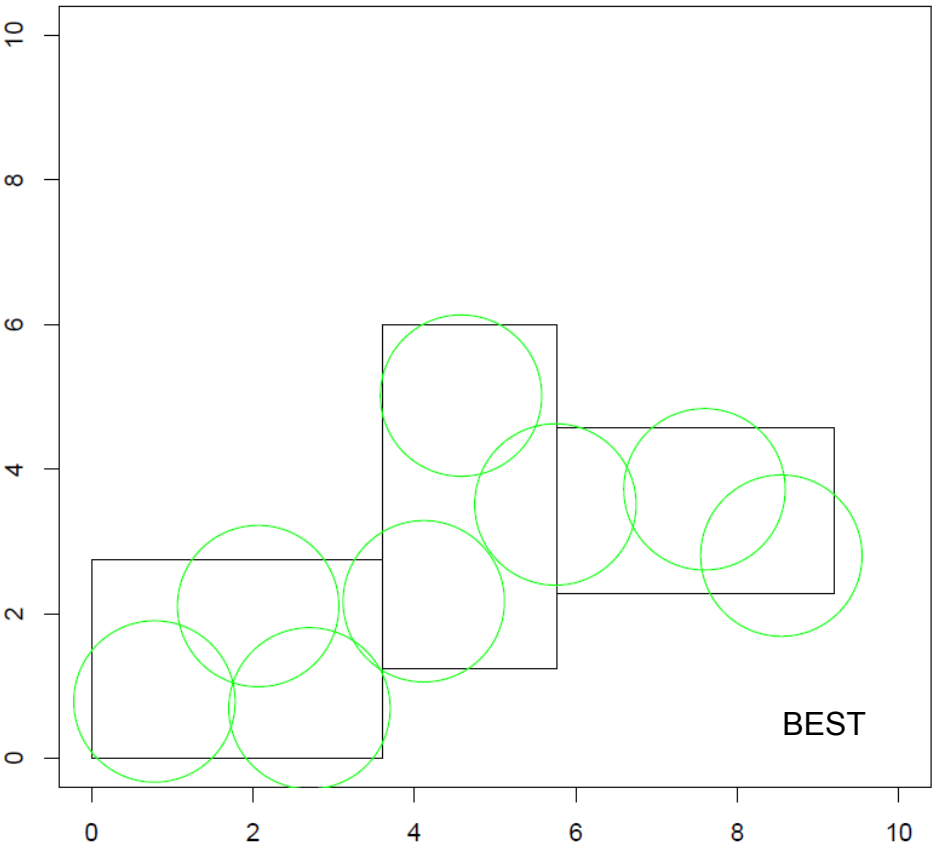
Uruchomienie **run2**: **run(1, 0.35, 0.65, 10, 3, 5, 2, 1000)**

Nr próby	quality	numberOfCameras
1	4.050945	8
2	4.370629	7
3	4.165238	9
4	4.114983	8
5	4.315992	9

Próba z najniższą wartością quality: próba nr 1



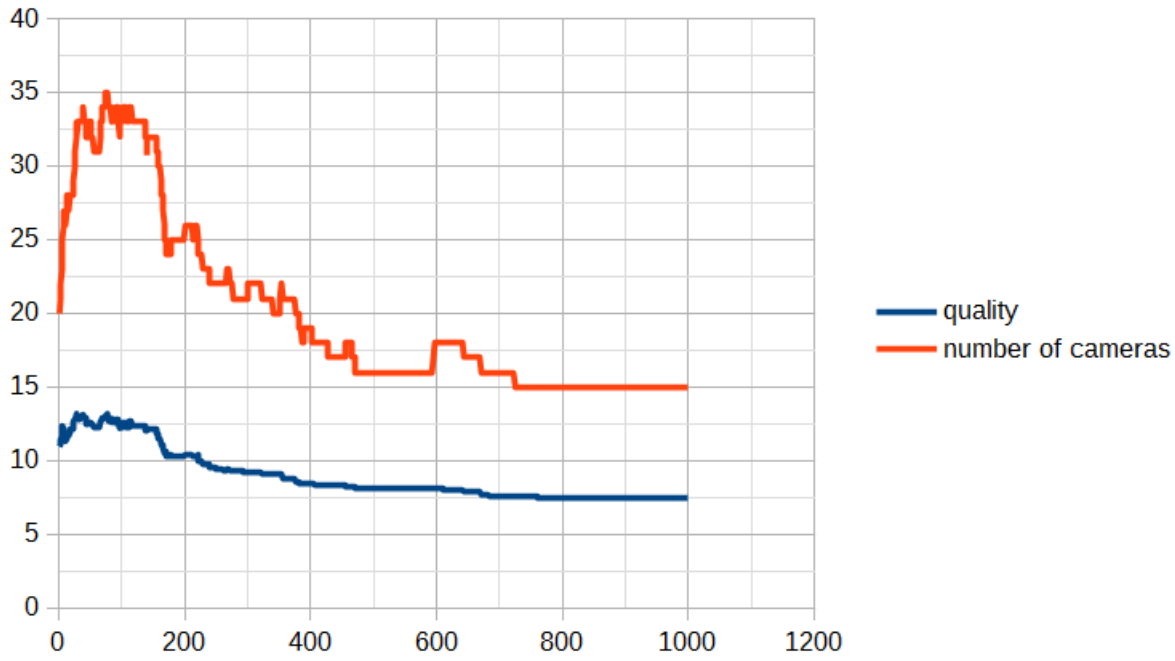

RANDOM
0
2
4
6
8
10

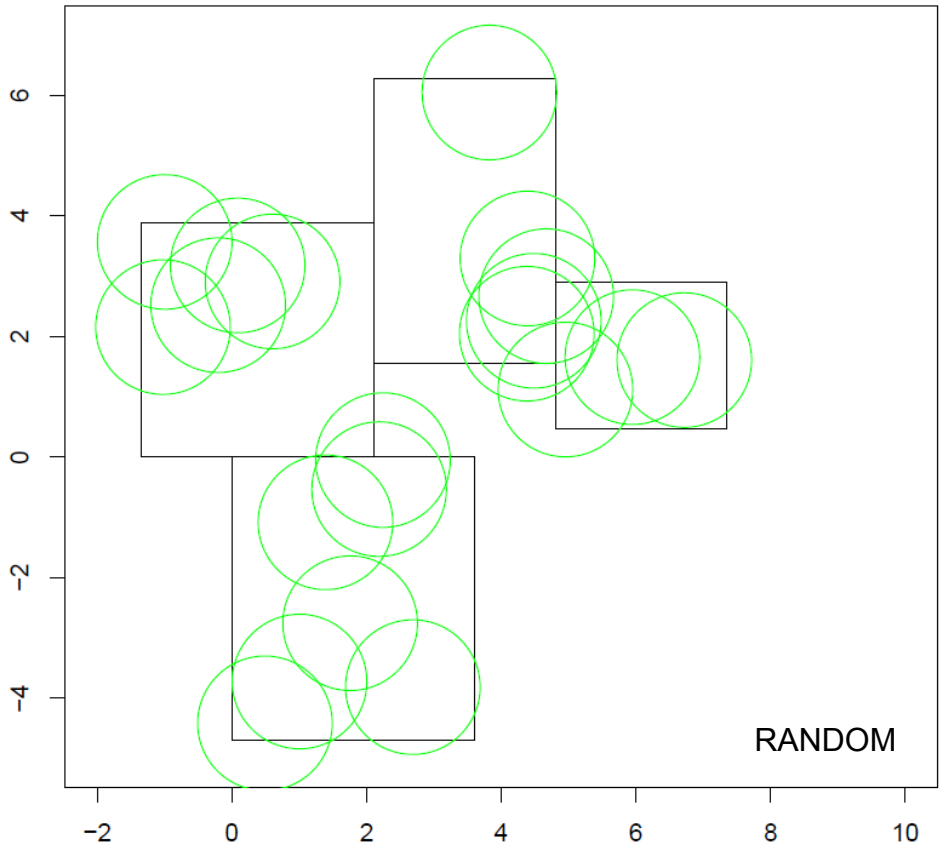

BEST
0
2
4
6
8
10

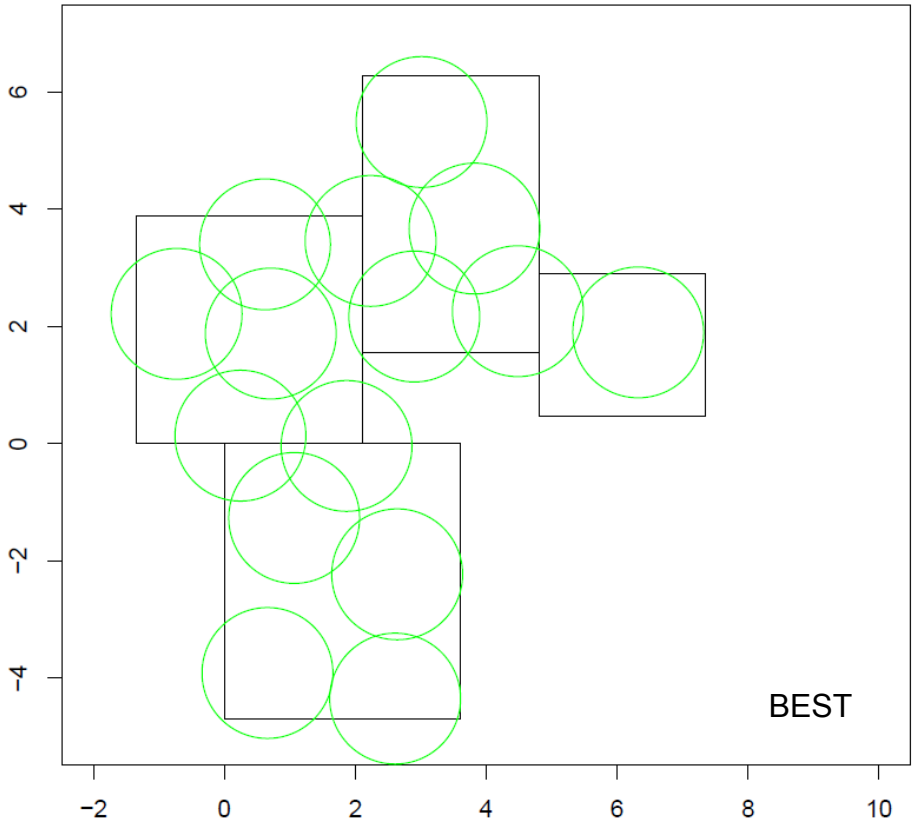
Uruchomienie **run3**: **run(1, 0.35, 0.65, 10, 4, 5, 2, 1000)**

Nr próby	quality	numberOfCameras
1	7.440328	15
2	7.927825	14
3	7.994994	16
4	7.645212	15
5	7.929186	17

Próba z najniższą wartością quality: próba nr 1



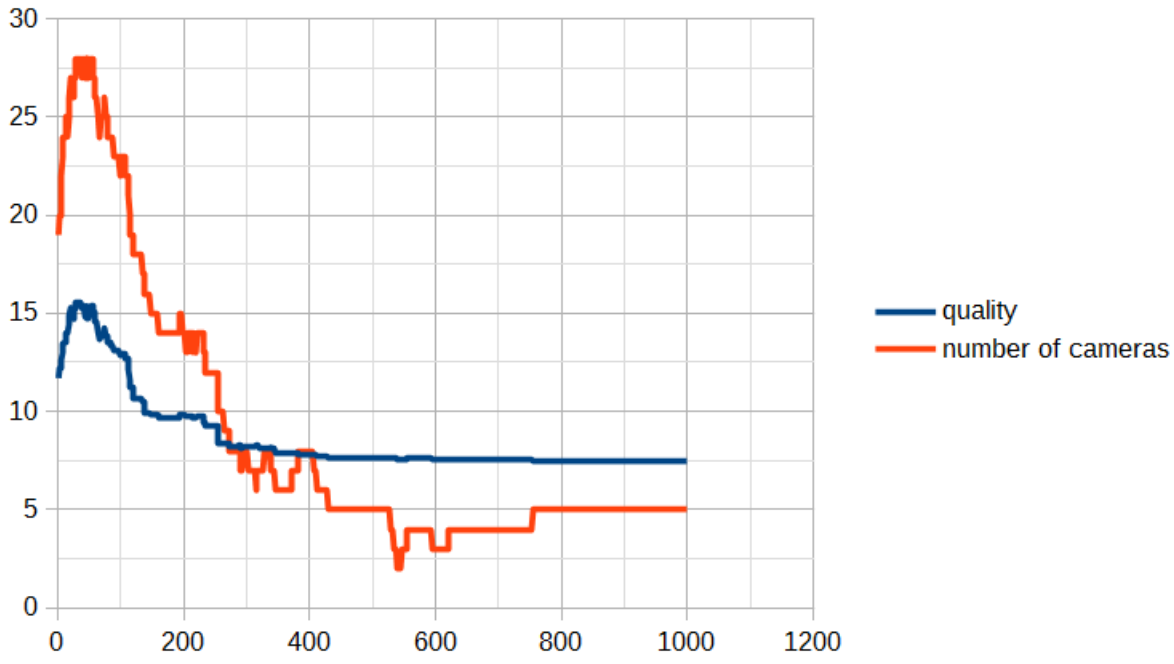

RANDOM
6
4
2
0
−2
−4
−2
0
2
4
6
8
10

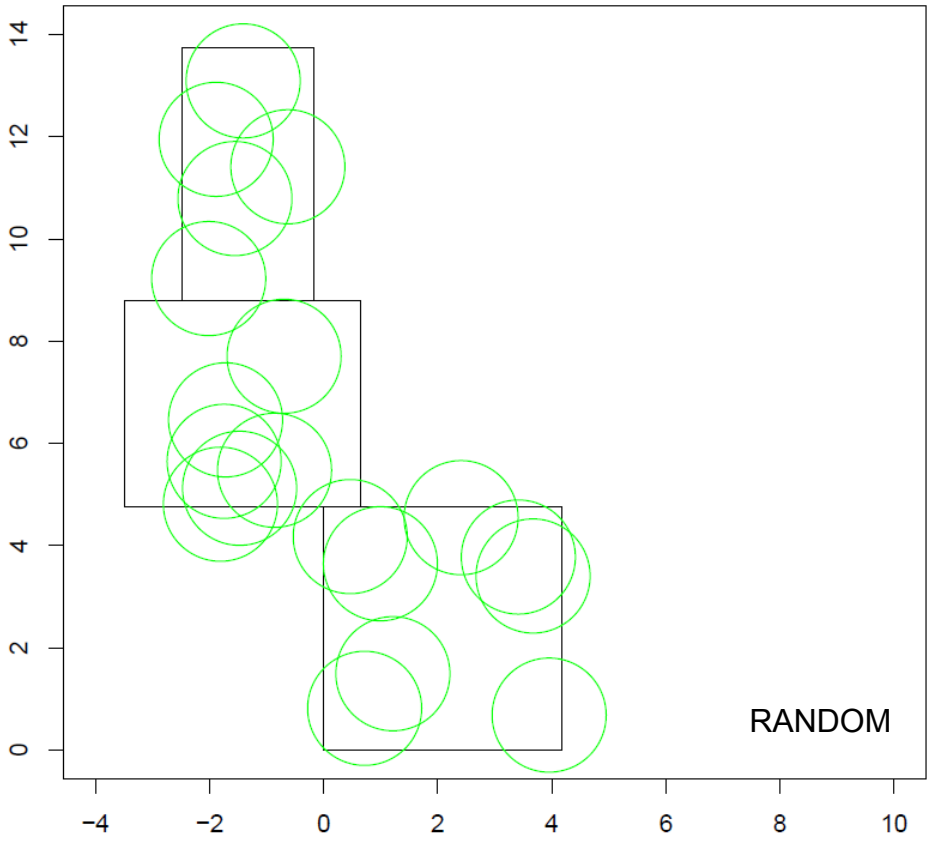

BEST
6
4
2
0
−2
−4
−2
0
2
4
6
8
10

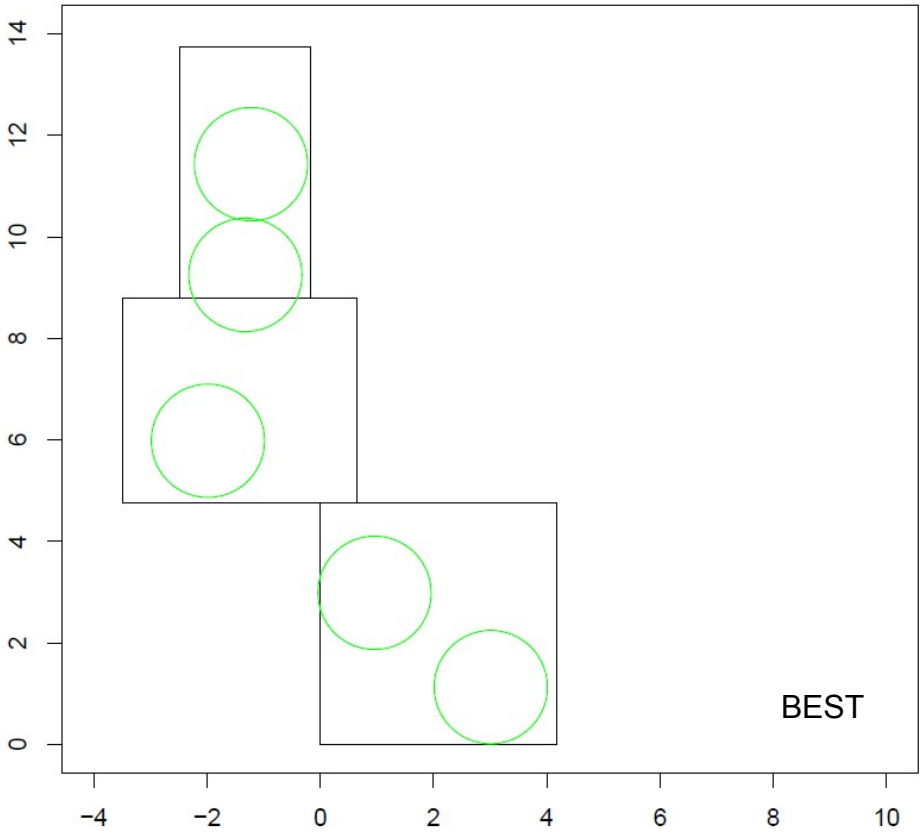
Uruchomienie **run4: run(1, 0.5, 0.5, 10, 2, 5, 2, 1000)**

Nr próby	quality	numberOfCameras
1	7.712998	7
2	7.523664	7
3	7.604331	6
4	7.508466	5
5	7.669207	5

Próba z najniższą wartością quality: próba nr 4



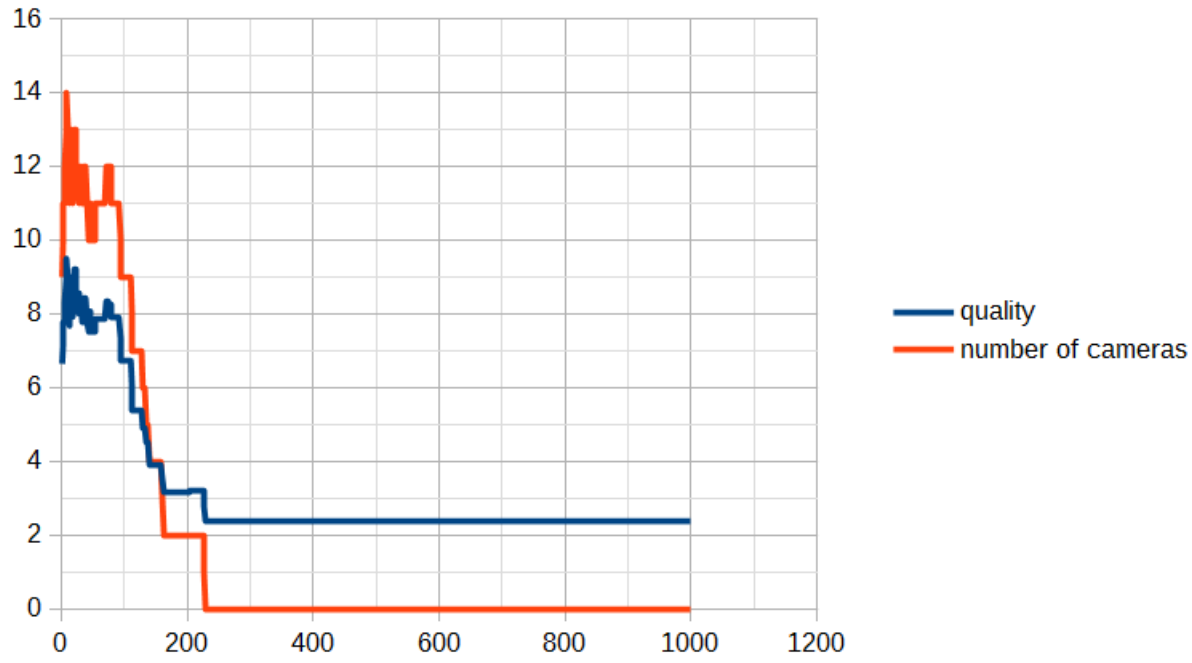

RANDOM

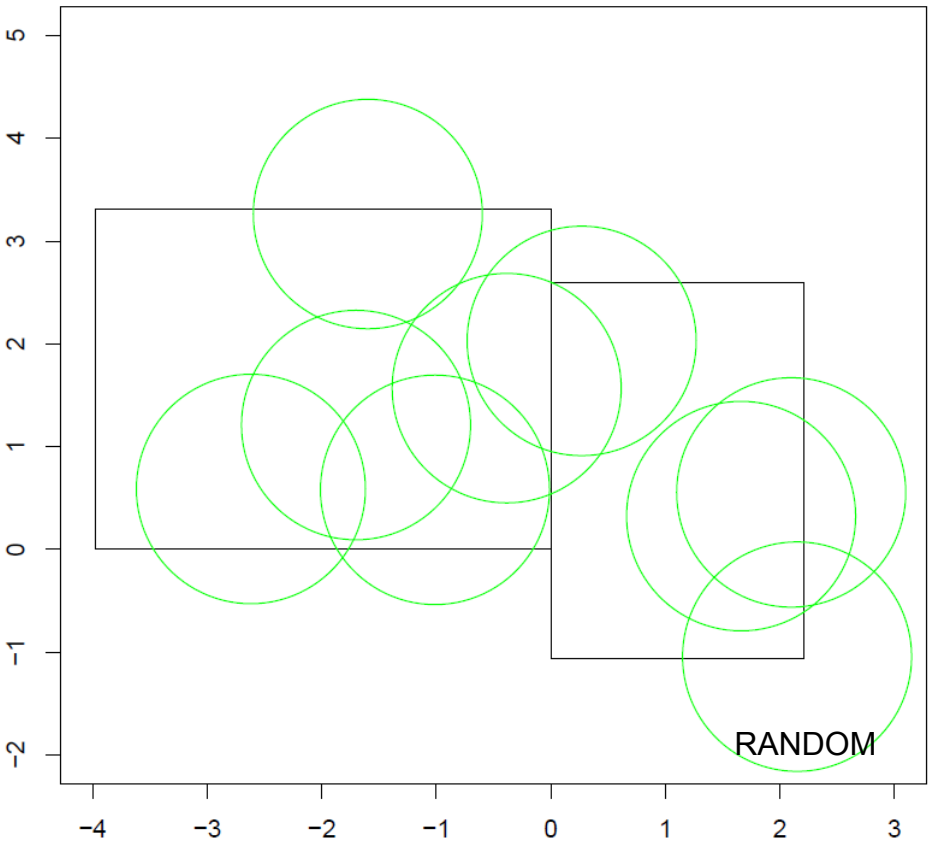

BEST

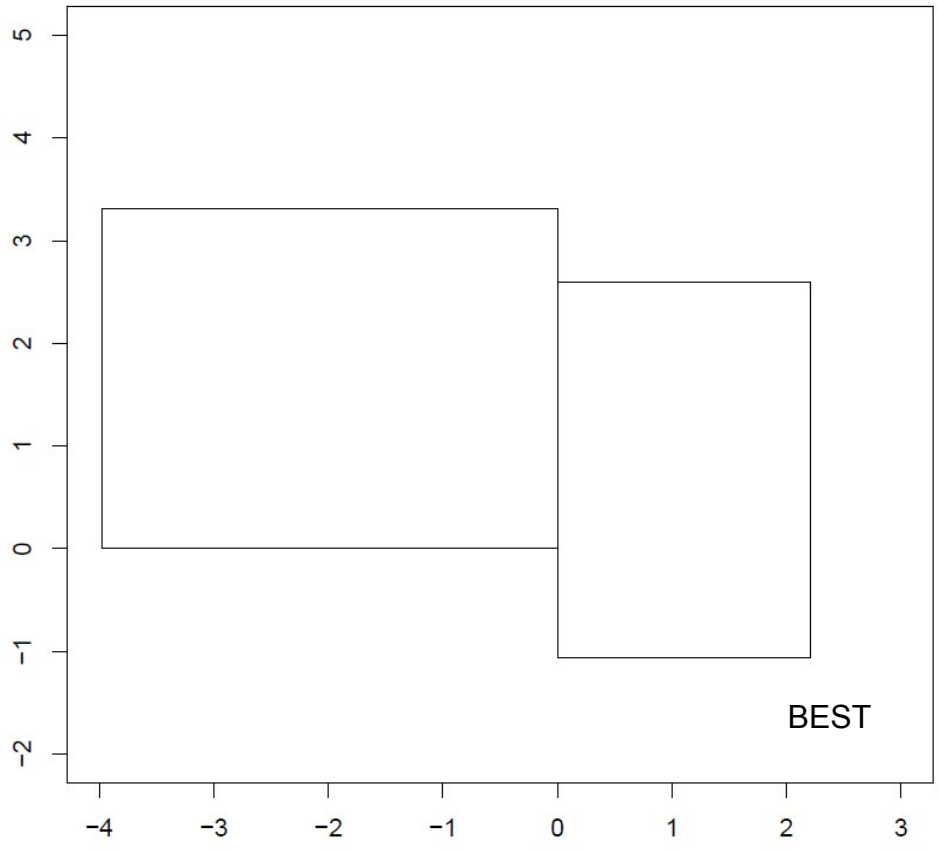
Uruchomienie **run5: run(1, 0.65, 0.35, 10, 2, 5, 2, 1000)**

Nr próby	quality	numberOfCameras
1	2.370405	0
2	2.370405	0
3	2.370405	0
4	2.370405	0
5	2.370405	0

Próba z najniższą wartością quality: próba nr 1 (przykładowa)



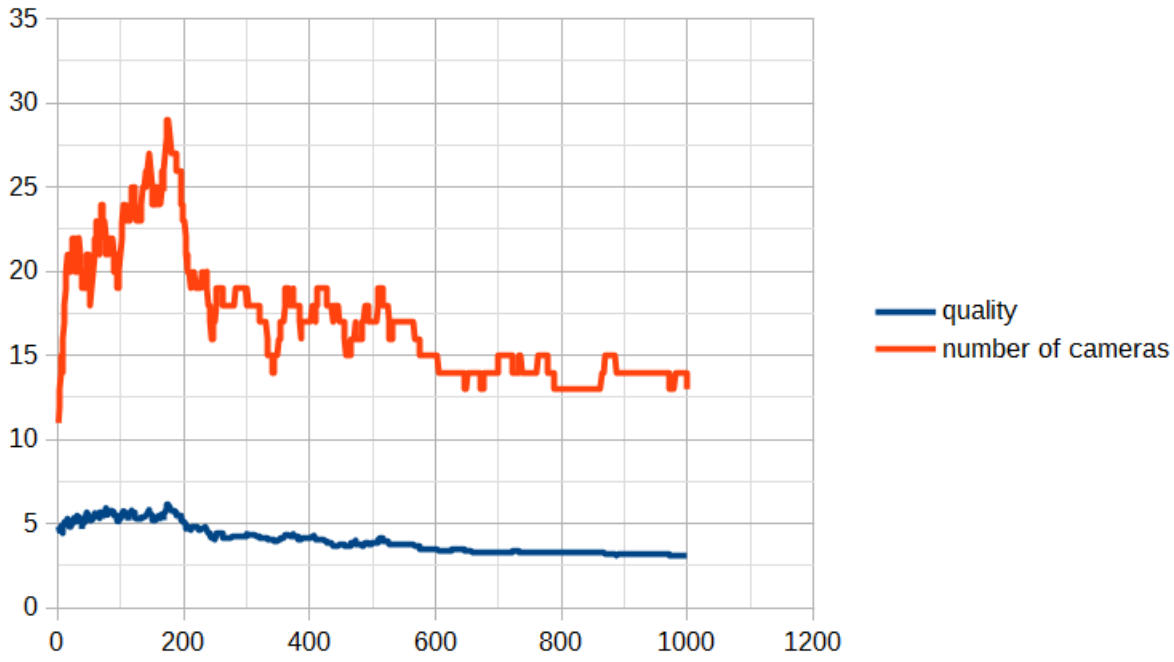

RANDOM

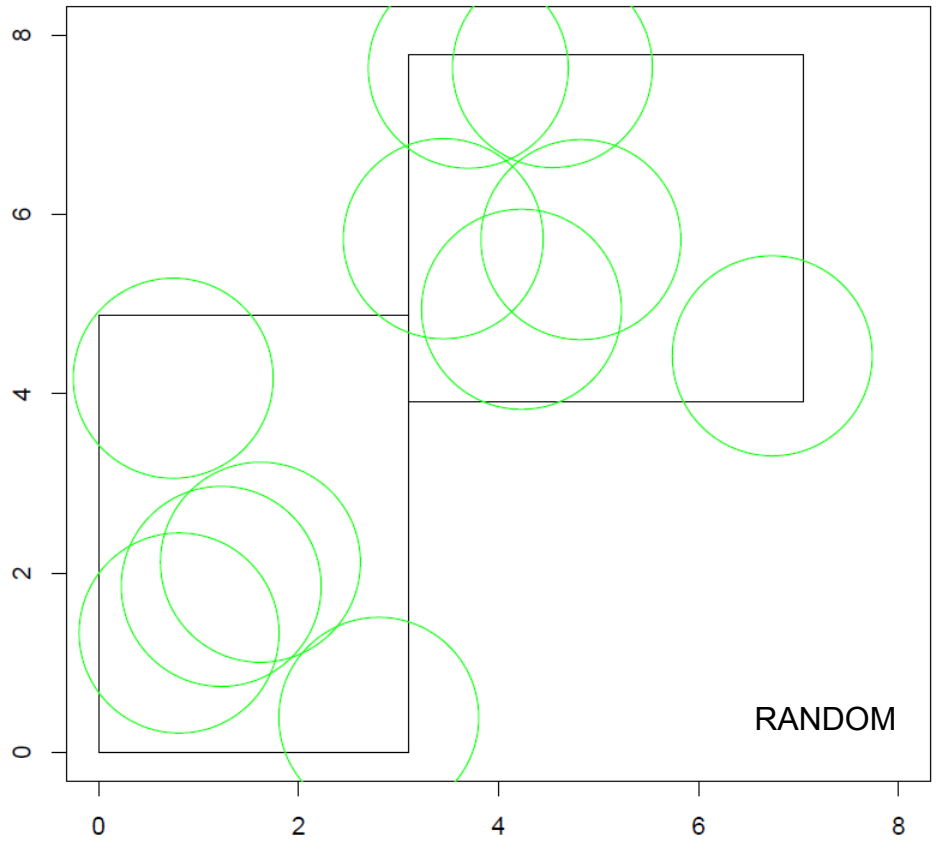

BEST

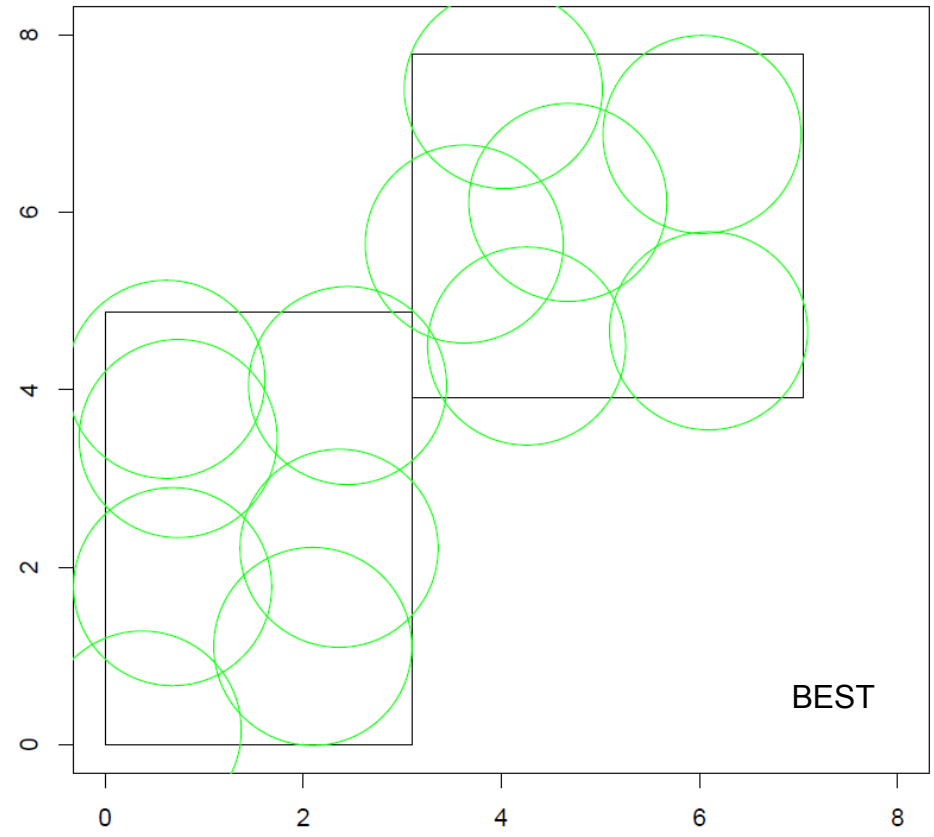
Uruchomienie **run6: run(1, 0.20, 0.80, 20, 2, 5, 2, 1000)**

Nr próby	quality	numberOfCameras
1	3.208539	13
2	3.335844	13
3	3.473595	15
4	3.101587	13
5	3.264559	13

Próba z najniższą wartością quality: próba nr 4



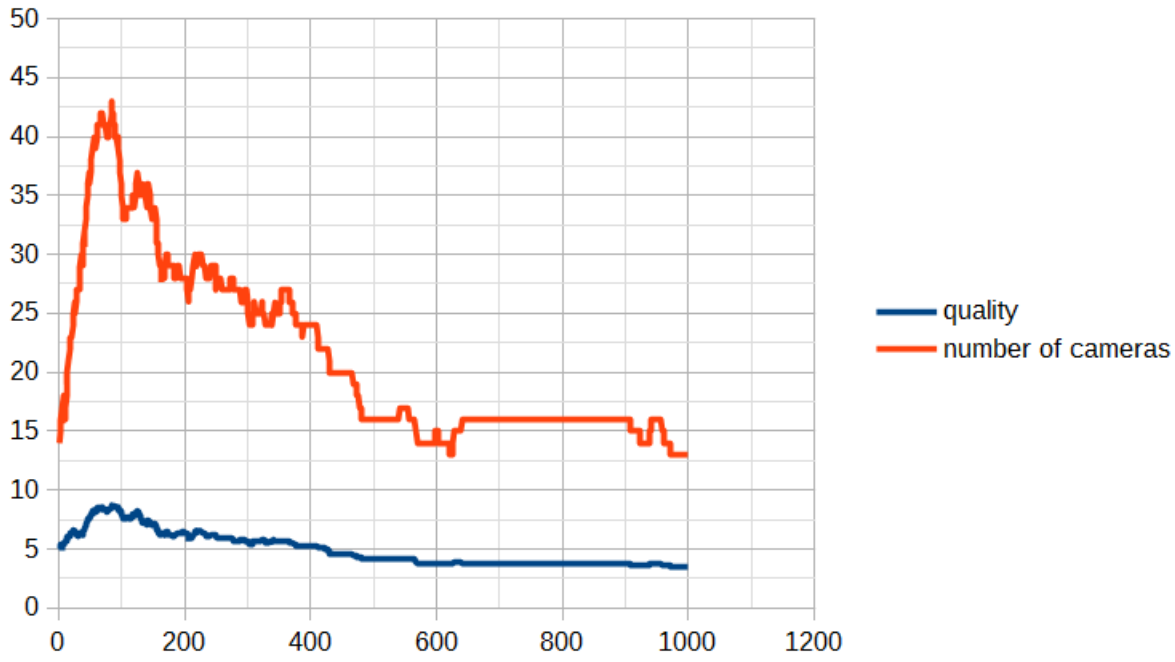

RANDOM
0
2
4
6
8

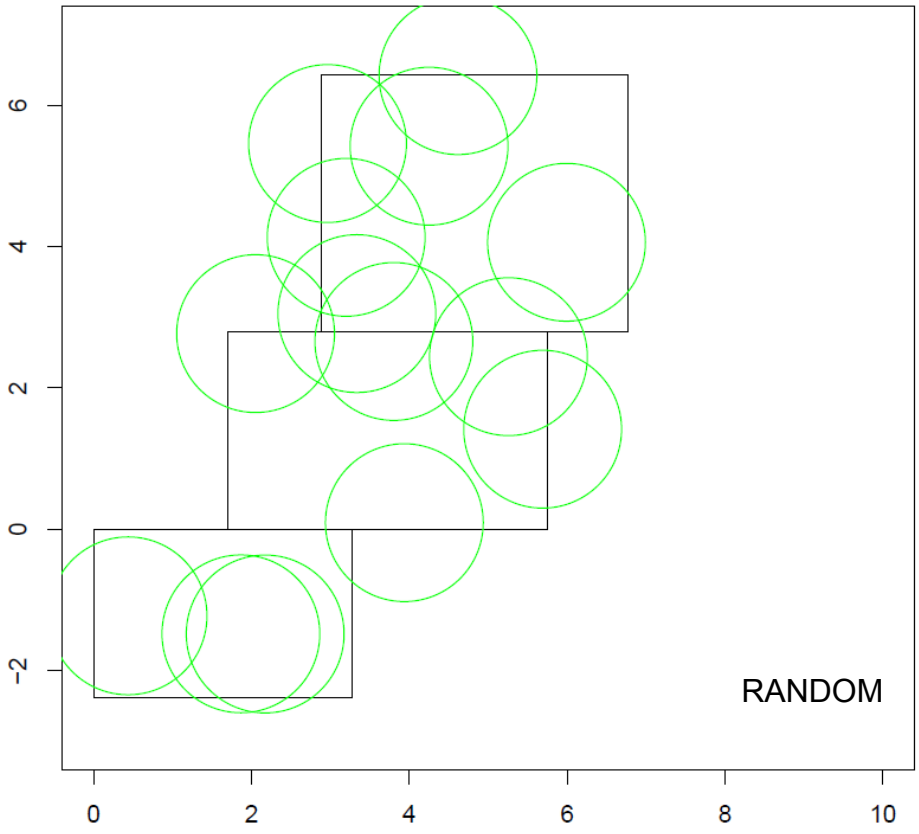

BEST
0
2
4
6
8

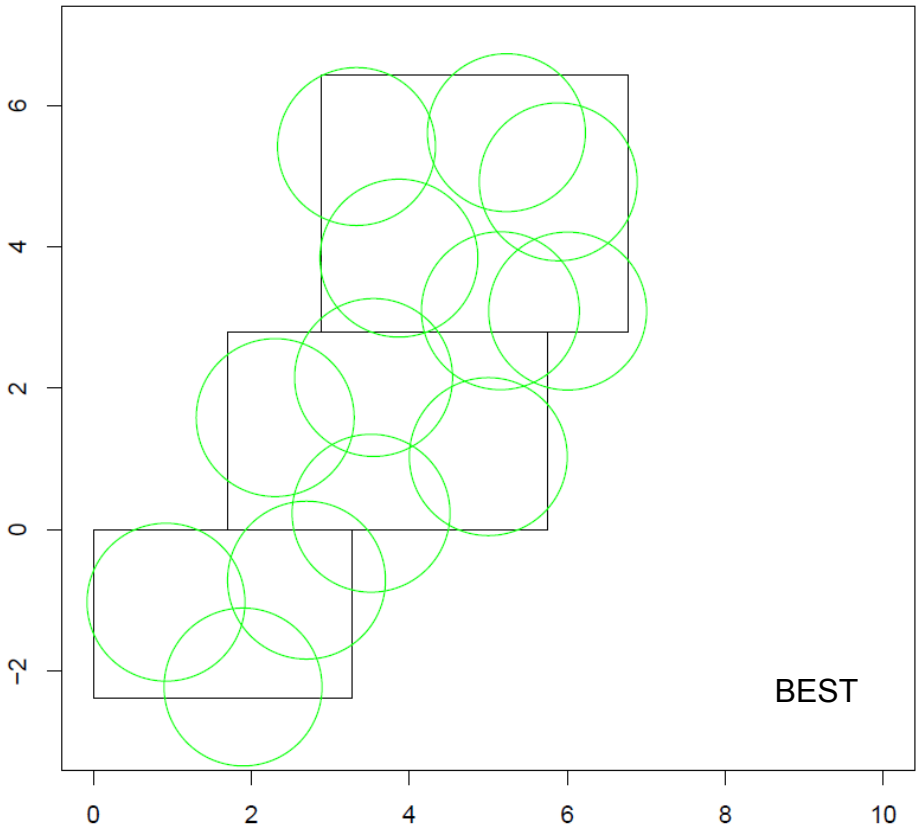
Uruchomienie **run7: run(1, 0.20, 0.80, 20, 3, 5, 2, 1000)**

Nr próby	quality	numberOfCameras
1	3.730526	15
2	3.452703	13
3	3.845067	15
4	3.652704	14
5	3.517802	14

Próba z najniższą wartością quality: próba nr 2



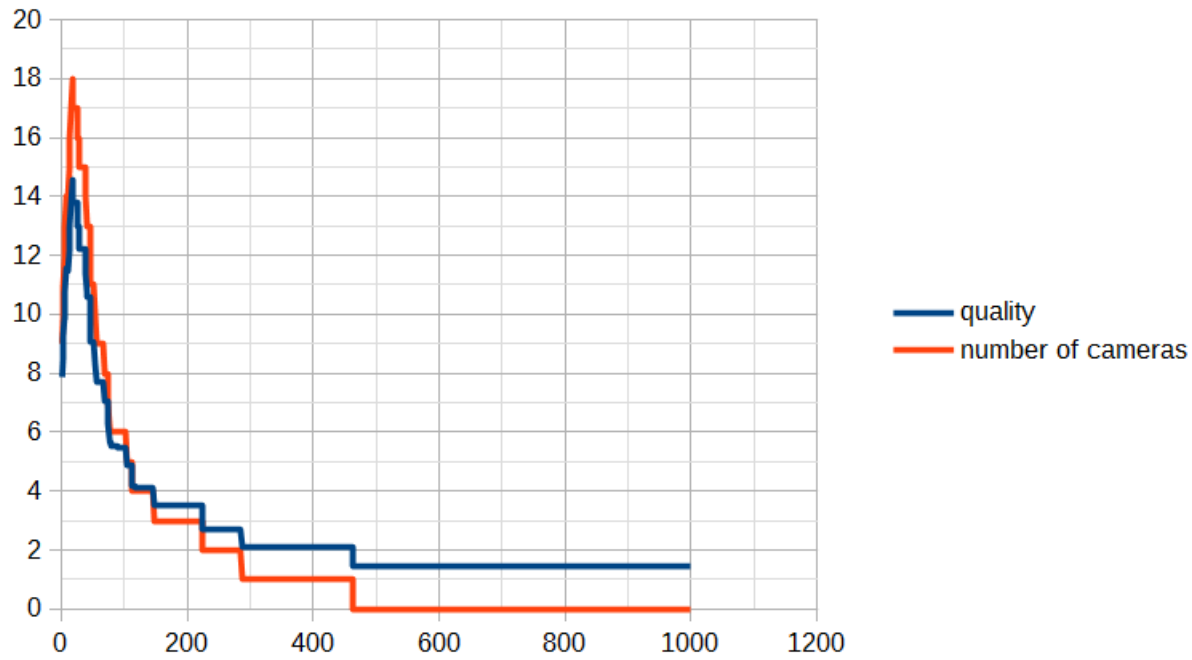

RANDOM
6
4
2
0
−2
0
2
4
6
8
10

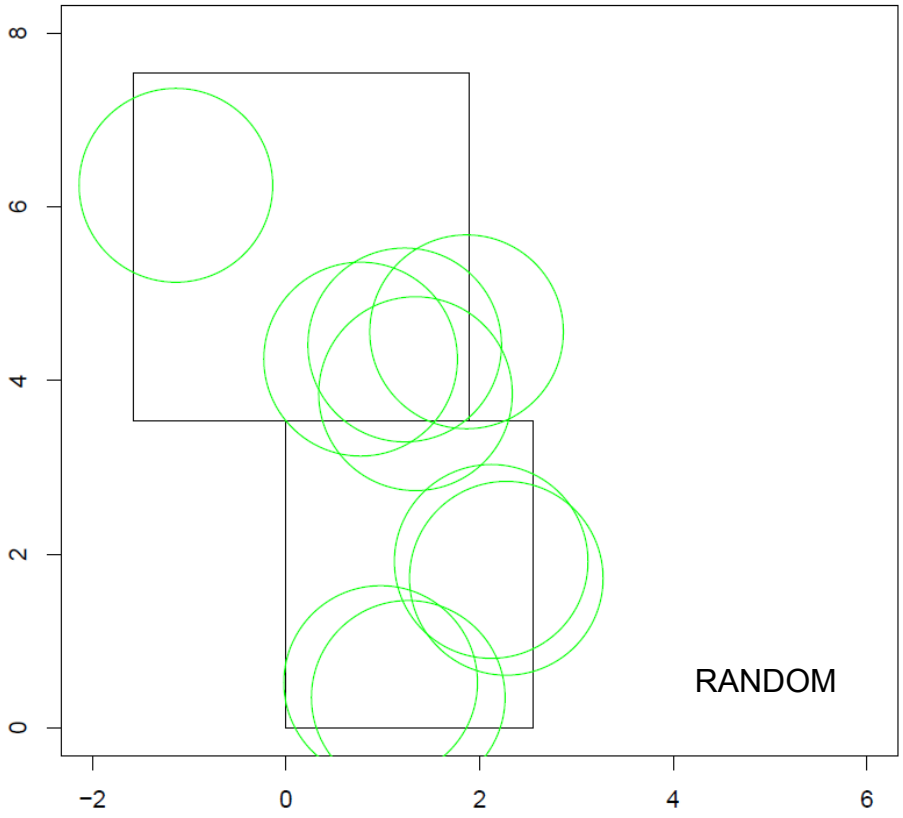

BEST
6
4
2
0
−2
0
2
4
6
8
10

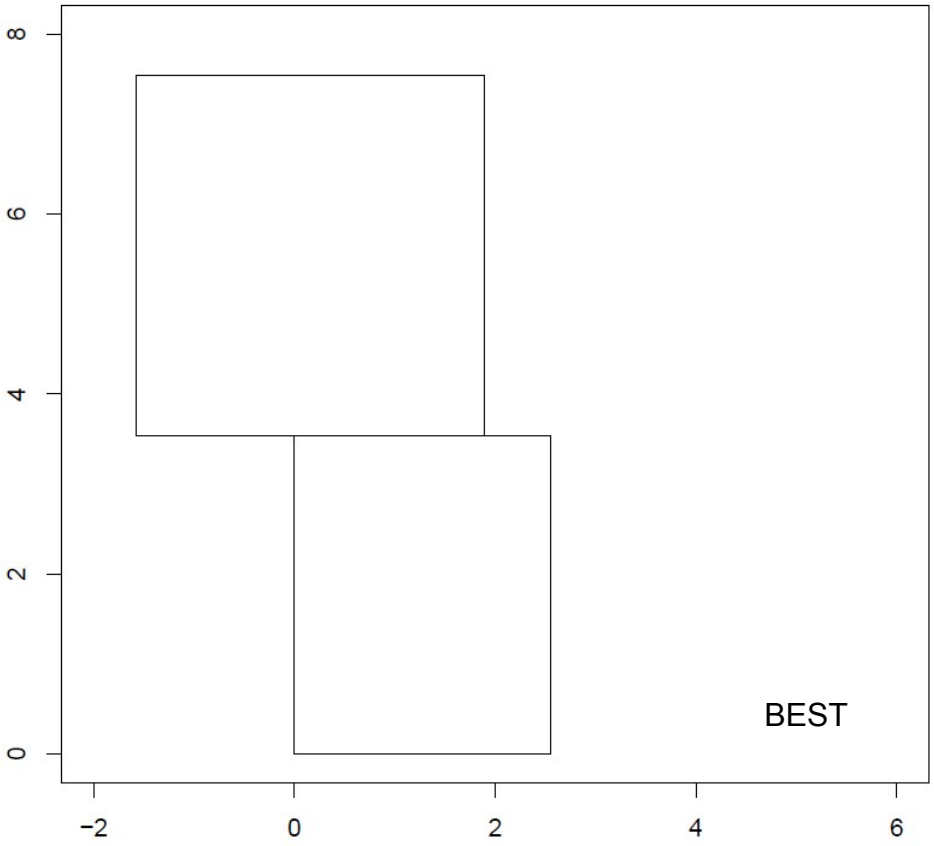
Uruchomienie **run8: run(1, 0.80, 0.20, 10, 2, 5, 2, 1000)**

Nr próby	quality	numberOfCameras
1	1.457910	0
2	1.457910	0
3	1.457910	0
4	1.457910	0
5	1.457910	0

Próba z najniższą wartością quality: próba nr 1 (przykładowa)




8
6
4
2
0
−2
0
2
4
6
RANDOM


8
6
4
2
0
−2
0
2
4
6
BEST

ANALIZA WYNIKÓW

Analiza temperatury:

Analizując wykresy quality dla każdego z badanych przypadków można zauważyć, że funkcja obrazująca wielkość quality nie jest monotonicznie nierosnąca. Na wykresach można zauważyć momenty, gdy wartość quality rośnie (czyli akceptowane jest gorsze rozwiązanie). Zdarza się to znacznie częściej na początku działania algorytmu niż później.
Dzieje się tak z powodu temperatury. Pozwala ona na akceptowanie gorszych rozwiązań z pewnym prawdopodobieństwem. W przypadku gdy temperatura jest wysoka, rozwiązania gorsze są akceptowane częściej. W przypadku gdy niższa, są one akceptowane rzadziej. W naszym algorytmie temperatura jest odwrotnie proporcjonalna do numeru iteracji. Dlatego tez na początku można zaobserwować częstsze akceptowanie rozwiązań gorszych, natomiast pod koniec działania algorytmu zdarza się to stosunkowo rzadko.
Ponadto porównując wykresy otrzymane dla przypadków z różnymi temperaturami można zauważyć, że czym większa jest temperatura początkowa, tym dłużej utrzymują się skoki w przebiegu zmiennej quality, tym więcej czasu potrzeba by przebiegi wartości zmiennej quality oraz liczby kamer się ustabilizowały.

Analiza współczynników A i B:

Możliwe są trzy przypadki: **A > B, A = B, A < B**

W przypadku gdy **A < B** bardziej zależy nam na stawianiu kamer niż na ich usuwaniu. Analizując wykresy dla tego typu przypadków zwykle możemy wyróżnić dwie fazy. W pierwszej z nich następuje gwałtowny wzrost liczby kamer. Ponieważ większe kary nakładane są za brak pokrycia obszaru, zwykle opłacalne jest dodanie kamery pokrywającej dany obszar. W momencie gdy odpowiednia część pomieszczenia zostanie pokryta i dodawanie kamer nie jest już tak opłacalne, zaczyna się druga faza. Polega ona na stopniowym przesuwaniu lub odejmowaniu kamer. Ponieważ niektóre kamery się nakładają lub też są nieefektywnie rozmieszczone, ich przesunięcie lub tez usunięcie jest opłacalne i zmniejsza wartość zmiennej quality. Fazy te są tym widoczniejsze, im większy jest iloraz B/A.
Wyniki eksperymentów potwierdzają opisane powyżej właściwości.

W przypadku gdy **A = B** jest nam obojętne, czy kamera zostanie stawiona czy też nie. W pierwszych fazach działania, gdy temperatura jest jeszcze duża, kamery są dodawana i usuwane z dużą losowością, wartość zmiennej quality nie jest czynnikiem decydującym w przypadku wyboru nowego najlepszego rozwiązania. Jednak po pewnej liczbie iteracji, gdy wpływ czynnika losowego zmaleje, można zauważyć że funkcja określająca liczbę kamer jest funkcją nierosnącą. Jest to spowodowane faktem, że żadne rozwiązanie polegające na dodaniu nowej kamery nie zwiększa wartości zmiennej quality. Ponadto eliminowane są przypadki, w których kamera nie znajduje się w całości w pomieszczeniu lub tez nakłada się na inną kamerę. Zatem w tym przypadku również można wyróżnić dwie fazy. Pierwsza z nich polega na przyjmowaniu losowych rozwiązań wynikających z dużej wartości temperatury. Druga to usuwanie kamer które nie leżą w całości w pomieszczeniu lub też się nakładają.
Wyniki eksperymentu potwierdzają opisane powyżej właściwości. Finalne rozwiązanie zwrócone przez algorytm składa się z kamer leżących całkowicie w pomieszczeniu, które się nie nakładają. Jeśli chodzi o wykres to można zauważyć w nim pewne odstępstwa od tych opisanych powyżej. Wynikają one z faktu, że do badania pokrytego pola wykorzystujemy metodę Monte Carlo, wykonując 100 próbek na jednostkę powierzchni. Zatem jest to niedeterministyczna metoda wyznaczania pola. W celu posiadania dokładniejszych wyników należało by zwiększyć liczbę próbek do 1000 lub 10000 (jednak są to wartości nieosiągalne dla naszych komputerów).

W przypadku gdy **A > B** większe kary nakładane są za stawianie kamer niż za ich brak. Dla takiej relacji tych parametrów proste jest przewidzenie wyniku końcowego działania algorytmu. Każda akcja polegająca na usunięciu kamery skutkuje zmniejszeniem, czyli polepszeniem zmiennej quality. Zatem ostatecznie powinniśmy otrzymać pusty pokój, bez jakiejkolwiek kamery. Ewentualne wzrosty liczby kamer zawsze powodowane są czynnikiem losowym (niezerową wartością temperatury). Tak jak i poprzednio również tutaj możemy wyróżnić dwie fazy, jednak nie są one tak widoczne jak w innych przypadkach. Pierwsza z nich następuje w momencie gdy temperatura jest jeszcze duża, liczba kamer momentami spada a momentami rośnie. Druga faza polega już tylko na systematycznym zmniejszaniu liczby kamer, aż osiągnie ona wartość zero.
Wyniki eksperymentów potwierdzają opisane powyżej właściwości.

Analiza wielkości pomieszczenia (ilość prostokątów)

Analizując wykresy liczby kamer oraz wartości zmiennej quality można zauważyć, że czym większe jest pomieszczenie (czym więcej prostokątów wchodzi w jego skład), tym więcej czasu, więcej iteracji potrzebuje algorytm aby osiągnąć stabilny stan. Jest to rzecz oczywista. Czym większy jest pokój, tym więcej kamer może w sobie pomieścić, zatem tym większy jest wektor reprezentujący maksymalną liczbę kamer. Analiza większych wektorów zajmuje więcej czasu niż analiza mniejszych.

Wnioski dotyczące dobierania parametrów A i B

Parametry A i B są parametrami, które w dużej mierze decydują o tym jak zachowywać się będzie algorytm i jaki punkt końcowy zostanie przez niego zwrócony. Po przeanalizowaniu różnych przypadków można dojść do wniosku, że jedynie uruchomienia w których A < B mają sens . Jeśli chodzi o przypadki, w których A = B, to kwestia jest sporna. Uruchomienia z A > B nie mają żadnego sensu, ponieważ ich wynik z góry jest znany.
W przypadku gdy A < B, czym większy jest iloraz B/A, tym więcej kamer będzie generowanych w pomieszczeniu. Nie istnieją jedyne poprawne wartości A i B. Są one zależne od tego w jakim stopniu zależy nam na pokryciu pomieszczenia kamerami. Gdy „koszty nie grają roli” iloraz B/A może być duży. W innym przypadku powinien być on odpowiednio mniejszy.