Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Wykład 12.

Algorytmy geometryczne

Opracowano z wykorzystaniem materiałów: http://wazniak.minuw.edu.pl

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Algorytmy geometryczne – podstawowe obiekty i problemy

Podstawowe obiekty geometryczne:

- Punkt p, reprezentowany przez parę współrzędnych (xp, yp) w układzie XOY,
- Odcinek p - q, reprezentowany przez parę punktów p oraz q,
- Wektor o początku w p a końcu w q, oznaczany jako p → q
- Prosta, reprezentowana przez dowolną parę różnych punktów leżących na niej.

Przykłady problemów:

- względne położenie punktów,
- po której stronie wektora p → q leży punkt r
- czy punkty x i y leżą po tej samej stronie prostej p - q
- czy punkt r należy do odcinka p - q
- czy odcinki p – q oraz r -s przecinają się
- czy punkt p leży wewnątrz wielokąta W,
- znajdowanie otoczki wypukłej zbioru punktów
- czy w zbiorze odcinków istnieją dwa odcinki przecinające się,
- wyznaczanie najmniejszej odległości w zbiorze punktów.

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Względne położenie punktów

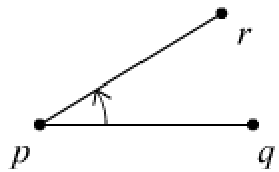
Atomową operacją używaną w algorytmach jest operacja wyznaczania względnego położenia trzech punktów:

p = (xp, yp), q = (xq, yq) , r = (xr, yr)

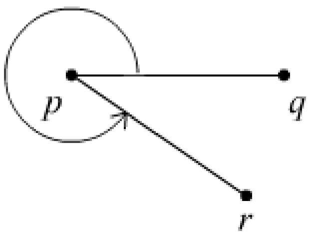
Konstruujemy wyznacznik: $\det(p,q,r) = \det \begin{bmatrix} x_p & y_p & 1 \\ x_q & y_q & 1 \\ x_r & y_r & 1 \end{bmatrix}$

Znak wyznacznika jest równy znakowi sinusa kąta między wektorami p → r oraz p → q.

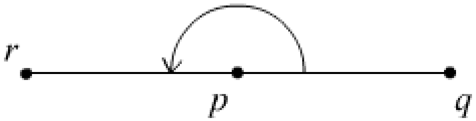
Punkt r leży po lewej stronie wektora p → q, jeżeli det(p,q r) > 0.



Punkt r leży po prawej stronie wektora p → q, jeżeli det(p,q r) < 0



Jeżeli det(p,q r) = 0 to powiemy, że punkty są współliniowe.



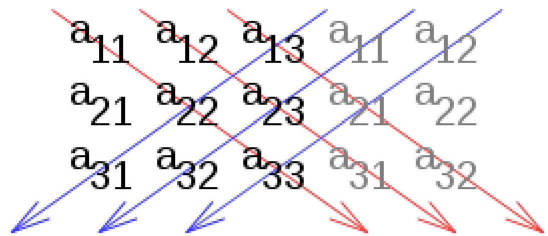
Politechnika Wrocławska, Wydział Informatyki i Zarządzania

Zbigniew Szpunar

Dygresja: obliczanie wyznacznika macierzy 3-stopnia (reguła Sarrusa)

Aby obliczyć wyznacznik:
$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

dopisuje się z prawej strony dwie pierwsze kolumny:



a następnie oblicza się sumę iloczynów wzdłuż czerwonych strzałek i odejmuje od niej sumę iloczynów wzdłuż niebieskich strzałek.

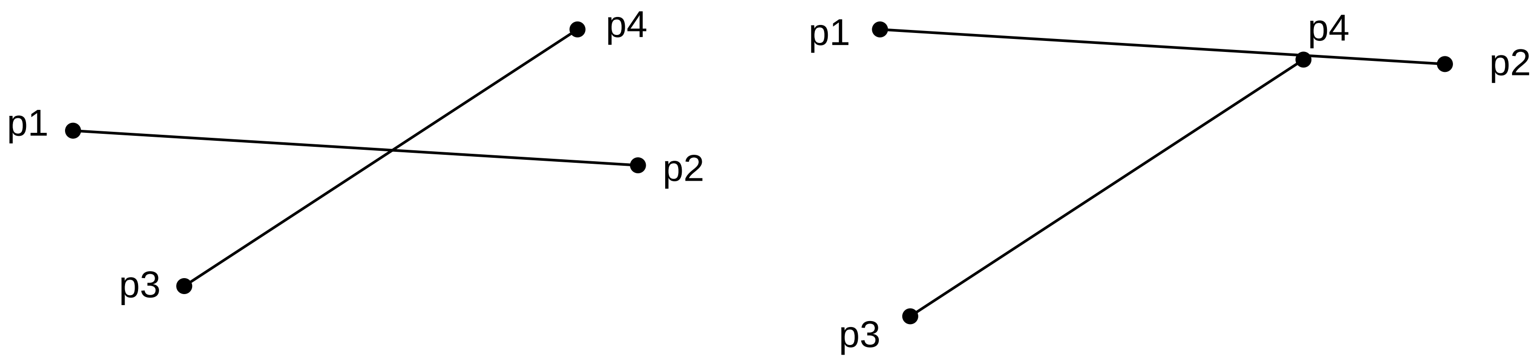
Ogólny wzór ma postać następującą:

$$(a_{11} \cdot a_{22} \cdot a_{33} + a_{12} \cdot a_{23} \cdot a_{31} + a_{13} \cdot a_{21} \cdot a_{32}) - (a_{13} \cdot a_{22} \cdot a_{31} + a_{11} \cdot a_{23} \cdot a_{32} + a_{12} \cdot a_{21} \cdot a_{33})$$

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Przecinanie się odcinków p1 – p2 oraz p3 – p4

Odcinek p1 – p2 *przekracza* odcinek p3 – p4, gdy punkt p1 leży po jednej stronie prostej *L* przechodzącej przez p3 i p4, a p2 leży po drugiej stronie prostej *L*.



Odcinek p1 – p2 przecina p3 – p4 wtedy, gdy zachodzi co najmniej jeden z warunków:

1. odcinek p1 – p2 przekracza odcinek p3 – p4 oraz
   odcinek p3 – p4 przekracza odcinek p1 – p2,

2. koniec jednego z odcinków leży na drugim odcinku.

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Przecinanie się odcinków p1 – p2 oraz p3 – p4

Idea metody zwracającej *true*, gdy odcinki się przecinają:

1. Wyznacz położenie każdego końca odcinka względem drugiego odcinka:
   d1 = det(p3,p4,p1);
   d2 = det(p3,p4,p2);
   d3 = det(p1,p2,p3);
   d4 = det(p1,p2,p4);
   Jeśli (d1*d2<0) i (d3*d4<0) /* czyli wyznaczniki, parami, są różnych znaków */
   to zwróć *true*
2. Zbadać położenie współliniowych punktów względem stosownych odcinków:
   jeśli (d1=0) i punktWspółliniowyLezyNaOdcinku(p3,p4, p1)
   lub jeśli (d2=0) i punktWspółliniowyLezyNaOdcinku(p3,p4, p2)
   lub jeśli (d3=0) i punktWspółliniowyLezyNaOdcinku(p1,p2, p3)
   lub jeśli (d4=0) i punktWspółliniowyLezyNaOdcinku(p1,p2, p4)
   to zwróć *true*
   else zwróć *false*

gdzie:

punktWspółliniowyLezyNaOdcinku(a, b, p) // p leży na odcinku a – b
Jeśli (xp >= min(xa, xb)) i (xp <= max(xa, xb)
i (yp >= min(ya, yb)) i (yp <= max(ya, yb) to zwróć true
else zwróć false

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Ogólne techniki konstrukcji algorytmów geometrycznych

Trzy podstawowe techniki konstrukcji algorytmów geometrycznych:

1. Zamiatanie polarne (biegunowe)

   Najpierw wybieramy jeden z punktów i porządkujemy resztę obiektów zgodnie z ich współrzędną polarną względem tego punktu. Następnie przeglądamy punkty zgodnie z ich uporządkowaniem.
   Przykład: konstrukcja algorytmu znajdowania otoczki wypukłej zbioru punktów.

2. Zamiatanie

   Zaczynamy od posortowania punktów zgodnie z jedną z ich współrzędnych, np. współrzędną x. Następnie przeglądamy punkty, przesuwając pionową prostą (tzw. miotłę) od lewej do prawej. W miotle pamiętamy informację o obiektach ją przecinających.
   Przykład: konstrukcja algorytmu sprawdzającego, czy w zbiorze odcinków istnieją dwa odcinki przecinające się.

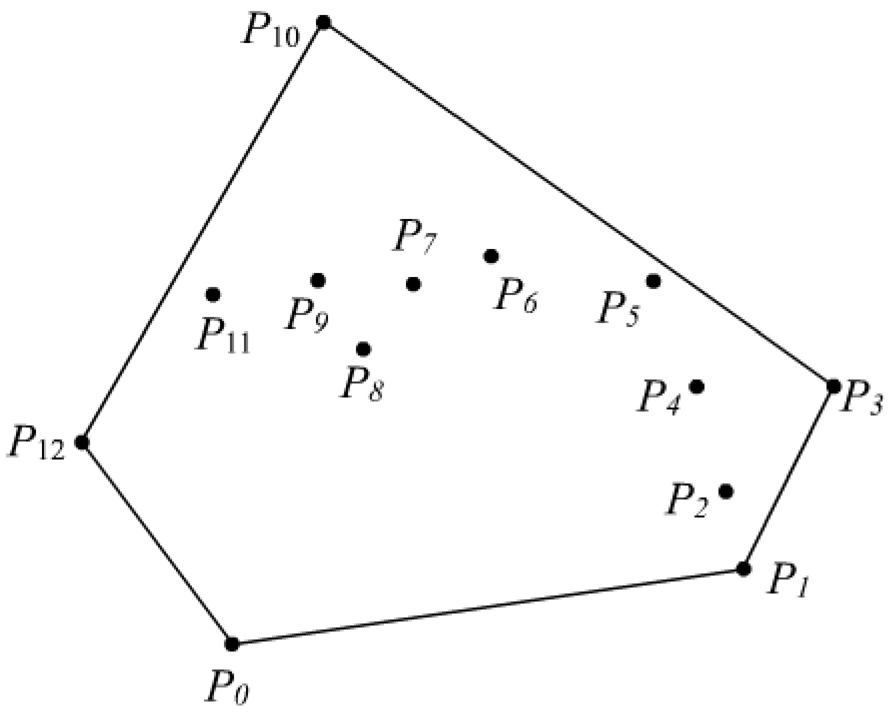
3. Dziel i zwyciężaj

   Podział problemu następuje tutaj zazwyczaj względem pewnej (pionowej) prostej, a następnie wyniki częściowe z dwóch mniejszych problemów są scalane.
   Przykład: konstrukcja algorytmu wyznaczającego najmniejszą odległość w zbiorze punktów.

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Znajdowanie otoczki wypukłej zbioru punktów

Otoczka wypukła O(S) skończonego zbioru punktów S to najmniejszy wypukły wielokąt P zawierający punkty zbioru S.
W problemie otoczki wypukłej mamy dany zbiór punktów i chcemy wyznaczyć wierzchołki otoczki wypukłej w kolejności ich występowania na jej obwodzie.



S = {p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12}

O(S) = {p0, p1, p3, p10, p12}

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Znajdowanie otoczki wypukłej zbioru punktów – algorytm Grahama

Idea algorytmu:

W algorytmie Grahama problem wypukłej otoczki jest rozwiązywany z użyciem stosu S, który zawiera kandydatów na wierzchołki otoczki.
Każdy punkt z wejściowego zbioru Q jest raz wkładany na stos, natomiast punkty nie będące wierzchołkami otoczki są ze stosu zdejmowane. W momencie zakończenia działania algorytmu stos S zawiera punkty występujące na otoczce w kolejności odwrotnej do ruchu wskazówek zegara.

Danymi wejściowymi do procedury GRAHAM jest co najmniej 3-elementowy zbiór punktów Q. Procedura ta używa funkcji:

- top(S) zwracającej wierzchołek stosu S,
- nextToTop(S) zwracającej drugi wierzchołek na stosie,
- pop(S) zdejmującej ze stosu element znajdujący się na szczycie stosu,
- push(p, S) kładącej wierzchołek p na szczyt stosu .

Algorytm wymaga na wstępie wybranie ze zbioru Q punktu „startowego” oraz posortowaniu pozostałych punktów względem ich współrzędnej polarnej (biegunowej).

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Znajdowanie otoczki wypukłej zbioru punktów – algorytm Grahama

GRAHAM(Q)

1. Niech p0 będzie punktem z o najmniejszej współrzędnej y; jeżeli jest kilka takich punktów, to tym najbardziej na lewo spośród nich.
2. Posortować pozostałe punkty ze zbioru Q rosnąco względem ich współrzędnych polarnych, w odniesieniu do p0. Niech *{p1, p2, … , pn}* będzie tym posortowanym ciągiem.
3. Jeżeli w ciągu *{p1, p2, … , pn}* występują dwa punkty o takiej samej współrzędnej polarnej, to pozostawić tylko jeden z nich, najbardziej odległy od p0; Niech *{p1, p2, … , pm}* będzie pozostałym ciągiem punktów.
4. push(p0,S); push(p1,S); push(p2,S);
5. dla i od 3 do m wykonuj:

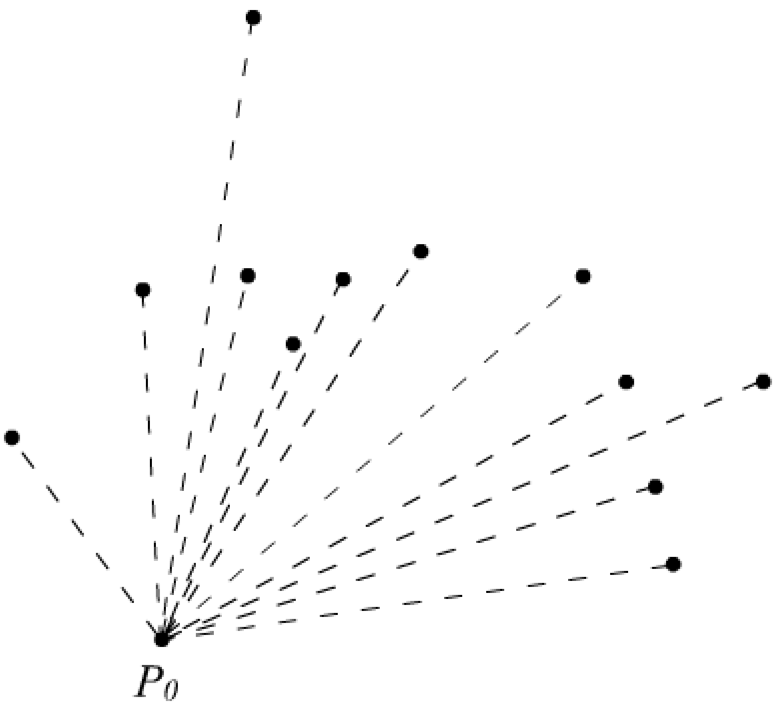
   - dopóki (punkt *pi* jest na lewo wektora: top(S) → nextToTop(S)) pop(S);
   - push(*p*i, S);
6. Zwróć S. // Stos S zawiera wierzchołki wielokąta w kolejności odwrotnej.

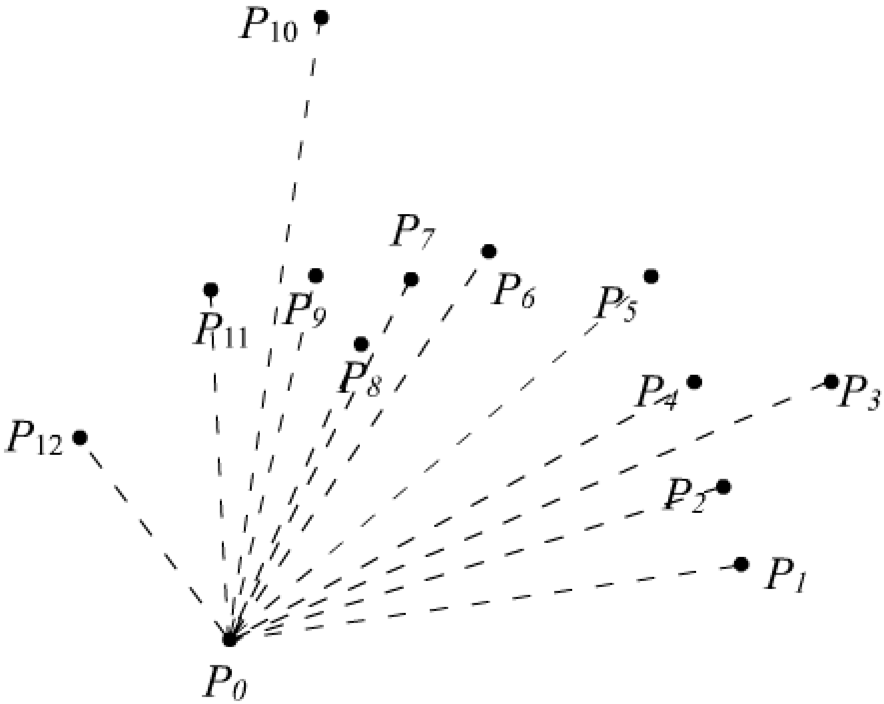
Algorytm o złożoności wyznaczonej przez złożoność sortowania, co najmniej liniowo-logarytmicznej. Wykorzystuje technikę zamiatania polarnego.

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Znajdowanie otoczki wypukłej zbioru punktów – algorytm Grahama

Ilustracja kroków algorytmu:





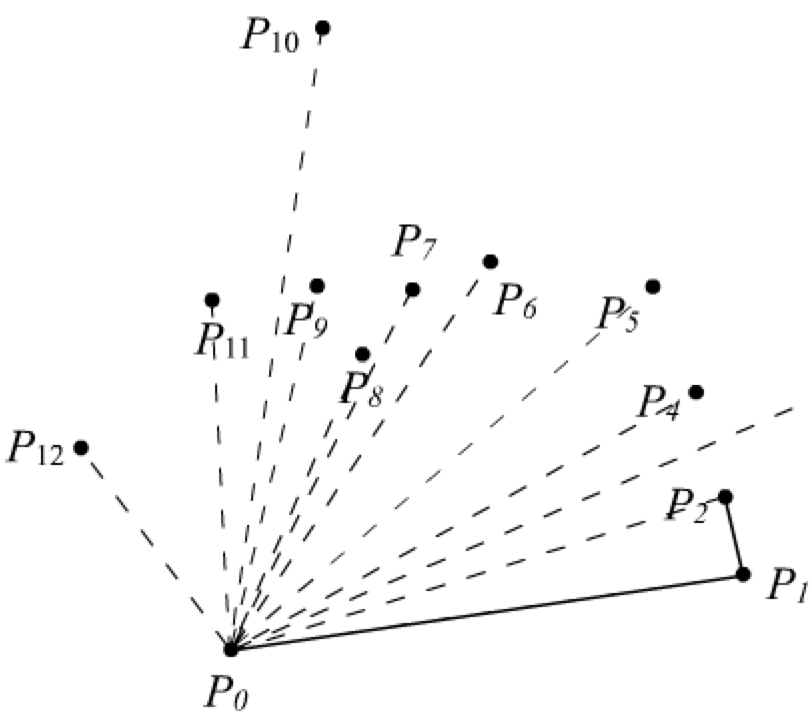
Wybieramy punkt *p0* jako leżący najniżej. Jeśli jest więcej takich punktów, wybieramy ten najbardziej z lewej strony.

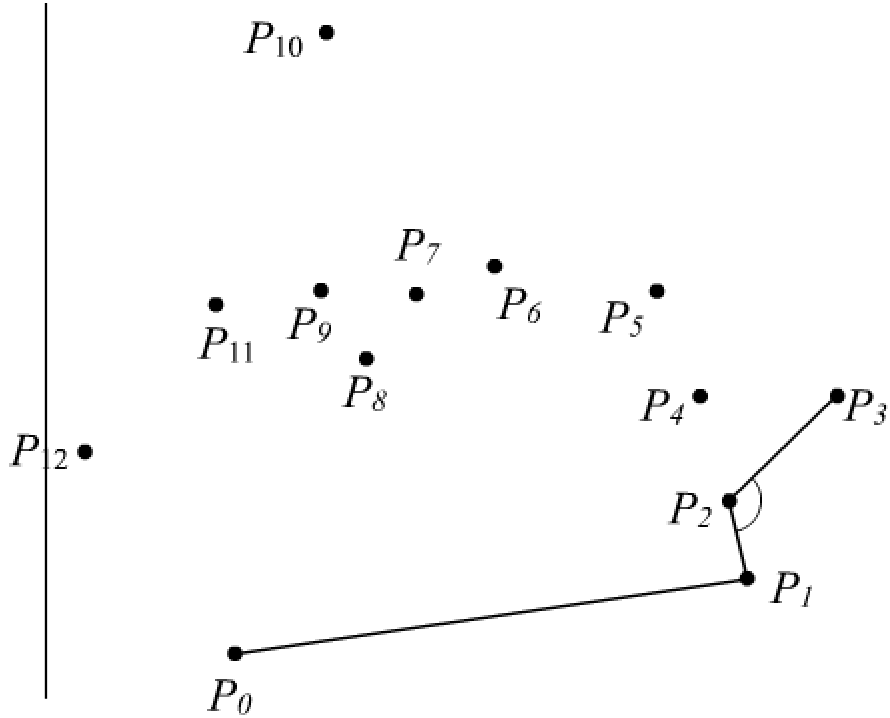
Sortujemy punkty w porządku rosnącej współrzędnej polarnej (w jakim będą przetwarzane).

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Znajdowanie otoczki wypukłej zbioru punktów – algorytm Grahama

Ilustracja kroków algorytmu:





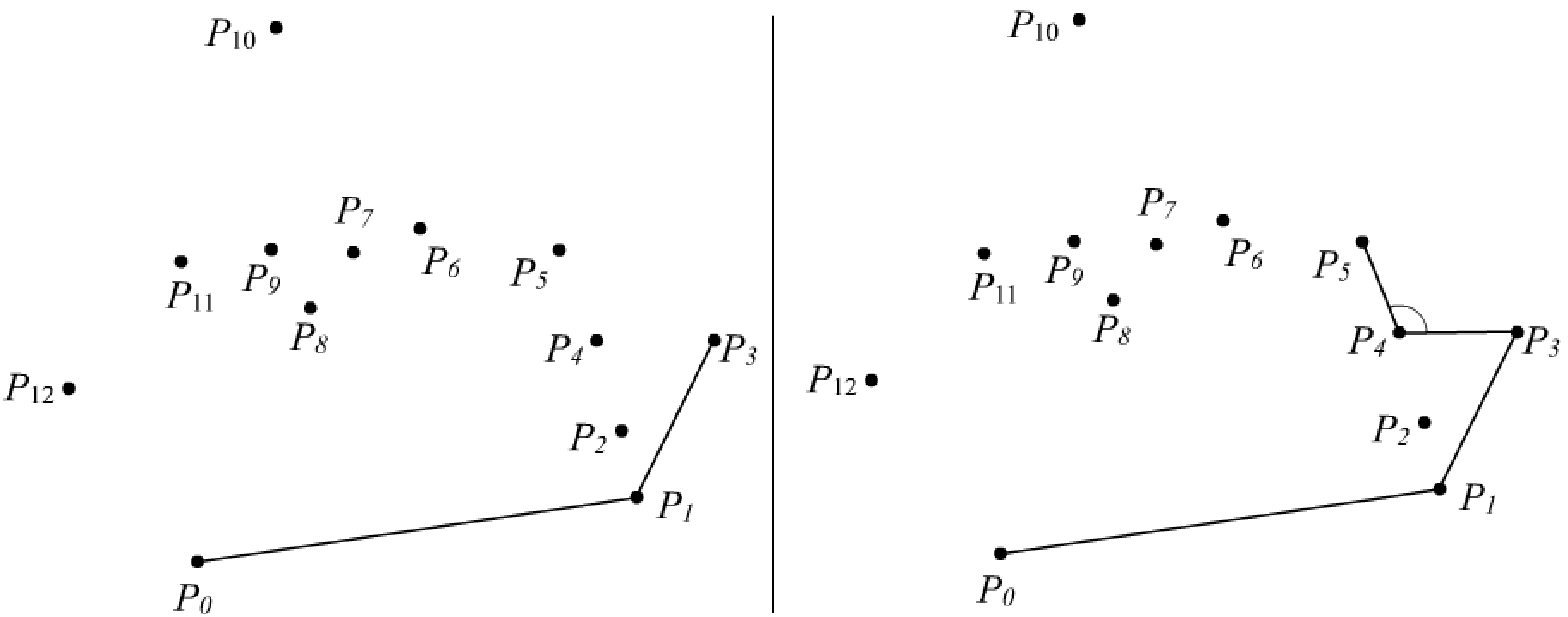
Konstruujemy otoczkę wypukłą złożoną z trzech pierwszych punktów: *{p0, p1, p2}*. W kolejnych krokach będziemy konstruować otoczkę wypukłą dla coraz większej liczby punktów.

Dodajemy połączenie otoczki z kolejnym punktem (*p3*). Otoczka *{p0, p1, p2, p3}* przestała być wypukła (punkt p3 leży na lewo od wektora p2 → p1.

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Znajdowanie otoczki wypukłej zbioru punktów – algorytm Grahama

Ilustracja kroków algorytmu:



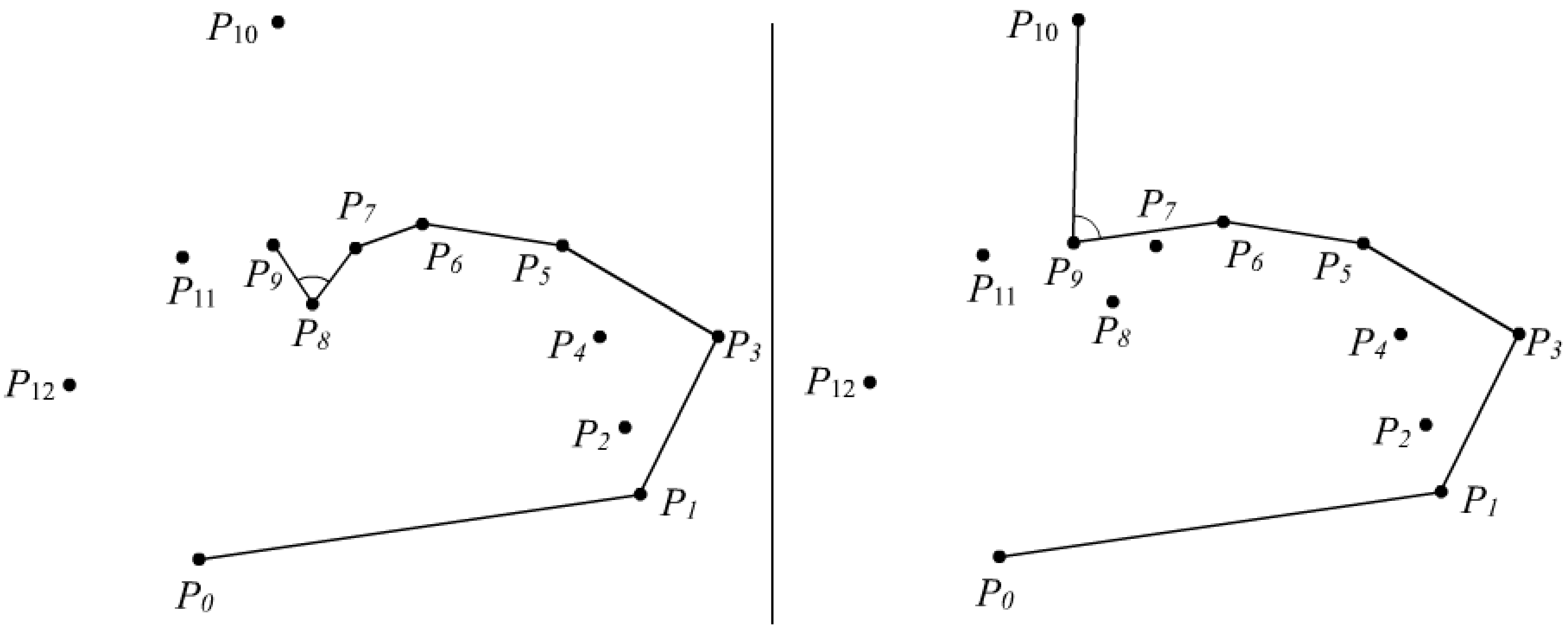
Usuwany punkt *p2* z otoczki (zastępujemy kąt wklęsły odcinkiem p1 – p3).

Po kolejnych krokach…

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Znajdowanie otoczki wypukłej zbioru punktów – algorytm Grahama

Ilustracja kroków algorytmu:



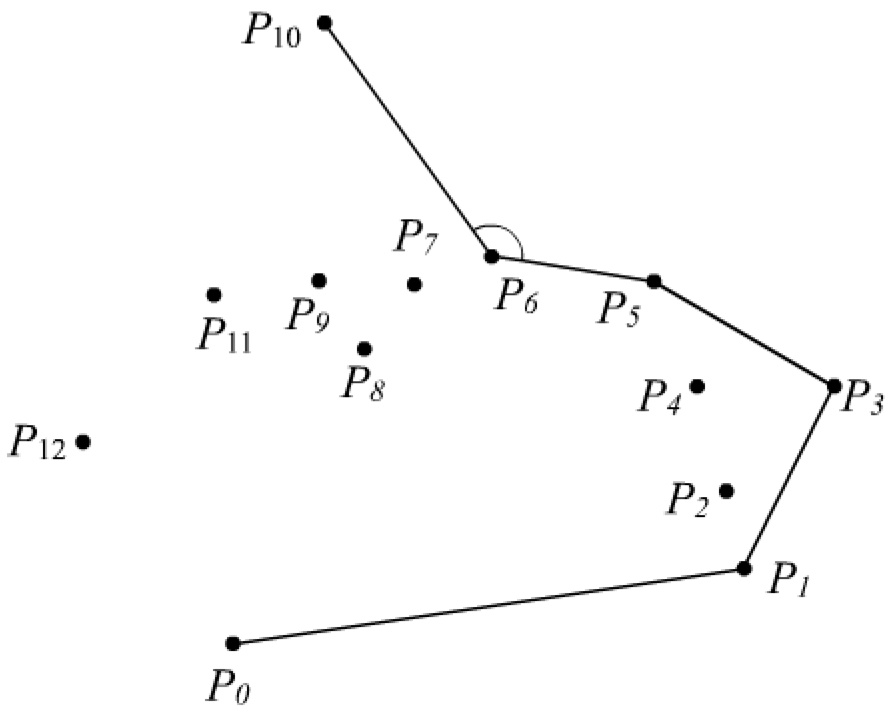
Po kilku kolejnych krokach…

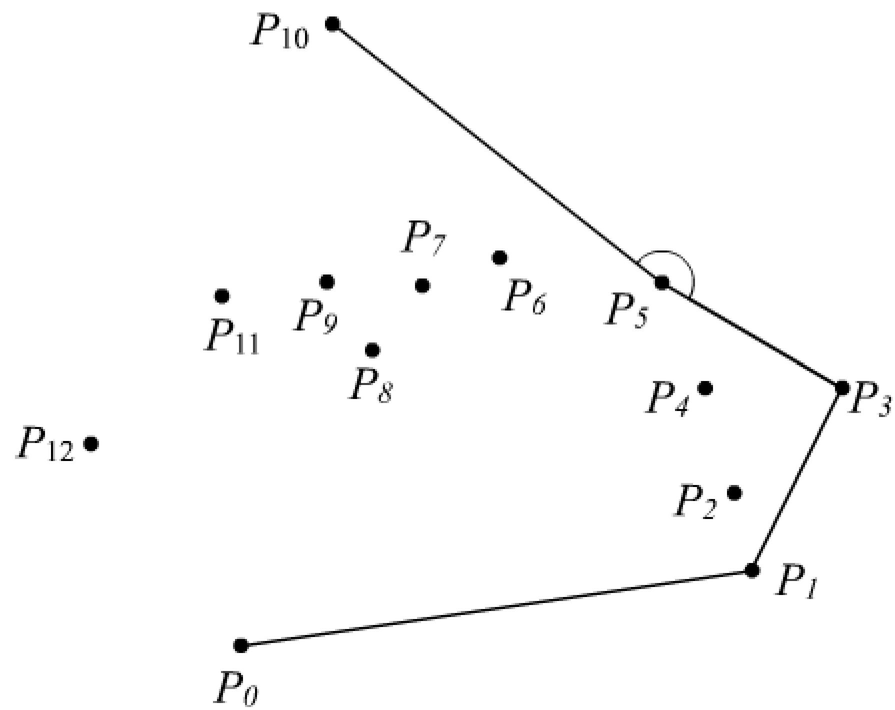
Po kilku kolejnych krokach…

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Znajdowanie otoczki wypukłej zbioru punktów – algorytm Grahama

Ilustracja kroków algorytmu:





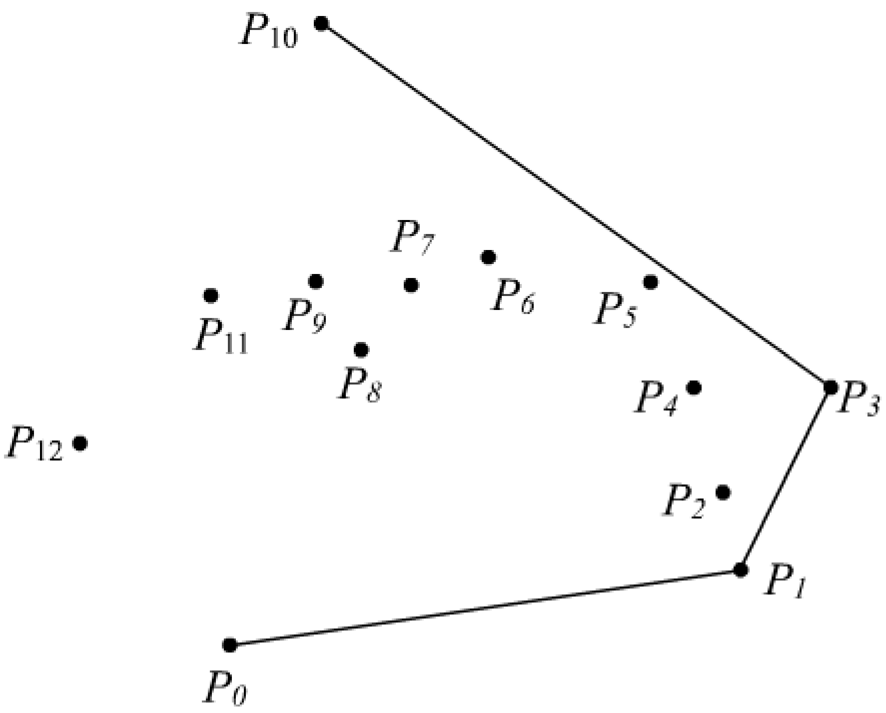
Po kilku kolejnych krokach…

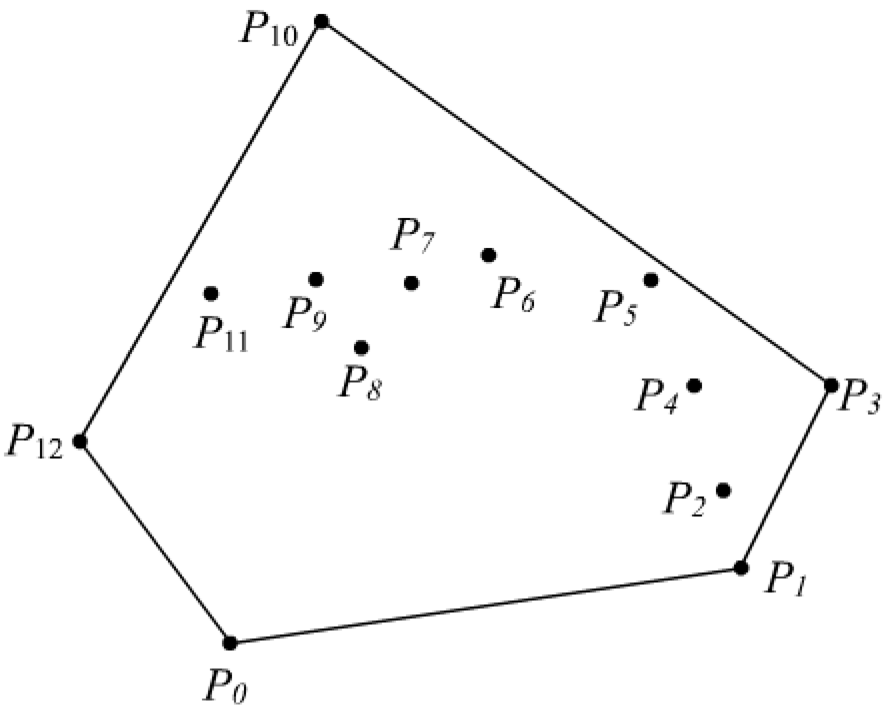
Po kilku kolejnych krokach…
Otoczka = {p0, p1, p3, p5, p10}

Politechnika Wrocławska, Wydział Informatyki i Zarządzania
Zbigniew Szpunar

Znajdowanie otoczki wypukłej zbioru punktów – algorytm Grahama

Ilustracja kroków algorytmu:





Po kilku kolejnych krokach…
Otoczka = {p0, p1, p3, p10}

Finalnie:
Otoczka = {p0, p1, p3, p10, p12}