

Krzysztof Bruniecki\*

## Porównanie algorytmów ważonego umieszczania grafów w grafach minimalizujących opóźnienia komunikacyjne

### 1. Wstęp

W celu optymalizacji przydziału procesów do środowiska docelowego pod względem opóźnień komunikacyjnych można rozważyć model grafowy, polegający na zastąpieniu sieci procesów grafem gościem, natomiast środowiska docelowego grafem gospodarzem.

Umieszczenie grafu  $G$  (gościa) w grafie  $H$  (gospodarza) polega na odwzorowaniu wierzchołków grafu  $G$  w wierzchołki grafu  $H$  oraz odwzorowaniu krawędzi gościa w odpowiednie ścieżki gospodarza. Graf gość reprezentuje sieć komunikujących się ze sobą procesów, natomiast graf gospodarz dostępną architekturę wykonania tych zadań.

Optymalizacja opóźnień komunikacyjnych polega wówczas na wybraniu odwzorowania, które minimalizuje koszty wynikające z potrzeby użycia zastępczych ścieżek w grafie docelowym.

Dotychczas wiele prac poświęcono przypadkowi klasycznemu, w którym grafy  $G$  i  $H$  są grafami prostymi. Ten model został zaproponowany przez Moniena i Sudborough [1], jednakże podobne rozważania dla pewnych szczególnych klas grafów były prowadzone wcześniej [2–4].

Podejście klasyczny nakłada ograniczenie związane z jednolitością krawędzi, zarówno dla grafu gościa, jak i gospodarza. Aby usunąć to ograniczenie, została zaproponowana modyfikacja, polegająca na zastosowaniu wag [5]. Umożliwia to użycie modelu do rozwiązywania rzeczywistych problemów, gdzie połączenia, zarówno wymagane, jak i dostępne, nie są jednorodne.

Model ważony umieszczania grafów w grafach był rozważany w pracach [5–6]. Zostały tam zaprezentowane algorytmy, rozwiązujące zagadnienie ważonej dylatacji dla pewnych szczególnych klas grafów. W odróżnieniu od wcześniejszych, niniejsza praca rozważa algorytmy działające dla dowolnych grafów, a nie tylko dla specyficznych topologii. Dodatkowo, rozważane algorytmy zostały zaimplementowane, przetestowane i porównane pod względem czasów działania i jakości uzyskiwanych rozwiązań.

---

\* Katedra Systemów Geoinformatycznych, Politechnika Gdańska

## 2. Podstawy modelu ważonego

Niech  $G = (V_G, E_G)$  jest grafem umieszczanym, a  $H = (V_H, E_H)$  grafem docelowym.

### Definicja 1

Parę funkcji  $(\varphi, \pi)$  nazywamy *umieszczeniem*, jeżeli  $\varphi$  przyporządkowuje różnowartościowo wierzchołki grafu  $G$  wierzchołkom grafu  $H$ , a funkcja  $\pi$  przyporządkowuje krawędziom grafu  $G$  ścieżki w grafie  $H$ . Oczywiście ścieżka  $\pi(\{v, u\})$  dla każdej krawędzi  $\{v, u\} \in E_G$  łączy wierzchołki  $\varphi(v)$  i  $\varphi(u)$  w grafie  $H$ .

Aby wprowadzić model ważony, konieczne jest dodanie do grafów parametrów różniących poszczególne krawędzie.

### Definicja 2

Funkcje  $w_G$  oraz  $w_H$  przyporządkowują krawędziom grafów  $G$  i  $H$  dodatnie wagi nazywane *długościami*. W przypadku klasycznym – bez wag – funkcje te można uważać za tożsamościowo równe 1. *Długością ścieżki* w grafie nazywamy sumę wag krawędzi wchodzących w jej skład.

Podstawowe kryterium jakości, dla modelu ważonego, związane jest z opóźnieniem wnikającym z zstąpienia krawędzi grafu gościa przez ścieżkę grafu gospodarza. Można zatem zdefiniować następujący cel optymalizacyjny.

### Definicja 3

*Dylatacją krawędzi*  $\{v, u\} \in E_G$  w umieszczeniu  $(\varphi, \pi)$  nazywamy długość ścieżki  $\pi(\{v, u\})$  podzieloną przez długość krawędzi  $\{v, u\}$ . *Dylatacją umieszczenia*  $(\varphi, \pi)$  nazywamy maksymalną dylatację krawędzi  $\{v, u\}$  po wszystkich krawędziach z  $E_G$ . *Optymalną dylatacją umieszczenia* grafu  $G$  w  $H$  nazywamy najmniejszą dylatację umieszczenia po wszystkich poprawnych umieszczeniach  $(\varphi, \pi)$  grafu  $G$  w graf  $H$ .

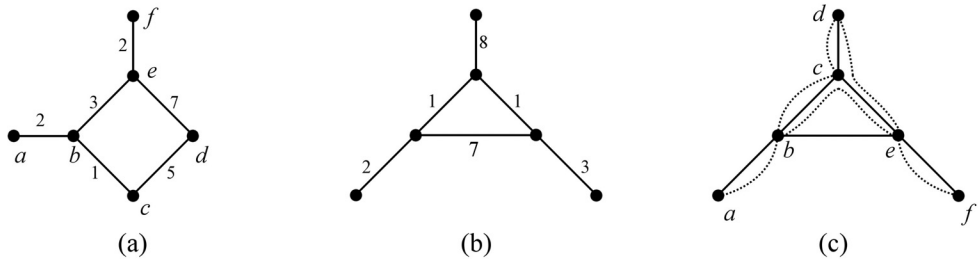
Odrębnym, wartym rozważenia, kryterium optymalizacyjnym jest uśredniona wersja dylatacji, która w naturalny sposób jest mniej wrażliwa na opóźnienia pojedynczych krawędzi.

### Definicja 4

*Średnią dylatacją umieszczenia*  $(\varphi, \pi)$  nazywamy iloraz sumy dylatacji krawędzi grafu gościa przez łączną liczbę krawędzi. *Optymalną średnią dylatacją umieszczenia* grafu  $G$  w  $H$  nazywamy najmniejszą średnią dylatację umieszczenia po wszystkich poprawnych umieszczeniach  $(\varphi, \pi)$  grafu  $G$  w graf  $H$ .

Na rysunku 1 przedstawione zostało umieszczenie przykładowego grafu (rys. 1a) w grafie (rys. 1b). Przykładowo, dylatacja krawędzi  $\{c, d\}$  wynosi  $8/5$  i jest największa spośród wszystkich krawędzi grafu gościa. Tym samym, dylatacja zaprezentowanego umieszczenia (rys. 1c) wynosi  $8/5$ . Warto zauważyć, że krawędź o wadze 7 w grafie gospodarza nie jest wykorzystywana (pomimo łączenia bezpośrednio obrazów wierzchołków  $b$  i  $e$ ), zamiast niej wykorzystywana jest ścieżka o krótszej długości. Dodatkowo, zgodnie z definicją, dopuszczalne jest wielokrotne wykorzystywanie tej samej krawędzi grafu gospodarza do zestawiania różnych połączeń.





Rys. 1. Przykładowe umieszczenie (c) ważonego grafu (a) w grafie (b)

### 3. Ogólne własności modelu

Zarówno optymalizacja dylatacji, jak i średniej dylatacji są problemami NP-trudnymi [6].

Można zauważyć, że optymalna wartość dylatacji przy umieszczaniu grafu w siebie jest równa jedności. W przypadku średniej dylatacji, optimum może być mniejsze od jedności. Ma to miejsce, w szczególności, gdy wagi na krawędziach nie spełniają nierówności trójkąta.

Stosując odległościowe kryterium optymalizacji, redukujemy problem do znalezienia odwzorowania wierzchołków – nie musimy zajmować się problemem przydziału ścieżek, gdyż wybór optymalnych połączeń w grafie gospodarza jest naturalny i polega na wyborze ścieżki o najkrótszej odległości pomiędzy danymi wierzchołkami.

W ogólnym przypadku, gdy nie mamy dodatkowej informacji o strukturze grafu gospodarza, warto zauważyć, że może on być rozważany, jako graf pełny z wagami na krawędziach spełniającymi nierówność trójkąta. W praktyce instancję grafu gospodarza  $H$  możemy zastąpić grafem pełnym  $H'$  z wagami na krawędziach równymi odległościom pomiędzy odpowiednimi wierzchołkami w grafie  $H$ . Taka zamiana nie zmienia wartości dylatacji ani średniej dylatacji.

Podobnie możemy postąpić z grafem  $G$ . Pełny graf ważony  $G'$  powstały na bazie wierzchołków  $G$  przez nadanie krawędziom wag równych najkrótszym ścieżkom pomiędzy odpowiednimi wierzchołkami nie zmienia wartości dylatacji. Należy zauważyć, że nie możemy w ten sposób zastępować grafu  $G$  w przypadku optymalizacji średniej ważonej dylatacji.

### 4. Zakres eksperymentu

Podstawowym zagadnieniem rozważanym w pracy jest porównanie algorytmów dla problemu umieszczania grafów w grafach optymalizujących dylatację oraz średnią dylatację. Zauważmy, że wagi przyporządkowane są zarówno do krawędzi grafu gościa, jak i do krawędzi grafu gospodarza.

#### 4.1. Testowane algorytmy

Dla problemu optymalizacji dylatacji zostały przetestowane algorytmy dokładne, natomiast dla problemu optymalizacji średniej dylatacji zostały przetestowane również zachłanne algorytmy heurystyczne.

##### Wyczerpujące przeglądanie

Algorytm polega na sprawdzeniu wartości dylatacji lub średniej dylatacji dla wszystkich możliwych odwzorowań wierzchołków grafu. Liczba wszystkich możliwych przyporządkowań wierzchołków grafu gościa w wierzchołki grafu gospodarza równa jest liczbie wariacji bez powtórzeń  $|V_G|$  elementowych ze zbioru  $V_H$ . Złożoność obliczeniową zwiększa dodatkowo potrzeba wyznaczenia wartości dylatacji ważonej dla każdego przyporządkowania.

Pomijając czas potrzebny na policzenie wszystkich odległości w grafie gospodarza można określić złożoność obliczeniową następująco:  $O(|E_G| \|V_H| |/(|V_H| - |V_G|)!)$ .

Algorytm wyczerpującego przeglądania oznaczany będzie dalej symbolem BF.

##### Wyczerpujące przeglądanie z backtrackingiem

Użycie backtrackingu jest modyfikacją polegającą na przyspieszeniu algorytmu przeglądającego wszystkie możliwe odwzorowania.

Możliwe odwzorowania wierzchołków generowane są w sposób rekurencyjny, tworząc drzewo przeszukiwań. Korzeniem drzewa jest odwzorowanie puste, w którym żaden wierzchołek grafu gościa nie został przypisany do wierzchołków grafu gospodarza. Liśćmi takiego drzewa są odwzorowania kompletne, w których wszystkie wierzchołki grafu gościa mają obrazy w postaci wierzchołków grafu gospodarza. Poszczególne ścieżki – od korzenia do liścia – reprezentują serię wyborów wierzchołów grafu gospodarza, w które odwzorowywane są kolejne wierzchołki grafu gościa. Węzły pośrednie reprezentują częściowe odwzorowania, w którym pewien początkowy podzbiór wierzchołków grafu gościa ma przyporządkowane wierzchołki w grafie gospodarza.

Podczas przeglądania możliwych odwzorowań zapamiętujemy najlepszą dotychczas wartość ważonej dylatacji (lub średniej dylatacji). Jeśli rozwiązanie cząstkowe, w pewnym węźle pośrednim drzewa rozwiązań, osiągnęło lub przekroczyło wartość dylatacji rozwiązania, jak dotąd najlepszego, wówczas możemy pominąć uzupełnienie takiego rozwiązania cząstkowego. Tym samym usuwamy część węzłów z drzewa przeszukiwań i redukujemy czas obliczeń potrzebny do uzyskania rozwiązania optymalnego.

Sposób generowania drzewa rozwiązań ma zasadnicze znaczenie w kontekście możliwości pomijania jak największych poddrzew. Intuicja podpowiada, że jeżeli dostatecznie szybko znajdziemy dobre rozwiązanie o niewielkiej dylatacji, wówczas większe poddrzewa będą pomijane. Ma to związek z zaniechaniem rozszerzania rozwiązania o umieszczenia pozostałych wierzchołków. Powodem zaniechania jest przekroczenie wartości dylatacji w częściowym rozwiązaniu.

Wpływ na generowanie drzewa rozwiązań ma między innymi kolejność wierzchołków grafu gościa, zgodnie z którą uzupełniane są rozwiązania cząstkowe.



Rozważane były modyfikacje algorytmu backtrackingu uwzględniające następujące reguły kolejności wierzchołków grafu gościa:

- losowa, (BAC);
- nierosnąco, według stopni wierzchołków, (BAC-LF);
- nierosnąco, według sum odwrotności wag incydentnych krawędzi, (BAC-wLF);
- nierosnąco, według stopni wierzchołków w minimalnym drzewie spinającym, (BAC-MST);
- nierosnąco, według sum odwrotności wag incydentnych krawędzi w minimalnym drzewie spinającym, (BAC-wMST);
- nierosnąco, według liczby sąsiadów w podgrafie indukowanym przez dotychczasowo wybrane wierzchołki (jako pierwszy jest wybierany wierzchołek o największym stopniu w całym grafie gościa), (BAC-SLF);
- nierosnąco, według sum odwrotności wag incydentnych krawędzi w podgrafie indukowanym przez dotychczasowo wybrane wierzchołki (jako pierwszy jest wybierany wierzchołek o największej sumie odwrotności wag incydentnych krawędzi w całym grafie gościa), (BAC-wSLF);
- iteracyjnie, jako ostatni wybierany jest wierzchołek o najmniejszym stopniu, następnie jest usuwany z grafu gościa i procedura jest powtarzana do wyczerpania wierzchołków (BAC-SL);
- iteracyjnie, jako ostatni wybierany jest wierzchołek o najmniejszej sumie odwrotności wag incydentnych krawędzi, następnie jest usuwany z grafu gościa i procedura jest powtarzana do wyczerpania wierzchołków, (BAC-wSL).

Formalnie, złożoność obliczeniowa jest taka jak dla algorytmu wyczerpującego przeszukiwania i wynosi  $O(|E_G| \cdot |V_H| / (|V_H| - |V_G|))$ .

### Podejście zachłanne

Jeżeli rozmiar problemu przekracza możliwości algorytmów dokładnych, może okazać się konieczne zastosowanie algorytmu szybkiego, niedającego gwarancji optymalności rozwiązania.

Powyższy postulat jest realizowany, między innymi, przez rodzinę algorytmów zachłannych. W algorytmach zachłannych decyzje cząstkowe, prowadzące do kompletnego rozwiązania, podejmowane są w oparciu o kryterium maksymalizacji lokalnej funkcji celu.

W problemach umieszczania grafów w grafach pojedyncza decyzja dotyczy uzupełnienia częściowego rozwiązania o umieszczenie kolejnego wierzchołka grafu gościa w wybrany, wolny wierzchołek grafu gospodarza.

Kolejność wierzchołków grafu gościa, zgodnie z którą rozszerzane jest umieszczenie, ustalana jest na początku algorytmu. Zgodnie z tą kolejnością ( $<$ ), dla wierzchołka  $v$  grafu gościa, wybierany jest lokalnie najlepszy, wolny wierzchołek  $x$  grafu gospodarza minimalizujący następujące wyrażenie:

$$\sum_{\substack{\{v,w\} \in E_G, \\ w < v}} \frac{w_H(x, \varphi(w))}{w_G(\{v,w\})}$$



Analogicznie jak w przypadku algorytmów backtrackingu, rozważane były modyfikacje algorytmów zachłanych ze względu na kolejność osadzania wierzchołków grafu gościa.

## 4.2. Dane testowe

Dane testowe do eksperymentów stanowiły grafy losowe o losowych wagach na krawędziach.

Dla algorytmów dokładnych przeprowadzono eksperymenty dla grafów o maksymalnie 16 wierzchołkach, natomiast dla algorytmów zachłanych liczba wierzchołków była maksymalnie równa 100. Dodatkowo, dla pojedynczej instancji problemu, liczby wierzchołków grafu gościa i gospodarza były zawsze równe.

Dla wszystkich przetestowanych grafów liczba krawędzi była dokładnie dwukrotnie większa niż liczba wierzchołków. Dla zagwarantowania istnienia umieszczenia, jako gospodarze, wybierano wyłącznie grafy spójne.

Wagi dla krawędzi były losowane w sposób równomierny w przedziale [1...100].

## 5. Wyniki

W tabeli 1 zostały zaprezentowane średnie czasy działania algorytmów dokładnych dla grafów o różnej liczbie wierzchołków. Dla każdej liczby wierzchołków zostało wykonane 100 testów z różnymi grafami losowymi, a następnie czasy działania zostały uśrednione. Dla wszystkich algorytmów stosowane były jednakowe zestawy grafów.

**Tabela 1**  
Średnie czasy działania dla algorytmów dokładnych optymalizujących dylatację

Czas działania [ms]		Liczba wierzchołków grafów gościa i gospodarza									
		7	8	9	10	11	12	13	14	15	16
Algorytm	BF	1,57	10,9	98,4	965	9670	–	–	–	–	–
	BAC	1,20	9,37	73,9	499	2970	28400	–	–	–	–
	BAC-LF	1,09	7,19	34,6	229	1350	11100	–	–	–	–
	BAC-wLF	0,47	0,94	1,41	5,16	83,9	17,03	31,2	457	652	46500
	BAC-MST	1,25	4,53	21,7	161	1010	7430	15400	–	–	–
	BAC-wMST	1,87	9,22	63,6	523	3630	38600	–	–	–	–
	BAC-SLF	1,25	6,72	29,7	293	922	14300	–	–	–	–
	BAC-wSLF	0,62	0,93	1,87	6,88	96,3	60,9	470	759	1470	–
	BAC-SL	1,41	5,31	23,0	292	848	43700	–	–	–	–
	BAC-wSL	<b>0,47</b>	<b>0,79</b>	<b>1,09</b>	<b>3,91</b>	<b>78,9</b>	<b>9,22</b>	<b>16,7</b>	<b>257</b>	<b>337</b>	<b>39500</b>



W tabeli 2 znalazło się porównanie czasów działania dla algorytmów dokładnych, optymalizujących średnią dylatację. Procedura testowa była taka sama jak w przypadku algorytmów optymalizujących dylatację.

**Tabela 2**

Średnie czasy działania dla algorytmów dokładnych optymalizujących średnią dylatację

Czas działania [ms]		Liczba wierzchołków grafów gościa i gospodarza							
		6	7	8	9	10	11	12	13
Algorytm	BF	<b>0,62</b>	3,13	23,5	239	2210	26200	–	–
	BAC	0,79	4,22	34,4	311	2870	14700	195000	–
	BAC-LF	0,70	3,60	23,5	217	1190	6080	78500	–
	BAC-wLF	<b>0,62</b>	<b>2,34</b>	14,1	95,3	267	2020	11300	84900
	BAC-MST	0,78	3,59	25,0	148	1220	6760	148000	–
	BAC-wMST	0,74	4,22	29,7	223	2850	13000	–	–
	BAC-SLF	0,94	3,60	23,4	155	1450	9750	89500	–
	BAC-wSLF	0,78	2,50	15,6	95,4	284	2090	21300	88100
	BAC-SL	0,78	3,75	23,4	175	1090	5490	183000	–
	BAC-wSL	0,63	<b>2,34</b>	<b>12,5</b>	<b>82,8</b>	<b>250</b>	<b>1870</b>	<b>10400</b>	<b>68500</b>

Wyniki porównujące algorytmy zachłanne, pod względem średniej dylatacji uzyskanych rozwiązań, zostały przedstawione w tabeli 3. W ostatnim wierszu, gdzie było to możliwe, podane zostały również wartości optymalne średniej dylatacji.

**Tabela 3**

Wartości średniej dylatacji uzyskanych rozwiązań dla algorytmów heurystycznych

Wartość średniej dylatacji		Liczba wierzchołków grafów gościa i gospodarza									
		5	6	7	8	9	10	11	20	50	100
Algorytm	GRD	2,06	1,88	2,36	2,31	2,83	2,68	2,69	3,59	4,37	4,38
	GRD-LF	2,06	1,81	2,10	2,08	2,50	2,27	2,35	3,01	3,59	3,61
	GRD-wLF	<b>1,19</b>	1,23	1,48	1,39	1,50	1,48	1,62	2,03	2,40	2,56
	GRD-MST	1,63	1,55	2,09	1,91	2,38	2,19	2,37	3,02	3,57	3,57
	GRD-wMST	1,83	1,76	2,23	2,16	2,58	2,38	2,55	3,34	4,26	4,27
	GRD-SLF	2,06	1,75	2,08	2,10	2,49	2,42	2,42	3,10	3,44	3,44
	GRD-wSLF	<b>1,19</b>	1,20	<b>1,43</b>	1,37	1,51	1,45	1,62	2,04	2,41	2,45
	GRD-SL	1,82	1,80	2,19	1,99	2,31	2,22	2,34	3,00	3,32	3,34
	GRD-wSL	<b>1,19</b>	<b>1,19</b>	1,46	<b>1,35</b>	<b>1,46</b>	<b>1,44</b>	<b>1,57</b>	<b>1,96</b>	<b>2,20</b>	<b>2,37</b>
Wartość optymalna		1,11	1,08	1,21	1,14	1,24	1,18	1,25	–	–	–



## 6. Wnioski

Algorytmy dokładne optymalizujące dylatację w czasie poniżej 200 sekund poradziły sobie z grafami o kilkunastu wierzchołkach. Najlepszym z przetestowanych algorytmów okazał się BAC-wSL. Modyfikacja „wSL” umożliwia rozpatrywanie instancji problemu większej o około 4 wierzchołki, w porównaniu do metody czystego backtrackingu, zachowując zbliżony czas poszukiwania rozwiązania. W efekcie umieszczanie grafów o 16 wierzchołkach zajmowało średnio 39,5 sekundy. Nieco gorszą efektywność zaprezentowała modyfikacja „wLF”. Modyfikacja „wSLF” poradziła sobie z instancjami o 15 wierzchołkach.

Warto w tym miejscu zwrócić uwagę na rozbieżności, jakie pojawiały się w czasach wykonania dla różnych instancji problemu o takiej samej liczbie wierzchołków. Około 90% wszystkich 16-wierzchołkowych grafów było umieszczanych optymalnie, przy użyciu algorytmu BAC-wSL, w czasie poniżej jednej sekundy.

W przypadku algorytmów dokładnych, optymalizujących średnią dylatację, wyznaczenie rozwiązania okazało się bardziej czasochłonne. Również w tym przypadku najszybszy okazał się algorytm BAC-wSL. Algorytm ten w czasie średnim 68,5 sekundy znajdował umieszczenia dokładne dla grafów o 13 wierzchołkach. Spadek wydajności, w stosunku do optymalizacji dylatacji, wynika głównie ze zmieszenia poddrzew drzewa rozwiązań, jakie są pomijane podczas przeglądania.

Niestety, zaprezentowane algorytmy dokładne potwierdzają swoją praktyczną nieużyteczność dla grafów o liczbie wierzchołków przekraczającej kilkanaście. Dla takich grafów najlepsze rezultaty, optymalizacji średniej ważonej dylatacji, osiągnął algorytm GRD-wSL.

Porównując wartości uzyskiwane przez algorytm GRD-wSL z optymalnymi, widać że wraz ze wzrostem liczby wierzchołków błąd względny ma tendencję do narastania.

Niniejsza praca absolutnie nie wyczerpuje tematyki związanej z analizą algorytmów umieszczania grafów w grafach. Warto dalszych badań wydają się zagadnienia związane z rozkładem czasu działania algorytmów dokładnych dla różnych instancji o jednakowej liczbie wierzchołków. Wartościowym uzupełnieniem przedstawionych wyników byłyby również analiza metaheurystyk w kontekście ważonego umieszczania grafów w grafach.

## Literatura

- [1] Monien B., Sudborough I.H., *Embedding one interconnection network in another*. Computing Supplementum, nr 7, 1990, 257–282.
- [2] Harper L.H., *Optimal assignments on numbers to vertices*. SIAM Journal, nr 12, 1964, 131–135.
- [3] Chvatalova J., *Optimal labelling of a product of two paths*. Discrete Mathematics, nr 11, 1975, 249–253.
- [4] Papadimitriou C.H., *The NP-completeness of the bandwidth minimization problem*. Computing, nr 16, 1976, 263–270.
- [5] Bruniecki K., *Modele umieszczania grafów w grafach i ich zastosowania*. Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej (Technologie Informacyjne), nr 13, 2007, 457–464.
- [6] Bruniecki K., *Ważone umieszczanie grafów w grafach jako model optymalizacji komunikacji w sieciach heterogenicznych*. Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej (Technologie Informacyjne), nr 16, 2008, 403–408.

