WOJSKOWA AKADEMIA TECHNICZNA

LABORATORIUM
DIAGNOSTYKA I WIARYGODNOŚĆ SYSTEMÓW KOMPUTEROWYCH

Stopień, imię i nazwisko prowadzącego	Stopień, imię i nazwisko słuchacza	Grupa szkoleniowa
mgr inż. Łukasz Strzelecki	***inż. Grzegorz Pol***	***I0G1S4***

Data wykonania ćwiczenia
19.04.2012 r.

SPRAWOZDANIE
Z
PRACY LABORATORYJNEJ
NR 3

Temat: Badanie własności m-diagnozowalnych struktur opiniowania diagnostycznego

Zadanie 1

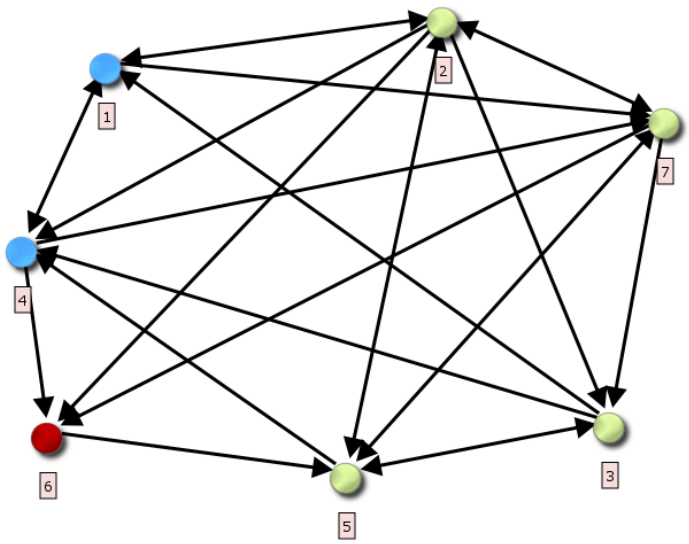
Wykorzystując warunki konieczne dla m-diagnozowalnej SOD wyznaczyć taką strukturę typu PMC. Opierając się na warunkach wystarczających Hakimi'ego ([9], [4]) wskazać przypadek (wyznaczyć konkretną strukturę w programie simdiag), w którym struktura spełniająca warunki konieczne nie jest m-diagnozowalna. Odpowiedzieć na pytanie: z czego wynika taka sytuacja?

- Zastosowana procedura postępowania przy wyznaczaniu m-diagnozowalnej SOD

ROZWIĄZANIE

Podczas tworzenia diagramu sprawdzałam zgodność z warunkami koniecznymi do zastosowania m-diagnozowalnej struktury:

1. $n \geq 2m + 1, dla\ m = 3, n \geq 7$
2. $\mu^{-}(e) \geq m = 3$



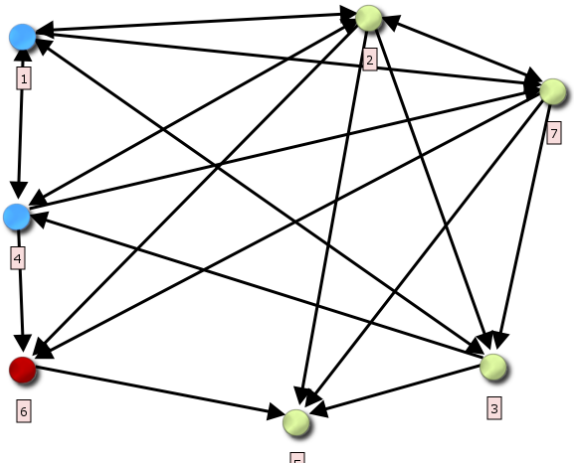
3. $(\forall\ 0 \leq p \leq m - 1)(\forall\ E' \in E: |E'| = |E| - \ 2m + p): \mathbb{\Gamma}(E') > p$

Dla m=3, $0 \leq p \leq m - 1$, tak więc p=0 v p=1 v p=2

Dla p=0, E’ = 7 – 2 x 3 + 0 = 1 – warunek niespełniony (wierzchołek nr 5 – rys na kolejnej stronie)

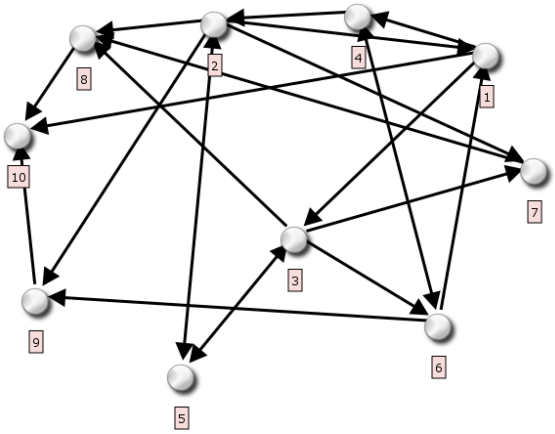
Dla p=1, E’ = 7 – 2 x 3 + 1 = 2

Dla p=2, E’ = 7 – 2 x 3 + 2 = 3



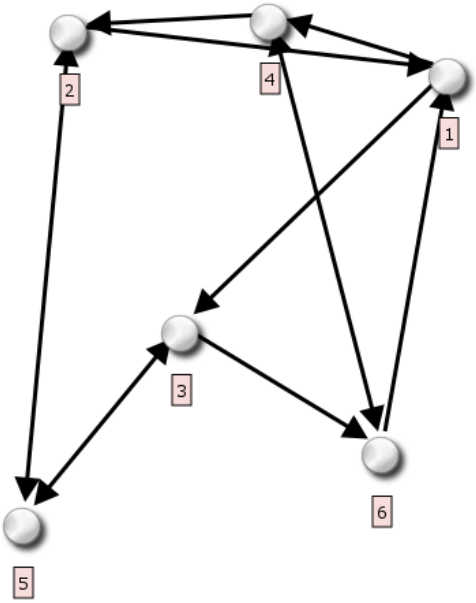
Zadanie 2

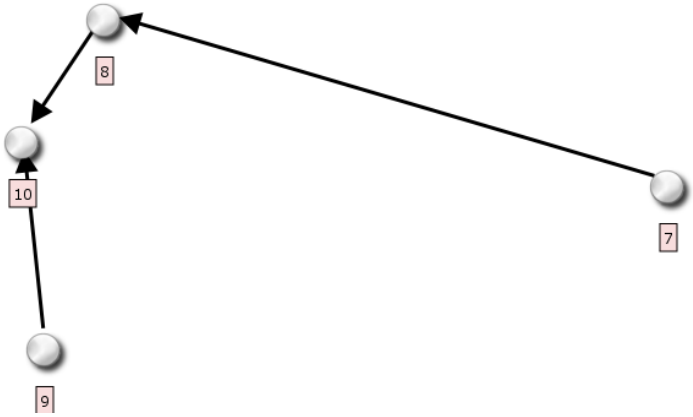
Wygenerować m-diagnozowalną strukturę opiniowania diagnostycznego typu PMC 10-tego rzędu o składowej spójności 6-tego rzędu, a następnie wskazać część acykliczną SOD oraz składową silnej spójności.



Po wygenerowaniu przez program powyższej struktury usuwam wierzchołki, które nie zakłócają składowej spójności (w moim przypadku są to wierzchołki 7, 8, 9,10). Rysunek tej struktury znajduje się po prawej stronie.

Częścią acykliczną będą usunięte w poprzedniej strukturze wierzchołki (poniżej):





Zadanie 3

Porównać efektywność (m.in. czas wykonania zadania, możliwości) poszczególnych generatorów struktur opiniowania diagnostycznego dostępnych w aplikacji Simdiag (patrz: punkt 7) oraz ich skuteczność (w przypadku metod heurystycznych).

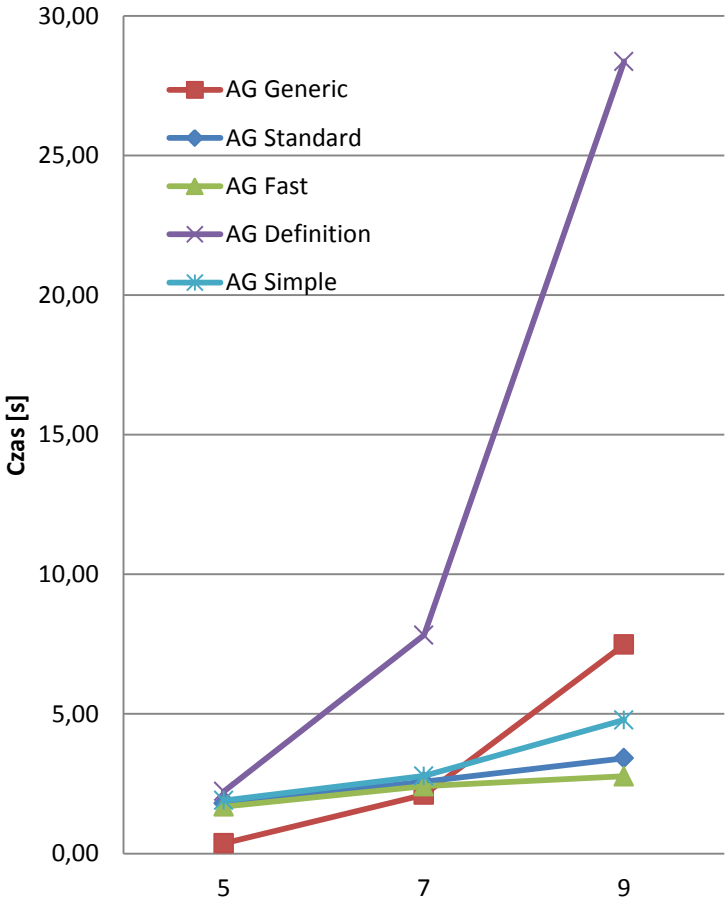
Algorytmy heurystyczne

AG Standard

	5	7	9
1	1,69	2,85	3,11
2	1,66	2,76	3,02
3	2,09	3,03	3,04
4	1,73	2,42	3,29
5	1,75	2,41	3,26
6	1,68	2,15	3,02
7	1,51	2,32	3,49
8	1,70	2,43	3,57
9	2,36	2,25	3,90
10	1,79	3,17	4,47
śr	**1,80**	**2,58**	**3,42**

AG Generic

	5	7	9
1	0,61	2,81	9,08
2	0,16	2,12	2,34
3	0,53	2,64	8,98
4	0,16	2,43	9,21
5	0,37	2,06	8,23
6	0,16	1,44	5,94
7	0,38	1,98	8,39
8	0,16	2,09	10,71
9	0,55	1,95	5,91
10	0,58	1,54	6,03
śr	**0,37**	**2,11**	**7,48**



AG Fast

	5	7	9
1	1,39	2,13	3,17
2	1,86	2,42	2,66
3	1,50	2,48	2,65
4	1,48	3,04	2,70
5	2,05	2,12	2,80
6	1,70	2,35	2,75
7	1,50	2,39	2,66
8	1,78	2,56	2,76
9	1,44	2,54	2,54
10	2,11	2,08	2,97
śr	**1,68**	**2,41**	**2,77**

AG Definition

	5	7	9
1	2,32	10,83	36,64
2	2,00	7,74	21,10
3	2,42	8,80	26,78
4	2,44	7,02	32,14
5	2,30	7,29	27,99
6	1,87	6,05	28,77
7	2,06	7,39	27,54
8	2,48	6,32	30,57
9	2,06	7,97	31,43
10	2,29	8,81	20,73
śr	**2,22**	**7,82**	**28,37**

AG Simple

	5	7	9
1	1,64	2,26	7,56
2	1,59	2,50	3,44
3	1,58	2,96	8,51
4	1,66	4,07	3,82
5	1,50	2,67	4,31
6	2,47	2,59	3,79
7	1,59	2,48	3,99
8	1,68	2,48	3,95
9	2,84	3,40	4,04
10	2,44	2,37	4,44
śr	**1,90**	**2,78**	**4,79**

Algorytmy analityczne i porównanie z heurystycznymi

Analitic Gradient

	5	7	9
1	0,23	0,53	9,48

Analitic Standard

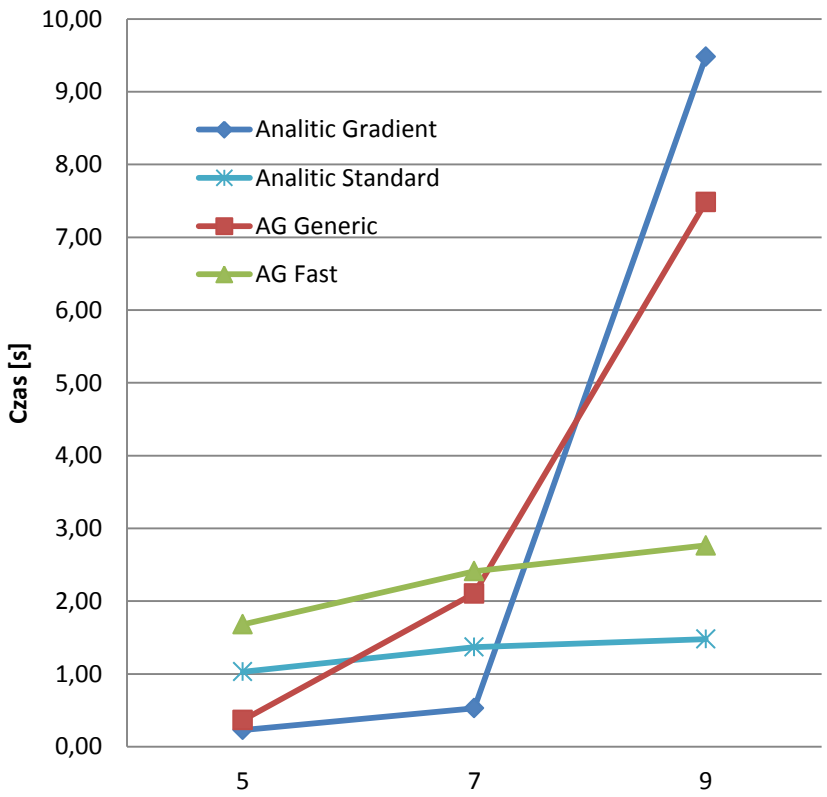
	5	7	9
1	1,03	1,37	1,48

Brute Force I

	5	7	9
1	70,88	błąd	błąd

Brute Force II

	5	7	9
1	23,38	błąd	błąd



Skuteczność algorytmów heurystycznych

algorytm	AG Standard			AG Generic			AG Fast			AG Definition			AG Simple		
rząd	5	7	9	5	7	9	5	7	9	5	7	9	5	7	9
prób	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
błędów	0	0	0	0	0	0	1	1	2	0	0	0	0	0	0
skuteczność	100%	100%	100%	100%	100%	100%	90%	90%	80%	100%	100%	100%	100%	100%	100%

Wnioski

Czas działania algorytmów jest uzależniony od mocy obliczeniowej komputera, na którym zostały przeprowadzone pomiary. W moim przypadku był to komputer przenośny z procesorem Intel Core i5 M460 2,53Ghz z pamięcią operacyjną 4GB DDR3.

Pomiary algorytmów heurystycznych wykazały, że dla rzędów 5 i 7 najszybszym algorytmem był AG Generic. Algorytmy AG Standard, AG Simple, AG Fast miały podobne wyniki, jednak najszybszy z nich za każdym razem był algorytm AG Fast. Niestety jako jedyny podczas pomiaru nie miał 100% skuteczności.

Algorytmy analityczne: Analitic Gradient i Analitic Standard wypadły podobnie, minimalnie lepiej od algorytmów heurystycznych. Algorytmy Brute Force niestety dla większych rzędów powodowały przywieszanie się aplikacji i nie byłem w stanie dokonać ich pełnego pomiaru. Mogę jedynie przypuszczać, że ich czas generowania jest zdecydowanie niesatysfakcjonujący.