

SPRAWOZDANIE

Lista 2 Sztuczna Inteligencja Laboratorium

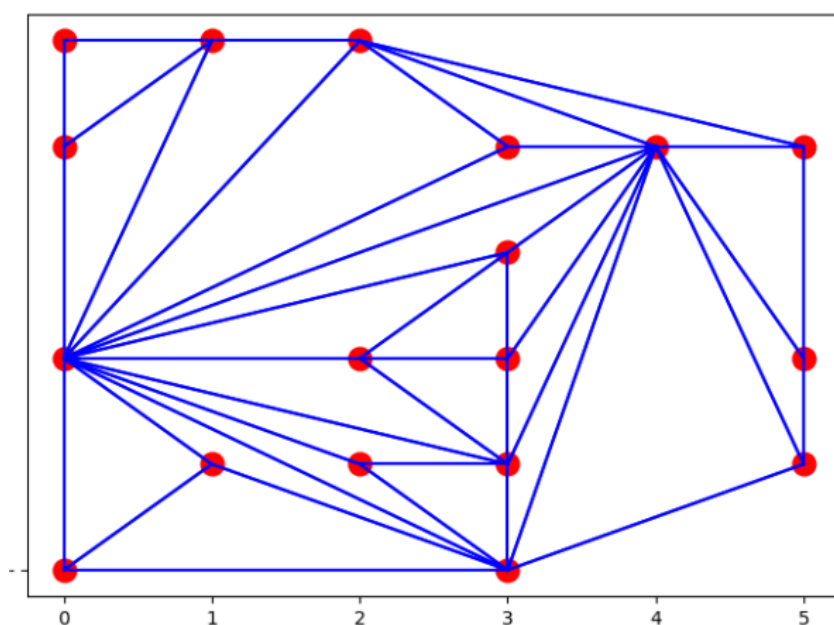
Spis treści

1. Wstęp.....	1
2. Zbadanie wpływu heurystyk na liczbę przeszukiwanych stanów.	2
2.1 Sprawdzenie wpływu heurystyki LCV.....	2
2.2 Sprawdzenie wpływu heurystyki MRV.....	4
3. Zbadanie metod sprawdzania wprzód oraz z nawrotami.....	6

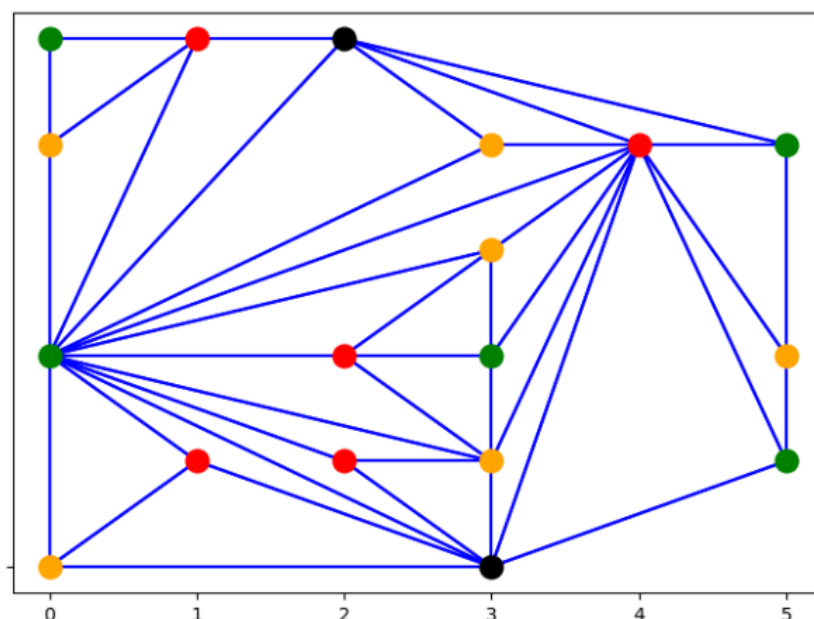
1. Wstęp

Zadanie polegało na napisaniu algorytmów rozwiązujących problemy spełniania ograniczeń (CSP). Jednym z problemów do rozwiązania w ramach listy był problem kolorowania mapy. Dla zadanej mapy posiadającej zbiór punktów (czyli regionów) oraz połączeń między nimi, należy pokolorować ją w taki sposób, żeby żaden sąsiadujący obszar nie był taki sam.

Przykład niepokolorowanej mapy:



Przykładowe rozwiązanie:



Zaimplementowałem następujące rozwiązania: Backtracking, Filtry Backtrackingu: Forward Checking oraz AC-3, Heurystyki: MRV (zmienna), LCV (wartość). W następnej części sprawozdania przedstawię ich wpływ na szybkość przeszukiwania drzewa oraz ilość odwiedzanych węzłów.

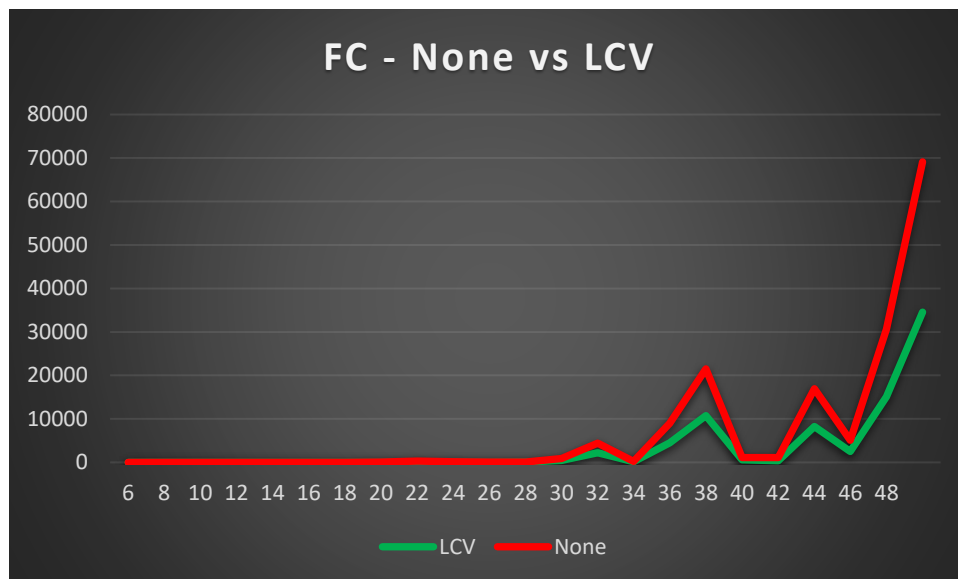
2. Zbadanie wpływu heurystyk na liczbę przeszukiwanych stanów.

Aby prawidłowo wykonać badanie wpływu heurystyk w moim przypadku MRV oraz LCV na czas potrzebny do rozwiązania zadania oraz ilość odwiedzanych węzłów, przeprowadziłem szereg doświadczeń. Wycinek tabeli poniżej obrazuje średnie otrzymane wartości:

Alg\Points	30	32	34	36	38	40
FC / None	2216,6	113,6	4509,4	10763,4	557,6	732,2
FC / MRV	33,8	33	35	38,2	39	41
FC / LCV	2212,4	114,8	4495,8	10726,8	557,2	383
FC / Both	33,8	33	35	38,2	39	41
AC3 / None	41,6	33	35,6	112,6	39,4	41,2
AC3 / MRV	32,6	33	35	37,4	39	41
AC3 / LCV	41,6	33,8	35,6	112,4	39	41
AC3 / Both	32,6	33	35	37,4	39	41

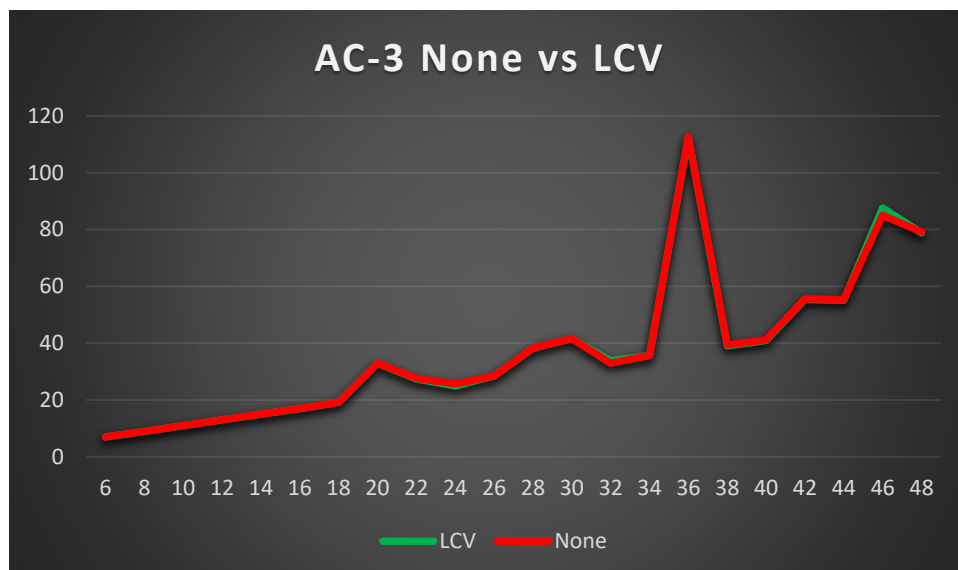
2.1 Sprawdzenie wpływu heurystyki LCV

Najpierw przeanalizujemy wpływ heurystyki LCV na ilość odwiedzonych węzłów, gdy używamy algorytmu Forward-Checking.



Wniosek: LCV zmniejsza ilość odwiedzanych węzłów przez Forward Checking, jednak widoczny wpływ widać dopiero przy większej ilości punktów (regionów) do pokolorowania. Do 30 punktów na mapie nie było widoczne żadne przyspieszenie.

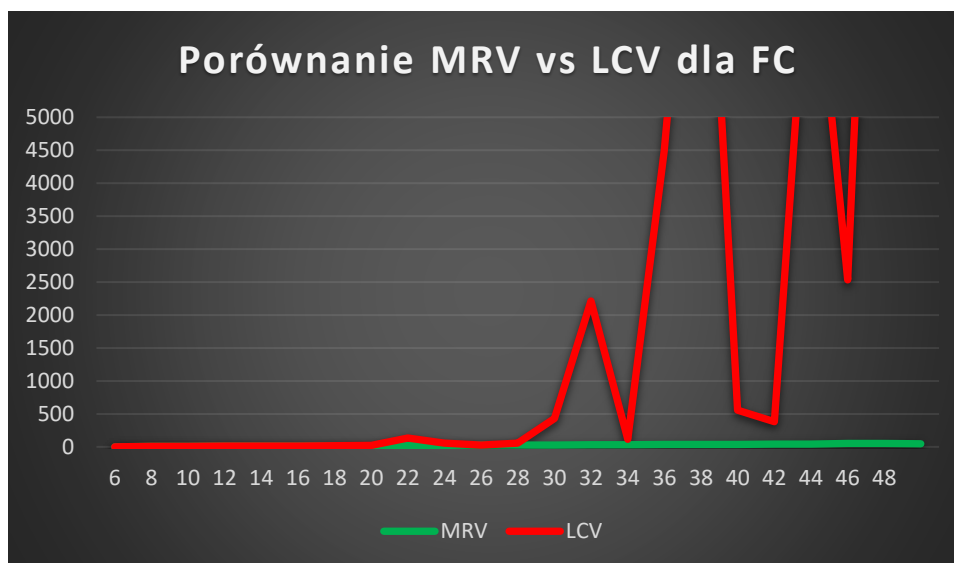
Teraz zbadajmy wpływ LCV na ilość odwiedzanych węzłów gdy używamy AC-3:



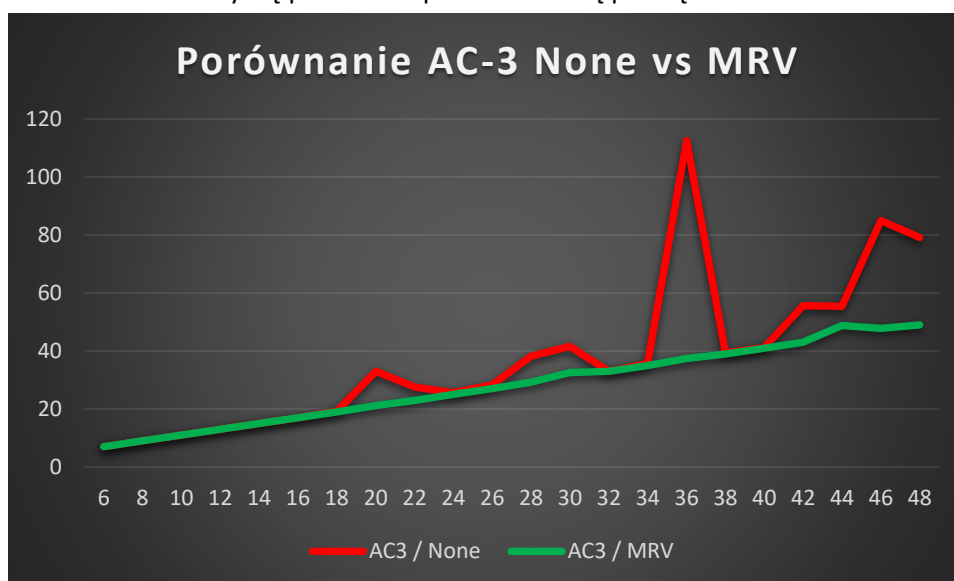
Wniosek: łatwo zauważyć, że w przypadku AC-3, heurystyka LCV nie wpływa na ilość odwiedzanych węzłów. Czasem może ją minimalnie zmniejszyć a czasem lekko pogorszyć. Sam algorytm LCV nie powinien być tutaj mocnym ulepszeniem.

2.2 Sprawdzenie wpływu heurystyki MRV

Z wyników umieszczonych w tabelce łatwo pokazać, że MRV znacząco zmniejsza ilość odwiedzanych węzłów zarówno dla Forward-Checkingu jak i AC-3.



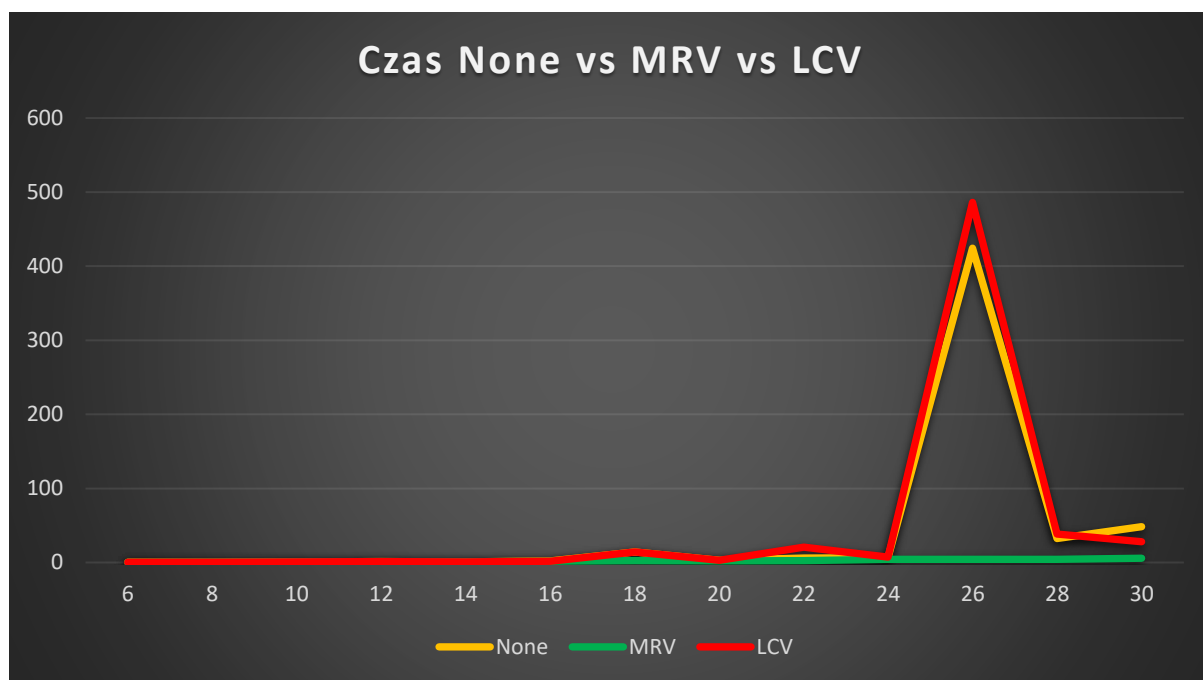
Widać, że w przeciwieństwie do LCV, MRV powoduje, że rozwiązania zawsze są bardzo dobre. Można by się pokusić o opisanie ich linią prostą.



MRV zapewniło nam również rozwiązanie, które odwiedziło znacznie mniejszą ilość węzłów nawet dla algorytmu AC-3.

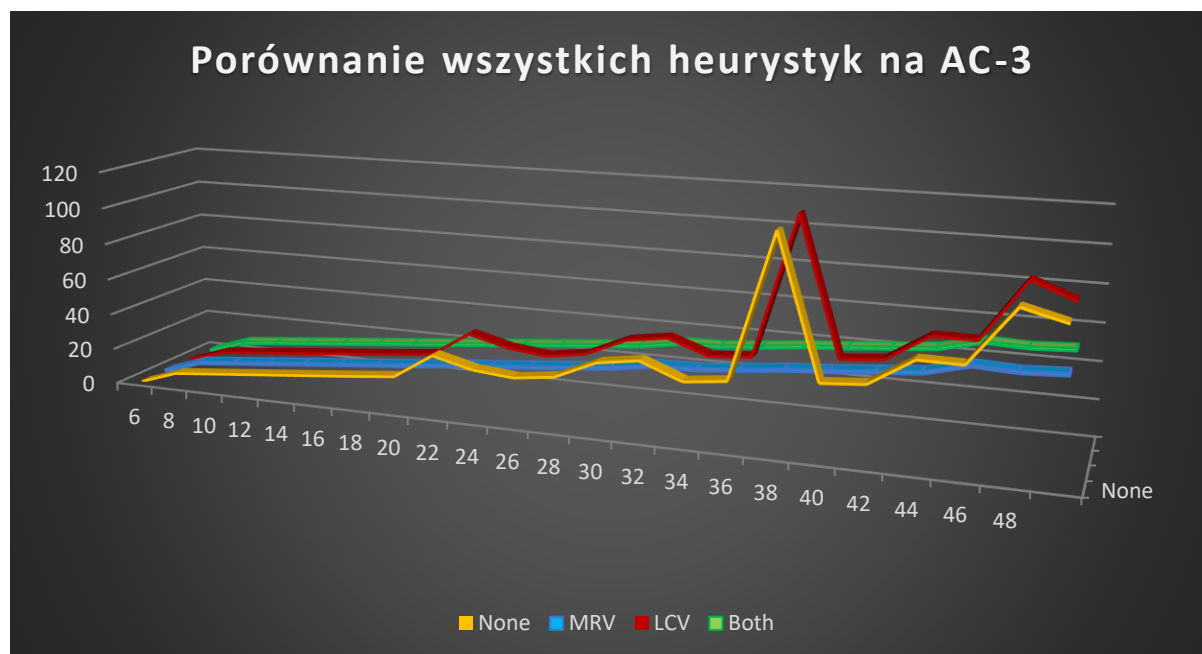
Wniosek: MRV znacznie zmniejsza ilość odwiedzanych węzłów zarówno dla F-C jak i AC-3.

Sprawdźmy jeszcze jaki wpływ heurystyki mają na czas szukania rozwiązania.



Łatwo zauważyć, że MRV wypada tutaj najlepiej. LCV wypada bardzo podobnie do samej czystej metody bez żadnych heurystyk. Nie można tutaj powiedzieć, czy wydłuża czy skraca ten czas, ponieważ wykresy te się przeplatają. Można tylko potwierdzić, że MRV wypada w tym aspekcie najlepiej.

Na sam koniec umieszczam wykres obrazujący ilość odwiedzanych węzłów dla wszystkich kombinacji używając AC-3 (Brak orderingu, Samo MRV, Samo LCV, Obie na raz).



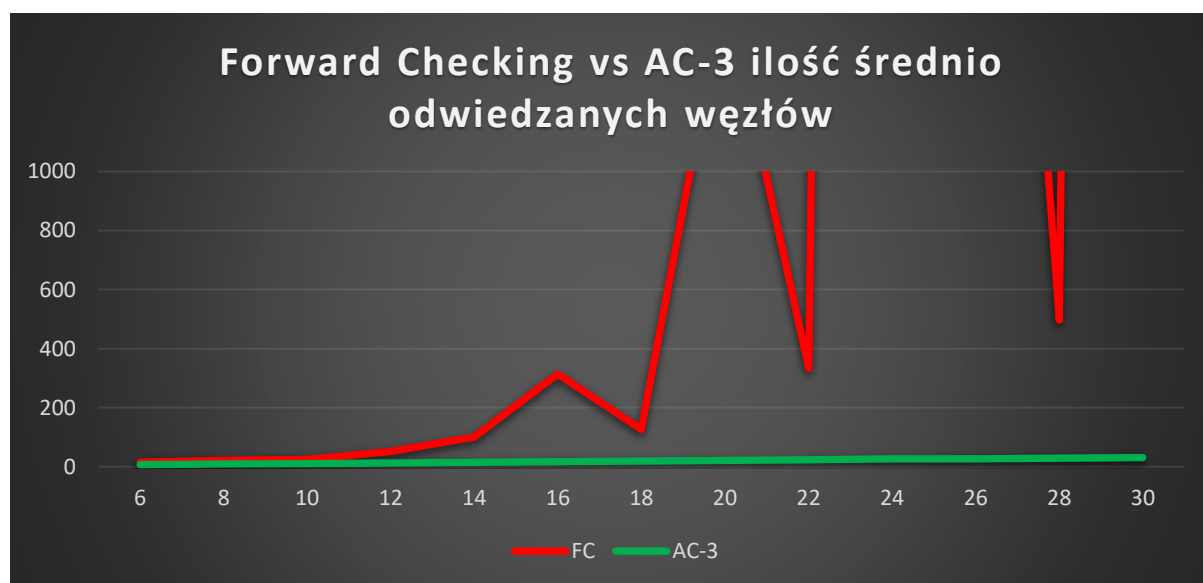
3. Zbadanie metod sprawdzania wprzód oraz z nawrotami

Na samym początku umieszczam przykład wycinka tabelki z wynikami z przeprowadzonych przeze mnie badań.

Alg\Points	6	8	10	12	14
BC	15,4	21,8	25	51,4	100
BC time	0	0,6004	1,1996	3,1998	6,0018
FC	15,4	21,8	25	51,4	100
FC time	0,3994	0,8	1,0245	1,1988	2,0008
FC Heur	7	9	11	15,6	23
FC Heur time	0	0,599	1,194	1,2008	1,798
AC3	7	9	11	13	15
AC3 time	0,8004	1,2	1,92718	2,5994	4,2006
AC3 Heur	7	9	11	13	19,4
AC3 Heur time	0,5998	1,1996	1,8008	2,6	4,2002

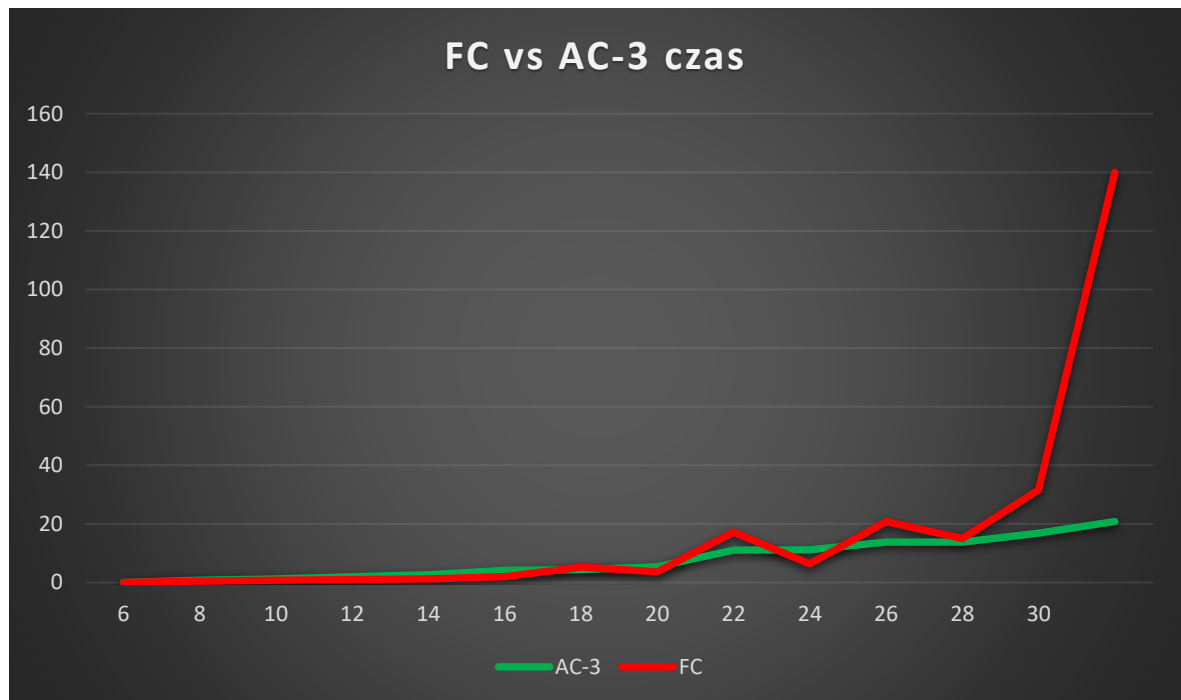
Przejdźmy zatem do analizy.

Na początku porównajmy FC oraz AC-3 pod względem ilości średnio odwiedzanych węzłów.



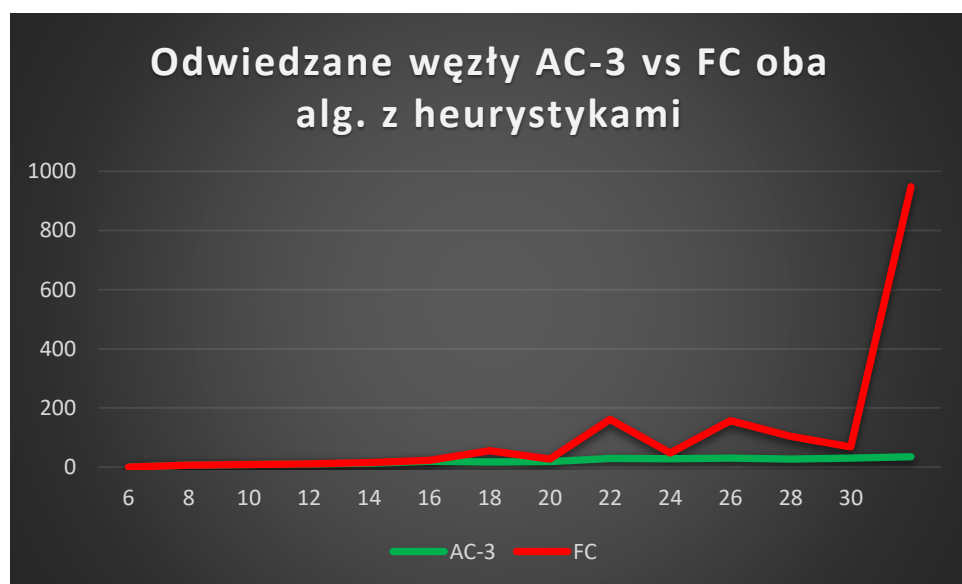
Wnioski: Z wykresu łatwo odczytać, że AC-3 wypada zdecydowanie lepiej w tym aspekcie. AC-3 jest pewnego rodzaju rozwinięciem i modyfikacją/ulepszeniem Forward Checkingu więc nie można tutaj było się spodziewać innego wyniku.

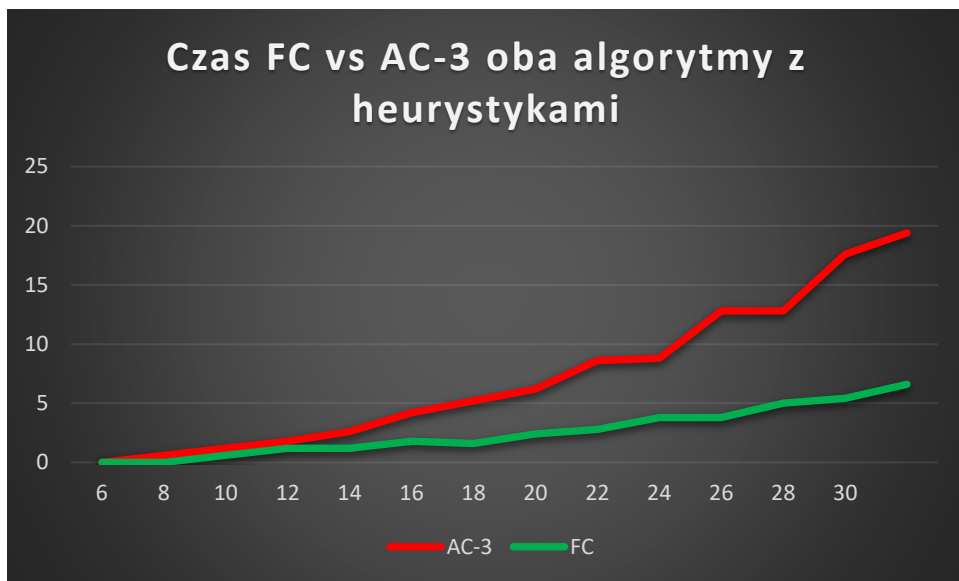
Jak jednak z czasem?



Tutaj widzimy już, że Forward Checking nawiązuje walkę z AC-3. Wykresy się przeplatają miejscami co oznacza, że dla niektórych węzłów Forward Checking znalazł rozwiązanie szybciej niż AC-3.

A jak to będzie wyglądać gdy do obu algorytmów dodamy heurystyki? W końcu to AC-3 z dwoma heurystykami wypadało zawsze najlepiej pod względem odwiedzanych wierzchołków.





Wnioski: Widzimy, że mimo, że AC-3 zdecydowanie przeważa jeżeli chodzi o ilość odwiedzanych węzłów to mimo to ilość operacji, które musi po drodze wykonać wpływają na to, że algorytm zwyczajnie zaczął przegrywać z Forward Checking z heurystykami. Pytanie zatem brzmi, Czy nie powinniśmy czasem stosować FC z heurystykami zamiast AC-3? To zależy jakie zadanie rozpatrujemy oraz jaka jest ilość zmiennych.

Na koniec umieściłem porównanie tych metod z Backtrackingiem. Wyraźnie widać, dlaczego implementowaliśmy te kolejne metody.

