Sierakowski Adam, 246662
19.04.2021, poniedziałek 17:05

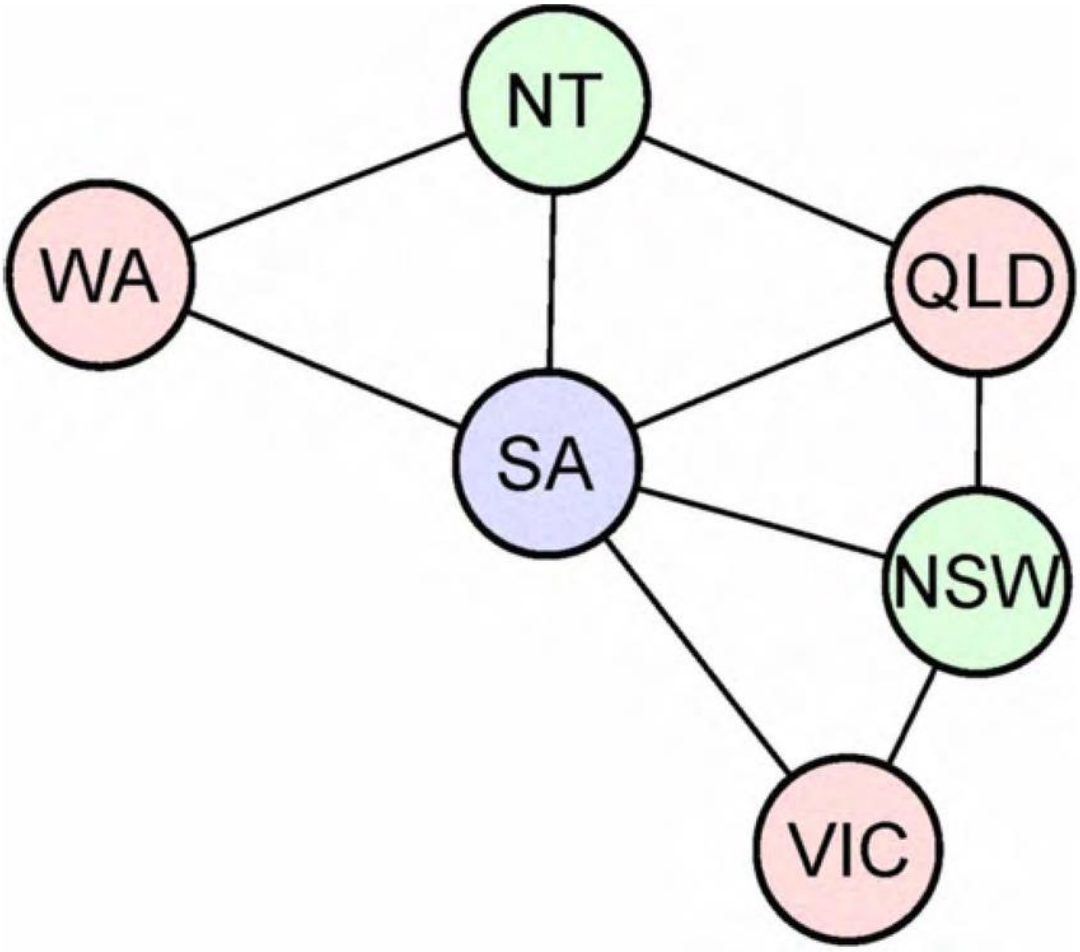
Problem CSP
Badania działania heurystyk oraz algorytmów

Spis treści

1. Przedstawiony problem

Omawianym problemem jest **CSP** z konkretnym przykładem w postaci kolorowania regionów.

1.1. Ograniczenia problemu



Rysunek 1 Źródło https://www.researchgate.net/figure/Map-coloring-as-constraint-graph-One-possible-solution-to-the-CSP-is-indicated-by-the_fig1_221534479

W problemie kolorowania map mamy następujące zasady:

- Sąsiadujące regiony nie mogą posiadać tego samego koloru (**WA** nie mógłby być koloru zielonego, ponieważ **NT** już taki kolor posiada i ograniczenie zostało by naruszone)
- Jest ograniczona liczba kolorów jakie możemy użyć.

1.2. Wykorzystywane algorytmy

- **Backtracking** – algorytm z nawrotami.
- **Forward checking** – algorytm z sprawdzeniem w przód.
- **AC3 –** algorytm utrzymania spójności łukowej, ze względu na swoje działanie, **AC3** w tym problemie, akurat nie znajdzie zastosowania, więc nie będzie brany pod uwagę podczas badań.

1.3. Wykorzystywane heurystyki

- **MRV** – heurystyka służąca do wyboru zmiennej z najmniejszą liczbą możliwych do wyboru wartości
- **LCV** – heurystyka służąca do wyboru wartości zmiennej, będzie wybierać wartość, która jak najmniej ograniczy jego sąsiadów

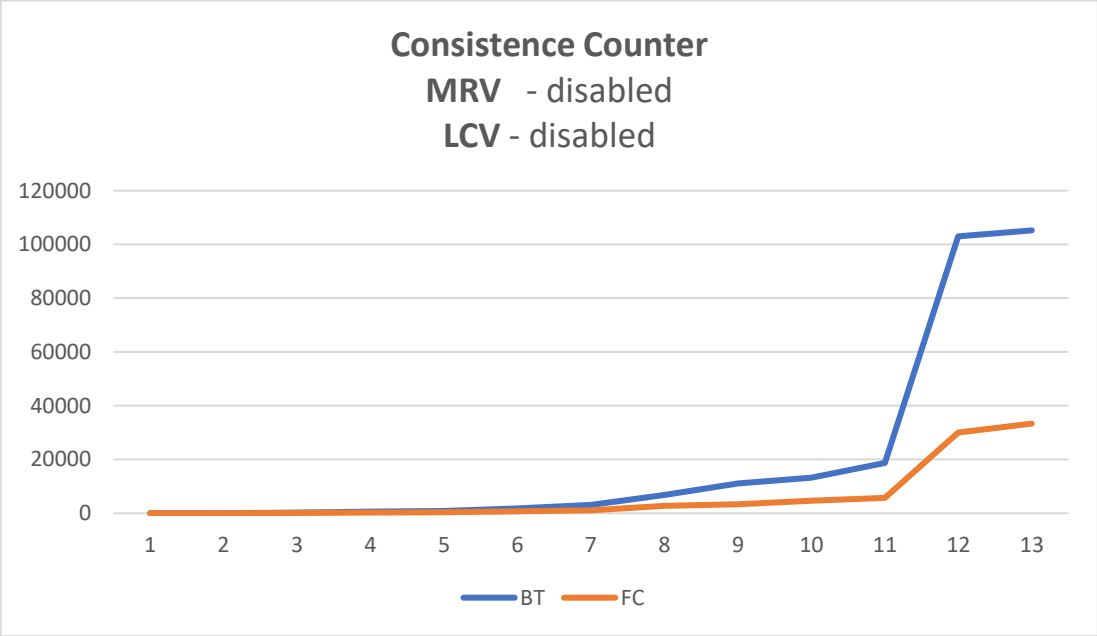
2. Porównania algorytmów wraz z zastosowanymi heurystykami – Problem kolorowania map (4 kolory)

2.1. Backtracking vs Forward Checking, MRV – disabled, LCV – disabled

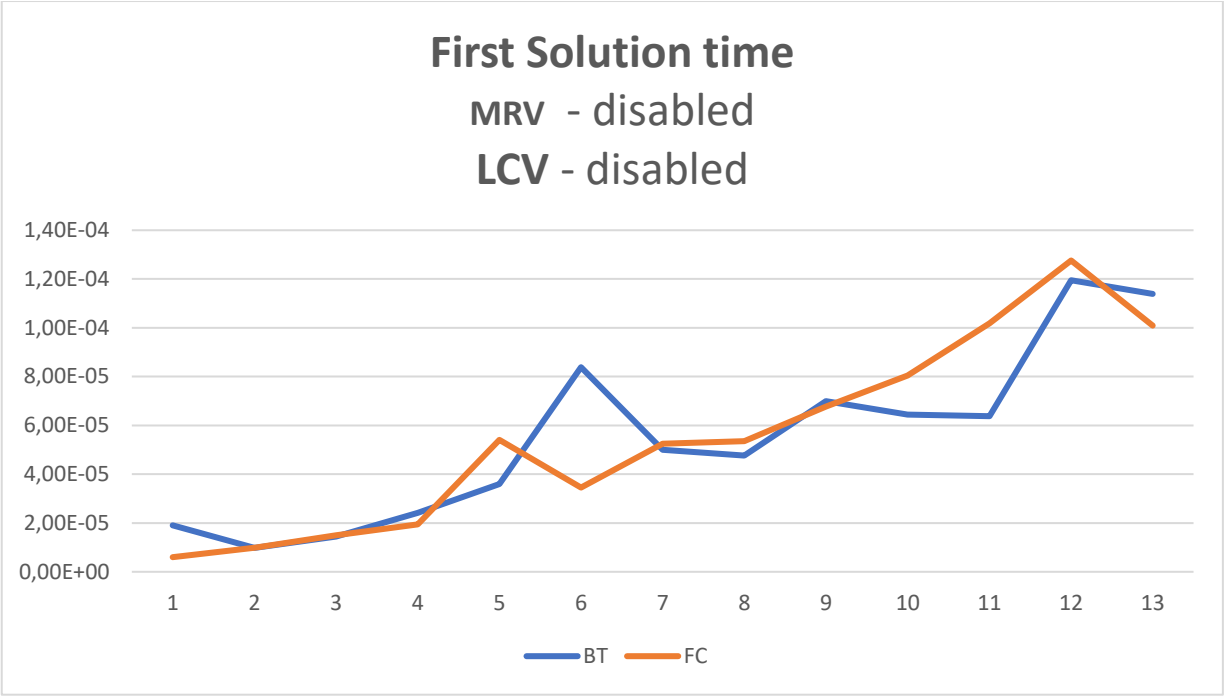
Consistence Counter (Nodes)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	20	68	228	660	836	1812	3140	6804	11124	13188	18708	102996	105204
FC	16	40	104	260	400	740	1072	2756	3404	4640	5732	29972	33356

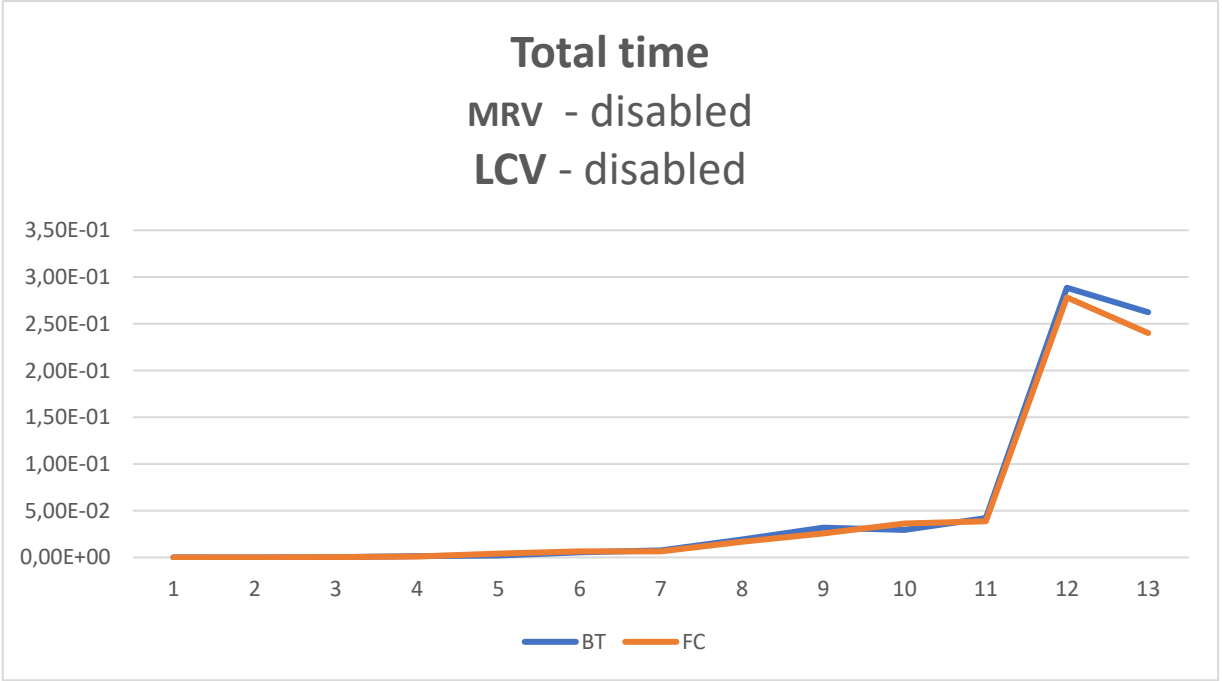
First Solution Time (s)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	1,91E-05	9,80E-06	1,45E-05	2,41E-05	3,60E-05	8,39E-05	5,00E-05	4,77E-05	6,99E-05	6,44E-05	6,38E-05	0,0001195	0,0001139
FC	6,00E-06	9,80E-06	1,50E-05	1,95E-05	5,41E-05	3,45E-05	5,25E-05	5,35E-05	6,77E-05	8,05E-05	0,0001018	0,0001276	0,0001009

Total Time (s)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	4,71E-05	0,0001259	0,0004997	0,001522	0,0019223	0,0055743	0,0074752	0,0191411	0,0318383	0,0295728	0,0421744	0,2885045	0,2621846
FC	3,62E-05	0,0001169	0,000374	0,0010477	0,0041751	0,00641	0,0063989	0,0168757	0,0256426	0,0363979	0,0385784	0,2781607	0,2401599



W bezpośrednim starciu możemy zauważyć przewagę algorytmu **Forward Checking,** jednak w całkowitym czasie, nie widzimy aż takiej różnicy, przyczynę opisałem poniżej w punkcie **2.2.**


First Solution time
MRV - disabled
LCV - disabled
1,40E-04
1,20E-04
1,00E-04
8,00E-05
6,00E-05
4,00E-05
2,00E-05
0,00E+00
1 2 3 4 5 6 7 8 9 10 11 12 13
BT
FC

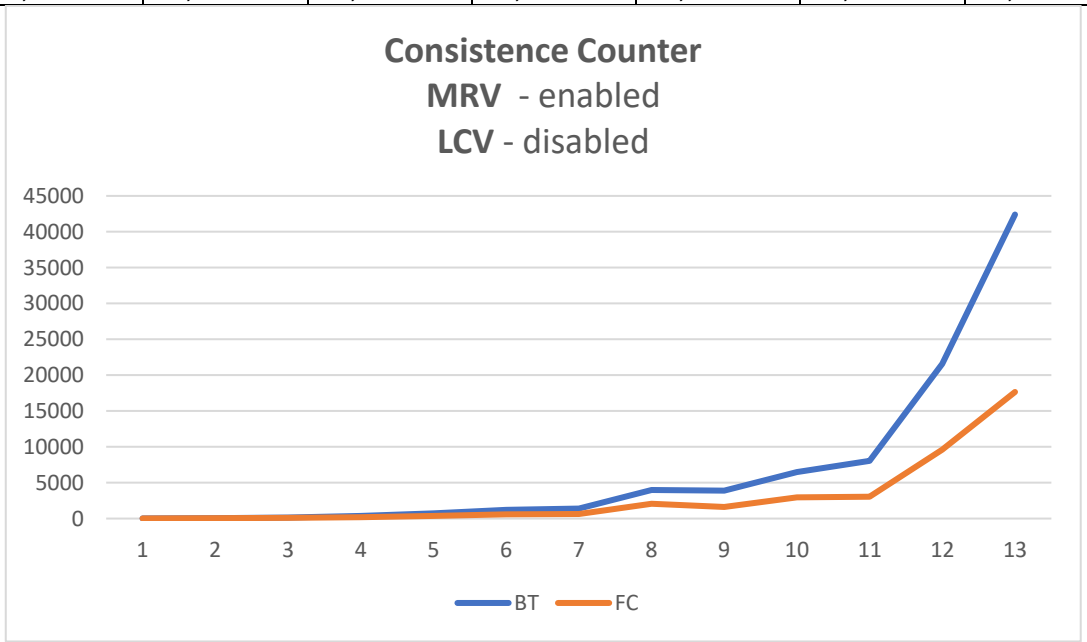

Total time
MRV - disabled
LCV - disabled
3,50E-01
3,00E-01
2,50E-01
2,00E-01
1,50E-01
1,00E-01
5,00E-02
0,00E+00
1 2 3 4 5 6 7 8 9 10 11 12 13
BT
FC

2.2. Backtracking vs Forward Checking, MRV – enabled, LCV - disabled

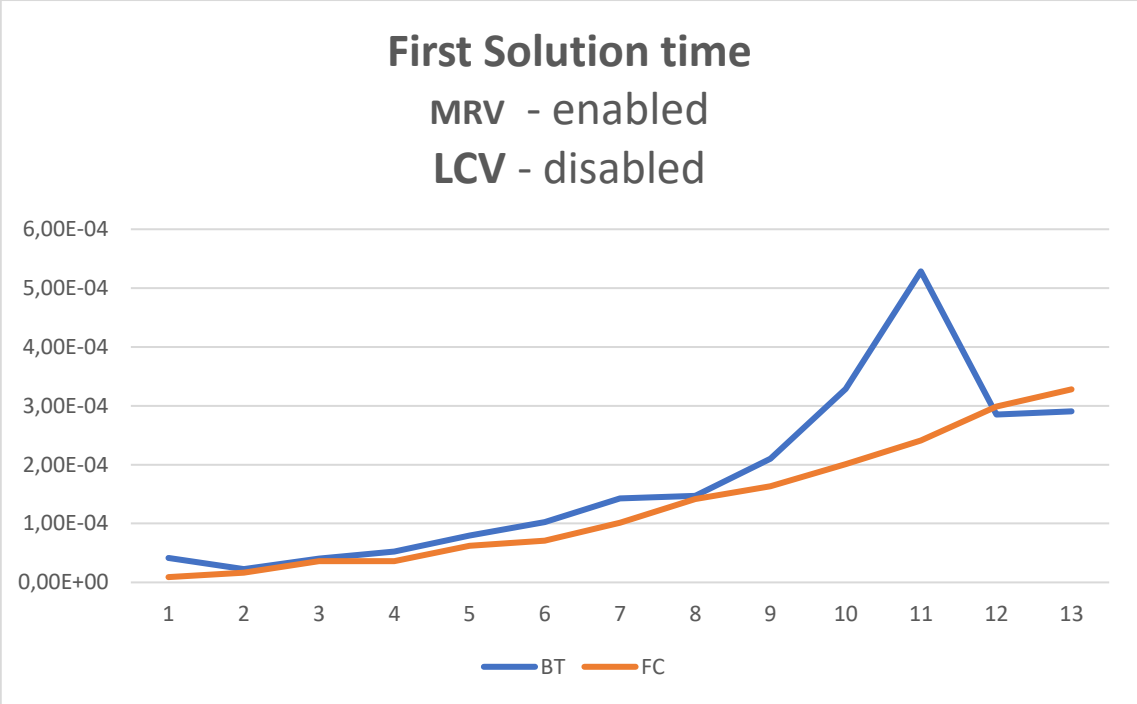
Consistence Counter (Nodes)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	20	68	164	356	740	1220	1412	4004	3908	6500	8036	21572	42404
FC	16	40	88	184	376	592	640	2056	1600	2968	3064	9616	17656

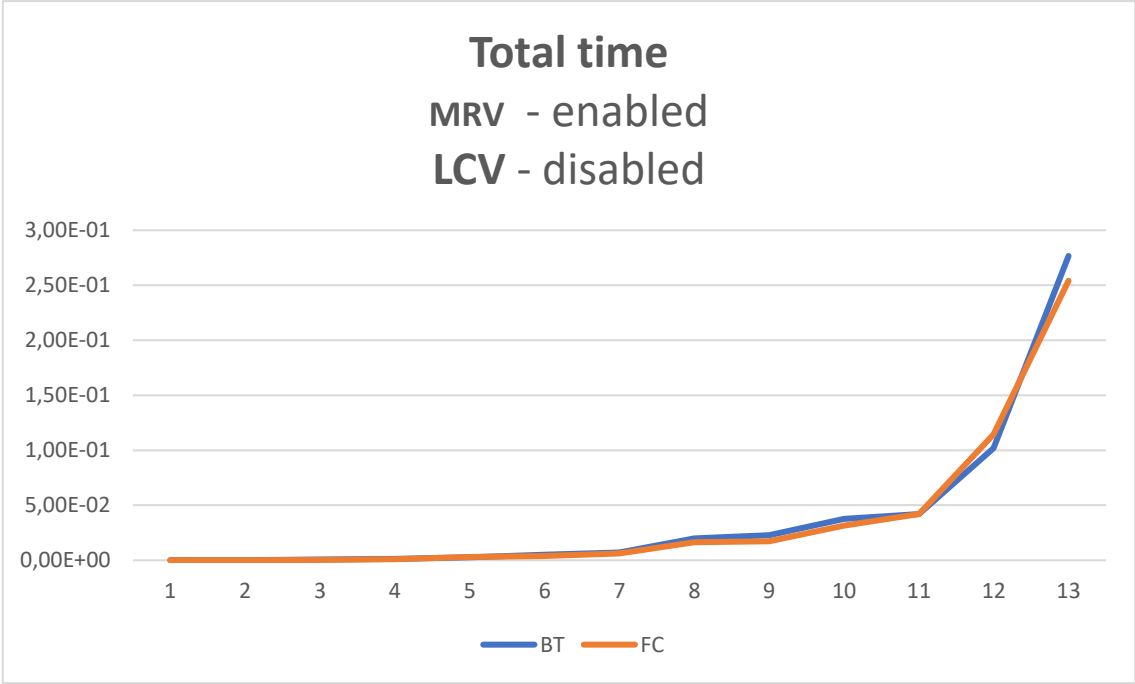
First Solution Time (s)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	4,14E-05	2,26E-05	4,05E-05	5,25E-05	7,94E-05	0,0001022	0,0001428	0,0001473	0,00021	0,0003285	0,0005284	0,0002849	0,0002905
FC	9,00E-06	1,64E-05	3,60E-05	3,60E-05	6,23E-05	7,10E-05	0,0001014	0,0001416	0,0001631	0,0002008	0,0002411	0,0002986	0,000328

Total Time (s)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	9,32E-05	0,0001629	0,0005099	0,0011799	0,0027891	0,0050323	0,0069929	0,0199723	0,0228671	0,0375439	0,0419859	0,1018339	0,2766268
FC	4,19E-05	0,0001551	0,000392	0,0008934	0,0028957	0,0037284	0,0060453	0,0163136	0,0173651	0,0314874	0,0420379	0,1143533	0,2540961



Wpływ **MRV** jest bardzo zauważalny, znacznie mniejsza liczba odwiedzonych zmiennych, lecz prędkość działania wcale nie jest lepsza, spowodowane jest to najpewniej nieoptymalnym zaimplementowaniem heurystyk. W **First Solution time** możemy zauważyć lekki chaos, ponieważ **BT** tam wyprzedził **FC**, jednak jest to dosyć losowa kategoria, w **Total time** czas jest zbliżony ze względu na moje nie dokońca poprawne zaimplementowanie **FC**, który tak naprawdę jest tutaj lepszą wersją **BT.**


First Solution time
MRV - enabled
LCV - disabled
6,00E-04
5,00E-04
4,00E-04
3,00E-04
2,00E-04
1,00E-04
0,00E+00
1 2 3 4 5 6 7 8 9 10 11 12 13
BT
FC

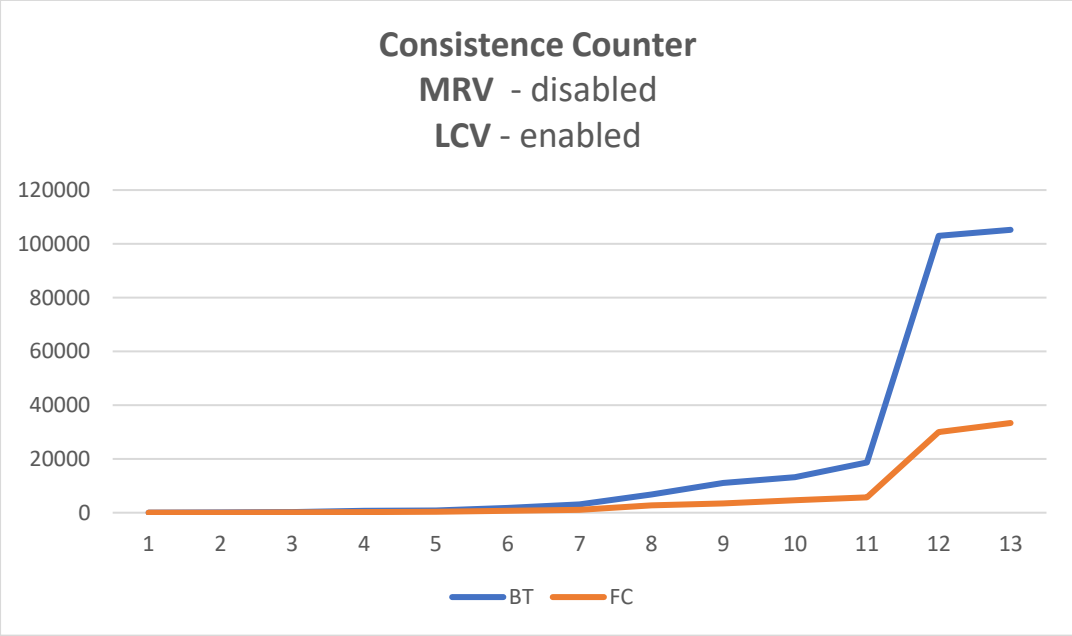

Total time
MRV - enabled
LCV - disabled
3,00E-01
2,50E-01
2,00E-01
1,50E-01
1,00E-01
5,00E-02
0,00E+00
1 2 3 4 5 6 7 8 9 10 11 12 13
BT
FC

2.3. Backtracking vs Forward Checking, MRV – disabled, LCV – enabled

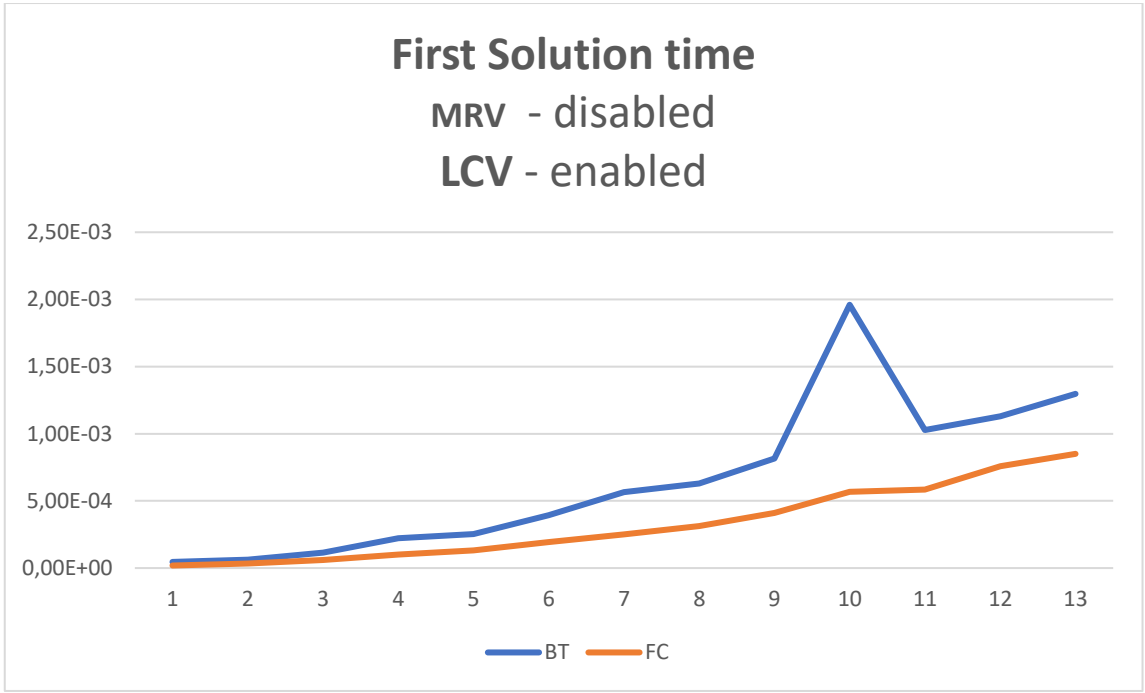
Consistence Counter (Nodes)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	20	68	228	660	836	1812	3140	6804	11124	13188	18708	102996	105204
FC	16	40	104	260	400	740	1072	2756	3404	4640	5732	29972	33356

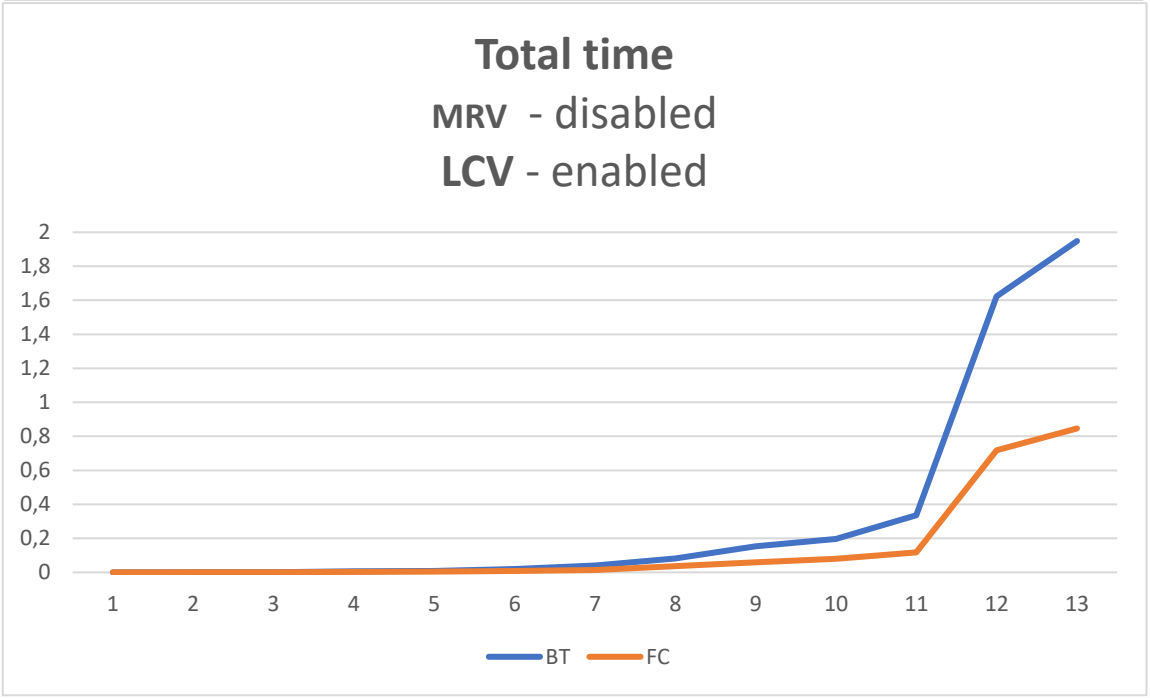
First Solution Time (s)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	4,58E-05	6,29E-05	0,0001148	0,0002216	0,0002531	0,000393	0,0005663	0,000629	0,0008158	0,0019599	0,0010271	0,0011291	0,001296
FC	1,89E-05	3,37E-05	5,96E-05	9,99E-05	0,0001318	0,0001934	0,0002518	0,0003128	0,0004095	0,0005677	0,0005848	0,0007574	0,0008502

Total Time (s)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	0,0001131	0,0003907	0,0016891	0,0056574	0,0083151	0,0201157	0,040416	0,0824295	0,1534596	0,1968417	0,335925	1,6242249	1,9483486
FC	5,75E-05	0,0002095	0,0006918	0,0020942	0,0039462	0,0087335	0,0139585	0,0359605	0,0596059	0,0803864	0,1178038	0,7188493	0,8463182



Tutaj możemy zauważyć, że **LCV**, również nie działa w praktyce najlepiej ze względu na bardzo kosztowny sposób, w jaki został zaimplementowany.


First Solution time
MRV - disabled
LCV - enabled
2,50E-03
2,00E-03
1,50E-03
1,00E-03
5,00E-04
0,00E+00
1 2 3 4 5 6 7 8 9 10 11 12 13
BT
FC

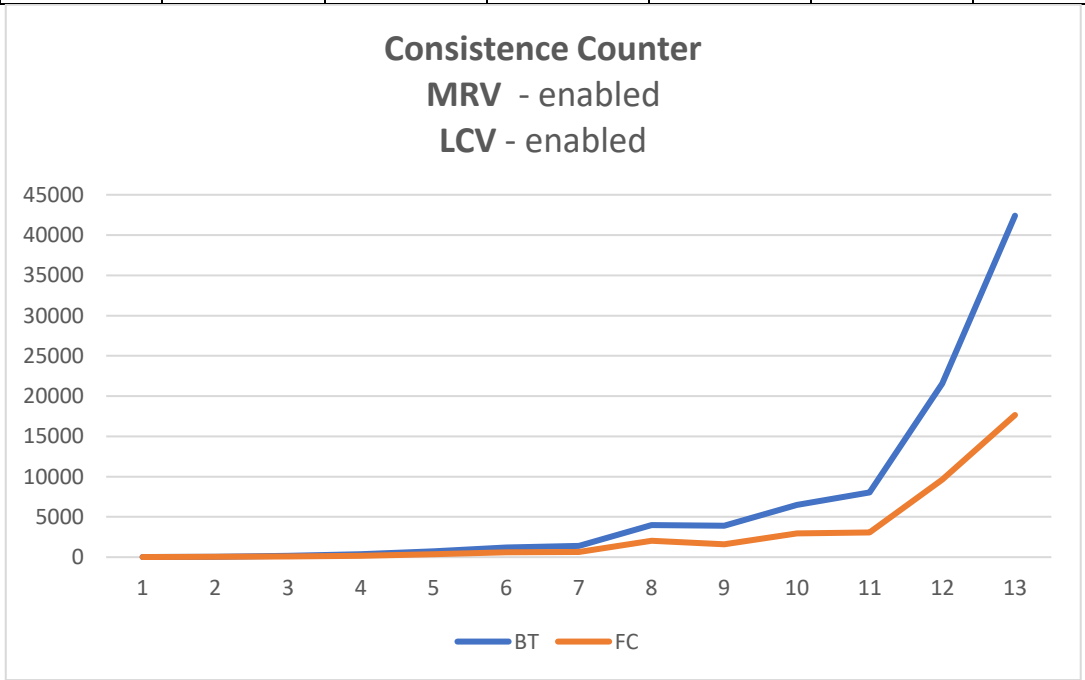

Total time
MRV - disabled
LCV - enabled
2
1,8
1,6
1,4
1,2
1
0,8
0,6
0,4
0,2
0
1 2 3 4 5 6 7 8 9 10 11 12 13
BT
FC

2.4. Backtracking vs Forward Checking, MRV – enabled, LCV - enabled

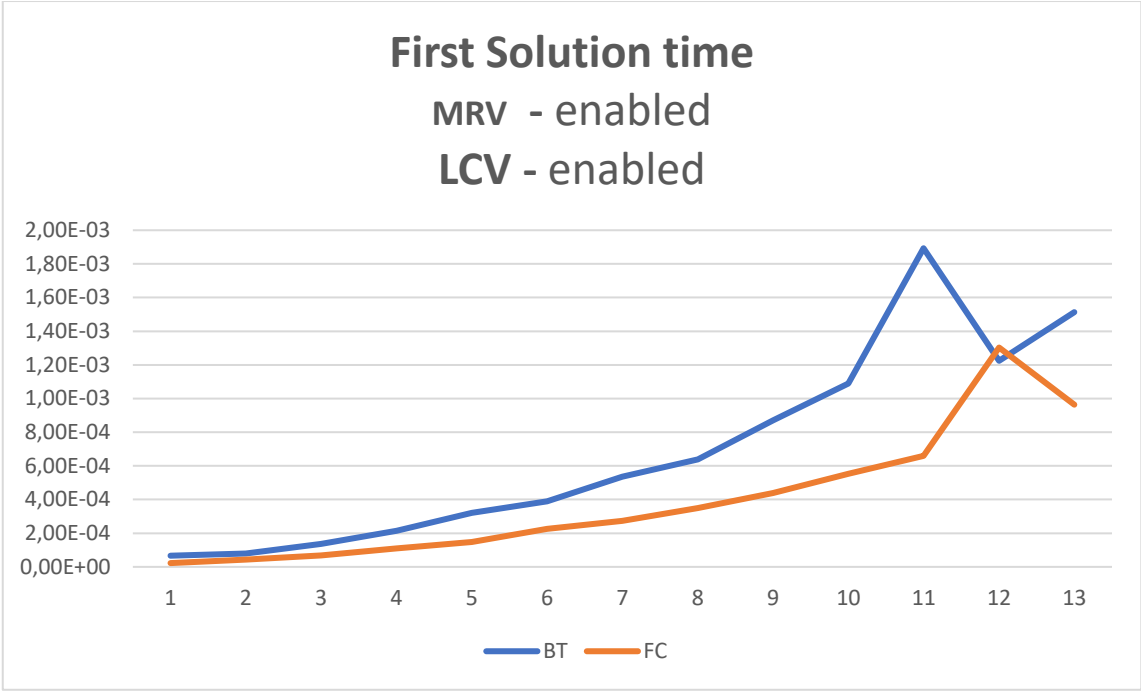
Consistence Counter (Nodes)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	20	68	164	356	740	1220	1412	4004	3908	6500	8036	21572	42404
FC	16	40	88	184	376	592	640	2056	1600	2968	3064	9616	17656

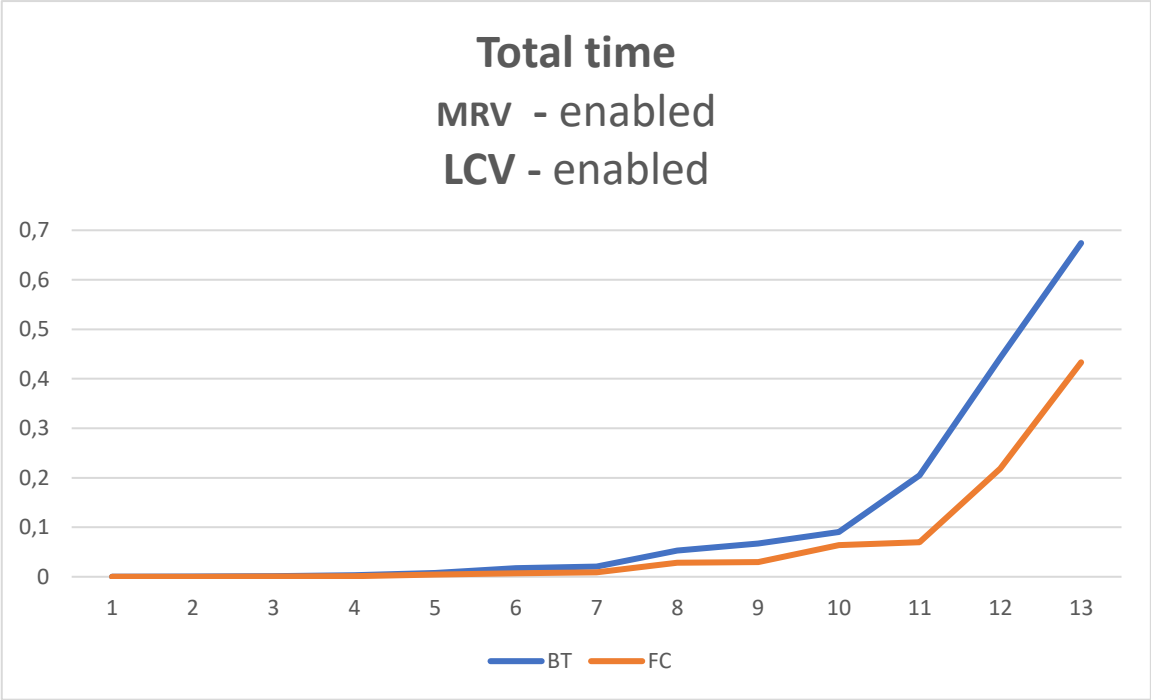
First Solution Time (s)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	6,52E-05	7,15E-05	0,0001367	0,0001956	0,0002984	0,0003684	0,000516	0,0006137	0,0008812	0,00236	0,0018343	0,0012356	0,0014406
FC	2,17E-05	3,97E-05	6,87E-05	0,0001023	0,000205	0,0002004	0,0002687	0,000341	0,0004338	0,0005544	0,0006525	0,0014868	0,000989

Total Time (s)													
n	2	3	4	5	6	7	8	9	10	11	12	13	14
BT	0,0001312	0,0004342	0,0013095	0,0031871	0,0074326	0,0137983	0,0193967	0,0595915	0,0626057	0,0980507	0,1467526	0,4007339	0,6678007
FC	6,39E-05	0,0002269	0,0006336	0,0015292	0,0045658	0,0068795	0,0085474	0,0284838	0,0296661	0,0782138	0,0696938	0,2190068	0,4349001



Połączenie obu heurystyk jest łączy silne strony ich obu, jednak wciąż samotny **MRV** będzie znacznie szybszy ze względu na nieoptymalną implementacje. Za to ma szybszy czas znalezienia pierwszego rozwiązania, więc **LCV** coś tutaj jedna zdziałał.


First Solution time
MRV - enabled
LCV - enabled
2,00E-03
1,80E-03
1,60E-03
1,40E-03
1,20E-03
1,00E-03
8,00E-04
6,00E-04
4,00E-04
2,00E-04
0,00E+00
1 2 3 4 5 6 7 8 9 10 11 12 13
BT FC


Total time
MRV - enabled
LCV - enabled
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
1 2 3 4 5 6 7 8 9 10 11 12 13
BT FC