Programme	Contre-exemple	Erreurs	LocFaults (≤ 3)	BugAssist
AbsMinusKO	$\{i = 0, j = 1\}$	17	{17}	{17}
AbsMinusKO2	$\{i = 0, j = 1\}$	11	{11},{17}	{17, 20, 16}
AbsMinusKO3	$\{i = 0, j = 1\}$	14	{20} {16(*Else*)},{14},{12}	{16, 20}
AbsMinusV2KO	$\{i = 0, j = 1\}$	13	{13}	{13}
AbsMinusV2KO2	$\{i = 0, j = 1\}$	11	{11},{13}	{13, 16, 12}
MinmaxKO	$\{in_1 = 2, in_2 = 1,$ $in_3 = 3\}$	19	{10},{19} {18(*If*)},{10}	{14, 19, 30}
MidKO	$\{a = 2, b = 1, c = 3\}$	19	{19} {14(*If*), 23(*If*), 26(*Else*)}	{14, 19, 30}
Maxmin6varKO	$\{a = 1, b = -4, c = -3,$ $d = -1, e = 0, f = -4\}$	27	{28} {15(*Else*)} {27(*If*)}	{15, 12, 27, 31, 166}
Maxmin6varKO2	$\{a = 1, b = -3, c = 0,$ $d = -2, e = -1, f = -2\}$	12	{65} {12(*Else*)}	{12, 64, 166}
Maxmin6varKO3	$\{a = 1, b = -3, c = 0,$ $d = -2, e = -1, f = -2\}$	12,15	{65} {12(*Else*), 15(*Else*)}	{12, 15, 64, 166}
Maxmin6varKO4	$\{a = 1, b = -3, c = -4,$ $d = -2, e = -1, f = -2\}$	12,15, 19	{116} {12(*Else*), 15(*If*), 19(*Else*)}	{12, 166}
TritypeKO	$\{i = 2, j = 3, k = 2\}$	54	{54} {26(*Else*)} {48(*Else*)},{30},{25} {29(*If*), 32(*Else*)} {53(*If*), 57(*Else*)},{30},{25}	{26, 27, 32, 33, 36, 48, 57, 68}
TritypeKO2	$\{i = 2, j = 2, k = 4\}$	53	{54} {21(*Else*)} {26(*If*)} {35(*Else*)},{27},{25} {53(*If*)},{27},{25} {29(*Else*), 57(*If*)} {32(*Else*), 44(*If*)}	{21, 26, 27, 29, 30, 32, 33, 35, 36, 53, 68}
TritypeKO2V2	$\{i = 1, j = 2, k = 1\}$	31	{50} {21(*Else*)} {26(*Else*)} {29(*If*)} {36(*Else*)},{31},{25} {49(*If*)},{31},{25} {33(*Else*), 45(*If*)}	{21, 26, 27, 29, 31, 33, 34, 36, 37, 49, 68}
TritypeKO3	$\{i = 1, j = 2, k = 1\}$	53	{54} {21(*Else*)} {29(*If*)} {35(*Else*)},{30},{25} {53(*If*)},{30},{25} {26(*Else*, 57(*If*} {32(*Else*), 44(*If*)}	{21, 26, 27, 29, 30, 32, 33, 35, 36, 48, 53, 68}
TritypeKO4	$\{i = 2, j = 3, k = 3\}$	45	{46} {45(*If*)},{33},{25} {26(*Else*, 32(*If*} {32(*If*), 35(*If*), 49(*Else*)} {32(*If*), 35(*If*), 53(*Else*)} {32(*If*), 35(*If*), 57(*Else*)}	{26, 27, 29, 30, 32, 33, 35, 45, 49, 68}
TritypeKO5	$\{i = 2, j = 3, k = 3\}$	32,45	{40} {26(*Else*)} {29(*Else*)} {32(*Else*), 45(*If*)} {35(*If*), 49(*Else*)},{25} {35(*If*), 53(*Else*)},{25} {35(*If*), 57(*Else*)},{25}	{26, 27, 29, 30, 32, 33, 35, 49, 68}
TritypeKO6	$\{i = 2, j = 3, k = 3\}$	32,33	{40} {26(*Else*)} {29(*Else*)} {35(*If*), 49(*Else*)},{25} {35(*If*), 53(*Else*)},{25} {35(*If*), 57(*Else*)},{25}	{26, 27, 29, 30, 32, 33 35, 49, 68}
TriPerimetreKO	$\{i = 2, j = 1, k = 2\}$	58	{58} {31(*If*)} {37(*Else*)},{32},{27}	{28, 29, 31, 32, 35, 37, 65, 72}
TriPerimetreKOV2	$\{i = 2, j = 3, k = 2\}$	34	{34},{60} {32(*If*)} {40(*Else*)},{33},{27}	{28, 32, 33, 34, 36, 38, 40, 41, 52, 55, 56, 60, 64, 67, 74}
TriPerimetreKO2	$\{i = 1, j = 1, k = 2\}$	57	{58} {22(*Else*)} {28(*If*)} {37(*Else*)},{29},{27} {57(*If*)},{29},{27} {31(*Else*), 61(*If*)} {34(*Else*), 48(*If*)}	{22, 28, 29, 31, 32, 34, 35, 37, 38, 48, 49, 52, 53, 57, 58, 61, 72}
TriPerimetreKO2V2	$\{i = 1, j = 2, k = 1\}$	33	{54} {22(*Else*)} {28(*Else*)} {31(*If*)} {38(*Else*)},{33},{27} {53(*If*)},{33},{27} {35(*Else*), 49(*If*)}	{22, 28, 72, 54, 53, 39, 33, 36, 38, 29, 31, 35, 49, 50}
TriPerimetreKO3	$\{i = 2, j = 1, k = 2\}$	57	{58} {22(*Else*)} {31(*If*)} {37(*Else*)},{32},{27} {57(*If*)},{32},{27} {28(*Else*), 61(*If*)} {34(*Else*), 48(*If*)}	{22, 28, 29, 31, 32, 34, 35, 37, 38, 49, 52, 57, 72}

TriPerimetreKO4	$\{i = 2, j = 3, k = 3\}$	49	{50} {34(*If*)} {37(*Else*)},{35},{27} {49(*If*)},{35},{27}	{37, 35, 72, 50, 49, 34, 28, 29, 32, 61, 65, 31}
TriPerimetreKO5	$\{i = 2, j = 2, k = 3\}$	34,49	{50} {34(*If*)} {37(*Else*)},{35},{27},{29} {49(*If*), 54(*Else*)},{35},{27},{29}	{37, 35, 32, 29, 72, 34, 31, 49, 53}
TriPerimetreKO6	$\{i = 2, j = 2, k = 3\}$	34,35	{50} {34(*If*)} {37(*Else*)},{35},{27},{29} {49(*If*), 53(*Else*)},{35},{27},{29}	{37, 72, 29, 32, 35, 34, 31, 49, 53}
TriMultPerimetreKO	$\{i = 2, j = 1, k = 2\}$	58	{58} {31(*If*)} {37(*Else*)}, {27}, {32}	{72, 37, 53, 49, 29, 35, 32, 31, 28, 65, 34, 62
TriMultPerimetreKO2	$\{i = 1, j = 1, k = 2\}$	57	{58} {22(*Else*)} {28(*If*)} {37(*Else*)},{27},{29} {57(*If*)},{29},{27} {31(*Else*), 61(*If*)} {34(*Else*), 48(*If*)}	{22, 37, 72, 58, 38, 52, 57, 49, 35, 32, 29, 28, 31, 65, 34}
TriMultPerimetreKO2V2	$\{i = 1, j = 2, k = 1\}$	32	{53} {21(*Else*)} {27(*Else*)} {30(*If*)} {37(*Else*)},{32},{26} {52(*If*)},{26},{32} {34(*Else*), 48(*If*)}	{21, 27, 71, 49, 52, 38, 53, 32, 35, 37, 28, 30, 34, 48}
TriMultPerimetreKO3	$\{i = 1, j = 2, k = 1\}$	56	{57} {21(*Else*)} {30(*If*)} {36(*Else*)},{26},{31} {56(*If*)},{31},{26} {27(*Else*), 60(*If*)} {33(*Else*), 47(*If*)}	{21, 71, 56, 51, 37, 57, 31, 28, 36, 34, 30, 27, 33, 47}
TriMultPerimetreKO4	$\{i = 2, j = 3, k = 3\}$	48	{49} {33(*If*)} {36(*Else*)},{26},{34} {48(*If*)},{34},{26}	{36, 34, 71, 49, 48, 33, 27, 28, 31, 53, 30, 60}
TriMultPerimetreKO5	$\{i = 2, j = 2, k = 3\}$	33,48	{49} {33(*If*)} {36(*Else*)},{34},{28},{26} {48(*If*), 52(*Else*)},{26},{34},{28}	{36, 34, 31, 28, 71, 49, 33, 30, 48, 52}
TriMultPerimetreKO6	$\{i = 2, j = 2, k = 3\}$	33,34	{48} 33(*If*) {36(*Else*)},{34},{26},{28} {47(*If*), 51(*Else*)},{26},{34},{28}	{36, 70, 48, 28, 31, 34, 33, 30, 47, 51}
HeronKO	$\{i = 3, j = 4, k = 3\}$	61	{61} {29(*If*)} {35(*Else*)}, {30}, {25}	{19, 61, 79, 35, 27, 33, 30, 42, 29, 26, 71, 32 48, 51, 54
HeronKO2	$\{i = 2, j = 2, k = 4\}$	59	{19}, {62} {26(*If*)} {35(*Else*)},{27},{25} {59(*If*)},{27},{25} {29(*Else*), 65(*If*)} {32(*Else*), 46(*If*)}	{62, 80, 19, 59, 36, 42, 33, 35, 30, 27, 26, 29, 68, 32, 48, 51, 54}
HeronV1	$\{i = 3, j = 4, k = 3\}$	61	{61} {29(*If*)} {35(*Else*)},{30},{25}	{79, 33, 30, 42, 35, 27, 61, 29, 26, 71, 32, 48, 36, 51}
HeronV2	$\{i = 2, j = 2, k = 4\}$	59	{62} {26(*If*)} {35(*Else*)},{25},{27} 59(*If*),{27},{25} {29(*Else*), 65(*If*)} {32(*Else*), 46(*If*)}	{62, 80, 59, 36, 42, 33, 35, 30, 27, 26, 29, 72, 32, 48, 51, 54}
HeronKO2V2	$\{i = 1, j = 2, k = 1\}$	31	{55} {26(*Else*)} {29(*If*)} {36(*Else*)},{25},{31} {52(*If*)},{31},{25} {33(*Else*), 47(*If*)}	{26, 19, 52, 80, 55, 43, 31, 34, 36, 27, 29, 33, 47, 49}
HeronKO3	$\{i = 1, j = 2, k = 1\}$	59	{62} {29(*If*)} {35(*Else*)},{25},{30} {59(*If*)},{30},{25} {26(*Else*), 65(*If*)} {32(*Else*), 46(*If*)}	{80, 42, 19, 59, 51, 62, 30, 27, 35, 33, 29, 26, 32, 46}
HeronKO4	$\{i = 2, j = 3, k = 3\}$	47	{49} {32(*If*)} {35(*Else*)},{33},{25} {47(*If*)},{33},{25}	{35, 33, 80, 49, 47, 32, 26, 19, 27, 30, 55, 29}
HeronKO5	$\{i = 2, j = 2, k = 3\}$	32,47	{49} {20(*Else*)} {32(*If*)} {35(*Else*)},{33},{25},{27} {47(*If*), 52(*Else*)},{25},{33},{27}	{20, 35, 33, 30, 27, 80, 49, 32, 29, 47, 52}
HeronKO6	$\{i = 2, j = 2, k = 3\}$	32,33	{48} {20(*Else*)} {32(*If*)} {35(*Else*)},{33},{27},{25} {46(*If*), 51(*Else*)},{27},{33},{25}	{20, 35, 79, 48, 27, 30, 33, 32, 29, 46, 51}

TABLE 1 – MCS identifiés par `LocFaults` [1] [2] pour des programmes sans boucles : résultats améliorés. Ce tableau présente aussi le résultat de `BugAssist` [3].

Programme	LocFaults									BugAssist	
	P	L								P	L
		$= 0$		≤ 1		≤ 2		≤ 3			
		V_1	V_2	V_1	V_2	V_1	V_2	V_1	V_2		
AbsMinusKO	0.719	0.024	0.021	0.024	0.026	0.021	0.028	0.027	0.033	0.01	0.02
AbsMinusKO2	0.726	0.035	0.042	0.029	0.031	0.037	0.04	0.034	0.036	0.02	0.06
AbsMinusKO3	0.708	0.02	0.026	0.032	0.069	0.092	0.045	0.059	0.05	0.02	0.07
AbsMinusV2KO	0.682	0.021	0.025	0.023	0.023	0.021	0.025	0.021	0.023	0.01	0.02
AbsMinusV2KO2	0.704	0.027	0.03	0.025	0.029	0.033	0.029	0.041	0.029	0.02	0.06
MinmaxKO	0.696	0.062	0.088	0.066	0.065	0.119	0.057	0.107	0.072	0.02	0.07
MidKO	0.695	0.027	0.022	0.023	0.029	0.021	0.022	0.028	0.037	0.02	0.10
Maxmin6varKO	0.856	0.032	0.028	0.04	0.052	0.041	0.05	0.061	0.052	0.07	1.50
Maxmin6varKO2	0.78	0.027	0.035	0.035	0.035	0.04	0.042	0.04	0.061	0.07	0.98
Maxmin6varKO3	0.791	0.027	0.037	0.03	0.036	0.046	0.045	0.049	0.056	0.07	1.79
Maxmin6varKO4	0.802	0.027	0.035	0.032	0.037	0.033	0.035	0.051	0.056	0.08	1.11
TritypeKO	0.735	0.028	0.024	0.071	0.081	0.161	0.163	0.19	0.151	0.03	0.40
TritypeKO2	0.757	0.031	0.029	0.135	0.116	0.159	0.146	0.177	0.153	0.02	0.69
TritypeKO2V2	0.743	0.011	0.014	0.061	0.062	0.129	0.096	0.275	0.102	0.03	0.80
TritypeKO3	0.735	0.025	0.024	0.1	0.11	0.169	0.148	0.234	0.18	0.03	0.77
TritypeKO4	0.739	0.028	0.028	0.069	0.058	0.085	0.063	0.188	0.17	0.02	0.37
TritypeKO5	0.74	0.022	0.022	0.042	0.034	0.161	0.157	0.153	0.158	0.02	0.39
TritypeKO6	0.752	0.023	0.028	0.038	0.039	0.182	0.163	0.187	0.146	0.03	0.34
TriPerimetreKO	0.787	0.028	0.027	0.071	0.061	0.099	0.074	0.219	0.092	0.02	0.98
TriPerimetreKOV2	0.751	0.086	0.068	0.177	0.185	0.186	0.162	0.243	0.19	0.04	1.78
TriPerimetreKO2	0.756	0.024	0.029	0.144	0.113	0.171	0.152	0.193	0.166	0.04	3.84
TriPerimetreKO2V2	0.722	0.151	0.041	0.182	0.166	0.13	0.136	0.157	0.164	0.03	2.18
TriPerimetreKO3	0.778	0.034	0.025	0.143	0.14	0.184	0.171	0.223	0.175	0.02	1.83
TriPerimetreKO4	0.786	0.023	0.025	0.117	0.127	0.122	0.092	0.172	0.122	0.04	1.12
TriPerimetreKO5	0.763	0.026	0.025	0.085	0.1	0.2	0.171	0.261	0.208	0.04	1.11
TriPerimetreKO6	0.75	0.029	0.028	0.105	0.078	0.247	0.167	0.232	0.203	0.04	0.85
TriMultPerimetreKO	0.723	0.058	0.054	0.13	0.137	0.145	0.147	0.168	0.148	0.04	3.23
TriMultPerimetreKO2	0.708	0.044	0.055	0.217	0.253	0.269	0.233	0.264	0.25	0.06	5.21
TriMultPerimetreKO2V2	0.728	0.055	0.053	0.231	0.239	0.25	0.276	0.227	0.246	0.06	3.97
TriMultPerimetreKO3	0.714	0.045	0.063	0.164	0.255	0.251	0.271	0.286	0.255	0.06	3.96
TriMultPerimetreKO4	0.724	0.048	0.058	0.228	0.208	0.196	0.15	0.162	0.142	0.04	2.78
TriMultPerimetreKO5	0.739	0.054	0.05	0.138	0.096	0.179	0.142	0.199	0.211	0.06	3.90
TriMultPerimetreKO6	0.722	0.056	0.054	0.133	0.132	0.218	0.2	0.24	0.229	0.05	2.65
HeronKO	0.793	0.13	0.119	0.23	0.221	0.224	0.231	0.212	0.238	0.06	6.78
HeronKO2	0.737	0.053	0.082	0.232	0.243	0.275	0.209	0.295	0.263	0.08	10.05
HeronV1	0.747	0.06	0.052	0.139	0.147	0.148	0.142	0.165	0.157	0.08	11.25
HeronV2	0.748	0.059	0.064	0.243	0.249	0.255	0.271	0.271	0.275	0.08	7.04
HeronKO2V2	0.757	0.126	0.124	0.298	0.337	0.317	0.268	0.199	0.33	0.08	5.84
HeronKO3	0.677	0.117	0.08	0.3	0.332	0.325	0.23	0.359	0.325	0.09	6.02
HeronKO4	0.743	0.046	0.047	0.144	0.149	0.152	0.217	0.174	0.196	0.08	4.52
HeronKO5	0.736	0.051	0.048	0.135	0.132	0.24	0.229	0.224	0.208	0.07	6.02
HeronKO6	0.754	0.051	0.055	0.139	0.14	0.22	0.225	0.251	0.252	0.08	4.58

TABLE 2 – Temps de calcul (en secondes). Les colonnes V_1 et V_2 correspondent respectivement aux résultats de `LocFaults` sans et avec l'usage du marquage des nœuds dans le CFG. Avec marquage des nœuds dans `LocFaults` : à une étape donnée, le nœud qui permet de détecter une déviation de correction minimale sera marqué par le cardinal de cette dernière ; pour qu'aux prochaines étapes, l'algorithme n'autorisera pas le balayage d'une liste d'adjacence de ce nœud. En ce qui concerne `LocFaults`, nous avons utilisé le solveur MIP de Cplex(`http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`) pour tous les programmes ; sauf les instances avec calcul non-linéaire, pour lesquels, nous avons employé le solveur IBM ILOG CP de Cplex(`http://www-01.ibm.com/software/commerce/optimization/cplex-cp-optimizer/`). Pour BugAssist, nous avons utilisé le solveur Max-SAT MSUnCore2 [4].

Références

[1] Bekkouche, Mohammed, Hélène Collavizza, and Michel Rueher. "Une approche CSP pour l'aideła localisation d'erreurs." arXiv preprint arXiv :1404.6567 (2014).

[2] Bekkouche, Mohammed, Hélène Collavizza, and Michel Rueher. "LocFaults : A new flow-driven and constraint-based error localization approach*." SAC'15, SVT track.

[3] Jose, Manu, and Rupak Majumdar. "Cause clue clauses : error localization using maximum satisfiability." ACM SIGPLAN Notices 46.6 (2011) : 437-446.

[4] Marques-Silva, Joao. "The msuncore maxsat solver." SAT 2009 competitive events booklet : preliminary version (2009) : 151.