

Risultati Sperimentali

Alessio Bardelli

Settembre 2020

In questo documento, sono riassunti i risultati sperimentali, generati tramite gli algoritmi che implementano le euristiche di Filippo e di Alessio; l'algoritmo che implementa l'euristica di Filippo migliorata, l'algoritmo enumerativo esatto e i test in cui si assegnano tutti i *don't care* rispettivamente a zero e a uno. Poiché l'algoritmo enumerativo esatto ha un elevato costo computazionale, i valori ottimi di autosimmetria non sempre saranno disponibili. Nella Tabella 1, a differenza delle Tabelle 2 e 3, è stato considerato ciascun output di una funzione come se fosse una funzione distinta. Di seguito si fornisce una breve descrizione dei dati presentati:

- *Funzione*: il nome della funzione presa in considerazione.
- n, m : rispettivamente il numero di input e output di una funzione.
- k : grado di autosimmetria medio per ciascun output di una funzione.
- k_1, k_2 : grado di autosimmetria rilevato per un output di una funzione, assegnando rispettivamente tutti i *don't care* a 0 e a 1.
- k_3 : grado di autosimmetria rilevato per un output di una funzione tramite l'euristica di Filippo.
- k_4 : grado di autosimmetria rilevato per un output di una funzione tramite l'euristica di Alessio.
- k_5 : grado di autosimmetria rilevato per un output di una funzione tramite l'euristica di Filippo migliorata.
- k_6 : grado di autosimmetria rilevato per un output di una funzione tramite l'algoritmo enumerativo esatto, quando disponibile.
- *Max*: massimo grado di autosimmetria rilevato per una funzione.
- *Test*: tempi di esecuzione espressi in millisecondi.
- -: l'esecuzione del test, dopo una soglia massima di 30 minuti, è stata abortita. Pertanto i risultati per quel test non sono disponibili.
- **Grassetto**: il test eseguito ha rilevato il grado di autosimmetria migliore per una certa funzione o per un suo specifico output.

Come si può osservare dalla Tabella 1, considerando ogni output come una funzione a se stante, l'euristica di Alessio rileva un grado di autosimmetria minore rispetto a quello rilevato tramite l'euristica di Filippo. Difatti il grado di autosimmetria rilevato con l'euristica di Alessio diminuisce del 3.29% rispetto a quello rilevato tramite l'euristica di Filippo. Continuando a osservare la tabella si evince che, l'euristica di Filippo migliorata presenta un aumento del grado di autosimmetria rilevato di circa lo 0.04% rispetto all'euristica di Filippo standard. Per il 36.1% delle funzioni testate non è disponibile il grado di autosimmetria ottimo, dato l'elevato costo computazionale dell'algoritmo enumerativo esatto. Infine, è interessante notare che, l'algoritmo enumerativo esatto ha rilevato un grado di autosimmetria maggiore rispetto a quello rilevato da tutti gli altri algoritmi solo per le funzioni *bench(4)*, *p1(2)*, *p1(7)*, *p1(16)*, *p3(2)*, *p3(7)*, *test4(18)*, a fronte delle 308 funzioni per cui è disponibile il grado di autosimmetria ottimo.

Per quanto riguarda la Tabella 2 e 3, si può vedere che l'euristica di Alessio risulta essere più efficiente rispetto all'euristica di Filippo. Difatti l'euristica di Alessio impiega il 14.9% di tempo in meno rispetto all'euristica di Filippo, per il calcolo del grado di autosimmetria sull'intero insieme di funzioni di test. Come già discusso precedentemente, l'algoritmo enumerativo esatto ha un elevato costo computazionale e, questo è il motivo per cui si ha un importante degrado delle prestazioni nel calcolo del grado di autosimmetria ottimo. Infatti l'algoritmo

enumerativo esatto impiega circa il 588% di tempo in più rispetto all’euristica di Alessio, per il calcolo del grado di autosimmetria sull’intero insieme di funzioni di test. Infine l’euristica di Filippo migliorata presenta un degrado delle prestazioni rispetto all’euristica di Alessio, in quanto impiega il 13.8% del tempo in più.

Table 1: Sono riportati i valori di autosimmetria rilevati degli algoritmi considerati su ciascun output delle funzioni.

Funzione (Output)	k_1	k_2	k_3	k_4	k_5	k_6
alu2(0)	6	0	7	7	7	7
alu2(1)	4	0	5	5	5	5
alu2(2)	2	0	3	3	3	3
alu2(3)	0	0	1	1	1	1
alu2(4)	1	1	1	1	1	1
alu2(5)	0	0	1	1	1	1
alu2(6)	3	2	3	3	3	3
alu2(7)	6	6	6	6	6	6
alu3(0)	6	0	6	6	6	6
alu3(1)	4	0	4	4	4	4
alu3(2)	2	0	2	2	2	2
alu3(3)	0	0	0	0	0	0
alu3(4)	1	1	1	1	1	1
alu3(5)	0	0	0	0	0	0
alu3(6)	3	2	3	3	3	3
alu3(7)	6	2	6	6	6	6
apla(0)	0	0	4	3	4	-
apla(1)	0	0	4	3	4	-
apla(2)	0	0	4	3	4	-
apla(3)	0	0	4	3	4	-
apla(4)	0	0	4	2	4	-
apla(5)	1	0	6	5	6	-
apla(6)	0	0	5	6	5	-
apla(7)	0	0	4	2	4	-
apla(8)	0	0	5	3	5	-
apla(9)	0	0	4	2	4	-
apla(10)	0	0	4	3	4	-
apla(11)	1	1	5	4	5	-
b10(0)	15	1	15	15	15	15
b10(1)	1	1	1	1	1	1
b10(2)	3	3	3	3	3	3
b10(3)	4	3	4	4	4	4
b10(4)	1	1	2	2	2	2
b10(5)	2	2	2	2	2	2
b10(6)	1	1	1	1	1	1
b10(7)	0	0	0	0	0	0
b10(8)	0	0	0	0	0	0
b10(9)	0	0	0	0	0	0
b10(10)	1	1	1	1	1	1
bcc(0)	11	11	11	11	11	11
bcc(1)	11	11	11	11	11	11
bcc(2)	11	11	11	11	11	11
bcc(3)	10	10	10	10	10	10
bcc(4)	10	10	10	10	10	10
bcc(5)	10	10	10	10	10	10
bcc(6)	10	10	10	10	10	10

bcc(7)	10	10	10	10	10	10
bcc(8)	10	10	10	10	10	10
bcc(9)	10	10	10	10	10	10
bcc(10)	10	10	10	10	10	10
bcc(11)	10	10	10	10	10	10
bcc(12)	10	10	10	10	10	10
bcc(13)	10	10	10	10	10	10
bcc(14)	11	11	11	11	11	11
bcc(15)	11	11	11	11	11	11
bcc(16)	10	10	10	10	10	10
bcc(17)	11	11	11	11	11	11
bcc(18)	10	10	10	10	10	10
bcc(19)	11	11	11	11	11	11
bcc(20)	11	11	11	11	11	11
bcc(21)	11	11	11	11	11	11
bcc(22)	11	11	11	11	11	11
bcc(23)	10	10	10	10	10	10
bcc(24)	11	11	11	11	11	11
bcc(25)	11	11	11	11	11	11
bcc(26)	11	11	11	11	11	11
bcc(27)	11	11	11	11	11	11
bcc(28)	11	11	11	11	11	11
bcc(29)	11	11	11	11	11	11
bcc(30)	11	11	11	11	11	11
bcc(31)	11	11	11	11	11	11
bcc(32)	11	11	11	11	11	11
bcc(33)	11	11	11	11	11	11
bcc(34)	11	11	11	11	11	11
bcc(35)	11	11	11	11	11	11
bcc(36)	11	11	11	11	11	11
bcc(37)	11	11	11	11	11	11
bcc(38)	11	11	11	11	11	11
bcc(39)	11	11	11	11	11	11
bcc(40)	11	11	11	11	11	11
bcc(41)	14	14	14	14	14	14
bcc(42)	11	11	11	11	11	11
bcc(43)	11	11	11	11	11	11
bcc(44)	11	11	11	11	11	11
bcd.div3(0)	2	1	2	2	2	2
bcd.div3(1)	0	0	1	1	1	1
bcd.div3(2)	0	0	2	2	2	2
bcd.div3(3)	0	0	1	1	1	1
bench1(0)	0	0	0	0	0	0
bench1(1)	0	0	0	0	0	0
bench1(2)	0	0	0	0	0	0
bench1(3)	0	0	0	0	0	0
bench1(4)	0	0	0	0	0	0
bench1(5)	0	0	0	0	0	0
bench1(6)	0	0	0	0	0	0
bench1(7)	0	0	0	0	0	0
bench1(8)	0	0	0	0	0	0
bench(0)	0	0	3	3	3	3
bench(1)	0	0	3	3	3	3

bench(2)	0	0	1	1	1	1
bench(3)	0	0	2	1	2	2
bench(4)	0	0	1	1	1	2
bench(5)	0	0	3	3	3	3
bench(6)	0	0	1	1	1	1
bench(7)	0	0	1	1	1	1
dekoder(0)	0	1	2	1	2	2
dekoder(1)	0	1	2	1	2	2
dekoder(2)	0	0	1	1	1	1
dekoder(3)	0	0	1	1	1	1
dekoder(4)	0	0	1	1	1	1
dekoder(5)	0	0	1	1	1	1
dekoder(6)	0	0	1	1	1	1
dk17(0)	0	0	5	5	5	-
dk17(1)	0	0	6	5	6	-
dk17(2)	0	0	5	4	5	-
dk17(3)	0	0	5	3	5	-
dk17(4)	0	0	6	4	6	-
dk17(5)	0	0	7	7	7	-
dk17(6)	0	0	7	7	7	-
dk17(7)	0	0	6	7	6	-
dk17(8)	0	0	5	2	5	-
dk17(9)	0	0	5	4	5	-
dk17(10)	1	1	6	6	6	-
dk27(0)	1	0	7	6	7	-
dk27(1)	1	0	7	6	7	-
dk27(2)	0	0	5	5	5	-
dk27(3)	0	0	6	7	6	-
dk27(4)	0	0	7	5	7	-
dk27(5)	0	0	6	7	6	-
dk27(6)	0	0	6	7	6	-
dk27(7)	0	0	6	5	6	-
dk27(8)	1	1	8	8	8	-
dk48(0)	0	0	12	13	12	-
dk48(1)	0	0	12	13	12	-
dk48(2)	0	0	12	9	12	-
dk48(3)	0	0	12	11	12	-
dk48(4)	0	0	11	9	11	-
dk48(5)	0	0	12	13	12	-
dk48(6)	0	0	11	11	11	-
dk48(7)	0	0	12	13	12	-
dk48(8)	0	0	11	13	11	-
dk48(9)	1	0	12	12	12	-
dk48(10)	0	0	12	13	12	-
dk48(11)	0	0	12	13	12	-
dk48(12)	0	0	12	13	12	-
dk48(13)	0	0	12	13	12	-
dk48(14)	0	0	10	11	10	-
dk48(15)	2	1	12	13	12	-
dk48(16)	0	0	10	10	10	-
ex1010(0)	0	0	0	0	0	0
ex1010(1)	0	0	0	0	0	0
ex1010(2)	0	0	0	0	0	0

ex1010(3)	0	0	0	0	0	0
ex1010(4)	0	0	0	0	0	0
ex1010(5)	0	0	0	0	0	0
ex1010(6)	0	0	0	0	0	0
ex1010(7)	0	0	0	0	0	0
ex1010(8)	0	0	0	0	0	0
ex1010(9)	0	0	0	0	0	0
exam(0)	0	1	4	3	4	-
exam(1)	0	10	10	10	10	-
exam(2)	0	0	2	3	2	-
exam(3)	0	0	2	2	2	-
exam(4)	0	0	3	3	3	-
exam(5)	0	0	3	2	3	-
exam(6)	0	0	3	3	3	-
exam(7)	0	0	2	2	3	-
exam(8)	10	0	10	10	10	-
exam(9)	0	0	2	2	2	-
exp(0)	0	0	3	3	3	3
exp(1)	0	0	0	0	0	0
exp(2)	0	0	0	0	0	0
exp(3)	8	0	8	8	8	8
exp(4)	8	0	8	8	8	8
exp(5)	0	0	0	0	0	0
exp(6)	1	0	1	1	1	1
exp(7)	0	0	0	0	0	0
exp(8)	0	0	0	0	0	0
exp(9)	0	0	0	0	0	0
exp(10)	0	0	1	1	1	1
exp(11)	0	0	1	1	1	1
exp(12)	2	0	2	2	2	2
exp(13)	1	0	2	2	2	2
exp(14)	0	0	0	0	0	0
exp(15)	0	0	0	0	0	0
exp(16)	0	0	0	0	0	0
exp(17)	0	0	0	0	0	0
exps(0)	0	0	0	0	0	0
exps(1)	0	0	0	0	0	0
exps(2)	0	0	0	0	0	0
exps(3)	0	0	0	0	0	0
exps(4)	0	0	0	0	0	0
exps(5)	1	1	1	1	1	1
exps(6)	1	1	1	1	1	1
exps(7)	0	0	0	0	0	0
exps(8)	0	0	0	0	0	0
exps(9)	0	0	0	0	0	0
exps(10)	0	0	0	0	0	0
exps(11)	0	0	0	0	0	0
exps(12)	0	0	0	0	0	0
exps(13)	0	0	0	0	0	0
exps(14)	0	0	0	0	0	0
exps(15)	0	0	0	0	0	0
exps(16)	1	1	1	1	1	1
exps(17)	0	0	0	0	0	0

exps(18)	1	1	1	1	1	1
exps(19)	1	1	1	1	1	1
exps(20)	0	0	0	0	0	0
exps(21)	0	0	0	0	0	0
exps(22)	0	0	0	0	0	0
exps(23)	0	0	0	0	0	0
exps(24)	0	0	0	0	0	0
exps(25)	0	0	0	0	0	0
exps(26)	0	0	0	0	0	0
exps(27)	0	0	0	0	0	0
exps(28)	0	0	0	0	0	0
exps(29)	0	0	0	0	0	0
exps(30)	0	0	3	1	3	-
exps(31)	1	0	3	1	3	3
exps(32)	1	0	4	3	4	4
exps(33)	0	0	0	0	0	0
exps(34)	0	0	0	0	0	0
exps(35)	0	0	0	0	0	0
exps(36)	0	0	0	0	0	0
exps(37)	0	0	0	0	0	0
fout(0)	0	0	0	0	0	0
fout(1)	0	0	0	0	0	0
fout(2)	0	0	0	0	0	0
fout(3)	0	0	0	0	0	0
fout(4)	0	0	1	1	1	1
fout(5)	0	0	0	0	0	0
fout(6)	0	0	0	0	0	0
fout(7)	0	0	0	0	0	0
fout(8)	0	0	0	0	0	0
fout(9)	0	0	0	0	0	0
inc(0)	1	1	1	1	1	1
inc(1)	1	1	1	1	1	1
inc(2)	0	0	0	0	0	0
inc(3)	0	0	0	0	0	0
inc(4)	0	1	4	4	4	4
inc(5)	1	1	3	3	3	3
inc(6)	0	1	4	4	4	4
inc(7)	1	0	1	1	1	1
inc(8)	3	3	3	3	3	3
l8err(0)	8	8	8	8	8	8
l8err(1)	8	8	8	8	8	8
l8err(2)	8	8	8	8	8	8
l8err(3)	0	0	0	0	0	0
l8err(4)	0	0	0	0	0	0
l8err(5)	0	0	0	0	0	0
l8err(6)	0	0	0	0	0	0
l8err(7)	0	8	8	8	8	8
p1(0)	0	0	6	7	6	7
p1(1)	0	0	2	1	2	2
p1(2)	0	0	2	1	2	3
p1(3)	8	0	8	8	8	8
p1(4)	8	0	8	8	8	8
p1(5)	0	0	2	1	2	2

p1(6)	1	0	2	1	2	2
p1(7)	0	0	2	1	2	3
p1(8)	0	0	1	1	1	1
p1(9)	0	0	3	2	3	3
p1(10)	0	0	3	2	3	3
p1(11)	0	0	3	2	3	3
p1(12)	2	0	5	6	5	6
p1(13)	1	0	6	6	6	6
p1(14)	0	0	1	1	1	1
p1(15)	0	0	1	1	1	1
p1(16)	0	0	2	1	2	3
p1(17)	0	0	1	1	1	1
p3(0)	0	0	6	7	6	7
p3(1)	0	0	2	1	2	2
p3(2)	0	0	2	1	2	3
p3(3)	8	0	8	8	8	8
p3(4)	8	0	8	8	8	8
p3(5)	0	0	2	1	2	2
p3(6)	1	0	2	1	2	2
p3(7)	0	0	2	1	2	3
p3(8)	0	0	1	1	1	1
p3(9)	0	0	3	2	3	3
p3(10)	0	0	3	2	3	3
p3(11)	0	0	3	2	3	3
p3(12)	2	0	5	6	5	6
p3(13)	1	0	6	6	6	6
pdc(0)	1	0	12	12	12	-
pdc(1)	1	0	12	12	12	-
pdc(2)	0	0	11	11	11	-
pdc(3)	0	0	4	4	4	-
pdc(4)	0	0	5	5	5	-
pdc(5)	0	0	7	5	7	-
pdc(6)	13	0	14	14	14	-
pdc(7)	0	0	11	11	11	-
pdc(8)	0	0	5	5	5	-
pdc(9)	0	0	6	6	6	-
pdc(10)	3	0	12	10	12	-
pdc(11)	4	0	11	11	11	-
pdc(12)	8	0	11	11	11	-
pdc(13)	8	0	12	11	12	-
pdc(14)	14	0	14	14	14	-
pdc(15)	13	0	13	13	13	-
pdc(16)	0	0	13	13	13	-
pdc(17)	0	0	13	13	13	-
pdc(18)	0	0	13	13	13	-
pdc(19)	0	0	13	13	13	-
pdc(20)	4	0	9	11	9	-
pdc(21)	7	0	12	12	12	-
pdc(22)	4	0	12	10	12	-
pdc(23)	0	0	7	7	7	-
pdc(24)	0	0	7	7	7	-
pdc(25)	9	0	12	11	12	-
pdc(26)	2	0	12	11	12	-

pdc(27)	8	0	12	12	12	-
pdc(28)	0	0	6	7	6	-
pdc(29)	0	0	7	7	7	-
pdc(30)	1	0	10	9	10	-
pdc(31)	5	0	8	6	8	-
pdc(32)	8	0	10	10	10	-
pdc(33)	0	0	6	7	6	-
pdc(34)	0	0	7	6	7	-
pdc(35)	0	0	7	6	7	-
pdc(36)	0	0	6	6	6	-
pdc(37)	0	0	9	10	9	-
pdc(38)	9	0	12	11	12	-
pdc(39)	9	0	12	11	12	-
spla(0)	0	0	0	0	0	-
spla(1)	0	0	0	0	0	-
spla(2)	1	1	2	1	2	-
spla(3)	1	1	2	2	2	-
spla(4)	1	1	2	2	2	-
spla(5)	0	0	1	1	1	-
spla(6)	0	0	1	1	1	-
spla(7)	0	0	2	2	2	-
spla(8)	13	13	13	13	13	-
spla(9)	0	0	2	2	2	-
spla(10)	0	0	1	1	1	-
spla(11)	0	0	1	1	1	-
spla(12)	0	0	2	2	2	-
spla(13)	7	7	9	8	9	-
spla(14)	8	8	10	10	10	-
spla(15)	5	5	7	6	7	-
spla(16)	4	4	6	6	6	-
spla(17)	8	8	10	10	10	-
spla(18)	8	8	10	10	10	-
spla(19)	0	0	1	1	1	-
spla(20)	0	0	1	1	1	-
spla(21)	0	0	2	1	2	-
spla(22)	0	0	2	1	2	-
spla(23)	1	1	2	1	2	-
spla(24)	9	9	11	11	11	-
spla(25)	9	9	11	11	11	-
spla(26)	0	0	1	1	1	-
spla(27)	0	0	1	1	1	-
spla(28)	0	0	1	1	1	-
spla(29)	0	0	2	1	2	-
spla(30)	0	0	1	1	1	-
spla(31)	8	8	9	9	9	-
spla(32)	0	0	1	1	1	-
spla(33)	0	0	1	1	1	-
spla(34)	9	9	11	11	11	-
spla(35)	0	0	2	2	2	-
spla(36)	0	0	2	2	2	-
spla(37)	0	0	2	2	2	-
spla(38)	0	0	2	2	2	-
spla(39)	3	3	5	4	5	-

spla(40)	0	0	2	1	2	-
spla(41)	4	4	5	4	5	-
spla(42)	8	8	10	8	10	-
spla(43)	5	5	7	6	7	-
spla(44)	4	4	6	6	6	-
spla(45)	0	0	1	1	1	-
t2(0)	5	5	5	5	5	-
t2(1)	5	5	5	5	5	-
t2(2)	6	6	6	6	6	-
t2(3)	8	8	8	8	8	-
t2(4)	10	10	10	10	10	-
t2(5)	6	6	6	6	6	-
t2(6)	4	4	7	6	7	-
t2(7)	7	7	9	7	9	-
t2(8)	11	10	12	12	12	-
t2(9)	10	10	11	10	11	-
t2(10)	11	11	12	12	12	-
t2(11)	11	11	12	12	12	-
t2(12)	12	10	13	12	13	-
t2(13)	11	9	12	11	12	-
t2(14)	11	9	12	11	12	-
t2(15)	10	10	12	12	12	-
t4(0)	3	3	6	6	6	-
t4(1)	1	1	6	5	6	-
t4(2)	3	3	8	7	8	-
t4(3)	1	1	6	6	6	-
t4(4)	1	1	5	6	5	-
t4(5)	3	3	8	8	8	-
t4(6)	3	3	8	8	8	-
t4(7)	3	3	8	10	8	-
test1(0)	0	0	0	0	0	0
test1(1)	0	0	0	0	0	0
test1(2)	0	0	0	0	0	0
test1(3)	0	0	0	0	0	0
test1(4)	0	0	0	0	0	0
test1(5)	0	0	0	0	0	0
test1(6)	0	0	0	0	0	0
test1(7)	0	0	0	0	0	0
test1(8)	0	0	0	0	0	0
test1(9)	0	0	0	0	0	0
test3(0)	0	0	0	0	0	0
test3(1)	0	0	0	0	0	0
test3(2)	0	0	0	0	0	0
test3(3)	0	0	0	0	0	0
test3(4)	0	0	0	0	0	0
test3(5)	0	0	0	0	0	0
test3(6)	0	0	0	0	0	0
test3(7)	0	0	0	0	0	0
test3(8)	0	0	0	0	0	0
test3(9)	0	0	0	0	0	0
test3(10)	0	0	0	0	0	0
test3(11)	0	0	0	0	0	0
test3(12)	0	0	0	0	0	0

test3(13)	0	0	0	0	0	0
test3(14)	0	0	0	0	0	0
test3(15)	0	0	0	0	0	0
test3(16)	0	0	0	0	0	0
test3(17)	0	0	0	0	0	0
test3(18)	0	0	0	0	0	0
test3(19)	0	0	0	0	0	0
test3(20)	0	0	0	0	0	0
test3(21)	0	0	0	0	0	0
test3(22)	0	0	0	0	0	0
test3(23)	0	0	0	0	0	0
test3(24)	0	0	0	0	0	0
test3(25)	0	0	0	0	0	0
test3(26)	0	0	0	0	0	0
test3(27)	0	0	0	0	0	0
test3(28)	0	0	0	0	0	0
test3(29)	0	0	0	0	0	0
test3(30)	0	0	0	0	0	0
test3(31)	0	0	0	0	0	0
test3(32)	0	0	0	0	0	0
test3(33)	0	0	0	0	0	0
test3(34)	0	0	0	0	0	0
test4(0)	0	0	0	0	0	0
test4(1)	0	0	0	0	0	0
test4(2)	0	0	1	1	1	1
test4(3)	0	0	0	0	0	0
test4(4)	0	0	1	1	1	1
test4(5)	0	0	1	1	1	1
test4(6)	0	0	1	1	1	1
test4(7)	0	0	1	1	1	1
test4(8)	0	0	1	1	1	1
test4(9)	0	0	1	1	1	1
test4(10)	0	0	0	0	0	0
test4(11)	0	0	1	1	1	1
test4(12)	0	0	1	1	1	1
test4(13)	0	0	1	1	1	1
test4(14)	0	0	1	2	1	2
test4(15)	0	0	0	0	0	0
test4(16)	0	0	0	0	0	0
test4(17)	0	0	1	1	1	1
test4(18)	0	0	1	1	1	2
test4(19)	0	0	0	0	0	0
test4(20)	0	0	1	1	1	1
test4(21)	0	0	1	1	1	1
test4(22)	0	0	1	1	1	1
test4(23)	0	0	1	1	1	1
test4(24)	0	0	0	0	0	0
test4(25)	0	0	1	1	1	1
test4(26)	0	0	1	1	1	1
test4(27)	0	0	0	0	0	0
test4(28)	0	0	0	0	0	0
test4(29)	0	0	0	0	0	0
x1dn(0)	11	3	11	11	11	11

x1dn(1)	9	3	9	9	9	9
x1dn(2)	9	3	13	13	13	-
x1dn(3)	13	3	16	16	16	-
x1dn(4)	13	3	16	16	16	-
x1dn(5)	0	3	10	13	10	-
wim(0)	0	0	1	1	1	1
wim(1)	0	0	1	1	1	1
wim(2)	0	1	2	1	2	2
wim(3)	0	0	1	1	1	1
wim(4)	1	1	2	1	2	2
wim(5)	0	0	1	1	1	1
wim(6)	0	0	1	1	1	1
Totale	1134	859	2094	2025	2095	851

			Euristica di Filippo			Euristica di Alessio			Test ottimo		
Funzione	<i>n</i>	<i>m</i>	<i>k</i>	<i>Max</i>	<i>Test</i>	<i>k</i>	<i>Max</i>	<i>Test</i>	<i>k</i>	<i>Max</i>	<i>Test</i>
alu2	10	8	3.375	7	4.177	3.375	7	4.909	3.375	7	5.027
alu3	10	8	2.75	6	3.288	2.75	6	3.451	2.75	6	3.396
apla	10	12	4.417	6	41.733	3.25	6	53.136	-	-	-
b10	15	11	2.636	15	10.006	2.636	15	11.351	2.636	15	54.84
bcc	26	45	10.755	14	59.558	10.755	14	53.864	10.755	14	60.919
bcd.div3	4	4	1.5	2	0.203	1.5	2	0.792	1.5	2	0.903
bench1	9	9	0	0	24.614	0	0	21.115	0	0	23.001
bench	6	8	1.875	3	1.937	1.75	3	3.759	2	3	210.973
dekoder	4	7	1.285	2	0.403	1	1	1.523	1.285	2	1.489
dk17	10	11	5.727	7	21.873	4.909	7	37.886	-	-	-
dk27	9	9	6.444	8	5.436	6.222	8	12.677	-	-	-
dk48	15	17	11.588	12	131.919	11.941	13	10212.934	-	-	-
ex1010	10	10	0	0	105.573	0	0	63.166	0	0	73.663
exam	10	10	4.1	10	28.322	4	10	47.231	-	-	-
exp	8	18	1.444	8	2.101	1.444	8	6.809	1.444	8	7.699
exps	8	38	0.395	4	2.49	0.263	3	13.52	-	-	-
fout	6	10	0.1	1	0.26	0.1	1	3.208	0.1	1	2.999
inc	7	9	1.888	4	0.778	1.888	4	2.415	1.888	4	2.179
l8err	8	8	4	8	0.432	4	8	2.006	4	8	1.826
p1	8	18	3.222	8	15.285	2.883	8	18.098	3.5	8	90018
p3	8	14	3.786	8	11.309	3.375	8	13.451	4.071	8	87273
pdc	16	40	9.875	14	757.339	9.6	14	5976.43	-	-	-
spla	16	46	3.978	13	27899.712	3.695	13	7807.671	-	-	-
t2	17	16	9.5	13	22.802	9.062	12	1035.178	-	-	-
t4	12	8	6.875	8	19.487	7	10	35.667	-	-	-
test1	8	10	0	0	13.05	0	0	11.126	0	0	9.521
test3	10	35	0	0	343.207	0	0	241.442	0	0	222.927
test4	8	30	0.633	1	19.609	0.666	2	28.654	0.7	2	34.74
x1dn	27	6	12.5	16	846.334	13	16	136.116	-	-	-
wim	4	7	1.285	2	0.361	1	1	1.35	1.285	2	2.591
Totale			115.933	200	30393.598	112.064	200	25860.935	41.289	90	178009.693

Table 2: Risultati sperimentali ottenuti tramite l'esecuzione dei tre algoritmi.

			All zeros test			All ones test			Euristica di Filippo migliorata		
Funzione	n	m	k	Max	$Test$	k	Max	$Test$	k	Max	$Test$
alu2	10	8	2.75	6	1.805	1.125	6	2.513	3.375	7	8.026
alu3	10	8	2.75	6	1.095	0.625	2	2.836	2.75	6	2.313
apla	10	12	0.166	1	1.439	0.083	1	2.713	4.417	6	38.419
b10	15	11	2.545	15	5.534	1.181	3	6.71	2.636	15	8.06
bcc	26	45	10.755	14	36.243	10.755	14	39.36	10.755	14	70.267
bcd.div3	4	4	0.5	2	0.13	0.25	1	0.13	1.5	2	0.227
bench1	9	9	0	0	8.855	0	0	19.351	0	0	0
bench	6	8	0	0	0.686	0	0	0.875	1.875	3	2.46
dekoder	4	7	0	0	0.252	0.285	1	0.133	1.285	2	0.352
dk17	10	11	0.09	1	0.877	0.09	1	1.029	5.727	7	25.767
dk27	9	9	0.333	1	0.468	0.111	1	0.503	6.444	8	5.346
dk48	15	17	0.176	2	1.62	0.058	1	1.805	11.588	12	119.311
ex1010	10	10	0	0	44.253	0	0	44.957	0	0	0.144
exam	10	10	1	10	3.342	1.1	10	4.545	4.2	10	33.144
exp	8	18	1.111	8	1.605	0	0	3.232	1.444	8	2.133
exps	8	38	0.184	1	5.946	0.131	1	7.379	0.395	4	2.49
fout	6	10	0	0	1.135	0	0	0.923	0.1	1	0.321
inc	7	9	0.777	3	0.496	0.888	3	0.524	1.888	4	0.679
l8err	8	8	3	8	0.884	4	8	0.501	4	8	0.52
p1	8	18	1.111	8	1.624	0	0	3.452	3.222	8	14.006
p3	8	14	1.428	8	1.153	0	0	1.625	3.786	8	9.376
pdc	16	40	3.275	14	10.387	0	0	22.038	9.875	14	757.339
spla	16	46	2.521	13	9.754	2.521	13	8.678	3.978	13	27127.78
t2	17	16	8.625	12	1.213	8.187	11	1.248	9.5	13	30.687
t4	12	8	2.25	3	0.81	2.25	3	0.9	6.875	8	20.698
test1	8	10	0	0	4.346	0	0	9.305	0	0	19.03
test3	10	35	0	0	104.855	0	0	227.414	0	0	550.411
test4	8	30	0	0	19.946	0	0	8.826	0.633	1	29.652
x1dn	27	6	9.166	13	2.894	3	3	5.923	12.5	16	557.006
wim	4	7	0.142	1	0.14	0.285	1	0.125	1.285	2	0.522
Totale			54.655	150	273.787	36.925	84	429.553	116.033	200	29436.486

Table 3: Risultati sperimentali ottenuti tramite l'esecuzione dei tre algoritmi.