

Άσκηση Διευκρινήσεις: Υπολογισμός stress-tests
με χρήση γενετικών αλγορίθμων - Υπολογισμός
φόρτου εργασίας μέγιστης κατανάλωσης ισχύος

Vasileios Tenentes

University of Ioannina

Παράμετροι

Παράμετροι για την άσκηση:

$L=2$; → το πλήθος των διανυσμάτων σε κάθε φόρτο-εργασίας/επίδοξο-stress-test/individual

$N=30$; → το πλήθος των individuals σε κάθε γενεά

$m=0.05$; → ρυθμός μετάλλαξης

$G=100$; → μέγιστο πλήθος γενεών

Ερώτημα 4.1

Παραγωγή ψευδοτυχαίων individuals και καταμέτρηση των εναλλαγών που προκαλούν

Διευκρινήσεις «δοκιμής» και «καταμέτρησης» εναλλαγών: η δοκιμή ενός φόρτου εργασίας μήκους $L=2$ στο κύκλωμα και η καταμέτρηση των εναλλαγών που προκαλεί γίνεται με τον εξής τρόπο: Αρχικά εφαρμόζουμε το πρώτο διάνυσμα στο κύκλωμα και αντιγράφουμε σε έναν πίνακα (νέος πίνακας SignalsBefore) τις τιμές των σημάτων που προκύπτουν από την εφαρμογή του πρώτου διανύσματος. Στην πορεία εφαρμόζουμε το δεύτερο διάνυσμα του φόρτου εργασίας και αντιγράφουμε τις τιμές των σημάτων που προκύπτουν από την εφαρμογή του δεύτερου διανύσματος σε έναν πίνακα (Signals). Στην πορεία συγκρίνοντας τους δύο πίνακες (SignalsBefore και Signals), κάνουμε καταμέτρηση του πλήθους των διαφορών τους ανά ψηφίο, το οποίο είναι και το πλήθος των εναλλαγών που προκλήθηκαν από το φόρτο εργασίας. **Προσοχή,** δεν καταμετράμε τις εναλλαγές στα `l1inputs`, γιατί τα `l1inputs` δεν είναι έξοδοι κάποιας πύλης του κυκλώματος.

Ερώτημα 4.1

Παραγωγή ψευδοτυχαίων individuals καταμέτρηση των εναλλαγών που προκαλούν

Η countswitches απλά βρίσκει το πλήθος διαφορών του πριν με το μετά:

```
function switchesnumber=countSwitches(obj)
    switchesnumber=0;
    for j=1:size(obj.Signals,2)
        if(obj.SignalsBefore(j)!=obj.Signals(j))
            switchesnumber++;
        endif
    endfor
endfunction
```

SignalsBefore [1 0 1 0 0 0 1 0 0 1]
 Signals [0 1 1 0 1 0 0 0 1 1]

obj) 5

Προσοχή το βάζω μετά την
αρχικοποίηση του TLPINPUTS
για το 2^ο διάνυσμα για να μην τα
προσμετρήσω στην σύγκριση μετά
, επειδή αυτά δεν είναι έξοδοι
στοιχείων

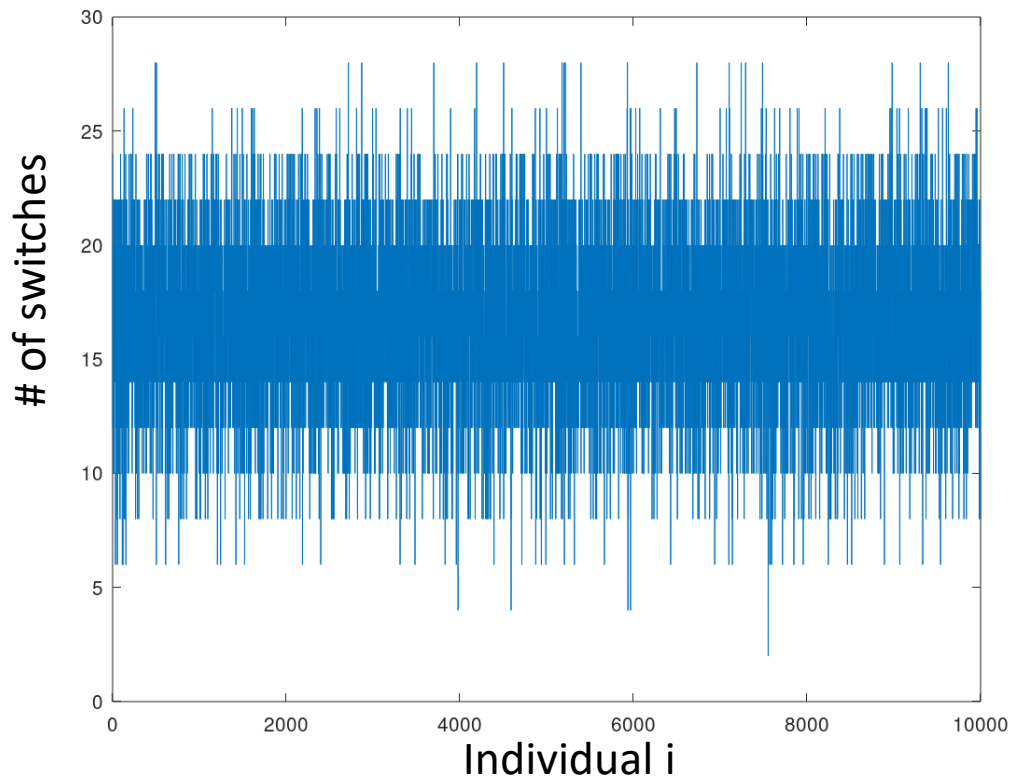
Καταμέτρηση σε έναν τυχαίο individual workload:

```
workload = gaGenerateRandomWorkload(inputsNumber, L);
circuit = circuit.applyTLPInputs(workload(1,:));
circuit = circuit.process();
circuit = circuit.applyTLPInputs(workload(2,:));
circuit.signalsBefore=circuit.signals;
circuit = circuit.process(fromFileCirc);
switches = circuit.countSwitches();
switchesTable=[switchesTable;switches];
```

Ερώτημα 4.1, τί αποτέλεσμα περιμένω;

Παραγωγή ψευδοτυχαίων individuals καταμέτρηση των εναλλαγών που προκαλούν

Το αποτέλεσμα δεν μπορεί να ξεφύγει από τα στατιστικά που προκαλεί η δομή του κυκλώματος. Πρόκειται στην ουσία για την εκτέλεση ενός Monte Carlo πειράματος. Επομένως, παρόλο που η εικόνα που θα βρει ο καθένας θα είναι διαφορετική, ο μέσος όρος και το variance των δεδομένων του πίνακα switchesTable πρέπει να είναι σχεδόν το ίδιο για όλους.



Στο octave να βάλετε το πακέτο statistics με τον εξής τρόπο:

`pkg install -forge statistics`

Και πρέπει να το φορτώσετε με τον εξής τρόπο:

`pkg load statistics`

Και να υπολογίσετε mean και variance του switchesTable

`averageSwitches=mean(montecarloswitches);`

`varianceSwitches=var(montecarloswitches);`

Δεν είναι ανάγκη να το τρέξετε για 5 λεπτά,
10000 individuals είναι αρκετοί

Ερώτημα 4.1, τί αποτέλεσμα περιμένω;

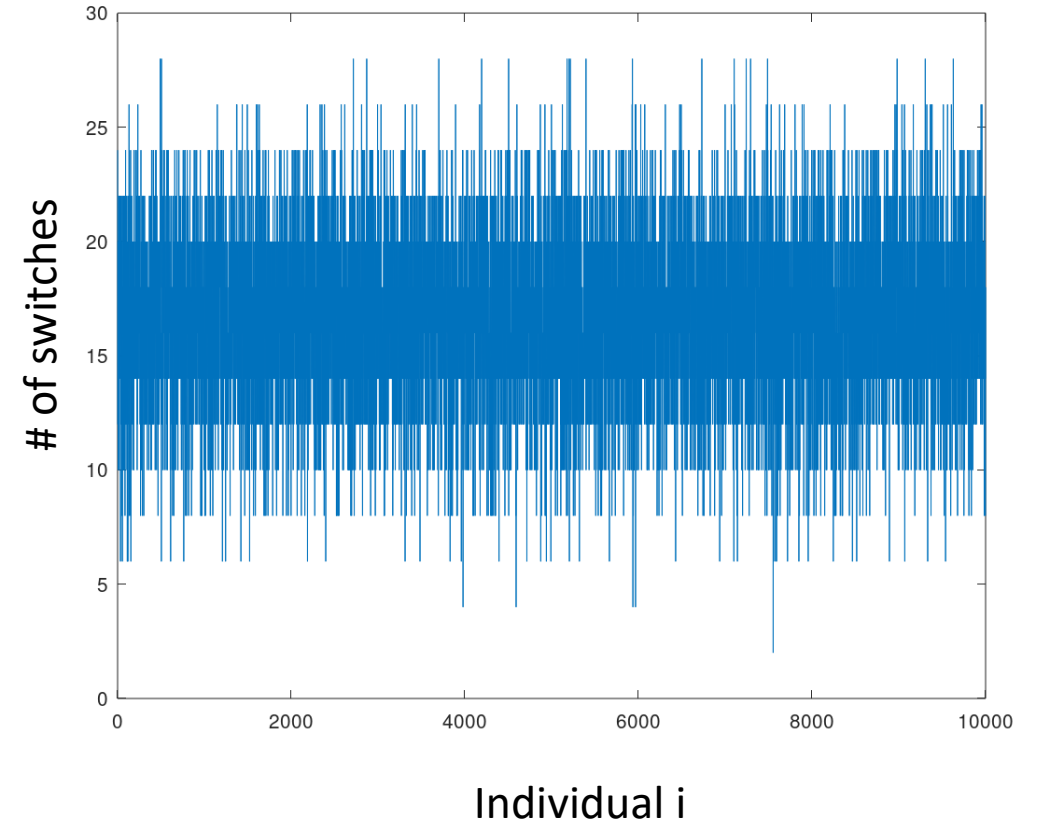
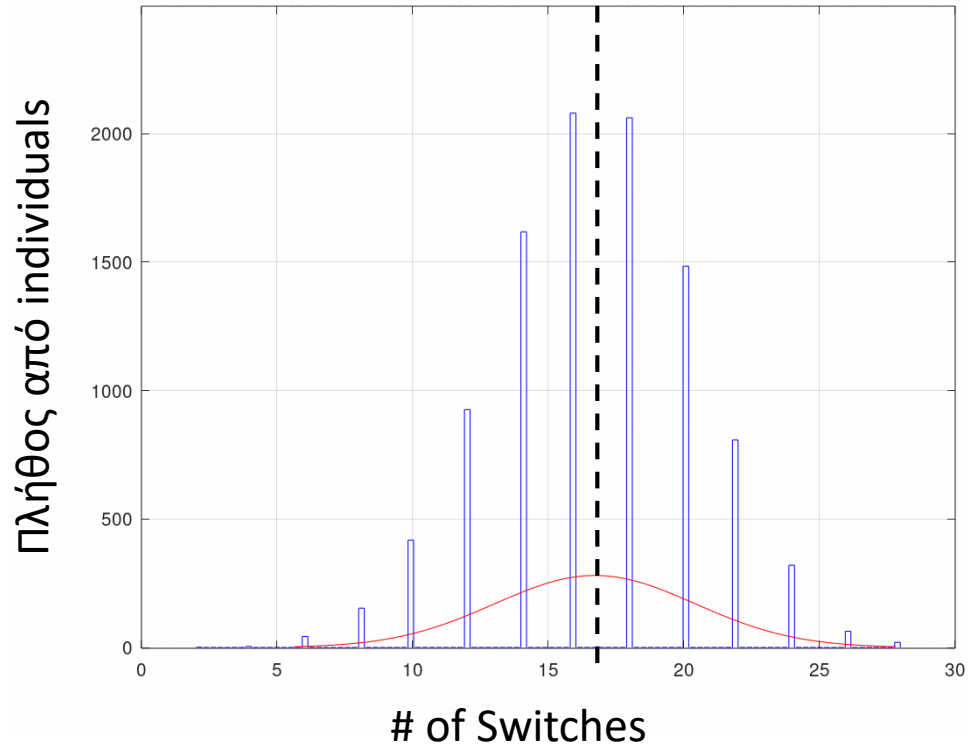
Παραγωγή ψευδοτυχαίων individuals καταμέτρηση των εναλλαγών που προκαλούν

Mean = 16.717

Variance = 13.681

To figure είναι με το qqplot

Mean
16.717

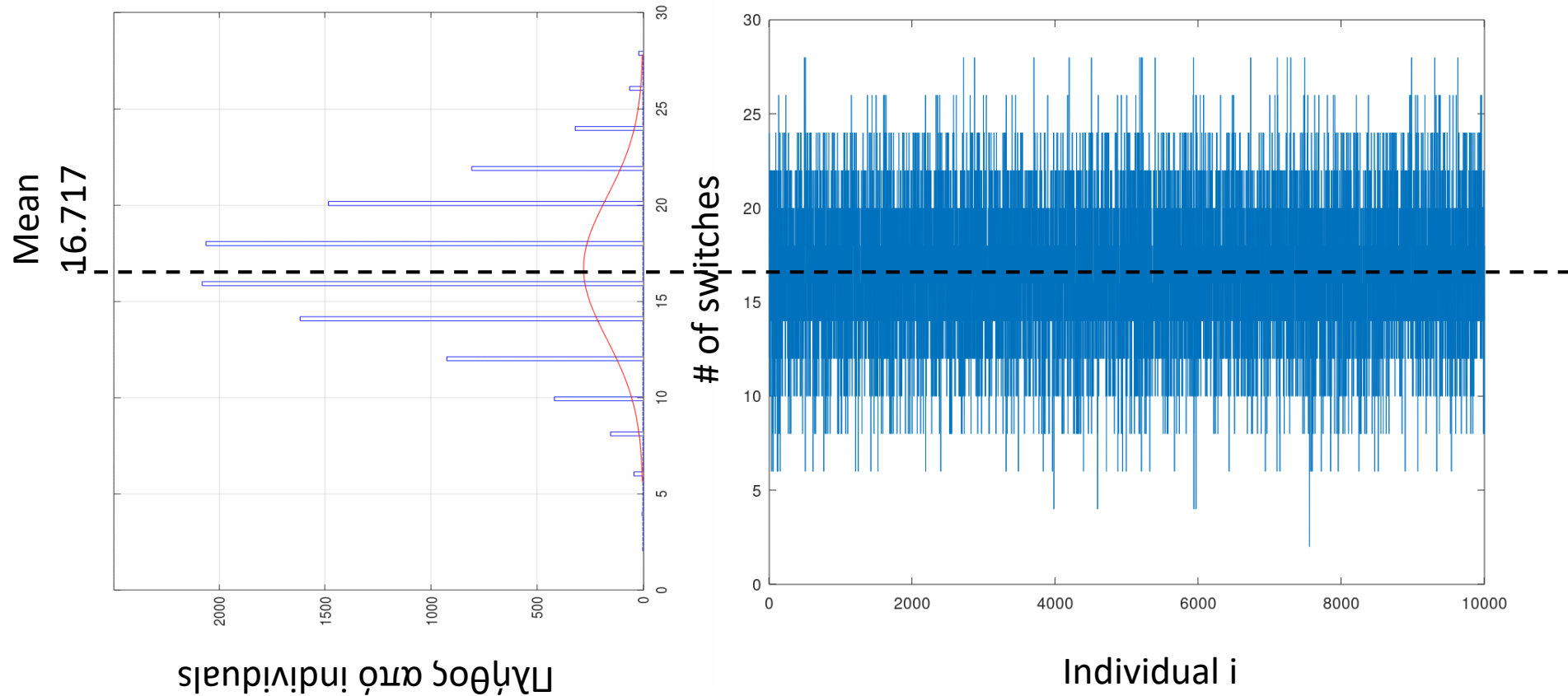


Ερώτημα 4.1, τί αποτέλεσμα περιμένω;

Παραγωγή ψευδοτυχαίων individuals καταμέτρηση των εναλλαγών που προκαλούν

Mean = 16.717

Variance = 13.681



Ερώτημα 4.2, επιλογή γονέων μετά το scoring

Γενετικός αλγόριθμος

Αφού γίνει η απόδοση του score σε κάθε individual, πρέπει να επιλεγούν οι δύο γονείς:

```
function [parent1, parent2, score1, score2]=gaSelectParents(scores, population, N, L)
    best=-1; % ο καλύτερος
    sbest=-1; % ο 2ος καλύτερος
    besti=-1; % το index του καλύτερου στον πίνακα
    sbesti=-1; % το index του 2ου καλύτερου στον πίνακα
    for i=1:size(scores,2)
        if(scores(1,i)>best)
            sbest=best;
            best=scores(1,i);
            sbesti=besti;
            besti=i;
        else
            if(scores(1,i)>=sbest)
                sbest=scores(1,i);
                sbesti=i;
            endif
        endif
    endfor
    parent1=gaGetWorkloadFromPopulation(N, L, population, besti);
    parent2=gaGetWorkloadFromPopulation(N, L, population, sbesti);
    score1=best;
    score2=sbest;
endfunction
```

Συχνό λάθος

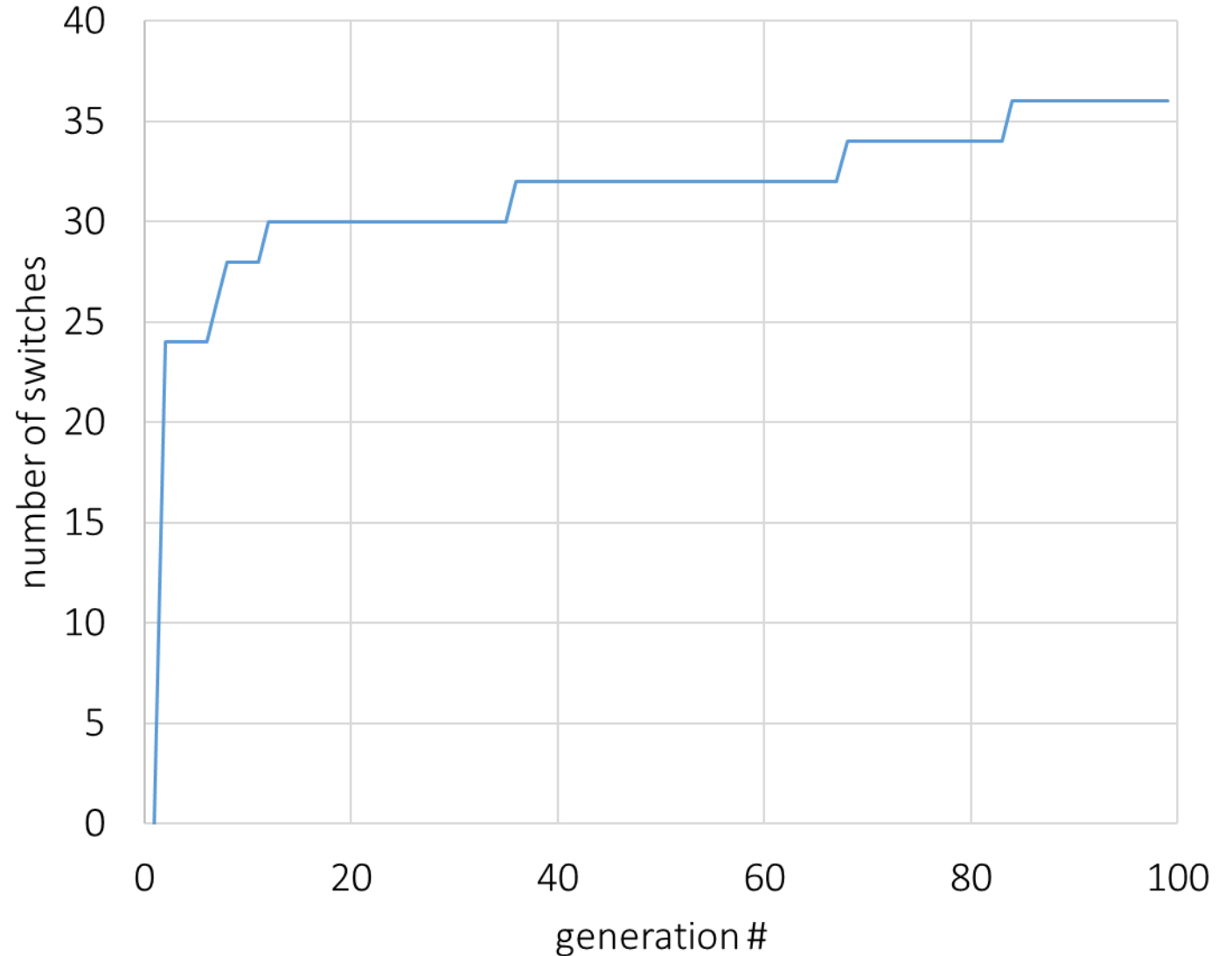
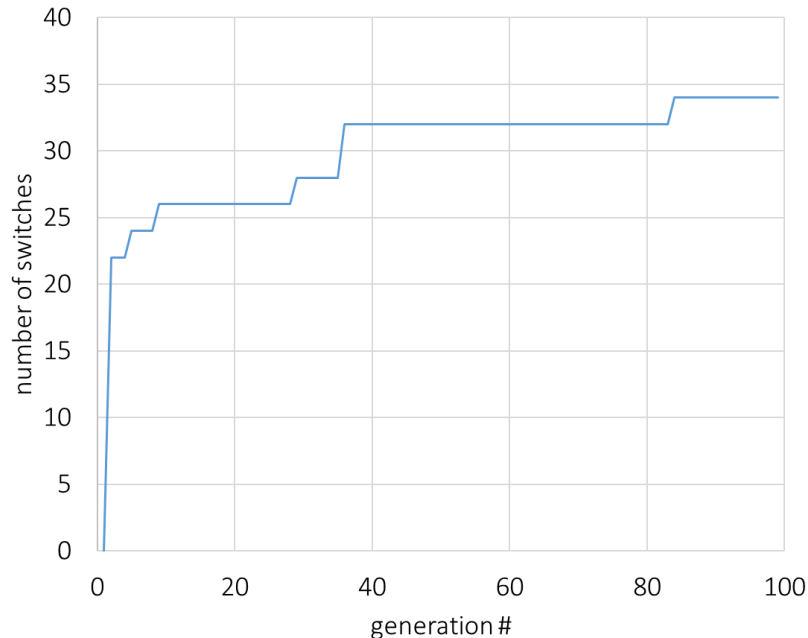
Να επιλέγουμε το ίδιο workload και για τους δύο γονείς.

Σιγουρευτείτε τυπώνοντας τους γονείς που επιλέχτηκαν ότι δεν επιλέγεται τους ίδιους και για τους δύο γονείς

Ερώτημα 4.2, τί αποτέλεσμα περιμένω;

Γενετικός αλγόριθμος

Το αποτέλεσμα που βλέπετε δίπλα το πήρα βάζοντας ένα πείραμα να εκτελέσει 100 φορές τον γενετικό αλγόριθμο (για 100 γενεές κάθε φορά). Αυτό ήταν το 83^ο πείραμα από τα 100. Τα υπόλοιπα κατέληγαν συνήθως στο 30, 32 και 34.



Κώδικας για process και ταξινόμηση κυκλώματος

Συνάρτηση process:

```
function updated=process(obj)
    obj.delays=zeros(size(obj.delays));
    obj.SignalsBefore=obj.Signals;
    if(size(obj.ProcessOrder)==0)
        obj=obj.findProcessOrder();
    endif
    for j=1:size(obj.ElementsSorted,2)
        ele=obj.Elements{obj.ElementsSorted(1,j)};
        inputValues=obj.Signals([ele.inputs{:,1}]);
        inputDelays=obj.delays([ele.inputs{:,1}]);
        obj.delays(ele.output)=max(inputDelays)+1.0;
        obj.Signals(ele.output)=spByType(ele.type, num2cell(inputValues){:});
    endfor
    updated=obj;
endfunction
```

Μπορείτε να ενσωματώσετε τον κώδικα της ταξινόμησης του κυκλώματος μέσα στον κώδικα προσομοίωσής του, ώστε να μην ξεχνάτε να το κάνετε

Κώδικας για ταξινόμηση κυκλώματος

```
function updated=findProcessOrder(obj)
    signalsUpdated=zeros(size(obj.Signals));
    elementsUpdated=zeros(size(obj.Elements,2));
    obj.ElementsSorted=[];
    for j=1:size(obj.TlpInputs, 2)
        signalsUpdated(obj.TlpInputs(j))=1;
    endfor
    updatedNumberOfElements=0;

    while(updatedNumberOfElements!=size(obj.Elements,2))
        for j=1:size(obj.Elements,2)
            if(elementsUpdated(j)==1)
                continue;
            endif
            ele=obj.Elements{j};
            updatedElement=1;
            for i=1:size(ele.inputs, 1)
                if(signalsUpdated(ele.inputs{i,1})==0)
                    updatedElement=0;
                    break;
                endif
            endfor
            if(updatedElement==0)
                continue;
            endif
            updatedNumberOfElements++;
            elementsUpdated(j)=1;
            obj.ElementsSorted=[obj. ElementsSorted, j];
            signalsUpdated(ele.output)=1;
        endfor
    endwhile
    updated=obj;
endfunction
```

Το κύκλωμα σορταρισμένο είναι:

```
TLPINPUTS i1 i2 i3 i4 i5 i6 i7 i8 i9 i10 i11 i12 i13 i14 i15 i16 i17 i18 i19 i20
AND t1 i12 i7
AND t2 i18 i14
XOR t3 i10 i9
AND t4 i15 i11
XNOR t6 i5 i11
XNOR t7 i20 i3
AND t8 i4 i17
XOR t9 i5 i6
AND t10 i10 i13
NOR t11 i11 i20
NAND t14 i17 i10 t9
XNOR t15 i20 i7 t10
XOR t16 i7 i10 t11
AND t19 i1 i12 t14
XOR t20 i9 i15 t15
XNOR t21 i19 i4 t16
XOR t24 i10 t8 t19
XNOR t25 i1 t9 t20
OR t26 i19 t1
NOR t27 i12 t2
XNOR t29 i15 t4
XNOR t31 i10 t6
AND t32 i10 t7
NAND t35 i9 t19
XOR t36 i7 t20
XNOR t37 i7 t21
OR t40 i8 t24
NOR t5 i9 i15 t20
AND t13 i16 i8 t14
NOR t18 i5 i2 t13
OR t23 i11 i5 t18
NOR t28 i12 t31
NAND t30 i4 t5
NAND t34 i4 t18
NAND t39 i11 t23
XOR o1 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
XNOR t12 i17 i5 t13
NOR t17 i4 i2 t12
AND t22 i6 i12 t17
XNOR t33 i18 t17
OR t38 i5 t22
XOR o2 t11 t12 t13 t14 t15 t16 t17 t18 t19 t20
XOR o3 t21 t22 t23 t24 t25 t26 t27 t28 t29 t30
XOR o4 t31 t32 t33 t34 t35 t36 t37 t38 t39 t40
```