

Ατομική Διπλωματική Εργασία

**ΚΑΤΑΝΕΜΗΜΕΝΗ ΛΗΨΗ ΑΠΟΦΑΣΕΩΝ :
ΤΡΕΙΣ ΠΡΑΚΤΟΡΕΣ, ΔΥΟ ΜΗΧΑΝΕΣ
ΚΑΙ ΚΑΤΑΝΟΜΗ POISSON**

Μιχάλης Καρατζής

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΚΑΤΑΝΕΜΗΜΕΝΟΙ ΛΗΨΗ ΑΠΟΦΑΣΕΩΝ:

ΤΡΕΙΣ ΠΡΑΚΤΟΡΕΣ , ΔΥΟ ΜΗΧΑΝΕΣ

ΚΑΙ ΚΑΤΑΝΟΜΗ POISSON

Μιχάλης Καρατζής

Επιβλέπων Καθηγητής

Μάριος Μαυρονικόλας

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του
Πανεπιστημίου Κύπρου

Μάιος 2010

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κύριο Μάριο Μαυρονικόλα για την υποστήριξη, την καθοδήγηση και την μεγάλη βοήθεια που μου πρόσφερε καθ' όλη την διάρκεια της εκπόνησης της διπλωματικής μου εργασίας, δίνοντας μου σημαντικές συμβουλές έτσι ώστε να καταστεί δυνατή η ολοκλήρωση της εργασίας αυτής.

Ακόμα θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη που μου παρείχε όλο αυτό τον καιρό για την εκπόνηση της διπλωματικής μου εργασίας.

Περίληψη

Στην εργασία αυτή μελετήθηκε ένα συγκεκριμένο μοντέλο κατανεμημένης λήψης αποφάσεων. Ένα μοντέλο με 3 Παίχτες(εργασίες) και 2 επεξεργαστές. Οι εργασίες είχαν μια τιμή (μέγεθος) με βάση την κατανομή Poisson και ο κάθε Παίχτης πρέπει να σκεφτεί και να αποφασίσει σε ποιο επεξεργαστή θα κατανέμει την εργασία του. Κάθε τριάδα από αποφάσεις των Παιχτών 1,2,3 είναι ένας Κατανεμημένος Αλγόριθμος.

Μελετήθηκαν συγκεκριμένα 8 διαφορετικές περιπτώσεις. Η κάθε περίπτωση , διαφέρει από την άλλη ανάλογα με το πόση επικοινωνία που υπάρχει ανάμεσα στους 3 Παίχτες. Για κάθε περίπτωση υπολογίζεται η πιθανότητα επιτυχίας του Αλγορίθμου για την περίπτωση αυτή και εξετάζεται κατά πόσο ο Αλγόριθμος είναι Βέλτιστος.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Γενικά	1
	1.2 Μοντέλο Κατανεμημένου Συστήματος	2
	1.3 Κατανομή Poisson	4
	1.4 Επικοινωνία στο Κατανεμημένο Σύστημα	5
	1.5 Απαρίθμηση όλων των Κατανεμημένων Αλγορίθμων	7
	1.6 Σύγκριση Κατανεμημένων Αλγορίθμων	9
Κεφάλαιο 2	Περίπτωση 1 : Καμία Επικοινωνία.....	10
	2.1 Περιγραφή	10
	2.2 Αλγόριθμος	11
	2.3 Πειράματα	11
	2.4 Ανάλυση Αλγορίθμου	12
	2.5 Παρατηρήσεις	15
	2.6 Βελτιστότητα Αλγορίθμου	15
Κεφάλαιο 3	Περίπτωση 2: Ο Παίκτης2 γνωρίζει τον Παίκτη3.....	20
	3.1 Περιγραφή	20
	3.2 Αλγόριθμος	21
	3.3 Πειράματα	21
	3.4 Ανάλυση Αλγορίθμου	21
	3.5 Παρατηρήσεις	23
	3.6 Βελτιστότητα Αλγορίθμου	24
Κεφάλαιο 4	Περίπτωση 3: Ο Παίκτης2 γνωρίζει τον Παίκτη3	
	Ο Παίκτης1 γνωρίζει τον Παίκτη3.....	25
	4.1 Περιγραφή	25
	4.2 Αλγόριθμος	26
	4.3 Πειράματα	26
	4.4 Ανάλυση Αλγορίθμου	27
	4.5 Παρατηρήσεις	32
	4.6 Βελτιστότητα Αλγορίθμου	32

Κεφάλαιο 5	Περίπτωση 4: Ο Παίχτης3 γνωρίζει τον Παίχτη1	
	Ο Παίχτης3 γνωρίζει τον Παίχτη2.....	34
	5.1 Περιγραφή	34
	5.2 Αλγόριθμος	35
	5.3 Πειράματα	35
	5.4 Ανάλυση Αλγορίθμου	36
	5.5 Παρατηρήσεις	39
	5.6 Βελτιστότητα Αλγορίθμου	39
Κεφάλαιο 6	Περίπτωση 5: Ο Παίχτης2 γνωρίζει τον Παίχτη3	
	Ο Παίχτης1 γνωρίζει τον Παίχτη2.....	40
	6.1 Περιγραφή	40
	6.2 Αλγόριθμος	41
	6.3 Πειράματα	41
	6.4 Ανάλυση Αλγορίθμου	42
	6.5 Παρατηρήσεις	49
	6.6 Βελτιστότητα Αλγορίθμου	50
Κεφάλαιο 7	Περίπτωση 6: Ο Παίχτης2 γνωρίζει τον Παίχτη3	
	Ο Παίχτης1 γνωρίζει τον Παίχτη2	
	Ο Παίχτης3 γνωρίζει τον Παίχτη1.....	51
	7.1 Περιγραφή	51
	7.2 Αλγόριθμος	52
	7.3 Πειράματα	52
	7.4 Ανάλυση Αλγορίθμου	52
	7.5 Παρατηρήσεις	52
	7.6 Βελτιστότητα Αλγορίθμου	52
Κεφάλαιο 8	Περίπτωση 7: Ο Παίχτης2 γνωρίζει τον Παίχτη1	
	Ο Παίχτης1 γνωρίζει τον Παίχτη2.....	53
	8.1 Περιγραφή	53
	8.2 Αλγόριθμος	54
	8.3 Πειράματα	54
	8.4 Ανάλυση Αλγορίθμου	55
	8.5 Παρατηρήσεις	63

8.6 Βελτιστότητα Αλγορίθμου	64
Κεφάλαιο 9 Περίπτωση 8: Ο Παίχτης2 γνωρίζει τον Παίχτη1	
Ο Παίχτης2 γνωρίζει τον Παίχτη3	
Ο Παίχτης1 γνωρίζει τον Παίχτη3.....	68
9.1 Περιγραφή	68
9.2 Αλγόριθμος	69
9.3 Πειράματα	70
9.4 Ανάλυση Αλγορίθμου	70
9.5 Παρατηρήσεις	70
9.6 Βελτιστότητα Αλγορίθμου	70
Κεφάλαιο 10 Συμπεράσματα.....	73
10.1 Σύνοψη	73
10.2 Συμπεράσματα	73
10.3 Μελλοντική Δουλειά	74
Βιβλιογραφία	75
Παράρτημα Α.....	Α-1
Παράρτημα Β.....	Β-1
Παράρτημα Γ.....	Γ-1

Κεφάλαιο 1

Εισαγωγή

1.1 Γενικά	1
1.2 Μοντέλο Κατανεμημένου Συστήματος	2
1.3 Κατανομή Poisson	4
1.4 Επικοινωνία στο Κατανεμημένο Σύστημα	5
1.5 Απαρίθμηση όλων των δυνατών Κατανεμημένων Αλγορίθμων	7
1.6 Σύγκριση Κατανεμημένων Αλγορίθμων	9

1.1 Γενικά

Τα Κατανεμημένα Συστήματα είναι συλλογές από ανεξάρτητους υπολογιστές οι οποίοι εμφανίζονται στους χρήστες τους ως ένα ενιαίο συνεκτικό σύστημα.

Η πληροφορία σένα Κατανεμημένο Σύστημα είναι ένας πολύ σημαντικός πόρος για τον υπολογισμό. Δηλαδή αναλόγως του πόση γνώση υπάρχει, το Κατανεμημένο Σύστημα θα δράσει καλύτερα ή χειρότερα. Στόχος είναι το Κατανεμημένο Σύστημα να αξιοποιήσει την γνώση του στο μέγιστο και να εκτελέσει τους υπολογισμούς του όσο το δυνατό αποδοτικότερα για να λύσει το συγκεκριμένο πρόβλημα.

Κάθε ένας από τους ανεξάρτητους Υπολογιστές του Κατανεμημένου Συστήματος θα τρέχει κάποιο αλγόριθμο. (A_i = Ο αλγόριθμος που τρέχει ο ανεξάρτητος Υπολογιστής i)
Μια ακολουθία από τους αλγορίθμους $\{A_1, A_2, A_3, \dots, A_n\}$ (αν έχω n τέτοιους υπολογιστές) είναι ένας Κατανεμημένος Αλγόριθμος.

Υπάρχουν πολλοί δυνατοί Κατανεμημένοι Αλγόριθμοι καθώς κάθε ανεξάρτητος υπολογιστής του Κατανεμημένου Συστήματος μπορεί να τρέξει διάφορους Αλγορίθμους.

Όταν ο κάθε ανεξάρτητος υπολογιστής τρέξει ένα αλγόριθμο ο οποίος είναι και βέλτιστος , τότε το Κατανεμημένο Σύστημα θα είναι και αυτό βέλτιστο για το συγκεκριμένο πρόβλημα.

Στην περίπτωση την δική μας το πρόβλημα μας είναι η ΛΗΨΗ ΑΠΟΦΑΣΕΩΝ. Δηλαδή αν κάθε ένας ανεξάρτητος Υπολογιστής ήταν ένας Παίχτης j ο οποίος έχοντας μια εργασία X_j υπό την ευθύνη του , θα έπρεπε να την κατανέμει σε κάποιο από τους επεξεργαστές.

Ο κάθε Παίχτης πρέπει να «παίξει» όσο το δυνατό καλύτερα , καθώς ο αριθμός και η χωρητικότητα των επεξεργαστών είναι περιορισμένα.

Στόχος είναι να ΜΗΝ ΞΕΧΕΙΛΙΣΟΥΝ οι επεξεργαστές.

Συνδέοντας τώρα τα κατανεμημένα συστήματα με το πρόβλημα της λήψης αποφάσεων , μπορούμε να πούμε ότι συνδυάζοντας τους τρόπους με τους οποίους μπορεί να παίζει ο κάθε παίχτης , παίρνουμε ένα κατανεμημένο αλγόριθμο.

Έτσι δημιουργείται το πρόβλημα της ΚΑΤΑΝΕΜΗΜΕΝΗΣ ΛΗΨΗΣ ΑΠΟΦΑΣΕΩΝ (DISTRIBUTED DECISION-MAKING) δηλαδή λήψης αποφάσεων σε κατανεμημένα συστήματα.

Συγκεκριμένα , πιο κάτω θα ασχοληθούμε με το να υπάρχουν 3 Παίχτες .Ο Παίχτης 1, ο Παίχτης 2 και ο Παίχτης 3 και 2 επεξεργαστές οι οποίοι πιο κάτω αναφέρονται σαν $bin0$ και $bin1$.

Τα κίνητρα για την μελέτη των προβλημάτων κατανεμημένης λήψης αποφάσεων είναι πολλά καθώς συναντάμε το πρόβλημα αυτό σε πολλούς τομείς της πληροφορικής σήμερα και ιδιαίτερα στα λειτουργικά συστήματα ιδιαίτερα τώρα με την αύξηση των πυρήνων στους επεξεργαστές.

1.2 Μοντέλο Κατανεμημένου Συστήματος

Το Μοντέλο Κατανεμημένου Συστήματος που μελετήθηκε στην βασική βιβλιογραφία για το πρόβλημα της κατανεμημένης λήψης αποφάσεων, [1] ήταν το εξής :

- Κάθε εργασία $X1, X2, X3$ είχε τυχαίο μέγεθος από το 0 ως το 1, με βάση την ΟΜΟΙΟΜΟΡΦΗ κατανομή. Δηλαδή με πιο απλά λόγια κάθε πραγματικός αριθμός στο διάστημα 0..1 έχει τις ίδιες πιθανότητες να έρθει με οποιοδήποτε άλλο αριθμό στο διάστημα αυτό.

- Το μέγεθος των δύο επεξεργαστών τέθηκε ίσο με 1.

Μια παραλλαγή του μοντέλου αυτού η οποία και θα μελετηθεί πιο κάτω είναι η εξής :

-Κάθε εργασία X_1, X_2, X_3 έχει τυχαίο μέγεθος με βάση την κατανομή POISSON. Συγκεκριμένα θα ασχοληθούμε με την ΔΙΑΚΡΙΤΗ κατανομή POISSON η οποία μας επιστρέφει ακέραιους αριθμούς.

Η κατανομή αυτή έχει τον πιο κάτω τύπο

$$P(X=j) = \frac{e^{-\mu} \mu^j}{j!}$$

Η παράμετρος μ τέθηκε ίση με 2 (ο λόγος θα εξηγηθεί πιο κάτω)

-Το μέγεθος των 2 επεξεργαστών τέθηκε ίσο με $2\mu=4$.

Ο λόγος που επιλέχθηκαν έτσι οι εργασίες , το μέγεθος των καλαθιών και το μέγεθος της παραμέτρου μ , είναι με βάση μια αντιστοιχία που έγινε με το μοντέλο του κατανεμημένου συστήματος που μελετήθηκε στην βασική βιβλιογραφία. Δηλαδή :

Ομοιόμορφη κατανομή :

Μέση τιμή = $1/2$

Εργασίες	X_1	X_2	X_3	
Μέση τιμή	$\frac{1}{2}$	$+$	$\frac{1}{2}$	$+$
			$\frac{1}{2}$	$=3/2$

$$\rightarrow (3/2) / 2 = 3/4 = 0.75$$

Επεξεργαστές	bin0		bin1	
Χωρητικότητα	1	$+$	1	$=2$

Έτσι με βάση την αναλογία του αθροίσματος της μέσης τιμής των εργασιών για την ομοιόμορφη κατανομή , ως προς το άθροισμα της χωρητικότητας των 2 επεξεργαστών πήραμε $3/4$.

Η χωρητικότητα των επεξεργαστών είναι η διπλάσια από την μέση τιμή της κατανομής. Με τον ανάλογο τρόπο τέθηκαν οι παράμετροι για το πρόβλημα της κατανεμημένης λήψης αποφάσεων με βάση την κατανομή Poisson

Κατανομή Poisson :

Μέση τιμή = μ

Εργασίες	X_1	X_2	X_3	
Μέση τιμή	μ	$+$	μ	$+$
			μ	$=3\mu$

$$\rightarrow 3\mu/4\mu = 3/4 = 0.75$$

Επεξεργαστές	bin0		bin1	
--------------	------	--	------	--

$$\text{Χωρητικότητα} \quad 2\mu \quad + \quad 2\mu \quad = 4\mu$$

Η χωρητικότητα των επεξεργαστών είναι διπλάσια από την μέση τιμή της κατανομής.

Για τους λόγους αυτούς επιλέχθηκε το μέγεθος των επεξεργαστών ίσο με 2μ για το πρόβλημα της κατανεμημένης λήψης αποφάσεων με βάση την κατανομή Poisson.

1.3 Κατανομή Poisson

$$P(X=j) = \frac{e^{-\mu} \mu^j}{j!}$$

[2]

Η κατανομή αυτή διαφέρει από την ομοιόμορφη κατανομή στο γεγονός ότι οι αριθμοί που θα έρθουν, έχουν διαφορετική πιθανότητα.

Παρατηρούμε ότι το παραγοντικό ($j!$) μεγαλώνει περισσότερο από το εκθετικό (μ^j).

Έτσι μεγάλοι αριθμοί (πχ 10,11,12,13) έχουν πιο λίγες πιθανότητες να έρθουν και αντίστοιχα οι μικροί αριθμοί έχουν πιο μεγάλες πιθανότητες να έρθουν (πχ 0,1,2,3,4).

Μερικά παραδείγματα

$$P(X=0) = \frac{e^{-\mu} \mu^0}{0!} = 0.1353 \quad = 13.53\%$$

$$P(X=1) = \frac{e^{-\mu} \mu^1}{1!} = 0.27067 \quad = 27.067\%$$

$$P(X=2) = \frac{e^{-\mu} \mu^2}{2!} = 0.27067 \quad = 27.067\%$$

$$P(X=3) = \frac{e^{-\mu} \mu^3}{3!} = 0.20300 \quad = 20.300\%$$

$$P(X=4) = \frac{e^{-\mu} \mu^4}{4!} = 0.09022 \quad = 9.022\%$$

$$P(X=5) = \frac{e^{-\mu} \mu^5}{5!} = 0.03608 \quad = 3.608\%$$

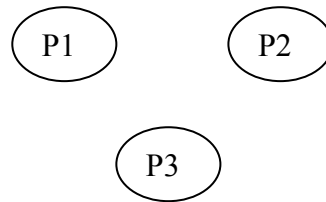
Παρατηρούμε ότι για να έρθει ένας από τους αριθμούς 0,1,2,3 η πιθανότητα είναι περίπου 88%. Όλοι οι άλλοι αριθμοί για να έρθουν έχουν πιθανότητα περίπου 12% όλοι μαζί.

Μπορούμε να παρομοιάσουμε την κατανομή Poisson ως την τροχαία κίνηση στους δρόμους , λέγοντας ότι η ώρα 2 υπάρχει η περισσότερη κίνηση και όσο περνά η ώρα η κίνηση αραιώνει και κάποια στιγμή πχ τα ξημερώματα , γίνεται αμελητέα.

1.4 Επικοινωνία στο Κατανεμημένο Σύστημα

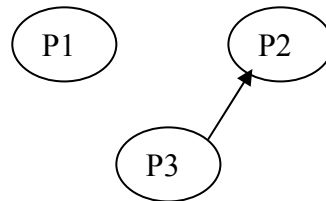
Στο συγκεκριμένο πρόβλημα κατανεμημένης λήψης αποφάσεων κάτω από την κατανομή Poisson , έχοντας 3 παίκτες και 2 μηχανές θα εξεταστούν οι πιο κάτω 8 περιπτώσεις

Περίπτωση 1:



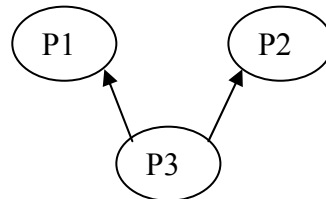
Σχήμα 1.4.1:Σχηματική αναπαράσταση Περίπτωσης 1

Περίπτωση 2:



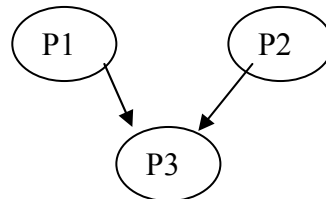
Σχήμα 1.4.2:Σχηματική αναπαράσταση Περίπτωσης 2

Περίπτωση 3:



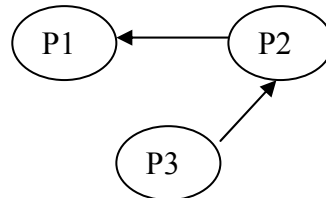
Σχήμα 1.4.3:Σχηματική αναπαράσταση Περίπτωσης 3

Περίπτωση 4:



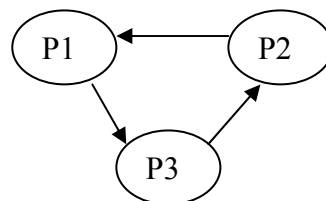
Σχήμα 1.4.4:Σχηματική αναπαράσταση Περίπτωσης 4

Περίπτωση 5:



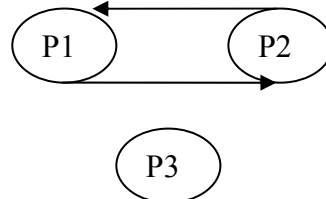
Σχήμα 1.4.5: Σχηματική αναπαράσταση Περίπτωσης 5

Περίπτωση 6:



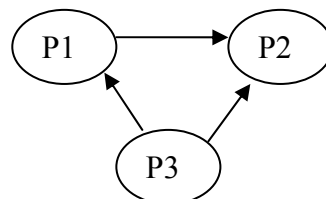
Σχήμα 1.4.6: Σχηματική αναπαράσταση Περίπτωσης 6

Περίπτωση 7:



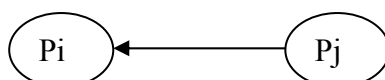
Σχήμα 1.4.7: Σχηματική αναπαράσταση Περίπτωσης 7

Περίπτωση 8:



Σχήμα 1.4.8: Σχηματική αναπαράσταση Περίπτωσης 8

Γενικά το



σημαίνει ότι

Σχήμα 1.4.9

Ο Παίχτης i γνωρίζει τον Παίχτη j . Δηλαδή το μέγεθος της εργασίας j , τον τρόπο που θα σκεφτεί να παίξει ο Παίχτης j άρα και σε ποιο επεξεργαστή θα καταλήξει. Οι 8 αυτές περιπτώσεις είναι αυτές που θα μελετηθούν αναλυτικά στα πιο κάτω κεφάλαια.

1.5 Απαρίθμηση όλων των δυνατών κατανεμημένων αλγορίθμων

Δεδομένου ότι βρήκα ένα αλγόριθμο για κάθε περίπτωση, το επόμενο βήμα είναι να δω αν αυτός ο αλγόριθμος είναι Βέλτιστος.

Αυτό μπορεί να γίνει συγκρίνοντας τον δικό μου κατανεμημένο Αλγόριθμο για κάθε περίπτωση με όλους τους δυνατούς κατανεμημένους Αλγόριθμους που μπορούν να προκύψουν.

Όπως αναφέραμε και πιο πριν ένας κατανεμημένος αλγόριθμος είναι 3άδες αλγορίθμων για τους Παίχτες 1,2,3.

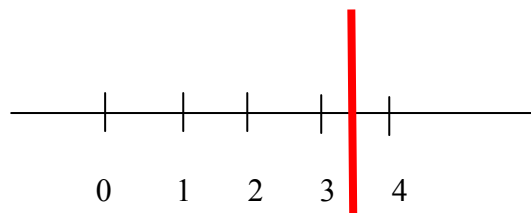
Έτσι όλοι οι πιθανοί κατανεμημένοι αλγόριθμοι είναι όλοι οι συνδυασμοί, όλων των δυνατών αλγορίθμων για κάθε παίχτη.

Κάθε αλγόριθμος για κάθε παίχτη είναι να αποφασίσει σε ποιο επεξεργαστή θα στείλει την εργασία του ο παίχτης αυτός. Οι δυνατοί αλγόριθμοι για κάθε Παίχτη εξαρτώνται από την γνώση που έχει ο Παίχτης αυτός.

Αν ο Παίχτης i δεν έχει καμία γνώση, τότε όλοι οι διαφορετικοί τρόποι με τους οποίους μπορεί να στείλει την εργασία του σε ένα από τους 2 επεξεργαστές, αναλόγως του μεγέθους της είναι συγκεκριμένοι.

Δηλαδή με πόσες ευθείες μπορώ να χωρίσω τον πιο κάτω μονοδιάστατο άξονα.

Η κάθε ευθεία στέλλει τα αριστερά της σημεία στο 0 και τα δεξιά της σημεία στο 1 και το αντίθετο.



Σχήμα 1.5.1

Για παράδειγμα η ευθεία αυτή διαχωρίζει τον άξονα σε δύο μέρη. Το ένα μέρος είναι τα μεγέθη των εργασιών τα οποία αν έρθουν θα καταλήξουν στον επεξεργαστή 0 και το

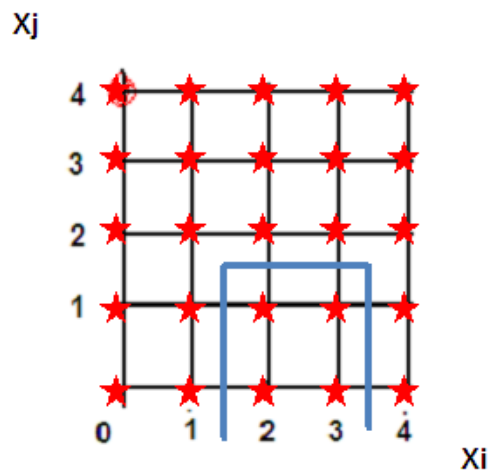
άλλο μέρος είναι τα μεγέθη των εργασιών τα οποία αν έρθουν θα καταλήξουν στον επεξεργαστή 1(και το αντίθετο)

Μέγεθος εργασίας	0	1	2	3	4
Επεξεργαστής	0	0	0	0	1
	1	1	1	1	0

Η περίπτωση όπου κάποιος Παίχτης γνωρίζει μόνο τον εαυτό του έχει 2^5 δυνατούς διαχωρισμούς. Άρα υπάρχουν 2^5 πιθανοί αλγόριθμοι για κάποιον Παίχτη που δεν έχει γνώση.

Δεν παίρνουμε τις περιπτώσεις όπου μια εργασία έχει μέγεθος ≥ 5 καθώς όποιος διαχωρισμός και να γίνει ο αλγόριθμος θα αποτυγχάνει αφού η χωρητικότητα του επεξεργαστή είναι 4.

Αν ο Παίχτης j έχει 1 γνώση, δηλαδή γνωρίζει τον εαυτό του και ακόμα ένα Παίχτη τότε μπορούν να αναπαρασταθούν όλοι οι δυνατοί διαχωρισμοί των εργασιών αναλόγως του μεγέθους τους με τον πιο κάτω τρόπο.



Σχήμα 1.5.2

Δηλαδή έχουμε τα σημεία X_i και X_j που αναπαριστούν όλους τους πιθανούς συνδυασμούς τιμών για τις εργασίες i και j . Ο πιο πάνω διαχωρισμός λέει ότι αν

$(X_j=2, X_i=0)$ τότε ο Παίχτης j θα στείλει την δική του εργασία στον επεξεργαστή 0.

$(X_j=2, X_i=1)$ τότε ο Παίχτης j θα στείλει την δική του εργασία στον επεξεργαστή 0.

$(X_j=3, X_i=0)$ τότε ο Παίχτης j θα στείλει την δική του εργασία στον επεξεργαστή 0.

$(X_j=3, X_i=1)$ τότε ο Παίχτης j θα στείλει την δική του εργασία στον επεξεργαστή 0.

Σε οποιοδήποτε άλλη περίπτωση ο Παίχτης j θα στείλει τη εργασία του στον επεξεργαστή 1 (και το αντίθετο).

Μπορούμε να πούμε ότι όλοι οι διαχωρισμοί είναι 2^{25} .

Είναι ένας τεράστιος αριθμός δυνατών διαχωρισμών.

Δεν παίρνουμε τις περιπτώσεις όπου μια εργασία έχει μέγεθος ≥ 5 καθώς όποιος διαχωρισμός και να γίνει ο αλγόριθμος θα αποτυγχάνει αφού η χωρητικότητα του επεξεργαστή είναι 4.

Αν ο Παίχτης j έχει 2 γνώσεις, δηλαδή γνωρίζει τα πάντα, τότε οι πιθανοί διαχωρισμοί είναι ακόμα πιο πολλοί καθώς θα ασχοληθούμε με 3 διαστάσεις και 125 σημεία δηλαδή με 2^{125} δυνατούς διαχωρισμούς.

Όταν βρεθούν για κάθε Παίχτη όλοι οι πιθανοί του αλγόριθμοι (ανάλογα με την γνώση του κάθε Παίχτη φυσικά) τότε έχουμε και όλους τους δυνατούς κατανεμημένους αλγόριθμους τους οποίους πρέπει να ελέγξουμε με τον κατανεμημένο αλγόριθμο που χρησιμοποιήσαμε για κάθε περίπτωση και να δούμε αν νικάει όλους τους άλλους πιθανούς κατανεμημένους αλγόριθμους. Αν όντως αυτό ισχύει, μόνο και μόνο τότε ο Αλγόριθμος μας είναι βέλτιστος.

1.6 Σύγκριση Κατανεμημένων Αλγορίθμων

Για να υπολογιστεί αν ο Αλγόριθμος που διάλεξα για κάθε περίπτωση είναι Βέλτιστος, πρέπει να χρησιμοποιηθεί ο αλγόριθμος «ΕΠΑΛΗΘΕΥΤΗΣ ΒΕΛΤΙΣΤΟΤΗΤΑΣ». Ο αλγόριθμος αυτός παίρνει σαν είσοδο όλους τους δυνατούς αλγορίθμους για κάθε Παίχτη από αρχεία. Για κάθε 3αδα αλγορίθμων των Παιχτών 1,2,3 εξετάζει τον συγκριμένο κατανεμημένο αλγόριθμο που δημιουργείται με τον δικό μου αλγόριθμο.

Αν για κάθε κατανεμημένο αλγόριθμο δηλαδή για κάθε πιθανή 3αδα αλγορίθμων των Παιχτών 1,2,3 ο αλγόριθμος μου δίνει καλύτερα αποτελέσματα τότε είναι Βέλτιστος χωρίς καμιά αμφιβολία.

Σε κάποιες περιπτώσεις δεν ήταν δυνατό ο αλγόριθμος «ΒΕΛΤΙΣΤΟΣ ΑΛΓΟΡΙΘΜΟΣ» να μας δώσει απάντηση λόγω του τεράστιου αριθμού πιθανών αλγορίθμων για κάθε Παίχτη που προκύπτουν ανάλογα με την κάθε Περίπτωση.

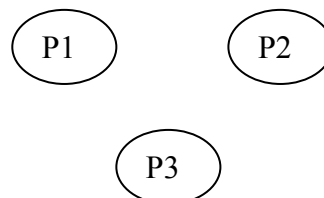
Κεφάλαιο 2

Περίπτωση 1 : «Καμία Επικοινωνία»

2.1 Περιγραφή	10
2.2 Αλγόριθμος	11
2.3 Πειράματα	11
2.4 Ανάλυση Αλγορίθμου	12
2.5 Παρατηρήσεις	15
2.6 Βελτιστότητα Αλγορίθμου	15

2.1 Περιγραφή

Είμαστε στην περίπτωση όπου δεν υπάρχει καμία επικοινωνία μεταξύ των παιχτών όπως φαίνεται στο πιο κάτω σχήμα



Σχήμα 2.1.1: Σχηματική αναπαράσταση Περίπτωσης 1

Για την περίπτωση αυτή οι παίκτες δεν έχουν και πολλές επιλογές καθώς γνωρίζουν μόνο τον εαυτό τους και αυτό είναι η μόνη πληροφορία που έχουν. Η σκέψη για τον αλγόριθμο αυτό είναι με βάση κάποιο κατώφλι, ο κάθε παίκτης να αποφασίζει αν η εργασία του είναι “μικρή” ή “μεγάλη” και ανάλογα να την τοποθετήσει στο καλάθι με τις “μικρές” εργασίες ή στο καλάθι με τις “μεγάλες” εργασίες. Το καλάθι για τις “μικρές” εργασίες είναι το καλάθι 0 και το καλάθι για τις “μεγάλες” εργασίες είναι το καλάθι 1.

2.2 Αλγόριθμος

IF $X3 \leq T$

$P3(X3) = \text{bin } 0$

ELSE

$P3(X3) = \text{bin } 1$

IF $X2 \leq T$

$P2(X2) = \text{bin } 0$

ELSE

$P2(X2) = \text{bin } 1$

IF $X1 \leq T$

$P1(X1) = \text{bin } 0$

ELSE

$P1(X1) = \text{bin } 1$

Το κατώφλι T επιλέχθηκε να είναι το 2 και ο λόγος που επιλέχθηκε ο αριθμός αυτός θα εξηγηθεί στο υποκεφάλαιο Αποδοτικότητα Αλγορίθμου.

2.3 Πειράματα

Με την χρήση της συνάρτησης της MATLAB `poissrnd(μ)` που επιστρέφει ένα ακέραιο αριθμό με βάση την κατανομή Poisson και παράμετρο $\mu=2$, έτρεξα τον αλγόριθμο αυτό για 1 εκατομμύριο επαναλήψεις και κρατούσα σε 2 μετρητές τις περιπτώσεις όπου ο αλγόριθμος αποτύχανε και τις περιπτώσεις όπου επιτύχανε. Η διαδικασία αυτή επαναλήφθηκε αρκετές φορές. Μερικά από τα πειραματικά αποτελέσματα φαίνονται στον πιο κάτω Πίνακα.

Επανάληψη	Πιθανότητα Επιτυχίας (%)
1	60.2740
2	60.2662
3	60.2148
4	60.1932

5	60.3098
6	60.2516

Πίνακας 2.3.1: Πειραματική δειγματοληψία Περίπτωσης 1

2.4 Ανάλυση Αλγορίθμου

P [να ξεχειλίσσει το καλάθι 0 ή να ξεχειλίσσει το καλάθι 1]

$=P$ [να είναι και οι 3 εργασίες μικρές και να μουν μαζί και να ξεχειλίσουν]

$\cup P$ [οι δύο να είναι μικρές και να μουν μαζί και η άλλη να είναι μεγάλη και να μπει μόνη της και να ξεχειλίσσει]

$\cup P$ [η μια να είναι μικρή και να μπει μόνη της και οι άλλες 2 μεγάλες και να μουν μαζί και να ξεχειλίσουν]

$\cup P$ [να είναι και οι τρεις εργασίες μεγάλες και να μουν όλες μαζί και να ξεχειλίσουν]

$= P [X1 \leq T \cap X2 \leq T \cap X3 \leq T \cap X1+X2+X3 > 2\mu]$

$\cup P [X1 \leq T \cap X2 \leq T \cap X3 > T \cap X3 > 2\mu]$

$\cup P [X1 \leq T \cap X2 > T \cap X3 \leq T \cap X2 > 2\mu]$

$\cup P [X1 > T \cap X2 \leq T \cap X3 \leq T \cap X1 > 2\mu]$

$\cup P [X1 > T \cap X2 > T \cap X3 \leq T \cap X1+X2 > 2\mu]$

$\cup P [X1 > T \cap X2 \leq T \cap X3 > T \cap X1+X3 > 2\mu]$

$\cup P [X1 \leq T \cap X2 > T \cap X3 > T \cap X2+X3 > 2\mu]$

$\cup P [X1 > T \cap X2 > T \cap X3 > T \cap X1+X2+X3 > 2\mu]$

Παρατηρούμε ότι τα ενδεχόμενα είναι ανεξάρτητα μεταξύ τους δηλαδή δεν έχουν τομές άρα όπου έχουμε το σύμβολο $\cup$ θα βάλουμε το +.

Ακόμα παρατηρούμε ότι κάποια ενδεχόμενα είναι ακριβώς τα ίδια.

$\rightarrow P[\text{αποτυχίας}] =$

$P [X1 \leq T \cap X2 \leq T \cap X3 \leq T \cap X1+X2+X3 > 2\mu]$

$+ 3 * P [X1 \leq T \cap X2 \leq T \cap X3 > T \cap X3 > 2\mu]$

$+ 3 * P [X1 > T \cap X2 > T \cap X3 \leq T \cap X1+X2 > 2\mu]$

$+ P [X1 > T \cap X2 > T \cap X3 > T \cap X1+X2+X3 > 2\mu]$

Θέτουμε

$$A = P [X_1 \leq T \cap X_2 \leq T \cap X_3 \leq T \cap X_1 + X_2 + X_3 > 2\mu]$$

$$B = P [X_1 \leq T \cap X_2 \leq T \cap X_3 > T \cap X_3 > 2\mu]$$

$$C = P [X_1 > T \cap X_2 > T \cap X_3 \leq T \cap X_1 + X_2 > 2\mu]$$

$$D = P [X_1 > T \cap X_2 > T \cap X_3 > T \cap X_1 + X_2 + X_3 > 2\mu]$$

$$\rightarrow P[\text{αποτυχίας}] = A + 3B + 3C + D$$

$$A = P [X_1 \leq 2 \cap X_2 \leq 2 \cap X_3 \leq 2 \cap X_1 + X_2 + X_3 > 2\mu]$$

Αναλυτικά οι περιπτώσεις όπου οι εργασίες είναι ≤ 2 και το άθροισμα τους είναι > 4

X_1	X_2	X_3
2	2	1
2	1	2
1	2	2
2	2	2

$$\begin{aligned} A &= P [X_1 = 2 \cap X_2 = 2 \cap X_3 = 1] + P [X_1 = 2 \cap X_2 = 1 \cap X_3 = 2] \\ &+ P [X_1 = 1 \cap X_2 = 2 \cap X_3 = 2] + P [X_1 = 2 \cap X_2 = 2 \cap X_3 = 2] \\ &= 3 * P [X_1 = 2 \cap X_2 = 2 \cap X_3 = 1] + P [X_1 = 2 \cap X_2 = 2 \cap X_3 = 2] \\ &= 3 * \left(\frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^1}{1!} \right) + \left(\frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^2}{2!} \right) \\ &= 3 * \left[\left(\frac{e^{-\mu} \mu^2}{2!} \right)^2 * \frac{e^{-\mu} \mu^1}{1!} \right] + \left(\frac{e^{-\mu} \mu^2}{2!} \right)^3 \end{aligned}$$

$$B = P [X_1 \leq 2 \cap X_2 \leq 2 \cap X_3 > 2 \cap X_3 > 2\mu]$$

$$= P [X_1 \leq 2 \cap X_2 \leq 2 \cap X_3 > 2\mu]$$

$$= P [X_1 \leq 2] * P [X_2 \leq 2] * P [X_3 > 2\mu]$$

$$= \sum_{\substack{j \geq 0: \\ j=0}}^2 \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j=0}}^2 \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!}$$

$$= \left(\sum_{\substack{j \geq 0: \\ j=0}}^2 \frac{e^{-\mu} \mu^j}{j!} \right)^2 * \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!}$$

$$C = P [X_1 > 2 \cap X_2 > 2 \cap X_3 \leq 2 \cap X_1 + X_2 > 2\mu]$$

$$= P [X_1 > 2 \cap X_2 > 2 \cap X_3 \leq 2]$$

$$= P [X_1 > 2] * P [X_2 > 2] * P [X_3 \leq 2\mu]$$

$$= \sum_{\substack{j \geq 0: \\ j > 2}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j > 2}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j = 0}}^2 \frac{e^{-\mu} \mu^j}{j!}$$

$$= \left(\sum_{\substack{j \geq 0: \\ j > 2}} \frac{e^{-\mu} \mu^j}{j!} \right)^2 * \sum_{\substack{j \geq 0: \\ j = 0}}^2 \frac{e^{-\mu} \mu^j}{j!}$$

$$D = P [X_1 > 2 \cap X_2 > 2 \cap X_3 > 2 \cap X_1 + X_2 + X_3 > 2\mu]$$

$$= P [X_1 > 2 \cap X_2 > 2 \cap X_3 > 2]$$

$$= P [X_1 > 2] * P [X_2 > 2] * P [X_3 > 2]$$

$$= \sum_{\substack{j \geq 0: \\ j > 2}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j > 2}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j > 2}} \frac{e^{-\mu} \mu^j}{j!}$$

$$= \left(\sum_{\substack{j \geq 0: \\ j > 2}} \frac{e^{-\mu} \mu^j}{j!} \right)^3$$

$$\rightarrow P[\text{αποτυχία}] = A + 3B + 3C + D$$

$$= 3 * \left[\left(\frac{e^{-\mu} \mu^2}{2!} \right)^2 * \frac{e^{-\mu} \mu^1}{1!} \right] + \left(\frac{e^{-\mu} \mu^2}{2!} \right)^3 + 3 * \left[\left(\sum_{\substack{j \geq 0: \\ j = 0}}^2 \frac{e^{-\mu} \mu^j}{j!} \right)^2 * \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} \right]$$

$$+ 3 * \left[\left(\sum_{\substack{j \geq 0: \\ j > 2}} \frac{e^{-\mu} \mu^j}{j!} \right)^2 * \sum_{\substack{j \geq 0: \\ j = 0}}^2 \frac{e^{-\mu} \mu^j}{j!} \right] + \left(\sum_{\substack{j \geq 0: \\ j > 2}} \frac{e^{-\mu} \mu^j}{j!} \right)^3$$

$$\rightarrow P[\text{αποτυχία}] = 0.3976632$$

$$= 39.76632\%$$

$$\rightarrow P[\text{επιτυχία}] = 1 - P[\text{αποτυχία}] = 0.6023368$$

$$= 60.23368\%$$

2.5 Παρατηρήσεις

Για την περίπτωση αυτή που κανένας παίχτης δεν έχει επικοινωνία με κανένα άλλο παίχτη βλέπουμε ότι ο αλγόριθμος αυτός έχει πιθανότητα να επιτύχει 60.23368%.

Είναι λογικό η περίπτωση αυτή να έχει την πιο μικρή πιθανότητα επιτυχίας από όλες τις άλλες περιπτώσεις όπου υπάρχει έστω και μια επικοινωνία μεταξύ δύο παιχτών.

Παρατηρούμε ότι η πιθανότητα επιτυχίας που πήραμε με την ανάλυση του αλγορίθμου , συμφωνεί με τα πειράματα που έγιναν.

2.6 Βελτιστότητα Αλγορίθμου

Με βάση αυτά που ορίστηκαν από την αρχή , δηλαδή ότι το μέγεθος των καλάθιών είναι 2μ , δηλαδή 4 , μπορούμε να συγκεκριμενοποιήσουμε τις περιοχές όπου για μια περίπτωση επικοινωνίας (πχ περίπτωση καμιάς επικοινωνίας) κάθε αλγόριθμος κερδίζει ή κάθε αλγόριθμος χάνει ή ένας αλγόριθμος είναι καλύτερος από κάποιο άλλο.

Παρατηρούμε ότι κάθε αλγόριθμος κερδίζει αν $X1+X2+X3 \leq 2\mu$ καθώς όπως και να μπου οι 3 εργασίες στα 2 καλάθια πάντα θα υπάρχει επιτυχία.

Είναι γεγονός ότι έχοντας 2 καλάθια και 3 εργασίες , σίγουρα στο ένα καλάθι θα μπου τουλάχιστον οι 2 εργασίες από τις 3 (Αρχή του Περιστερώνα).

Έτσι παρατηρούμε ότι αν $X1+X2+X3 \geq 9$ κανένας αλγόριθμος , ακόμα και ο βέλτιστος για κάθε περίπτωση , δεν θα μπου να επιτύχει, καθώς είτε μια εργασία από τις τρεις είναι μεγαλύτερη του 2μ ,είτε ανά 2 οι εργασίες έχουν άθροισμα μεγαλύτερο του 2μ.

Ο πιο κάτω πίνακας δείχνει όλες τις περιπτώσεις όπου $X1+X2+X3=9$. Εννοείται ότι οι τιμές των $X1, X2, X3$ μπου να έρθουν με οποιαδήποτε σειρά.

X1	X2	X3
0	9	0
0	8	1
0	7	2
0	6	3
0	5	4
1	4	4
1	5	3

1	2	6
1	1	7
2	4	3
2	5	2
3	3	3

Πίνακας 2.6.1 : Περιπτώσεις όπου $X1+X2+X3=9$

Οι περιπτώσεις όπου ένας Αλγόριθμος μπορεί να έχει διαφορετικά αποτελέσματα από ένα άλλο είναι οι περιπτώσεις όπου

$$X1+X2+X3=5$$

$$X1+X2+X3=6$$

$$X1+X2+X3=7$$

$$X1+X2+X3=8$$

και πιο συγκεκριμένα κάποιες υποπεριπτώσεις των περιπτώσεων αυτών , καθώς για τον ίδιο λόγο που ανέφερα πιο πάνω , όλοι οι αλγόριθμοι αποτυγχάνουν.

X1	X2	X3
2	2	1
3	2	0
3	1	1
4	1	0
5	0	0

X1	X2	X3
2	2	2
4	1	1
3	3	0
3	2	1
4	2	0
6	0	0
5	1	0

X1	X2	X3
3	3	1
4	2	1
2	2	3
4	3	0
5	1	1
5	2	0
6	1	0
7	0	0

X1	X2	X3
4	4	0
4	3	1
2	2	4
3	3	2
7	0	1
6	2	0
6	1	1
8	0	0
5	3	0
5	2	1

Πίνακας 2.6.2: Περιπτώσεις όπου οι αλγόριθμοι έχουν διαφορετικά αποτελέσματα

Τώρα θα αναλύσουμε τον αλγόριθμο για κατώφλι $T=1, T=2, T=3, T \geq 4$.

Θα εξεταστούν μόνο οι περιπτώσεις για τις οποίες οι αλγόριθμοι μπορούν να έχουν διαφορετικό αποτέλεσμα δηλαδή οι περιπτώσεις που μας δίνει ο πιο πάνω πίνακας.

$$1) X_1 + X_2 + X_3 = 5$$

X1	X2	X3	T=1	T=2	T=3	T>=4
2	2	1	✓	×	×	×
3	2	0	×	✓	×	×
3	1	1	✓	✓	×	×
4	1	0	✓	✓	✓	×

Πίνακας 2.6.3: Συγκρίσεις κατωφλίων για $X_1 + X_2 + X_3 = 5$

$$2) X_1 + X_2 + X_3 = 6$$

X1	X2	X3	T=1	T=2	T=3	T>=4
2	2	2	×	×	×	×
4	1	1	✓	✓	✓	×
3	3	0	×	×	×	×
3	2	1	×	✓	×	×
4	2	0	×	✓	✓	×

Πίνακας 2.6.4: Συγκρίσεις κατωφλίων για $X_1 + X_2 + X_3 = 6$

$$3) X_1 + X_2 + X_3 = 7$$

X1	X2	X3	T=1	T=2	T=3	T>=4
3	3	1	×	×	×	×
4	2	1	×	✓	✓	×
2	2	3	×	✓	×	×
4	3	0	×	×	✓	×

Πίνακας 2.6.5: Συγκρίσεις κατωφλίων για $X_1 + X_2 + X_3 = 7$

$$4) X_1 + X_2 + X_3 = 8$$

X1	X2	X3	T=1	T=2	T=3	T>=4
4	3	1	×	×	×	×
4	2	2	×	✓	✓	×
4	4	0	×	×	×	×

Πίνακας 2.6.6: Συγκρίσεις κατωφλίων για $X_1 + X_2 + X_3 = 8$

Θα συγκρίνουμε τα κατώφλια μεταξύ τους για να δούμε πιο κατώφλι μπορεί να θεωρηθεί βέλτιστο για τον αλγόριθμο αυτό.

Παρατηρούμε ότι για $T \geq 4$ ποτέ δεν κερδίζει ο αλγόριθμος.

Αν συγκρίνουμε τα κατώφλια $T=1$ και $T=2$ από τους πιο πάνω πίνακες παρατηρούμε ότι η μόνη περίπτωση όπου το $T=1$ είναι καλύτερο από το $T=2$ είναι όταν $X1=2$ και $X2=2$ και $X3=1$ ή $X1=2$ και $X2=1$ και $X3=2$ ή $X1=1$ και $X2=2$ και $X3=2$.

$P[\text{ο } T=1 \text{ νικάει το } T=2]$

$$= P[X1=2] * P[X2=2] * P[X3=1] + P[X1=2] * P[X2=1] * P[X3=2] \\ + P[X1=1] * P[X2=2] * P[X3=2] \\ = 3 * (P[X1=2] * P[X2=2] * P[X3=1])$$

$$= 3 * \left(\frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^1}{1!} \right)$$

$$= 3 * \left(\frac{e^{-3\mu} \mu^5}{2! * 2!} \right) = 3 * \left(\frac{e^{-6} 2^5}{4} \right) = 3 * 32/4 * e^{-6} = 24 * e^{-6}$$

Δηλαδή οι περιπτώσεις όπου το $T=1$ είναι καλύτερο από το $T=2$ έχουν πιθανότητα να έρθουν 5.9490052%.

Τώρα θα δούμε τις περιπτώσεις όπου το $T=2$ είναι καλύτερο από το $T=1$ με βάση τα αποτελέσματα των πιο πάνω πινάκων.

Το $T=2$ είναι καλύτερο από το $T=1$ για τον αλγόριθμο αυτό στις περιπτώσεις όπου

$X1=3$ και $X2=2$ και $X3=0$ ή $X1=3$ και $X2=2$ και $X3=1$ ή
 $X1=4$ και $X2=2$ και $X3=0$ ή $X1=4$ και $X2=2$ και $X3=1$ ή
 $X1=2$ και $X2=2$ και $X3=3$ ή $X1=4$ και $X2=2$ και $X3=2$

$P[\text{ο } T=2 \text{ νικάει το } T=1]$

$$= 6 * (P[X1=3] * P[X2=2] * P[X3=0]) + 6 * (P[X1=3] * P[X2=2] * P[X3=1]) \\ + 6 * (P[X1=4] * P[X2=2] * P[X3=0]) + 6 * (P[X1=4] * P[X2=2] * P[X3=1]) \\ + 3 * (P[X1=2] * P[X2=2] * P[X3=3]) + 3 * (P[X1=4] * P[X2=2] * P[X3=2])$$

$$= 6 * \left(\frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^0}{0!} \right) + 6 * \left(\frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^1}{1!} \right) +$$

$$6 * \left(\frac{e^{-\mu} \mu^4}{4!} * \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^0}{0!} \right) + 6 * \left(\frac{e^{-\mu} \mu^4}{4!} * \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^1}{1!} \right) +$$

$$3 * \left(\frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^3}{3!} \right) + 3 * \left(\frac{e^{-\mu} \mu^4}{4!} * \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^2}{2!} \right)$$

$$=6*\left(\frac{e^{-3\mu}\mu^5}{3!*2!}\right) + 6*\left(\frac{e^{-3\mu}\mu^6}{3!*2!}\right) + 6*\left(\frac{e^{-3\mu}\mu^6}{4!*2!}\right) + 6*\left(\frac{e^{-3\mu}\mu^7}{4!*2!}\right) + 3*\left(\frac{e^{-3\mu}\mu^7}{3!*2!*2!}\right) + 3*\left(\frac{e^{-3\mu}\mu^8}{4!*2!*2!}\right)$$

$$=16 * e^{-6} + 32 * e^{-6} + 8 * e^{-6} + 16 * e^{-6} + 16 * e^{-6} + 8 * e^{-6} = 96 * e^{-6} > 24 * e^{-6}$$

➔ P [ο T=2 νικάει το T=1] > P [ο T=1 νικάει το T=2]

➔ Κατώφλι T=2 καλύτερο από το κατώφλι T=1

Τώρα θα συγκρίνουμε τα κατώφλια T=2 και T=3.

Παρατηρούμε ότι ο αλγόριθμος είναι καλύτερος για T=3 μόνο στο σημείο

X1=4 και X2=3 και X3=0

➔ P [ο T=3 νικάει το T=2] =

$$6* (P[X1=4] * P[X2=3] * P[X3=0])$$

$$=6*\left(\frac{e^{-\mu}\mu^4}{4!} * \frac{e^{-\mu}\mu^3}{3!} * \frac{e^{-\mu}\mu^0}{0!}\right) = 6* \frac{e^{-3\mu}\mu^7}{4!*3!} = 5.33 * e^{-6}$$

➔ P [ο T=2 νικάει το T=3] =

$$6* (P[X1=3] * P[X2=2] * P[X3=0]) + 3* (P[X1=3] * P[X2=1] * P[X3=1]) +$$

$$6* (P[X1=3] * P[X2=2] * P[X3=1]) + 3* (P[X1=2] * P[X2=2] * P[X3=3])$$

$$=6*\left(\frac{e^{-\mu}\mu^3}{3!} * \frac{e^{-\mu}\mu^2}{2!} * \frac{e^{-\mu}\mu^0}{0!}\right) + 3*\left(\frac{e^{-\mu}\mu^3}{3!} * \frac{e^{-\mu}\mu^1}{1!} * \frac{e^{-\mu}\mu^1}{1!}\right) +$$

$$6*\left(\frac{e^{-\mu}\mu^3}{3!} * \frac{e^{-\mu}\mu^2}{2!} * \frac{e^{-\mu}\mu^1}{1!}\right) + 3*\left(\frac{e^{-\mu}\mu^2}{2!} * \frac{e^{-\mu}\mu^2}{2!} * \frac{e^{-\mu}\mu^3}{3!}\right)$$

$$=6*\left(\frac{e^{-3\mu}\mu^5}{3!*2!}\right) + 3*\left(\frac{e^{-3\mu}\mu^5}{3!}\right) + 6*\left(\frac{e^{-3\mu}\mu^6}{3!*2!}\right) + 6*\left(\frac{e^{-3\mu}\mu^7}{3!*2!*2!}\right)$$

$$=16 * e^{-6} + 16 * e^{-6} + 32 * e^{-6} + 32 * e^{-6} = 96 * e^{-6} > 5.33 * e^{-6}$$

➔ P [ο T=2 νικάει το T=3] > P [ο T=3 νικάει το T=2]

➔ Κατώφλι T=2 καλύτερο από το κατώφλι T=3.

Για τους πιο πάνω λόγους το κατώφλι T για την περίπτωση της Καμίας Επικοινωνίας επιλέχθηκε T=2.

Τρέξαμε τον Αλγόριθμο «ΒΕΛΤΙΣΤΟΣ ΑΛΓΟΡΙΘΜΟΣ» ο οποίος συγκρίνει αυτόν τον Αλγόριθμο με όλους τους πιθανούς άλλους Αλγόριθμους για την Περίπτωση αυτή και πήραμε την απάντηση ότι ο Αλγόριθμος είναι Βέλτιστος.

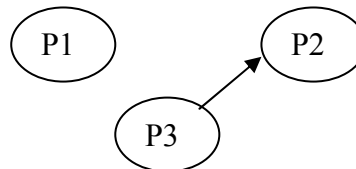
Κεφάλαιο 3

Περίπτωση 2 : Ο Παίχτης 2 γνωρίζει τον Παίχτη 3

3.1 Περιγραφή	20
3.2 Αλγόριθμος	21
3.3 Πειράματα	21
3.4 Ανάλυση Αλγορίθμου	21
3.5 Παρατηρήσεις	23
3.6 Βελτιστότητα Αλγορίθμου	24

3.1 Περιγραφή

Είμαστε στην περίπτωση όπου ο Παίχτης 2 γνωρίζει τον Παίχτη 3 και ο Παίχτης 1 και ο Παίχτης 3 δεν έχουν καμιά γνώση, όπως φαίνεται στο πιο κάτω σχήμα.



Σχήμα 3.1.1: Σχηματική αναπαράσταση Περίπτωσης 2

Για την περίπτωση αυτή θα μπορούσαν οι Παίχτες να παίξουν ως εξής.

Ο Παίχτης 3 να πηγαίνει πάντα σένα από τα δυο καλάθια και έτσι τώρα το παιχνίδι με 3 παίχτες και 2 ίσα καλάθια, θα αναχθεί σε παιχνίδι με 2 παίχτες και 2 άνισα καλάθια. Έτσι ο Παίχτης 1 θα πάει στο αντίθετο καλάθι από τον Παίχτη 3 δηλαδή στο “μεγάλο” καλάθι και ο Παίχτης 2 θα παίξει όσο πιο καλά μπορεί. Δηλαδή αν χωράει με το Παίχτη 3 θα μπει μαζί του αλλιώς θα μπει με τον Παίχτη 1 ελπίζοντας ότι δεν θα υπερχειλίσει το καλάθι.

3.2 Αλγόριθμος

$P3(X3) = \text{bin } 0$

$P1(X1) = \text{bin } 1$

IF $X3+X2 \leq 2\mu$

$P2(X2, X3) = \text{bin } 0$

ELSE

$P2(X2, X3) = \text{bin } 1$

3.3 Πειράματα

Με την χρήση της συνάρτησης της MATLAB `poissrnd(μ)` που επιστρέφει ένα ακέραιο αριθμό με βάση την κατανομή Poisson και παράμετρο $\mu=2$, έτρεξα τον αλγόριθμο αυτό για 1 εκατομμύριο επαναλήψεις και κρατούσα σε 2 μετρητές τις περιπτώσεις όπου ο αλγόριθμος αποτύχανε και τις περιπτώσεις όπου επιτύχανε. Η διαδικασία αυτή επαναλήφθηκε αρκετές φορές. Μερικά από τα πειραματικά αποτελέσματα φαίνονται στον πιο κάτω Πίνακα.

Επανάληψη	Πιθανότητα Επιτυχίας (%)
1	71.5932
2	71.6167
3	71.4505
4	71.6198
5	71.5862
6	71.5731

Πίνακας 3.3.1: Πειραματική δειγματοληψία Περίπτωσης 2

3.4 Ανάλυση Αλγορίθμου

P [να ξεχειλίσει το καλάθι 0 ή να ξεχειλίσει το καλάθι 1]

$= P [X3 > 2\mu \cup X1 > 2\mu \cup (X2 + X3 > 2\mu \cap X1 + X2 > 2\mu)]$

$= P [(X3 > 2\mu \cup X1 > 2\mu) \cup (X2 + X3 > 2\mu \cap X1 + X2 > 2\mu)]$

$$= P [X_3 > 2\mu \cup X_1 > 2\mu] + P [X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu] \\ - P [(X_3 > 2\mu \cup X_1 > 2\mu) \cap (X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu)]$$

Θέτουμε

$$A = P [X_3 > 2\mu \cup X_1 > 2\mu]$$

$$B = P [X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu]$$

$$C = P [(X_3 > 2\mu \cup X_1 > 2\mu) \cap (X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu)]$$

$$\rightarrow P[\text{αποτυχία}] = A + B - C$$

$$A = P [X_3 > 2\mu \cup X_1 > 2\mu]$$

$$= P [X_3 > 2\mu] + P[X_1 > 2\mu] - P [X_3 > 2\mu \cap X_1 > 2\mu]$$

$$= \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} + \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} - \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!}$$

$$= 2 * \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} - \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!}$$

$$= \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} * (2 - \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!})$$

$$B = P [X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu]$$

$$= \sum_{b \geq 0} (P [X_2 = b] \cap P [X_3 > 2\mu - b \cap X_1 > 2\mu - b])$$

$$= \sum_{b \geq 0} (P [X_2 = b] * P [X_3 > 2\mu - b] * P [X_1 > 2\mu - b])$$

$$= \sum_{b \geq 0} \left(\frac{e^{-\mu} \mu^b}{b!} * \sum_{\substack{j \geq 0: \\ j > 2\mu - b}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j > 2\mu - b}} \frac{e^{-\mu} \mu^j}{j!} \right)$$

$$= \sum_{b \geq 0} \left(\frac{e^{-\mu} \mu^b}{b!} * \left(\sum_{\substack{j \geq 0: \\ j > 2\mu - b}} \frac{e^{-\mu} \mu^j}{j!} \right)^2 \right)$$

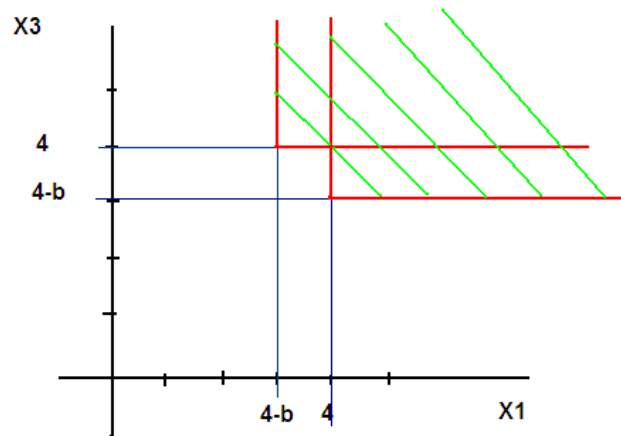
$$C = P [(X_3 > 2\mu \cup X_1 > 2\mu) \cap (X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu)]$$

$$= \sum_{b \geq 0} (P [X_2 = b] \cap P [X_3 > 2\mu \cup X_1 > 2\mu] \cap (X_3 > 2\mu - b \cap X_1 > 2\mu - b))$$

$$= \sum_{b \geq 0} (P[X_2=b] \cap P[(X_3 > 2\mu \cap X_3 > 2\mu-b \cap X_1 > 2\mu-b) \cup (X_3 > 2\mu \cap X_3 > 2\mu-b \cap X_1 > 2\mu-b)])$$

$$= \sum_{b \geq 0} (P[X_2=b] \cap P[(X_3 > 2\mu \cap X_1 > 2\mu-b) \cup (X_3 > 2\mu \cap X_1 > 2\mu-b)])$$

Θα αποδοθεί στις 2 διαστάσεις η πιο πάνω πιθανότητα ώστε να έχουμε ένα πιο απλοποιημένο ενδεχόμενο



Σχήμα 3.4.1: Αναπαράσταση πιθανότητας στις 2 διαστάσεις

$$= \sum_{b \geq 0} (P[X_2=b] \cap P[(2\mu-b < X_3 \leq 2\mu \cap X_1 > 2\mu) \cup (X_3 > 2\mu \cap X_1 > 2\mu-b)])$$

$$= \sum_{b \geq 0} (P[X_2=b] * P[(2\mu-b < X_3 \leq 2\mu * X_1 > 2\mu) + (X_3 > 2\mu * X_1 > 2\mu-b)])$$

$$= \sum_{b \geq 0} \left(\frac{e^{-\mu} \mu^b}{b!} * \left(\sum_{\substack{j \geq 0: \\ j > 2\mu-b}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} + \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j > 2\mu-b}} \frac{e^{-\mu} \mu^j}{j!} \right) \right)$$

$$= \sum_{b \geq 0} \left[\frac{e^{-\mu} \mu^{-b}}{b!} * \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} * \left(\sum_{\substack{j \geq 0: \\ j > 2\mu-b}} \frac{e^{-\mu} \mu^j}{j!} + \sum_{\substack{j \geq 0: \\ j > 2\mu-b}} \frac{e^{-\mu} \mu^j}{j!} \right) \right]$$

$$\rightarrow P[\text{αποτυχία}] = A+B-C=0.2841915 \quad = 28.41915\%$$

$$\rightarrow P[\text{επιτυχία}] = 1 - P[\text{αποτυχία}] = 0.7158085 \quad = 71.58085\%$$

3.5 Παρατηρήσεις

Καταρχήν παρατηρούμε ότι το αποτέλεσμα που βγήκε κατά την ανάλυση είναι πολύ κοντά στα αποτελέσματα που πήραμε με τα πειράματα.

Ακόμα για την περίπτωση αυτή , μιας και υπάρχει η γνώση του Παίχτη 2 , έπρεπε η γνώση αυτή να αξιοποιηθεί κατάλληλα έτσι ώστε να έχω πιθανότητα επιτυχίας μεγαλύτερη από την πιθανότητα επιτυχίας της Περίπτωσης 1 που δεν υπήρχε καμιά επικοινωνία. Φαίνεται να αξιοποιήθηκε κατά πολύ η γνώση αυτή και από 60.23368% πιθανότητα επιτυχίας που υπήρχε στην Περίπτωση 1 , τώρα η πιθανότητα επιτυχίας αυξήθηκε στο 71.58085% στην Περίπτωση 2.

Έτσι μπορούμε να πούμε ότι η ακμή μεταξύ του Παίχτη 2 και του Παίχτη 3, δηλαδή η μια επιπλέον γνώση, αξίζει 11.34717% .

3.6 Βελτιστότητα Αλγορίθμου

Με το να τοποθετείται πάντα ο Παίχτης 3 στο καλάθι 0, το παιχνίδι μετατρέπεται σε παιχνίδι με 2 παίκτες και 2 άνισα καλάθια χωρητικότητας 2μ και 2μ-X3 αντίστοιχα.

Έτσι μπορούμε να συνεχίσουμε λέγοντας ότι οι δύο παίκτες που απέμειναν θα πράξουν όσο καλύτερα μπορούν. Ο Παίχτης 1 ο οποίος δεν έχει καμιά γνώση θα πάει στο καλάθι το οποίο έχει λιγότερες πιθανότητες να ξεχειλίζει και αυτό είναι το καλάθι 1 και ο Παίχτης 2 θα εκμεταλλευτεί την γνώση του για να δει αν μπορεί να μπει μαζί με τον Παίχτη 3 στο καλάθι 0. Αν αυτό δεν μπορεί να γίνει η μόνη επιλογή που έχει ο Παίχτης 2 είναι να μπει στο καλάθι 1 μαζί με τον Παίχτη 1 ελπίζοντας ότι θα χωράει μαζί του και δεν θα ξεχειλίζει το καλάθι 1.

Ενδέχεται όμως ότι ο αλγόριθμος αυτός είναι Βέλτιστος και για αυτό θα συγκριθεί με όλους τους άλλους πιθανούς καταναμεμημένους αλγόριθμους χρησιμοποιώντας το πρόγραμμα «ΕΠΑΛΗΘΕΥΤΗΣ ΒΕΛΤΙΣΤΟΤΗΤΑΣ» ώστε να αποφανθούμε αν όντως είναι βέλτιστος.

Κεφάλαιο 4

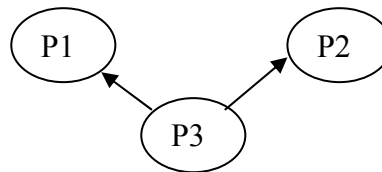
Περίπτωση 3 : Ο Παίχτης 2 γνωρίζει τον Παίχτη 3

Ο Παίχτης 1 γνωρίζει τον Παίχτη 3

4.1 Περιγραφή	25
4.2 Αλγόριθμος	26
4.3 Πειράματα	26
4.4 Ανάλυση Αλγορίθμου	27
4.5 Παρατηρήσεις	32
4.6 Βελτιστότητα Αλγορίθμου	32

4.1 Περιγραφή

Είμαστε στην περίπτωση όπου ο Παίχτης 2 γνωρίζει τον Παίχτη 3, ο Παίχτης 1 γνωρίζει τον Παίχτη 3 ενώ ο Παίχτης 3 δεν έχει καμιά γνώση, όπως φαίνεται στο πιο κάτω σχήμα.



Σχήμα 4.1.1: Σχηματική αναπαράσταση Περίπτωσης 3

Μια σκέψη για την περίπτωση αυτή είναι όπως ο Παίχτης 3 που δεν γνωρίζει κανένα, να πηγαίνει πάντα σε συγκεκριμένο καλάθι. Οι Παίχτες 1 και 2 θα έχουν την ίδια τακτική λόγω του γεγονότος ότι γνωρίζουν τον ίδιο παίχτη. Η τακτική αυτή θα είναι να ελέγχουν να δουν αν μπορούν να μπουν μαζί με τον Παίχτη 3 και αν όντως μπορούν τότε να μπουν, αλλιώς να μπουν στο αντίθετο καλάθι από τον Παίχτη 3. Με μια πρώτη ματιά ο αλγόριθμος αυτός φαίνεται να μειονεκτεί όταν και οι δύο Παίχτες 1 και 2

μπορούν να μπουν με τον Παίχτη 3 και έτσι όλοι οι παίκτες να μπουν τελικά στο ίδιο καλάθι και να έχουμε ένα καλάθι με 3 εργασίες και ένα καλάθι με καμιά εργασία.

4.2 Αλγόριθμος

$P3(X3) = \text{bin0}$

IF $X1+X3 \leq 2\mu$

$P1(X1, X3) = \text{bin0}$

ELSE

$P1(X1, X3) = \text{bin1}$

IF $X2+X3 \leq 2\mu$

$P2(X2, X3) = \text{bin0}$

ELSE

$P2(X2, X3) = \text{bin1}$

4.3 Πειράματα

Με την χρήση της συνάρτησης της MATLAB $\text{poissrnd}(\mu)$ που επιστρέφει ένα ακέραιο αριθμό με βάση την κατανομή Poisson και παράμετρο $\mu=2$, έτρεξα τον αλγόριθμο αυτό για 1 εκατομμύριο επαναλήψεις και κρατούσα σε 2 μετρητές τις περιπτώσεις όπου ο αλγόριθμος αποτύχανε και τις περιπτώσεις όπου επιτύχανε. Η διαδικασία αυτή επαναλήφθηκε αρκετές φορές. Μερικά από τα πειραματικά αποτελέσματα φαίνονται στον πιο κάτω Πίνακα.

Επανάληψη	Πιθανότητα Επιτυχίας (%)
1	57.3979
2	57.3978
3	57.4221
4	57.3183
5	57.3062
6	57.3684

Πίνακας 4.3.1: Πειραματική δειγματοληψία Περίπτωσης 3

4.4 Ανάλυση Αλγορίθμου

$P[\text{να ξεχειλίσσει το καλάθι 0 ή να ξεχειλίσσει το καλάθι 1}] =$

$$P[(X_3 > 2\mu) \cup (X_1 + X_3 \leq 2\mu \cap X_2 + X_3 \leq 2\mu \cap X_1 + X_2 + X_3 > 2\mu) \cup (X_1 + X_3 \leq 2\mu \cap X_2 + X_3 > 2\mu \cap X_2 > 2\mu) \cup (X_1 + X_3 > 2\mu \cap X_2 + X_3 \leq 2\mu \cap X_1 > 2\mu) \cup (X_3 \leq 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu)]$$

Παρατηρούμε ότι τα πιο πάνω ενδεχόμενα είναι ανεξάρτητα έτσι μπορεί να αντικατασταθεί το σύμβολο της $\cup$ με το $+$.

$$\begin{aligned} \rightarrow P[\text{αποτυχίας}] &= P[X_3 > 2\mu] + \\ &P[X_1 + X_3 \leq 2\mu \cap X_2 + X_3 \leq 2\mu \cap X_1 + X_2 + X_3 > 2\mu] + \\ &P[X_1 + X_3 \leq 2\mu \cap X_2 + X_3 > 2\mu \cap X_2 > 2\mu] + \\ &P[X_1 + X_3 > 2\mu \cap X_2 + X_3 \leq 2\mu \cap X_1 > 2\mu] + \\ &P[X_3 \leq 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu] \end{aligned}$$

Θέτουμε

$$\begin{aligned} A &= P[X_3 > 2\mu] \\ B &= P[X_1 + X_3 \leq 2\mu \cap X_2 + X_3 \leq 2\mu \cap X_1 + X_2 + X_3 > 2\mu] \\ C &= P[X_1 + X_3 \leq 2\mu \cap X_2 + X_3 > 2\mu \cap X_2 > 2\mu] \\ D &= P[X_1 + X_3 > 2\mu \cap X_2 + X_3 \leq 2\mu \cap X_1 > 2\mu] \\ E &= P[X_3 \leq 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu] \end{aligned}$$

$$\rightarrow P[\text{αποτυχίας}] = A + B + C + D + E$$

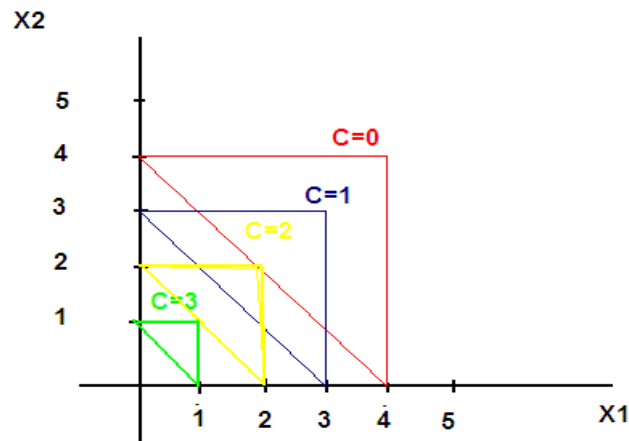
$$A = P[X_3 > 2\mu]$$

$$= \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!}$$

$$B = P [X_1 + X_3 \leq 2\mu \cap X_2 + X_3 \leq 2\mu \cap X_1 + X_2 + X_3 > 2\mu]$$

$$= \sum_{c \geq 0} (P[X_3 = c] \cap P[X_1 \leq 2\mu - c \cap X_2 \leq 2\mu - c \cap X_1 + X_2 > 2\mu - c])$$

Θα αποδώσουμε στις 2 διαστάσεις την πιο πάνω πιθανότητα ώστε να απλοποιηθεί κατάλληλα και να μπορεί να υπολογιστεί.



Σχήμα 4.4.1: Αναπαράσταση πιθανότητας στις 2 διαστάσεις

Παρατηρούμε ότι για $c \geq 4$ δεν ορίζεται η πιθανότητα καθώς δεν υπάρχει τομή στα ενδεχόμενα.

$$\begin{aligned} \rightarrow B &= \sum_{c=0}^3 \left(\frac{e^{-\mu} \mu^c}{c!} * \left(\sum_{a=1}^{2\mu-c} \sum_{b=2\mu-c-a+1}^{2\mu-c} P[X_1=a] \cap P[X_2=b] \right) \right) \\ &= \sum_{c=0}^3 \left(\frac{e^{-\mu} \mu^c}{c!} * \left(\sum_{a=1}^{2\mu-c} \sum_{b=2\mu-c-a+1}^{2\mu-c} P[X_1=a] * P[X_2=b] \right) \right) \\ &= \sum_{c=0}^3 \left(\frac{e^{-\mu} \mu^c}{c!} * \left(\sum_{a=1}^{2\mu-c} \sum_{b=2\mu-c-a+1}^{2\mu-c} \frac{e^{-\mu} \mu^a}{a!} * \frac{e^{-\mu} \mu^b}{b!} \right) \right) \end{aligned}$$

$$C = P [X_1 + X_3 \leq 2\mu \cap X_2 + X_3 > 2\mu \cap X_2 > 2\mu]$$

$$= P [X_1 + X_3 \leq 2\mu \cap X_2 > 2\mu]$$

$$= P[X_2 > 2\mu] * P [X_1 + X_3 \leq 2\mu]$$



Σχήμα 4.4.2: Αναπαράσταση πιθανότητας στις δύο διαστάσεις

$$\begin{aligned} \rightarrow C &= \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{a=0}^{2\mu} \sum_{b=0}^{2\mu-a} P[X1=a] * P[X2=b] \\ &= \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{a=0}^{2\mu} \sum_{b=0}^{2\mu-a} \frac{e^{-\mu} \mu^a}{a!} * \frac{e^{-\mu} \mu^b}{b!} \end{aligned}$$

$$D = P [X1+X3 > 2\mu \cap X2+X3 \leq 2\mu \cap X1 > 2\mu]$$

Παρατηρούμε ότι το D είναι το ίδιο ενδεχόμενο με το C

$$\rightarrow D = \sum_{\substack{j \geq 0: \\ j > 2\mu}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{a=0}^{2\mu} \sum_{b=0}^{2\mu-a} \frac{e^{-\mu} \mu^a}{a!} * \frac{e^{-\mu} \mu^b}{b!}$$

$$E = P [X3 \leq 2\mu \cap X1+X3 > 2\mu \cap X2+X3 > 2\mu \cap X1+X2 > 2\mu]$$

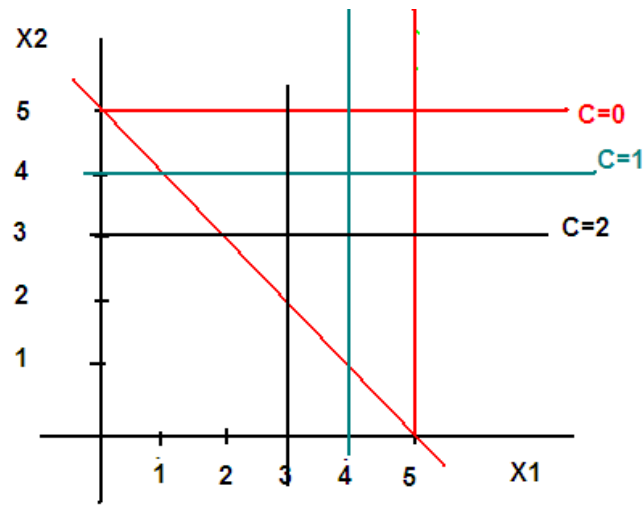
$$= \sum_{c=0}^{2\mu} P [X3=c] * P [X1 > 2\mu-c \cap X2 > 2\mu-c \cap X1+X2 > 2\mu]$$

Παρατηρούμε ότι πρέπει να για κάθε τιμή του c από το 0 μέχρι το 2μ να βρούμε την πιθανότητα του ενδεχομένου που δημιουργείται.

$$\begin{aligned} c=0 \rightarrow E1 &= P [X3=0] * P [X1 > 2\mu \cap X2 > 2\mu \cap X1+X2 > 2\mu] \\ &= P [X3=0] * P [X1 \geq 2\mu+1 \cap X2 \geq 2\mu+1 \cap X1+X2 \geq 2\mu+1] \end{aligned}$$

$$\begin{aligned} c=1 \rightarrow E2 &= P [X3=1] * P [X1 > 2\mu-1 \cap X2 > 2\mu-1 \cap X1+X2 > 2\mu] \\ &= P [X3=1] * P [X1 \geq 2\mu \cap X2 \geq 2\mu \cap X1+X2 \geq 2\mu+1] \end{aligned}$$

$$\begin{aligned}
c=2 \rightarrow E3 &= P[X3=2] * P[X1 > 2\mu-2 \cap X2 > 2\mu-2 \cap X1+X2 > 2\mu] \\
&= P[X3=2] * P[X1 \geq 2\mu-1 \cap X2 \geq 2\mu-1 \cap X1+X2 \geq 2\mu+1] \\
c=3 \rightarrow E4 &= P[X3=3] * P[X1 > 2\mu-3 \cap X2 > 2\mu-3 \cap X1+X2 > 2\mu] \\
&= P[X3=3] * P[X1 \geq 2\mu-2 \cap X2 \geq 2\mu-2 \cap X1+X2 \geq 2\mu+1] \\
c=4 \rightarrow E5 &= P[X3=4] * P[X1 > 2\mu-4 \cap X2 > 2\mu-4 \cap X1+X2 > 2\mu] \\
&= P[X3=4] * P[X1 \geq 2\mu-3 \cap X2 \geq 2\mu-3 \cap X1+X2 \geq 2\mu+1] \\
E &= E1+E2+E3+E4+E5
\end{aligned}$$

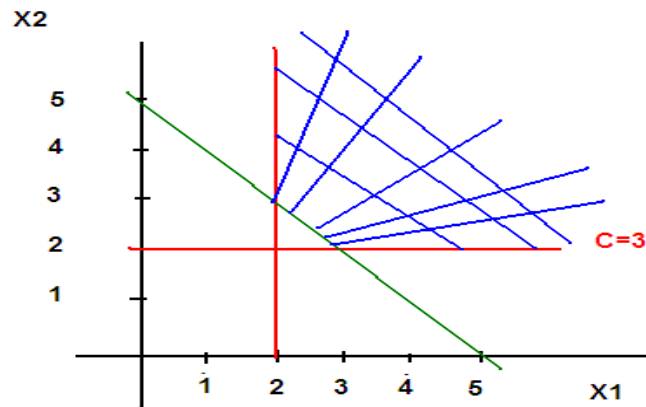


Σχήμα 4.4.3: Αναπαράσταση πιθανότητας στις δύο διαστάσεις

$$\begin{aligned}
\rightarrow E1 &= P[X3=0] * P[X1 \geq 2\mu+1 \cap X2 \geq 2\mu+1] \\
&= P[X3=0] * P[X1 \geq 2\mu+1] * P[X2 \geq 2\mu+1] \\
&= \frac{e^{-\mu} \mu^0}{0!} * \sum_{\substack{j \geq 0; \\ j \geq 2\mu+1}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0; \\ j \geq 2\mu+1}} \frac{e^{-\mu} \mu^j}{j!} \\
&= \frac{e^{-\mu} \mu^0}{0!} * \left(\sum_{\substack{j \geq 0; \\ j \geq 2\mu+1}} \frac{e^{-\mu} \mu^j}{j!} \right)^2 \\
\rightarrow E2 &= \frac{e^{-\mu} \mu^1}{1!} * \left(\sum_{\substack{j \geq 0; \\ j \geq 2\mu}} \frac{e^{-\mu} \mu^j}{j!} \right)^2
\end{aligned}$$

$$\rightarrow E3 = \frac{e^{-\mu} \mu^2}{2!} * \left(\sum_{\substack{j \geq 0: \\ j \geq 2\mu-1}} \frac{e^{-\mu} \mu^j}{j!} \right)^2$$

$$\rightarrow E4 = P[X3=3] * P[X1 \geq 2\mu-2 \cap X2 \geq 2\mu-2 \cap X1+X2 \geq 2\mu+1]$$



Σχήμα 4.4.4: Αναπαράσταση πιθανότητας στις δύο διαστάσεις

$$= P[X3=3] * (P[X2 \geq 3 \cap X1 \geq 2] + P[X1 \geq 3 \cap X2=2])$$

$$= P[X3=3] * (P[X2 \geq 3] * P[X1 \geq 2] + P[X1 \geq 3] * P[X2=2])$$

$$= \frac{e^{-\mu} \mu^3}{3!} * \left(\sum_{\substack{j \geq 0: \\ j \geq 3}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j \geq 2}} \frac{e^{-\mu} \mu^j}{j!} + \sum_{\substack{j \geq 0: \\ j \geq 3}} \frac{e^{-\mu} \mu^j}{j!} * \frac{e^{-\mu} \mu^2}{2!} \right)$$

$$\rightarrow E5 = P[X3=4] * P[X1 \geq 2\mu-3 \cap X2 \geq 2\mu-3 \cap X1+X2 \geq 2\mu+1]$$



Σχήμα 4.4.5 : Αναπαράσταση πιθανότητας στις 2 διαστάσεις

$$\begin{aligned}
&= P[X_3=4] * (P[X_1 \geq 4 \cap 1 \leq X_2 \leq 3] + P[X_2 \geq 4 \cap X_1 \geq 1] + \\
&\quad P[X_1=2 \cap X_2=3] + P[X_1=3 \cap X_2=3] + P[X_1=3 \cap X_2=2]) \\
&= P[X_3=4] * (P[X_1 \geq 4] * P[1 \leq X_2 \leq 3] + P[X_2 \geq 4] * P[X_1 \geq 1] + \\
&\quad P[X_1=2] * P[X_2=3] + P[X_1=3] * P[X_2=3] + P[X_1=3] * P[X_2=2]) \\
&= \frac{e^{-\mu} \mu^4}{4!} * (\sum_{\substack{j \geq 0: \\ j \geq 4}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j=1}}^3 \frac{e^{-\mu} \mu^j}{j!} + \sum_{\substack{j \geq 0: \\ j \geq 4}} \frac{e^{-\mu} \mu^j}{j!} * \sum_{\substack{j \geq 0: \\ j \geq 1}} \frac{e^{-\mu} \mu^j}{j!} + \\
&\quad \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^3}{3!} + \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^3}{3!} + \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^2}{2!})
\end{aligned}$$

$$E = E_1 + E_2 + E_3 + E_4 + E_5$$

$$\rightarrow P[\text{αποτυχία}] = A + B + C + D + E = 0.4263066 \quad = 42.63066\%$$

$$\rightarrow P[\text{επιτυχία}] = 1 - P[\text{αποτυχία}] = 0.5736934 \quad = 57.36934\%$$

4.5 Παρατηρήσεις

Για ακόμα μια φορά το αποτέλεσμα που πήραμε με την ανάλυση για να βρούμε την πιθανότητα επιτυχίας του αλγορίθμου συμφωνεί με τα πειράματα που έγιναν.

Οι αρχικές ανησυχίες ότι ο αλγόριθμος θα αποτυγχάνει σε περιπτώσεις όπου ο Παίχτης 1 και ο Παίχτης 2 μπορούν να μουν στο ίδιο καλάθι με τον Παίχτη 3 πράγματι επιβεβαιώθηκαν και έτσι παρόλο που είχαμε μια επιπλέον γνώση σε σχέση με την Περίπτωση 2, δεν μας βοήθησε έτσι ώστε να αυξηθεί η πιθανότητα επιτυχίας του αλγορίθμου μας.

4.6 Βελτιστότητα Αλγορίθμου

Είναι φανερό ότι ο αλγόριθμος που αναλύθηκε δεν είναι καθόλου αποδοτικός, παρόλο που χρησιμοποιεί μια επιπλέον γνώση σε σύγκριση με την Περίπτωση 2.

Παρατηρούμε ότι η Περίπτωση 2 είναι υποπερίπτωση της Περίπτωσης 3. Έτσι ο Αλγόριθμος που επιλέχθηκε για την Περίπτωση 2 (ο οποίος είναι και βέλτιστος) μπορεί να χρησιμοποιηθεί και για την Περίπτωση 3.

Αν όντως δεν υπάρχει κάποιος αλγόριθμος που να χρησιμοποιεί και την επιπλέον γνώση και να τα πηγαίνει καλύτερα από τον αλγόριθμο της Περίπτωσης 2, τότε μπορούμε να πούμε ότι η συγκεκριμένη επιπλέον γνώση δεν αξίζει τίποτα και η

Περίπτωση 3 μπορεί να χρησιμοποιεί τον αλγόριθμο της Περίπτωσης 2 που παρόλο που χρησιμοποιεί λιγότερη γνώση , έχει καλύτερα αποτελέσματα.

Ενδέχεται όμως ότι ο αλγόριθμος αυτός είναι Βέλτιστος και για αυτό θα συγκριθεί με όλους τους άλλους πιθανούς καταναμημένους αλγόριθμους χρησιμοποιώντας το πρόγραμμα «ΕΠΑΛΗΘΕΥΤΗΣ ΒΕΛΤΙΣΤΟΤΗΤΑΣ» ώστε να αποφανθούμε αν όντως είναι βέλτιστος.

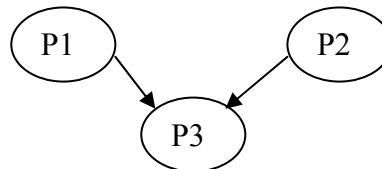
Κεφάλαιο 5

Περίπτωση 4 : Ο Παίχτης 3 γνωρίζει τον Παίχτη 1 και τον Παίχτη 2

5.1 Περιγραφή	34
5.2 Αλγόριθμος	35
5.3 Πειράματα	35
5.4 Ανάλυση Αλγορίθμου	36
5.5 Παρατηρήσεις	39
5.6 Βελτιστότητα Αλγορίθμου	39

5.1 Περιγραφή

Είμαστε στην περίπτωση όπου ο Παίχτης 3 γνωρίζει και τους δύο άλλους Παίχτες 1 και 2 όπως φαίνεται στο πιο κάτω σχήμα.



Σχήμα 5.1.1: Σχηματική αναπαράσταση Περίπτωσης 4

Από την στιγμή που ο Παίχτης 3 γνωρίζει και τους δύο άλλους Παίχτες , μπορούμε να πούμε ότι η Περίπτωση αυτή εξαρτάται από το πόσο καλά θα παίξουν οι Παίχτες 1 και 2 οι οποίοι δεν έχουν καμία γνώση. Έτσι η περίπτωση αυτή , αναδημιουργείται στην περίπτωση όπου υπάρχουν μόνο 2 Παίχτες, αφού ο Παίχτης 3 θα παίξει βέλτιστα.

Αφού οι 2 αυτοί Παίχτες δεν έχουν καμία γνώση για να επιλέξουν σε ποιο καλάθι θα μπουν , ο μόνος τρόπος που έχουν για να σκεφτούν είναι με βάση κάποιο κατώφλι. Για τους λόγους που εξηγήθηκαν σε προηγούμενο Κεφάλαιο το κατώφλι και για τους 2 Παίχτες θα είναι το 2 που θα ξεχωρίζει τις «μικρές» από τις «μεγάλες» εργασίες.

5.2 Αλγόριθμος

```
IF (X1<=2)
    P1(X1) = bin0
ELSE
    P1(X1) = bin1
IF (X2<=2)
    P2(X2) = bin0
ELSE
    P2(X2) = bin1
IF (X1<=2 && X2<=2)
    P3(X3, X2, X1) = bin1
ELSE
    P3(X3, X2, X1) = bin0
```

5.3 Πειράματα

Με την χρήση της συνάρτησης της MATLAB `poissrnd(μ)` που επιστρέφει ένα ακέραιο αριθμό με βάση την κατανομή Poisson και παράμετρο $\mu=2$, έτρεξα τον αλγόριθμο αυτό για 1 εκατομμύριο επαναλήψεις και κρατούσα σε 2 μετρητές τις περιπτώσεις όπου ο αλγόριθμος αποτύχανε και τις περιπτώσεις όπου επιτύχανε. Η διαδικασία αυτή επαναλήφθηκε αρκετές φορές. Μερικά από τα πειραματικά αποτελέσματα φαίνονται στον πιο κάτω Πίνακα.

Επανάληψη	Πιθανότητα Επιτυχίας (%)
1	72.7353
2	72.8424
3	72.8512
4	72.8426
5	72.8511
6	72.7906

Πίνακας 5.3.1: Πειραματική δειγματοληψία Περίπτωσης 4

5.4 Ανάλυση Αλγορίθμου

$$P[\text{αποτυχίας}] =$$

$$\begin{aligned} & (X1 \leq T \cap X2 > T \cap (X1 + X3 > 2\mu \cup X2 > 2\mu)) \quad \cup \\ & (X2 \leq T \cap X1 > T \cap (X2 + X3 > 2\mu \cup X1 > 2\mu)) \quad \cup \\ & (X1 \leq T \cap X2 \leq T \cap (X1 + X2 > 2\mu \cup X3 > 2\mu)) \quad \cup \\ & (X1 > T \cap X2 > T \cap (X3 > 2\mu \cup X1 + X2 > 2\mu)) \end{aligned}$$

Θέτουμε :

$$A = (X1 \leq T \cap X2 > T \cap (X1 + X3 > 2\mu \cup X2 > 2\mu))$$

$$B = (X2 \leq T \cap X1 > T \cap (X2 + X3 > 2\mu \cup X1 > 2\mu))$$

$$C = (X1 \leq T \cap X2 \leq T \cap (X1 + X2 > 2\mu \cup X3 > 2\mu))$$

$$D = (X1 > T \cap X2 > T \cap (X3 > 2\mu \cup X1 + X2 > 2\mu))$$

Παρατηρούμε ότι δεν υπάρχουν τομές μεταξύ των A,B,C,D

$$\rightarrow P[\text{αποτυχίας}] = A+B+C+D$$

$$A = P[X1 \leq T \cap X2 > T \cap (X1 + X3 > 2\mu \cup X2 > 2\mu)]$$

$$A = P[(X1 \leq T \cap X2 > T \cap X1 + X3 > 2\mu) \cup (X1 \leq T \cap X2 > T \cap X2 > 2\mu)]$$

$$\begin{aligned} &= P(X1 \leq T \cap X2 > T \cap X1 + X3 > 2\mu) + (X1 \leq T \cap X2 > T \cap X2 > 2\mu) - \\ & P(X1 \leq T \cap X2 > T \cap X1 + X3 > 2\mu \cap X1 \leq T \cap X2 > T \cap X2 > 2\mu) \end{aligned}$$

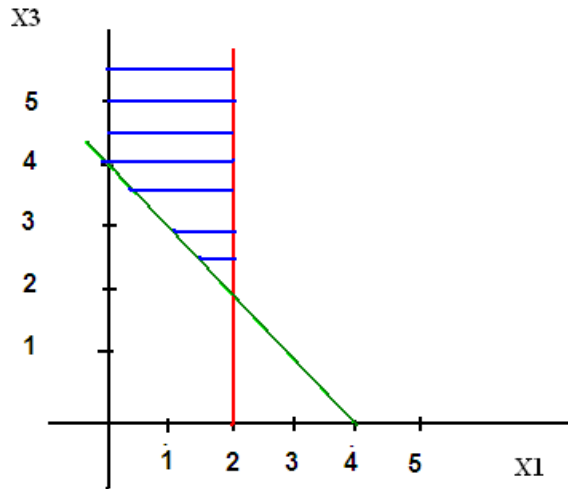
$$\begin{aligned} &= P(X1 \leq T \cap X2 > T \cap X1 + X3 > 2\mu) + (X1 \leq T \cap X2 > 2\mu) - \\ & P(X1 \leq T \cap X1 + X3 > 2\mu \cap X2 > 2\mu) \end{aligned}$$

$$\begin{aligned} &= P(X2 > T) * P(X1 \leq T \cap X1 + X3 > 2\mu) + P(X1 \leq T \cap X2 > 2\mu) - \\ & P(X2 > 2\mu) * P(X1 \leq T \cap X1 + X3 > 2\mu) \end{aligned}$$

$$= P(X1 \leq T \cap X2 > 2\mu) + P(X1 \leq T \cap X1 + X3 > 2\mu) * (P(X2 > T) - P(X2 > 2\mu))$$

$$\rightarrow A = P(X1 \leq T \cap X2 > 2\mu) + P(X1 \leq T \cap X1 + X3 > 2\mu) * [P(X2 > T) - P(X2 > 2\mu)]$$

Την πιθανότητα $P(X1 \leq T \cap X1+X3 > 2\mu)$ θα την αναπαράστησουμε στις 2 διαστάσεις για να γίνει πιο εύκολη η εύρεση της.



Σχήμα 5.4.1 : Αναπαράσταση πιθανότητας στις 2 διαστάσεις.

$$P(X1 \leq T \cap X1+X3 > 2\mu) = P(X1 \leq T \cap X3 > 2\mu) + \left(\sum_{a=1}^2 \sum_{b=2\mu-a+1}^{2\mu} P(X1=a) * P(X2=\beta) \right)$$

$$\rightarrow A = P(X1 \leq T \cap X2 > 2\mu) + P(P(X1 \leq T \cap X3 > 2\mu) + \left(\sum_{a=1}^2 \sum_{b=2\mu-a+1}^{2\mu} P(X1=a) * \right. \\ \left. P(X2=\beta) \right) * [P(X2 > T) - P(X2 > 2\mu)]$$

$$= \sum_{0 \leq j \leq 2} \frac{e^{-\mu} \mu^j}{j!} * \sum_{j > 2\mu} \frac{e^{-\mu} \mu^j}{j!} + \left[\sum_{0 \leq j \leq 2} \frac{e^{-\mu} \mu^j}{j!} * \sum_{j > 2\mu} \frac{e^{-\mu} \mu^j}{j!} + \right. \\ \left. \left(\sum_{a=1}^2 \sum_{b=2\mu-a+1}^{2\mu} \frac{e^{-\mu} \mu^a}{a!} * \frac{e^{-\mu} \mu^b}{b!} \right) * \left(\sum_{j > 2} \frac{e^{-\mu} \mu^j}{j!} - \sum_{j > 2\mu} \frac{e^{-\mu} \mu^j}{j!} \right) \right]$$

Παρατηρούμε ότι η πιθανότητα του B είναι η ίδια με την πιθανότητα A.

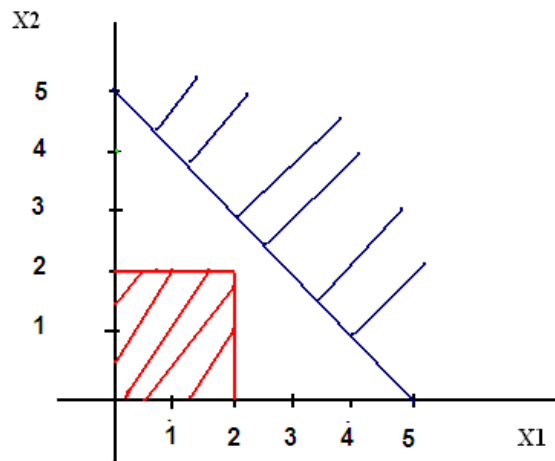
$$\rightarrow B = \sum_{0 \leq j \leq 2} \frac{e^{-\mu} \mu^j}{j!} * \sum_{j > 2\mu} \frac{e^{-\mu} \mu^j}{j!} + \left[\sum_{0 \leq j \leq 2} \frac{e^{-\mu} \mu^j}{j!} * \sum_{j > 2\mu} \frac{e^{-\mu} \mu^j}{j!} + \right. \\ \left. \left(\sum_{a=1}^2 \sum_{b=2\mu-a+1}^{2\mu} \frac{e^{-\mu} \mu^a}{a!} * \frac{e^{-\mu} \mu^b}{b!} \right) * \left(\sum_{j > 2} \frac{e^{-\mu} \mu^j}{j!} - \sum_{j > 2\mu} \frac{e^{-\mu} \mu^j}{j!} \right) \right]$$

$$\Gamma = P[X1 \leq T \cap X2 \leq T \cap (X1+X2 > 2\mu \cup X3 > 2\mu)]$$

$$= P[X1 \leq T \cap X2 \leq T \cap X1+X2 > 2\mu] \cup (X1 \leq T \cap X2 \leq T \cap X3 > 2\mu)]$$

$$\begin{aligned}
&= P[X_1 \leq T \cap X_2 \leq T \cap X_1 + X_2 > 2\mu] + P[X_1 \leq T \cap X_2 \leq T \cap X_3 > 2\mu] - \\
&\quad P[X_1 \leq T \cap X_2 \leq T \cap X_1 + X_2 > 2\mu \cap X_1 \leq T \cap X_2 \leq T \cap X_3 > 2\mu] \\
&= P[X_1 \leq T \cap X_2 \leq T \cap X_1 + X_2 > 2\mu] + P[X_1 \leq T \cap X_2 \leq T \cap X_3 > 2\mu] - \\
&\quad P[X_1 \leq T \cap X_2 \leq T \cap X_1 + X_2 > 2\mu \cap X_3 > 2\mu] \\
&= P[X_1 \leq T \cap X_2 \leq T \cap X_1 + X_2 > 2\mu] + P[X_1 \leq T \cap X_2 \leq T \cap X_3 > 2\mu] - \\
&\quad P[X_3 > 2\mu] * P[X_1 \leq T \cap X_2 \leq T \cap X_1 + X_2 > 2\mu] \\
&= P[X_1 \leq T \cap X_2 \leq T \cap X_3 > 2\mu] + P[X_1 \leq T \cap X_2 \leq T \cap X_1 + X_2 > 2\mu] * \\
&\quad [1 - P(X_3 > 2\mu)]
\end{aligned}$$

Παρατηρούμε στο πιο κάτω σχήμα ότι η πιθανότητα $P[X_1 \leq T \cap X_2 \leq T \cap X_1 + X_2 > 2\mu]$ είναι 0 καθώς δεν υπάρχει τομή μεταξύ των ενδεχομένων.



Σχήμα 5.4.2 : Αναπαράσταση πιθανότητας στις 2 διαστάσεις

$$\Rightarrow \Gamma = P[X_1 \leq T \cap X_2 \leq T \cap X_3 > 2\mu]$$

$$\Gamma = \left(\sum_{0 \leq j \leq 2} \frac{e^{-\mu} \mu^j}{j!} \right)^2 * \sum_{j > 2\mu} \frac{e^{-\mu} \mu^j}{j!}$$

$$\Delta = P[X_1 > 2 \cap X_2 > 2 \cap (X_3 > 2\mu \cup X_1 + X_2 > 2\mu)]$$

$$= P[(X_1 > 2 \cap X_2 > 2 \cap X_3 > 2\mu) \cup (X_1 > 2 \cap X_2 > 2 \cap X_1 + X_2 > 2\mu)]$$

$$= P[(X_1 > 2 \cap X_2 > 2 \cap X_3 > 2\mu) + P(X_1 > 2 \cap X_2 > 2 \cap X_1 + X_2 > 2\mu)] -$$

$$P[(X_1 > 2 \cap X_2 > 2 \cap X_3 > 2\mu \cap X_1 > 2 \cap X_2 > 2 \cap X_1 + X_2 > 2\mu)]$$

$$= P[(X_1 > 2 \cap X_2 > 2 \cap X_3 > 2\mu) + P(X_1 > 2 \cap X_2 > 2 \cap X_1 + X_2 > 2\mu)] -$$

$$\begin{aligned}
& P[(X1>2 \cap X2>2 \cap X3>2\mu \cap X1+X2>2\mu)] \\
& = P[X1>2 \cap X2>2 \cap X3>2\mu] + P[(X1>2 \cap X2>2] - \\
& \quad P[X1>2 \cap X2>2 \cap X3>2\mu] \\
& = P[X1>2 \cap X2>2] * [P(X3>2\mu) + 1 - P(X3>2\mu)] \\
& = P[X1>2 \cap X2>2] = \left(\sum_{j>2} \frac{e^{-\mu} \mu^j}{j!} \right)^2
\end{aligned}$$

$$P[\text{αποτυχία}] = A+B+C+D = 0.2720731 = 27.20731\%$$

$$P[\text{επιτυχία}] = 1 - P[\text{αποτυχία}] = 0.7279269 = 72.79269\%$$

5.5 Παρατηρήσεις

Παρατηρούμε ότι το αποτέλεσμα επιτυχίας που πήραμε συμφωνεί και πάλι με τα πειράματα που έγιναν.

Φαίνεται ότι αν υπάρχουν 2 συνολικά ακμές , δηλαδή 2 γνώσεις , είναι καλύτερα να τις έχει και τις 2 ένας Παίχτης όπως στη περίπτωση αυτή όπου ο Παίχτης 3 ξέρει τους Παίχτες 1 και 2 , παρά να έχει κάποιος Παίχτης την μια γνώση και κάποιος άλλος Παίχτης την άλλη όπως στην Περίπτωση 4.

5.6 Βελτιστότητα Αλγορίθμου

Αφού ισχύει ότι κάθε Βέλτιστος καταναμημένος αλγόριθμος , έχει την Βέλτιστη 3άδα από Αλγόριθμους και στην Περίπτωση αυτή ο Παίχτης 3 μπορεί να παίζει βέλτιστα , δεν χρειάζεται να δώσουμε σαν είσοδο στο πρόγραμμα «ΕΠΑΛΗΘΕΥΤΗΣ ΒΕΛΤΙΣΤΟΤΗΤΑΣ» όλους τους δυνατούς αλγορίθμους για τον Παίχτη3,απλά θα βάλουμε τον Παίχτη 3 να παίζει πάντα βέλτιστα. Έτσι εξοικονομήθηκε και αρκετός χρόνος για να τρέξει το πρόγραμμα αυτό. Τρέχοντας λοιπόν τον αλγόριθμο «ΒΕΛΤΙΣΤΟΣ ΑΛΓΟΡΙΘΜΟΣ» μας που σύγκρινε τον δικό μας αλγόριθμο με όλους τους άλλους δυνατούς αλγορίθμους μας απάντησε ότι δεν υπάρχει κάποιος καταναμημένος αλγόριθμος καλύτερος από τον δικό μας. Δηλαδή ο Αλγόριθμος για την Περίπτωση 4, είναι καλύτερος από όλους τους πιθανούς Αλγορίθμους που μπορούν να υπάρξουν στην Περίπτωση αυτή. Έτσι ο Αλγόριθμος αυτός είναι Βέλτιστος.

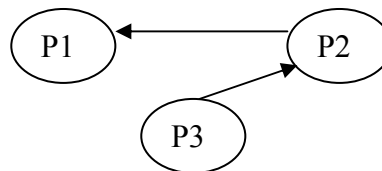
Κεφάλαιο 6

Περίπτωση 5 : Ο Παίχτης 2 γνωρίζει τον Παίχτη 3 Ο Παίχτης 1 γνωρίζει τον Παίχτη 2

6.1 Περιγραφή	40
6.2 Αλγόριθμος	41
6.3 Πειράματα	41
6.4 Ανάλυση Αλγορίθμου	42
6.5 Παρατηρήσεις	49
6.6 Βελτιστότητα Αλγορίθμου	50

6.1 Περιγραφή

Είμαστε στην περίπτωση όπου ο Παίχτης 2 γνωρίζει τον Παίχτη 3 και ο Παίχτης 1 γνωρίζει τον Παίχτη 2 όπως φαίνεται στο πιο κάτω σχήμα.



Σχήμα 6.1.1: Σχηματική αναπαράσταση Περίπτωσης 5

Από την στιγμή που ο Παίχτης 3 δεν έχει καμία γνώση θα τοποθετηθεί στο καλάθι 0. Οι Παίχτες 1 και 2 που γνωρίζουν από ένα Παίχτη ο καθένας τους και δεν είναι ο ίδιος αυτός ο Παίχτης, θα έχουν μια στρατηγική που θα συνδυάζει και την γνώση που έχουν και κάποιο κατώφλι. Το κατώφλι θα είναι το $T=2$ για τους ίδιους λόγους που αναφέρθηκαν σε προηγούμενα κεφάλαια.

6.2 Αλγόριθμος

P3(X3) = bin0

IF (X2+X3<=4 && X2<=2)

 P2(X2, X3) = bin0

ELSE

 P2(X2, X3) = bin1

IF (X2>2 && X1+X2>4)

 P1(X1, X2) = bin0

ELSE

 P1(X1, X2) = bin1

6.3 Πειράματα

Με την χρήση της συνάρτησης της MATLAB `poissrnd(μ)` που επιστρέφει ένα ακέραιο αριθμό με βάση την κατανομή Poisson και παράμετρο $\mu=2$, έτρεξα τον αλγόριθμο αυτό για 1 εκατομμύριο επαναλήψεις και κρατούσα σε 2 μετρητές τις περιπτώσεις όπου ο αλγόριθμος αποτύχανε και τις περιπτώσεις όπου επιτύχανε. Η διαδικασία αυτή επαναλήφθηκε αρκετές φορές. Μερικά από τα πειραματικά αποτελέσματα φαίνονται στον πιο κάτω Πίνακα.

Επανάληψη	Πιθανότητα Επιτυχίας (%)
1	76.0353
2	75.9390
3	76.0526
4	75.9648
5	75.9875
6	76.0421

Πίνακας 6.3.1: Πειραματική δειγματοληψία Περίπτωσης 5

6.4 Ανάλυση Αλγορίθμου

$P[\text{να ξεχειλίσσει το καλάθι 0 ή να ξεχειλίσσει το καλάθι 1}] =$

$$P[(X_3 > 2\mu) \cup (X_3 \leq 2\mu \cap (X_2 > 2 \cup X_2 + X_3 > 2\mu) \cap (X_2 \leq 2 \cup X_1 + X_2 \leq 2\mu) \cap X_1 + X_2 > 2\mu) \cup (X_3 \leq 2\mu \cap (X_2 > 2 \cup X_2 + X_3 > 2\mu) \cap (X_2 > 2 \cap X_1 + X_2 > 2\mu) \cap (X_1 + X_3 > 2\mu \cup X_2 > 2\mu) \cup (X_3 \leq 2\mu \cap (X_2 \leq 2 \cap X_2 + X_3 \leq 2\mu) \cap (X_2 \leq 2 \cup X_1 + X_2 \leq 2\mu) \cap (X_2 + X_3 > 2\mu \cup X_1 > 2\mu))]$$

Θέτουμε

$$A = P[(X_3 > 2\mu)]$$

$$B = P[(X_3 \leq 2\mu \cap (X_2 > 2 \cup X_2 + X_3 > 2\mu) \cap (X_2 \leq 2 \cup X_1 + X_2 \leq 2\mu) \cap X_1 + X_2 > 2\mu)]$$

$$C = P[X_3 \leq 2\mu \cap (X_2 > 2 \cup X_2 + X_3 > 2\mu) \cap (X_2 > 2 \cap X_1 + X_2 > 2\mu) \cap (X_1 + X_3 > 2\mu \cup X_2 > 2\mu)]$$

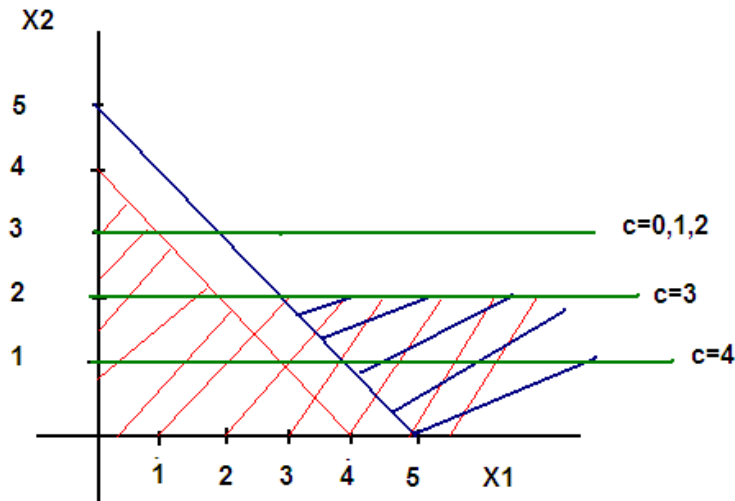
$$D = P[(X_3 \leq 2\mu \cap (X_2 \leq 2 \cap X_2 + X_3 \leq 2\mu) \cap (X_2 \leq 2 \cup X_1 + X_2 \leq 2\mu) \cap (X_2 + X_3 > 2\mu \cup X_1 > 2\mu)]$$

Παρατηρούμε ότι τα πιο πάνω ενδεχόμενα A,B,C,D δεν έχουν ανά δύο τομή μεταξύ τους και έτσι μπορούμε να πούμε ότι

$$P[\text{αποτυχίας}] = A + B + C + D$$

$$A = P[(X_3 > 2\mu)] = \sum_{j > 2\mu} \frac{e^{-\mu} \mu^j}{j!}$$

$$\begin{aligned} B &= P[(X_3 \leq 2\mu \cap (X_2 > 2 \cup X_2 + X_3 > 2\mu) \cap (X_2 \leq 2 \cup X_1 + X_2 \leq 2\mu) \cap X_1 + X_2 > 2\mu)] \\ &= \sum_{0 \leq c \leq 4} P[(X_3 = c] * P[(X_2 > 2 \cup X_2 > 2\mu - c) \cap (X_2 \leq 2 \cup X_1 + X_2 \leq 2\mu) \cap X_1 + X_2 > 2\mu] \end{aligned}$$



Σχήμα 6.4.1: Αναπαράσταση πιθανότητας στον χώρο

Για $c=0,1,2$ δεν υπάρχει τομή

$c = 3$:

$$B1 = P(X1 \geq 3 \cap X2 = 2) * P(X3 = 3) = \sum_{j \geq 3} \frac{e^{-\mu} \mu^j}{j!} * \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^3}{3!}$$

$c = 4$:

$$B2 = [P(X1 \geq 4 \cap 1 \leq X2 \leq 2) + P(X1 = 3 \cap X2 = 2)] * P(X3 = 4)$$

$$= \left[\sum_{j \geq 4} \frac{e^{-\mu} \mu^j}{j!} * \sum_{1 \leq j \leq 2} \frac{e^{-\mu} \mu^j}{j!} + \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^2}{2!} \right] * \frac{e^{-\mu} \mu^4}{4!}$$

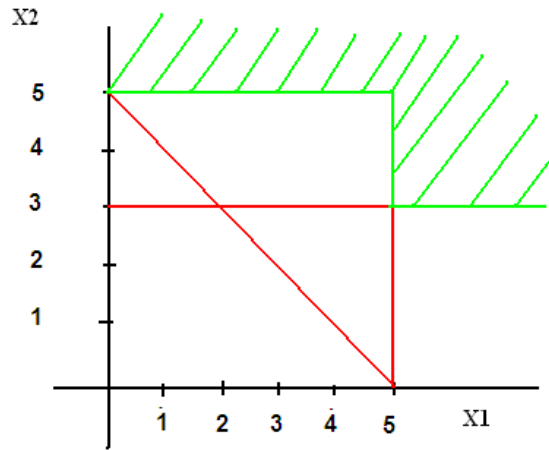
$$B = B1 + B2$$

$$C = P \left[(X3 \leq 2\mu \cap (X2 > 2 \cup X2 + X3 > 2\mu)) \cap (X2 > 2 \cap X1 + X2 > 2\mu) \cap (X1 + X3 > 2\mu \cup X2 > 2\mu) \right]$$

$$C = \sum_{c=0}^{2\mu} P[X3=c] * P \left[(X2 > 2 \cup X2 > 2\mu - c) \cap (X2 > 2 \cap X1 + X2 > 2\mu) \cap (X1 > 2\mu - c \cup X2 > 2\mu) \right]$$

$$c = 0: C1 = P[(X2 \geq 3 \cup X2 \geq 2\mu + 1) \cap (X2 \geq 3 \cap X1 + X2 \geq 2\mu + 1) \cap (X1 \geq 2\mu + 1 \cup X2 \geq 2\mu + 1)] * P[(X3 = 0)]$$

$$= P[(X_2 \geq 3) \cap (X_2 \geq 3 \cap X_1 + X_2 \geq 2\mu + 1) \cap (X_1 \geq 2\mu + 1 \cup X_2 \geq 2\mu + 1)] * P[X_3 = 0]$$

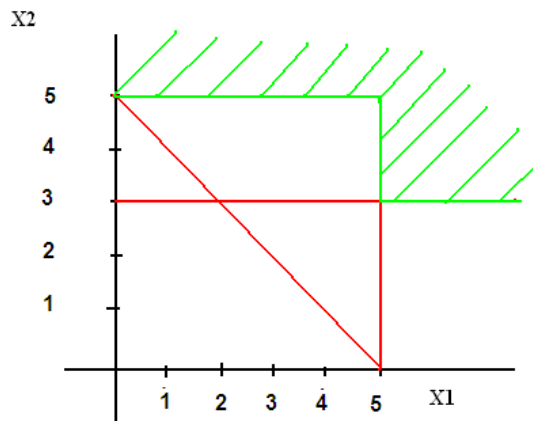


Σχήμα 6.4.2: Αναπαράσταση πιθανότητας στον χώρο

$$= [P(X_2 \geq 5) + P(X_1 \geq 5 \cap 3 \leq X_2 \leq 4)] * P(X_3 = 0)$$

$$\rightarrow C1 = \left[\sum_{j \geq 5} \frac{e^{-\mu} \mu^j}{j!} + \sum_{j \geq 5} \frac{e^{-\mu} \mu^j}{j!} * \sum_{3 \leq j \leq 4} \frac{e^{-\mu} \mu^j}{j!} \right] * \frac{e^{-\mu} \mu^0}{0!}$$

$$c = 1: C2 = P[(X_2 \geq 3 \cup X_2 \geq 2\mu) \cap (X_2 \geq 3 \cap X_1 + X_2 \geq 2\mu + 1) \cap (X_1 \geq 2\mu \cup X_2 \geq 2\mu + 1)]$$

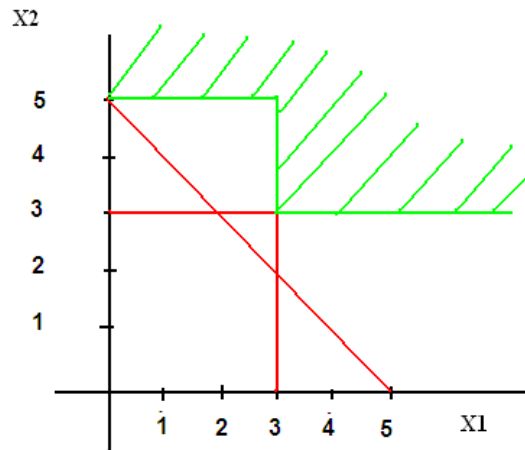


Σχήμα 6.4.3: Αναπαράσταση πιθανότητας στον χώρο

$$C2 = [P(X_2 \geq 5) + P(X_1 \geq 4 \cap 3 \leq X_2 \leq 4)] * P(X_3 = 1)$$

$$\rightarrow C2 = \left[\sum_{j \geq 5} \frac{e^{-\mu} \mu^j}{j!} + \sum_{j \geq 4} \frac{e^{-\mu} \mu^j}{j!} * \sum_{3 \leq j \leq 4} \frac{e^{-\mu} \mu^j}{j!} \right] * \frac{e^{-\mu} \mu^1}{1!}$$

$$c = 2: C3 = P[(X2 \geq 3 \cup X2 \geq 3) \cap (X2 \geq 3 \cap X1 + X2 \geq 2\mu + 1) \cap (X1 \geq 3 \cup X2 \geq 2\mu + 1)]$$

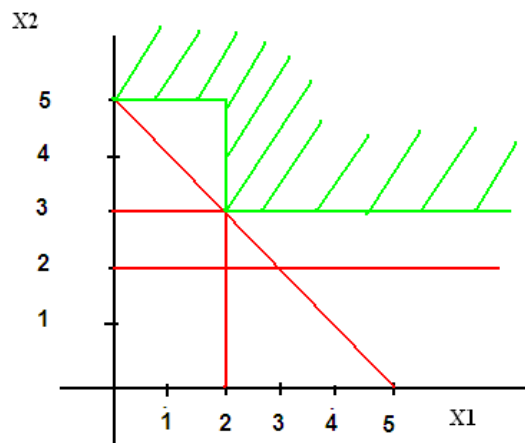


Σχήμα 6.4.4: Αναπαράσταση πιθανότητας στον χώρο

$$[P(X2 \geq 5) + P(X1 \geq 3 \cap 3 \leq X2 \leq 4)] * P(X3 = 2)$$

$$= \left[\sum_{j \geq 5} \frac{e^{-\mu} \mu^j}{j!} + \sum_{j \geq 3} \frac{e^{-\mu} \mu^j}{j!} * \sum_{3 \leq j \leq 4} \frac{e^{-\mu} \mu^j}{j!} \right] * \frac{e^{-\mu} \mu^2}{2!}$$

$$c = 3: C4 = P[(X2 \geq 3 \cup X2 \geq 2) \cap (X2 \geq 3 \cap X1 + X2 \geq 2\mu + 1) \cap (X1 \geq 2 \cup X2 \geq 2\mu + 1)]$$

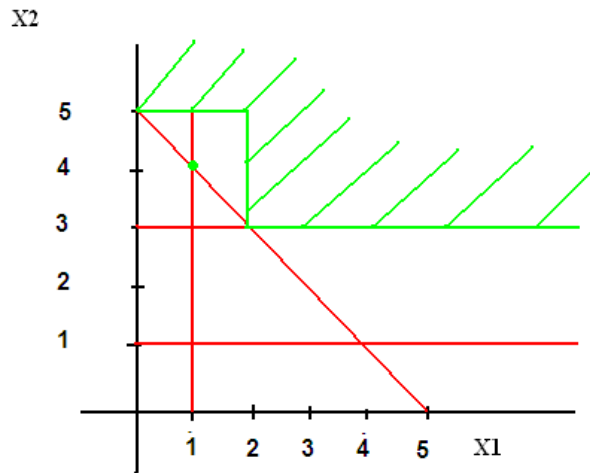


Σχήμα 6.4.5: Αναπαράσταση πιθανότητας στον χώρο

$$[P(X_2 \geq 5) + P(X_1 \geq 2 \cap 3 \leq X_2 \leq 4)] * P(X_3 = 3)$$

$$\rightarrow C_4 = \left[\sum_{j \geq 5} \frac{e^{-\mu} \mu^j}{j!} + \sum_{j \geq 2} \frac{e^{-\mu} \mu^j}{j!} * \sum_{3 \leq j \leq 4} \frac{e^{-\mu} \mu^j}{j!} \right] * \frac{e^{-\mu} \mu^3}{3!}$$

$$c = 4: C_5 = P[(X_2 \geq 3 \cup X_2 \geq 1) \cap (X_2 \geq 3 \cap X_1 + X_2 \geq 2\mu + 1) \cap (X_1 \geq 1 \cup X_2 \geq 2\mu + 1)]$$



Σχήμα 6.4.6: Αναπαράσταση πιθανότητας στον χώρο

$$[P(X_2 \geq 5) + P(X_1 \geq 2 \cap 3 \leq X_2 \leq 4) + P(X_1 = 1) \cap P(X_2 = 4)] * P(X_3 = 4)$$

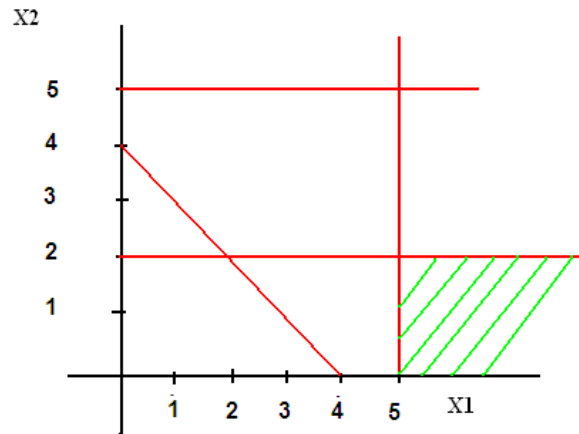
$$\rightarrow C_5 = \left[\sum_{j \geq 5} \frac{e^{-\mu} \mu^j}{j!} + \sum_{j \geq 2} \frac{e^{-\mu} \mu^j}{j!} * \sum_{3 \leq j \leq 4} \frac{e^{-\mu} \mu^j}{j!} + \frac{e^{-\mu} \mu^1}{1!} * \frac{e^{-\mu} \mu^4}{4!} \right] * \frac{e^{-\mu} \mu^4}{4!}$$

$$C = C_1 + C_2 + C_3 + C_4 + C_5$$

$$D = P[(X_3 \leq 2\mu \cap (X_2 \leq 2 \cap X_2 + X_3 \leq 2\mu) \cap (X_2 \leq 2 \cup X_1 + X_2 \leq 2\mu) \cap (X_2 + X_3 > 2\mu \cup X_1 > 2\mu)]$$

$$D = \sum_{c=0}^{2\mu} P(X_3 = c) * P[(X_2 \leq 2 \cap X_2 \leq 2\mu - c) \cap (X_2 \leq 2 \cup X_1 + X_2 \leq 2\mu) \cap (X_2 > 2\mu - c \cup X_1 > 2\mu)]$$

$$\begin{aligned} c=0: \rightarrow D_1 &= P(X_3 = 0) * P[(X_2 \leq 2 \cap X_2 \leq 2\mu) \cap (X_2 \leq 2 \cup X_1 + X_2 \leq 2\mu) \cap \\ &(X_2 > 2\mu + 1 \cup X_1 > 2\mu + 1)] \\ &= P[(X_2 \leq 2) \cap (X_2 \leq 2 \cup X_1 + X_2 \leq 2\mu) \cap (X_2 > 2\mu + 1 \cup X_1 > 2\mu + 1)] \\ &* P(X_3 = 0) \end{aligned}$$



Σχήμα 6.4.7: Αναπαράσταση πιθανότητας στον χώρο

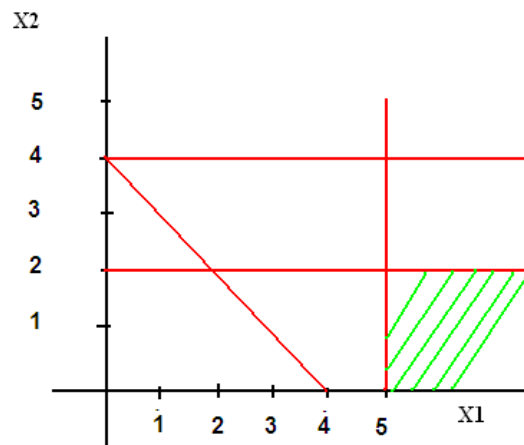
$$D1 = P(X1 \geq 5 \cap 0 \leq X2 \leq 2) * P(X3=0)$$

$$\rightarrow D1 = \sum_{j \geq 5} \frac{e^{-\mu} \mu^j}{j!} * \sum_{0 \leq j \leq 2} \frac{e^{-\mu} \mu^j}{j!} * \frac{e^{-\mu} \mu^0}{0!}$$

$$c=1 : \rightarrow D2 = P[(X2 \leq 2 \cap X2 \leq 2\mu-1) \cap (X2 \leq 2 \cup X1+X2 \leq 2\mu)$$

$$\cap (X2 \geq 2\mu \cup X1 \geq 2\mu+1)] * P(X3=1)$$

$$= P[(X2 \leq 2) \cap (X2 \leq 2 \cup X1+X2 \leq 2\mu) \cap (X2 \geq 2\mu \cup X1 \geq 2\mu+1)] * P(X3=1)$$

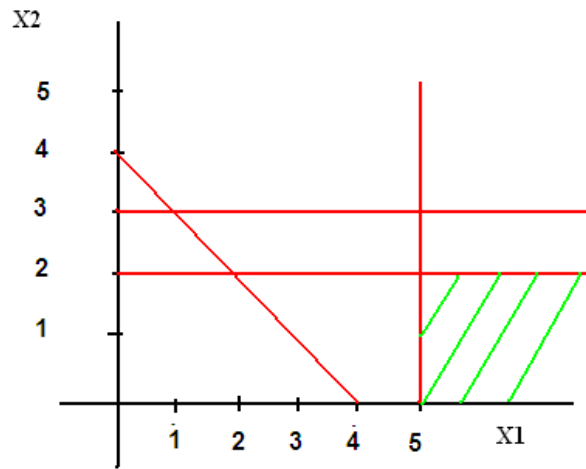


Σχήμα 6.4.8: Αναπαράσταση πιθανότητας στον χώρο

$$D2 = P(X1 \geq 5 \cap 0 \leq X2 \leq 2) * P(X3=1)$$

$$D2 = \sum_{j \geq 5} \frac{e^{-\mu} \mu^j}{j!} * \sum_{0 \leq j \leq 2} \frac{e^{-\mu} \mu^j}{j!} * \frac{e^{-\mu} \mu^1}{1!}$$

$$c=2: \rightarrow D3=P[(X2 \leq 2 \cap X2 \leq 2\mu-2) \cap (X2 \leq 2 \cup X1+X2 \leq 2\mu) \cap (X2 \geq 2\mu-1 \cup X1 \geq 2\mu+1)] * P[X3=2]$$

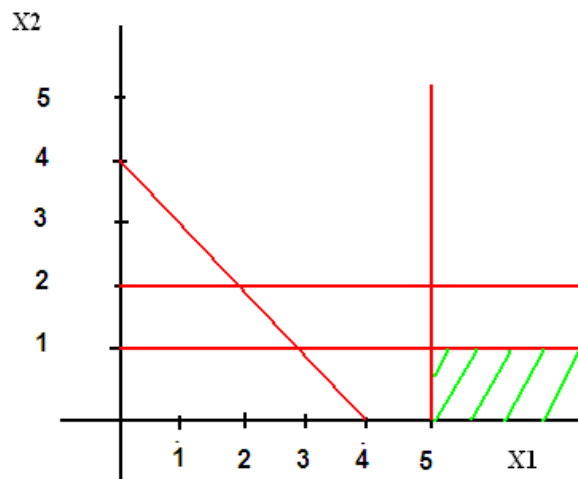


Σχήμα 6.4.9: Αναπαράσταση πιθανότητας στον χώρο

$$=P(X1 \geq 5 \cap 0 \leq X2 \leq 2) * P(X3=2)$$

$$\rightarrow D3 = \sum_{j \geq 5} \frac{e^{-\mu} \mu^j}{j!} * \sum_{0 \leq j \leq 2} \frac{e^{-\mu} \mu^j}{j!} * \frac{e^{-\mu} \mu^2}{2!}$$

$$c=3: D4=P[(X2 \leq 2 \cap X2 \leq 2\mu-3) \cap (X2 \leq 2 \cup X1+X2 \leq 2\mu) \cap (X2 \geq 2\mu-2 \cup X1 \geq 2\mu+1)] * P[X3=3]$$



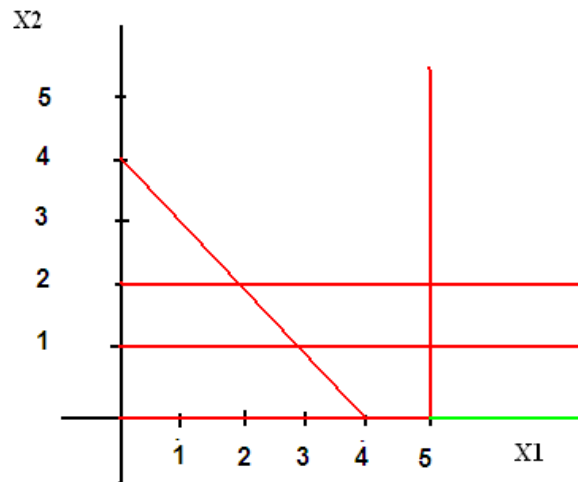
Σχήμα 6.4.10: Αναπαράσταση πιθανότητας στον χώρο

$$=P(X1 \geq 5 \cap 0 \leq X2 \leq 1) * P(X3=3)$$

$$D4 = \sum_{j \geq 5} \frac{e^{-\mu} \mu^j}{j!} * \sum_{0 \leq j \leq 1} \frac{e^{-\mu} \mu^j}{j!} * \frac{e^{-\mu} \mu^3}{3!}$$

$$c=4: D5 = P[(X2 \leq 2 \cap X2 \leq 2\mu-4) \cap (X2 \leq 2 \cup X1+X2 \leq 2\mu) \cap (X2 > 2\mu-4 \cup X1 \geq 2\mu+1)]$$

$$= P[(X2 \leq 2 \cap X2 \leq 0) \cap (X2 \leq 2 \cup X1+X2 \leq 2\mu) \cap (X2 > 1 \cup X1 \geq 2\mu+1)]$$



Σχήμα 6.4.11: Αναπαράσταση πιθανότητας στον χώρο

$$= P(X1 \geq 5 \cap X2 = 0) * P(X3 = 4)$$

$$\rightarrow D5 = \sum_{j \geq 5} \frac{e^{-\mu} \mu^j}{j!} * \frac{e^{-\mu} \mu^0}{0!} * \frac{e^{-\mu} \mu^4}{4!}$$

$$\rightarrow D = D1 + D2 + D3 + D4 + D5$$

$$P[\text{αποτυχίας}] = 0.2401247$$

$$= 24.01247\%$$

$$P[\text{επιτυχίας}] = 0.7598753$$

$$= 75.98753\%$$

6.4 Παρατηρήσεις

Η ανάλυση συμφωνεί για ακόμα μια φορά με τα πειράματα που έγιναν.

Παρατηρούμε ότι αν υπάρχουν διαθέσιμες 2 γνώσεις η πιθανότητα επιτυχίας εξαρτάται από το πώς αξιοποιούνται αυτές οι γνώσεις.

Αν 2 παίκτες ξέρουν τον ίδιο παίκτη (Περίπτωση 3) είναι χειρότερο από το να ξέρει ένας παίκτης και τους δύο άλλους Παίχτες (Περίπτωση 4) το οποίο με την σειρά του

είναι χειρότερο από το να ξέρουν δύο Παίχτες από ένα διαφορετικό Παίχτη(Περίπτωση 5) .

6.5 Βελτιστότητα Αλγορίθμου

Ενδέχεται όμως ότι ο αλγόριθμος αυτός είναι Βέλτιστος και για αυτό θα συγκριθεί με όλους τους άλλους πιθανούς καταναεμημένους αλγόριθμους χρησιμοποιώντας το πρόγραμμα «ΕΠΑΛΗΘΕΥΤΗΣ ΒΕΛΤΙΣΤΟΤΗΤΑΣ» ώστε να αποφανθούμε αν όντος είναι βέλτιστος.

Κεφάλαιο 7

Περίπτωση 6 : Ο Παίχτης 2 γνωρίζει τον Παίχτη 3

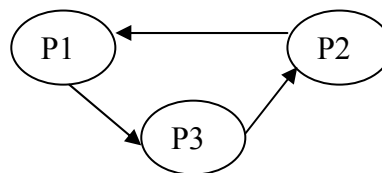
Ο Παίχτης 1 γνωρίζει τον Παίχτη 2

Ο Παίχτης 3 γνωρίζει τον Παίχτη 1

7.1 Περιγραφή	51
7.2 Αλγόριθμος	52
7.3 Πειράματα	52
7.4 Ανάλυση Αλγορίθμου	52
7.5 Παρατηρήσεις	52
7.6 Βελτιστότητα Αλγορίθμου	52

7.1 Περιγραφή

Είμαστε στην περίπτωση όπου ο Παίχτης 2 γνωρίζει τον Παίχτη 3 ,ο Παίχτης 1 γνωρίζει τον Παίχτη 2 και ο Παίχτης 3 γνωρίζει τον Παίχτη 1, όπως φαίνεται στο πιο κάτω σχήμα.



Σχήμα 7.1.1: Σχηματική αναπαράσταση Περίπτωσης 6 (cycle)

Είναι πολύ κοντινή περίπτωση με την Περίπτωση 6 ,με την διαφορά ότι ο Παίχτης 3 έχει και αυτός γνώση. Όμως παρατηρούμε ότι έχοντας 2 μηχανές και 3 εργασίες , τότε σίγουρα μια μηχανή θα έχει τουλάχιστον 2 εργασίες. Μήπως δεν μας βοηθάει κάπου αυτή η επιπλέον γνώση ; Μήπως επειδή από κάποια εργασία πρέπει να ξεκινήσει η διαδικασία ,το να έχουν όλοι από μια γνώση δεν βοηθά σε κάτι η μία γνώση ;

Επειδή δεν υπάρχει κάποιος προφανής Αλγόριθμος για την περίπτωση αυτή δοκιμάστηκε το εξής.

Αφού η Περίπτωση 5 είναι υποπερίπτωση της Περίπτωσης 6 , τότε ο αλγόριθμος της μπορεί να χρησιμοποιηθεί και για την Περίπτωση 6. Δεν θα χρησιμοποιείται όμως η επιπλέον γνώση της Περίπτωσης 6.

7.2 Αλγόριθμος

Ο ίδιος με την Περίπτωση 5

7.3 Πειράματα

Τα ίδια με την Περίπτωση 5 καθώς χρησιμοποιείται ο ίδιος Αλγόριθμος

7.4 Ανάλυση Αλγορίθμου

Είναι η ίδια με την Ανάλυση του Αλγορίθμου της Περίπτωσης 5.

7.5 Παρατηρήσεις

Είναι φανερό ότι ο Αλγόριθμος της Περίπτωσης 5 δουλεύει και για την Περίπτωση 6.

Θα χρησιμοποιήσουμε τον Αλγόριθμο αυτό σαν είσοδο στον Αλγόριθμο «ΕΠΑΛΗΘΕΥΤΗΣ ΒΕΛΤΙΣΤΟΤΗΤΑΣ» για να δούμε αν υπάρχει κάποιος άλλος καλύτερος για την περίπτωση αυτή. Αν όντως είναι βέλτιστος , μπορούμε να πούμε ότι η επιπλέον ακμή στην Περίπτωση αυτή που διαφοροποιεί την περίπτωση 5 από την Περίπτωση 6 , δεν αξίζει τίποτα.

7.6 Βελτιστότητα Αλγορίθμου

Ενδέχεται όμως ότι ο αλγόριθμος αυτός είναι Βέλτιστος και για αυτό θα συγκριθεί με όλους τους άλλους πιθανούς καταναμημένους αλγόριθμους χρησιμοποιώντας το πρόγραμμα «ΕΠΑΛΗΘΕΥΤΗΣ ΒΕΛΤΙΣΤΟΤΗΤΑΣ» ώστε να αποφανθούμε αν όντως είναι βέλτιστος.

Κεφάλαιο 8

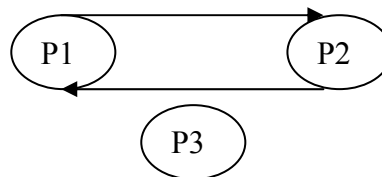
Περίπτωση 7 : Ο Παίχτης 1 γνωρίζει τον Παίχτη 2

Ο Παίχτης 2 γνωρίζει τον Παίχτη 1

8.1 Περιγραφή	53
8.2 Αλγόριθμος	54
8.3 Πειράματα	54
8.4 Ανάλυση Αλγορίθμου	55
8.5 Παρατηρήσεις	63
8.6 Βελτιστότητα Αλγορίθμου	64

8.1 Περιγραφή

Είμαστε στην περίπτωση όπου ο Παίχτης 2 γνωρίζει τον Παίχτη 1 και Παίχτης 1 γνωρίζει τον Παίχτη 2 .Ο Παίχτης 3 δεν έχει καμιά γνώση, όπως ακριβώς και το πιο κάτω σχήμα.



Σχήμα 8.1.1: Σχηματική αναπαράσταση Περίπτωσης 7

Είναι μια διαφορετική περίπτωση από τις άλλες καθώς 2 Παίχτες γνωρίζονται μεταξύ τους. Είναι ακόμα μια περίπτωση με 2 συνολικά γνώσεις. Είναι όντως καλύτερα 2 Παίχτες να γνωρίζονται μεταξύ τους παρά να γνωρίζουν τον ίδιο Παίχτη και οι δύο ή να γνωρίζουν από ένα διαφορετικό Παίχτη ; Αυτό θα φανεί στην Ανάλυση του Αλγορίθμου μας. Ένας λογικός Αλγόριθμος θα ήταν ο να μπει ο Παίχτης 3 μόνος του

και οι άλλοι 2 Παίχτες να μπουν μαζί αν μπορούν αλλιώς να μπουν ξεχωριστά με την μικρότερη εργασία $\min(x_1, x_2)$ να μπει μαζί με τον Παίχτη 3.

8.2 Αλγόριθμος

$P_3(X_3) = 0$

IF $(X_1 + X_2 \leq 2\mu)$

$P_1(X_1, X_2) = \text{bin1}$

$P_2(X_2, X_1) = \text{bin1}$

ELSE

IF $(X_1 \leq X_2)$

$P_1(X_1, X_2) = \text{bin0}$

$P_2(X_2, X_1) = \text{bin1}$

ELSE

$P_1(X_1, X_2) = \text{bin1}$

$P_2(X_2, X_1) = \text{bin0}$

8.3 Πειράματα

Με την χρήση της συνάρτησης της MATLAB $\text{poissrnd}(\mu)$ που επιστρέφει ένα ακέραιο αριθμό με βάση την κατανομή Poisson και παράμετρο $\mu=2$, έτρεξα τον αλγόριθμο αυτό για 1 εκατομμύριο επαναλήψεις και κρατούσα σε 2 μετρητές τις περιπτώσεις όπου ο αλγόριθμος αποτύχανε και τις περιπτώσεις όπου επιτύχανε. Η διαδικασία αυτή επαναλήφθηκε αρκετές φορές. Μερικά από τα πειραματικά αποτελέσματα φαίνονται στον πιο κάτω Πίνακα.

Επανάληψη	Πιθανότητα Επιτυχίας (%)
1	76.4306
2	76.7073
3	76.4228
4	76.5667

5	76.4299
6	76.4512

Πίνακας 8.3.1: Πειραματική δειγματοληψία Περίπτωσης 7

8.4 Ανάλυση Αλγορίθμου

Η ανάλυση για την περίπτωση αυτή θα γίνει με ένα διαφορετικό τρόπο.

Θα δημιουργηθεί ένας πίνακας που θα έχει στήλες όλες τις δυνατές περιπτώσεις για τις οποίες ο αλγόριθμος αυτός αποτυγχάνει και τις συνθήκες ελέγχου του αλγορίθμου.

Κάποιες περιπτώσεις δημιουργούν αντίφαση και για αυτό και διαγράφονται και κάποιες άλλες περιπτώσεις δεν ξεχειλίζουν και δεν λαμβάνονται υπόψη καθώς ψάχνουμε την πιθανότητα αποτυχίας του αλγορίθμου. Ο πίνακας δίνεται πιο κάτω.

$X3 > 2\mu$	$X2 > 2\mu$	$X1 > 2\mu$	$X1 + X3 > 2\mu$	$X2 + X3 > 2\mu$	$X1 + X2 > 2\mu$	$X1 \leq X2$
0	0	0	0	0		
0	0	0	0	1		
0	0	0	1	0		
0	0	0	1	1		
0	0	1	0	0		
0	0	1	0	1		
0	0	1	1	0		
0	0	1	1	1		
0	1	0	0	0		
0	1	0	0	1		
0	1	0	1	0		
0	1	0	1	1		
0	1	1	0	0		
0	1	1	0	1		
0	1	1	1	0		
0	1	1	1	1		
1	0	0	0	0		
1	0	0	0	1		

1	0	0	1	0		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	0	1		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	0		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		
1	1	1	1	0		
1	1	1	1	1		

Πίνακας 8.4.1: Ανάλυση όλων των περιπτώσεων αποτυχίας αλγορίθμου σε πίνακα.

Στον πιο κάτω πίνακα οι περιπτώσεις που έχουν δίπλα $\times$ είναι αυτές που δεν ξεχειλίζουν ενώ αυτές που έχουν δίπλα $\checkmark$ είναι αυτές που ξεχειλίζουν.

Και πάλι όσα είναι διαγραμμένα είναι αυτά που δημιουργούν αντιφάσεις.

Η πρώτη στήλη κρατά σε ποιο καλάθι θα μπει η κάθε εργασία και ανάλογος έχει $\checkmark$ ή $\times$ αν αυτή η κατανομή των εργασιών ξεχειλίζει ή όχι.

$\Xi EX.????$	$X3 > 2\mu$	$X2 > 2\mu$	$X1 > 2\mu$	$X1 + X3 > 2\mu$	$X2 + X3 > 2\mu$	$X1 + X2 > 2\mu$	$X1 \leq X2$
	0	0	0	0	1	0	0
1 1 0 $\times$						0	1
						1	0
0 1 0 $\times$						1	1
1 0 0 $\times$	0	0	0	1	0	0	0
						0	1
1 0 0 $\times$						1	0
						1	1
1 1 0 $\times$	0	0	0	1	1	0	0

1 1 0 ×						0	1
1 0 0 ✓						1	0
0 1 0 ✓						1	1
	0	0	1	1	0	0	0
						0	1
1 0 0 ✓						1	0
						1	1
	0	0	1	1	1	0	0
						0	1
1 0 0 ✓						1	0
						1	1
	0	1	0	0	1	0	0
						0	1
						1	0
0 1 0 ✓						1	1
	0	1	0	1	1	0	0
						0	1
						1	0
0 1 0 ✓						1	1
	0	1	1	1	1	0	0
						0	1
0 1 0 ✓						1	0
1 0 0 ✓						1	1
1 0 0 ✓	1	0	0	1	1	0	0
0 1 0 ✓						0	1
1 0 0 ✓						1	0
0 1 0 ✓						1	1
	1	0	1	1	1	0	0
						0	1
1 0 0 ✓						1	0
						1	1
	1	1	0	1	1	0	0

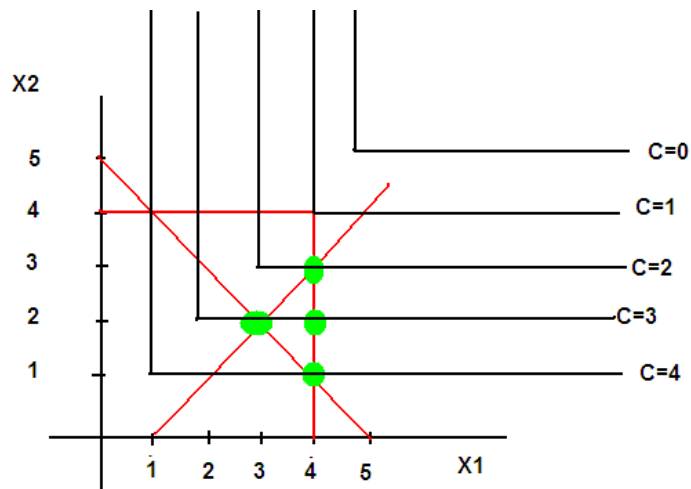
						0	1
						1	0
0 1 0 ✓						1	1
	1	1	1	1	1	0	0
						0	1
1 0 0 ✓						1	0
0 1 0 ✓						1	1

Πίνακας 8.4.2 :Αναπαράσταση περιπτώσεων αποτυχίας σε πίνακα

Τώρα για κάθε περίπτωση που έχει δίπλα σε αυτή ✓ , θα υπολογίσουμε την πιθανότητα της. Το άθροισμα όλων των περιπτώσεων με ✓ είναι η πιθανότητα αποτυχίας του αλγορίθμου. Πιο κάτω παίρνουμε αναλυτικά τις περιπτώσεις αυτές.

$$1. P(X_3 \leq 2\mu \cap X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 > X_2)$$

$$\sum_{\gamma=0}^{2\mu} P(X_3 = \gamma) * P(X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 > 2\mu - \gamma \cap X_2 > 2\mu - \gamma \cap X_1 + X_2 > 2\mu \cap X_1 > X_2)$$



Σχήμα 8.4.1 : Αναπαράσταση πιθανότητας στον χώρο

$\gamma=0,1 \rightarrow$ Δεν υπάρχει τομή

$\gamma=2$:

$$A1 = P(X1=4 \cap X2=3) * P(X3=2) = \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^4}{4!} * \frac{e^{-\mu} \mu^2}{2!} =$$

$\gamma=3$:

$$A2 = [P(X1=4 \cap X2=3) + P(X1=3 \cap X2=2) + P(X1=4 \cap X2=2)] * P(X3=3) \\ = \left(\frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^4}{4!} + \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^3}{3!} + \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^4}{4!} \right) * \frac{e^{-\mu} \mu^3}{3!}$$

$\gamma=4$:

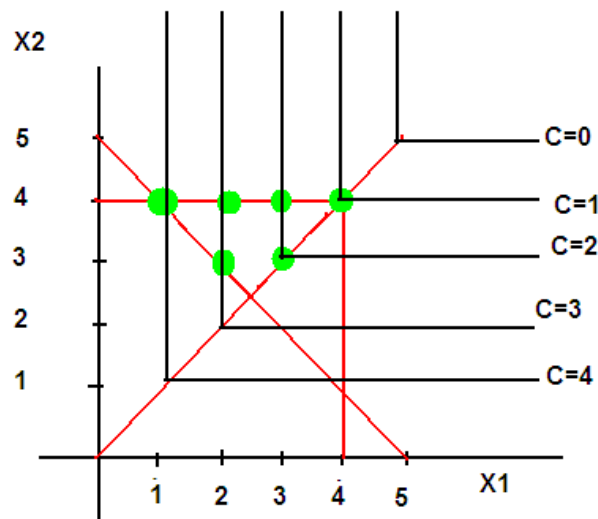
$$A3 = [P(X1=4 \cap X2=3) + P(X1=3 \cap X2=2) + P(X1=4 \cap X2=2) + P(X1=4 \cap X2=1)] * P(X3=4)$$

$$\left(\frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^4}{4!} + \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^3}{3!} + \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^4}{4!} + \frac{e^{-\mu} \mu^1}{1!} * \frac{e^{-\mu} \mu^4}{4!} \right) * \frac{e^{-\mu} \mu^4}{4!}$$

$$A = A1 + A2 + A3$$

$$2. P(X3 \leq 2\mu \cap X2 \leq 2\mu \cap X1 \leq 2\mu \cap X1 + X3 > 2\mu \cap X2 + X3 > 2\mu \cap X1 + X2 > 2\mu \cap X1 \leq X2)$$

$$\sum_{\gamma=0}^{2\mu} P(X3 = \gamma) * P(X2 \leq 2\mu \cap X1 \leq 2\mu \cap X1 > 2\mu - \gamma \cap X2 > 2\mu - \gamma \cap X1 + X2 > 2\mu \cap X1 \leq X2)$$



Σχήμα 8.4.2 : Αναπαράσταση πιθανότητας στον χώρο

$c=\gamma=0 \rightarrow$ δεν υπάρχει τομή

$$\gamma=1 : \rightarrow B1 = P(X1=4 \cap X2=4) * P(X3=1) = \frac{e^{-\mu} \mu^4}{4!} * \frac{e^{-\mu} \mu^4}{4!} * \frac{e^{-\mu} \mu^1}{1!} =$$

$$\gamma=2 : \rightarrow B2 = [P(X1=4 \cap X2=4) + P(X1=3 \cap X2=3) + P(X1=3 \cap X2=4)] * P(X3=2)$$

$$= \left(\frac{e^{-\mu} \mu^4}{4!} * \frac{e^{-\mu} \mu^4}{4!} + \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^3}{3!} + \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^4}{4!} \right) * \frac{e^{-\mu} \mu^2}{2!}$$

$$\gamma=3 : \rightarrow B3 = [P(X1=4 \cap X2=4) + P(X1=3 \cap X2=3) + P(X1=3 \cap X2=4) + P(X1=2 \cap X2=3) + P(X1=2 \cap X2=4)] * P(X3=3)$$

$$\left(\frac{e^{-\mu} \mu^4}{4!} * \frac{e^{-\mu} \mu^4}{4!} + \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^3}{3!} + \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^4}{4!} + \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^2}{2!} + \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^4}{4!} \right) * \frac{e^{-\mu} \mu^3}{3!}$$

$$\gamma=4 : \rightarrow B4 = [P(X1=4 \cap X2=4) + P(X1=3 \cap X2=3) + P(X1=3 \cap X2=4) + P(X1=2 \cap X2=3) + P(X1=2 \cap X2=4) + P(X1=1 \cap X2=4)] * P(X3=4)$$

$$\left(\frac{e^{-\mu} \mu^4}{4!} * \frac{e^{-\mu} \mu^4}{4!} + \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^3}{3!} + \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^4}{4!} + \frac{e^{-\mu} \mu^3}{3!} * \frac{e^{-\mu} \mu^2}{2!} + \frac{e^{-\mu} \mu^2}{2!} * \frac{e^{-\mu} \mu^4}{4!} + \frac{e^{-\mu} \mu^1}{1!} * \frac{e^{-\mu} \mu^4}{4!} \right) * \frac{e^{-\mu} \mu^4}{4!}$$

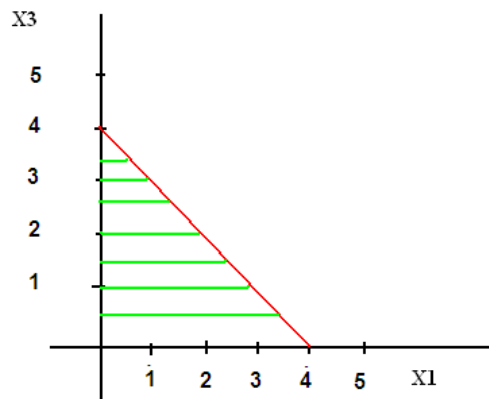
$$B = B1 + B2 + B3 + B4$$

$$3. P(X3 \leq 2\mu \cap X2 \leq 2\mu \cap X1 > 2\mu \cap X1 + X3 > 2\mu \cap X2 + X3 \leq 2\mu \cap X1 + X2 > 2\mu \cap X1 > X2)$$

$$= P(X3 \leq 2\mu \cap X2 \leq 2\mu \cap X1 > 2\mu \cap X2 + X3 \leq 2\mu)$$

$$= P(X1 > 2\mu) * P(X3 \leq 2\mu \cap X2 \leq 2\mu \cap X2 + X3 \leq 2\mu)$$

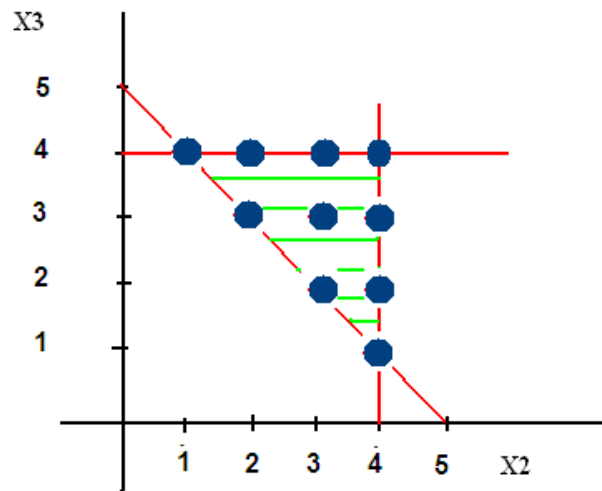
$$= P(X1 > 2\mu) * P(X2 + X3 \leq 2\mu)$$



Σχήμα 8.4.3 : Αναπαράσταση πιθανότητας στον χώρο

$$C = \sum_{j>4} \frac{e^{-\mu} \mu^j}{j!} * \sum_{a=0}^4 \sum_{b=0}^{4-a} \frac{e^{-\mu} \mu^a}{a!} * \frac{e^{-\mu} \mu^b}{b!}$$

$$\begin{aligned} 4. & P(X_3 \leq 2\mu \cap X_2 \leq 2\mu \cap X_1 > 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap \\ & X_1 > X_2) \\ & = P(X_3 \leq 2\mu \cap X_2 \leq 2\mu \cap X_1 > 2\mu \cap X_2 + X_3 > 2\mu) \\ & = P(X_1 > 2\mu) * P(X_3 \leq 2\mu \cap X_2 \leq 2\mu \cap X_2 + X_3 > 2\mu) \end{aligned}$$



Σχήμα 8.4.4 : Αναπαράσταση πιθανότητας στον χώρο

$$\begin{aligned} \rightarrow D = & P[(X_2=4 \cap X_3=1) + P(X_2=4 \cap X_3=2) + P(X_2=4 \cap X_3=3) + P(X_2=4 \cap X_3=4) \\ & + P(X_2=3 \cap X_3=2) + P(X_2=3 \cap X_3=3) + (X_2=3 \cap X_3=4) + P(X_2=2 \cap X_3=3) + \\ & P(X_2=2 \cap X_3=4) + P(X_2=1 \cap X_3=4)] * P(X_1 > 2\mu) \end{aligned}$$

$$\begin{aligned} 5. & P(X_3 \leq 2\mu \cap X_2 > 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 \leq 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap \\ & X_1 \leq X_2) \\ & = P(X_3 \leq 2\mu \cap X_2 > 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 \leq 2\mu) \\ & = P(X_2 > 2\mu) * P(X_3 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 \leq 2\mu) \end{aligned}$$

$$\rightarrow E = P(X_2 > 2\mu) * P(X_1 + X_3 \leq 2\mu) = C$$

$$\begin{aligned} 6. & P(X_3 \leq 2\mu \cap X_2 > 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap \\ & X_1 \leq X_2) \\ & = P(X_3 \leq 2\mu \cap X_2 > 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 > 2\mu) \\ & F = P(X_2 > 2\mu) * P(X_3 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 > 2\mu) \rightarrow F = D \end{aligned}$$

$$7. P(X_3 \leq 2\mu \cap X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 > X_2)$$

$$= P(X_3 \leq 2\mu \cap X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 > X_2)$$

$$= P(X_3 \leq 2\mu) * P(X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 > X_2)$$

$$8. P(X_3 \leq 2\mu \cap X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 \leq X_2)$$

$$= P(X_3 \leq 2\mu \cap X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 \leq X_2)$$

$$= P(X_3 \leq 2\mu) * P(X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 \leq X_2)$$

$$\rightarrow GH = G + H = 7 + 8 = P(X_3 \leq 2\mu) * P(X_2 > 2\mu \cap X_1 > 2\mu)$$

$$= P(X_3 \leq 2\mu) * P(X_2 > 2\mu) * P(X_1 > 2\mu)$$

$$9. P(X_3 > 2\mu \cap X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 \leq 2\mu \cap X_1 > X_2)$$

$$= P(X_3 > 2\mu \cap X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_2 \leq 2\mu \cap X_1 > X_2)$$

$$= P(X_3 > 2\mu) * P(X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_2 \leq 2\mu \cap X_1 > X_2)$$

$$10. P(X_3 > 2\mu \cap X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 \leq 2\mu \cap X_1 \leq X_2)$$

$$= P(X_3 > 2\mu \cap X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_2 \leq 2\mu \cap X_1 \leq X_2)$$

$$= P(X_3 > 2\mu) * P(X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_2 \leq 2\mu \cap X_1 \leq X_2)$$

$$= P(X_3 > 2\mu) * P(X_1 + X_2 \leq 2\mu \cap X_1 \leq X_2)$$

$$11. P(X_3 > 2\mu \cap X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 > X_2)$$

$$= P(X_3 > 2\mu \cap X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 > X_2)$$

$$= P(X_3 > 2\mu) * P(X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_2 > 2\mu + 1 \cap X_1 > X_2)$$

$$12. P(X_3 > 2\mu \cap X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 \leq X_2)$$

$$= P(X_3 > 2\mu \cap X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 \leq X_2)$$

$$= P(X_3 > 2\mu) * P(X_2 \leq 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 \leq X_2)$$

$$IJKL = I + J + K + L = 9 + 10 + 11 + 12 = P(X_3 > 2\mu) * P(X_2 \leq 2\mu) * P(X_1 \leq 2\mu)$$

$$13. P(X_3 > 2\mu \cap X_2 \leq 2\mu \cap X_1 > 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 > X_2)$$

$$= P(X_3 > 2\mu \cap X_2 \leq 2\mu \cap X_1 > 2\mu)$$

$$= P(X_3 > 2\mu) * P(X_2 \leq 2\mu \cap X_1 > 2\mu)$$

$$\rightarrow M = P(X_3 > 2\mu) * P(X_2 \leq 2\mu \cap X_1 > 2\mu)$$

$$14. P(X_3 > 2\mu \cap X_2 > 2\mu \cap X_1 \leq 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 \leq X_2)$$

$$= P(X_3 > 2\mu \cap X_2 > 2\mu \cap X_1 \leq 2\mu)$$

$$\rightarrow N = P(X_3 > 2\mu) * P(X_2 > 2\mu) * P(X_1 \leq 2\mu) = M$$

$$15. P(X_3 > 2\mu \cap X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 > X_2)$$

$$= P(X_3 > 2\mu \cap X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 > X_2)$$

$$= P(X_3 > 2\mu) * P(X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 > X_2)$$

$$16. P(X_3 > 2\mu \cap X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 + X_3 > 2\mu \cap X_2 + X_3 > 2\mu \cap X_1 + X_2 > 2\mu \cap X_1 \leq X_2)$$

$$= P(X_3 > 2\mu \cap X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 \leq X_2)$$

$$= P(X_3 > 2\mu) * P(X_2 > 2\mu \cap X_1 > 2\mu \cap X_1 \leq X_2)$$

$$\rightarrow OP = O + P = 15 + 16 = P(X_3 > 2\mu) * P(X_2 > 2\mu) * P(X_1 > 2\mu)$$

$$P[\text{αποτυχίας}] = A + B + C + D + E + F + G + H + I + J + K + L + M + N + O + P = 0.2357181 = 23.57181\%$$

$$P[\text{επιτυχίας}] = 0.7642819 = 76.42819\%$$

8.5 Παρατηρήσεις

Για ακόμα μια φορά τα πειράματα συμφώνησαν με την πραγματική πιθανότητα που πήραμε στην ανάλυση.

Για την περίπτωση αυτή έχουμε την μέγιστη πιθανότητα επιτυχίας. Και στην περίπτωση αυτή υπάρχουν 2 συνολικά γνώσεις όπως και στην περίπτωση 3 και στην περίπτωση 4 αλλά η περίπτωση αυτή έχει την πιο μεγάλη πιθανότητα επιτυχίας.

8.6 Βελτιστότητα Αλγορίθμου

Αν επιλέγαμε να αποδείξουμε βελτιστότητα για την Περίπτωση αυτή με την χρήση του αλγορίθμου «ΕΠΑΛΗΘΕΥΤΗΣ ΒΕΛΤΙΣΤΟΤΗΤΑΣ» (βλέπε ΚΕΦ 1.1.6) θα ήταν χρονοβόρα διαδικασία για τους λόγους που εξηγήθηκαν και πιο πριν και έτσι θα χρησιμοποιηθεί ένας άλλος τρόπος που έχει να κάνει με το να πάρουμε όλες τις δυνατές τιμές για τις εργασίες X_1, X_2, X_3 και να βγάλουμε κάποια συμπεράσματα. Ο τρόπος είναι παρόμοιος με αυτόν που χρησιμοποιήθηκε για την εύρεση του καλύτερου κατωφλίου για την Περίπτωση 1.

Παρατηρούμε ότι κάθε αλγόριθμος κερδίζει αν $X_1 + X_2 + X_3 \leq 2\mu$ καθώς όπως και να μουν οι 3 εργασίες στα 2 καλάθια πάντα θα υπάρχει επιτυχία.

Είναι γεγονός ότι έχοντας 2 καλάθια και 3 εργασίες, σίγουρα στο ένα καλάθι θα μουν τουλάχιστον οι 2 εργασίες από τις 3 (Αρχή του Περιστερώνα).

Έτσι παρατηρούμε ότι αν $X_1 + X_2 + X_3 \geq 9$ κανένας αλγόριθμος, ακόμα και ο βέλτιστος για κάθε περίπτωση, δεν θα μπορεί να επιτύχει, καθώς είτε μια εργασία από τις τρεις είναι μεγαλύτερη του 2μ , είτε ανά 2 οι εργασίες έχουν άθροισμα μεγαλύτερο του 2μ .

Ο πιο κάτω πίνακας δείχνει όλες τις περιπτώσεις όπου $X_1 + X_2 + X_3 = 9$. Εννοείται ότι οι τιμές των X_1, X_2, X_3 μπορεί να έρθουν με οποιαδήποτε σειρά.

X_1	X_2	X_3
0	9	0
0	8	1
0	7	2
0	6	3
0	5	4
1	4	4
1	5	3
1	2	6

1	1	7
2	4	3
2	5	2
3	3	3

Πίνακας 8.6.1 : Περιπτώσεις όπου $X1+X2+X3=9$

Οι περιπτώσεις όπου ένας Αλγόριθμος μπορεί να έχει διαφορετικά αποτελέσματα από ένα άλλο είναι οι περιπτώσεις όπου

$$X1+X2+X3=5$$

$$X1+X2+X3=6$$

$$X1+X2+X3=7$$

$$X1+X2+X3=8$$

και πιο συγκεκριμένα κάποιες υποπεριπτώσεις των περιπτώσεων αυτών , καθώς για τον ίδιο λόγο που ανέφερα πιο πάνω , όλοι οι αλγόριθμοι αποτυγχάνουν.

X1	X2	X3
2	2	1
3	2	0
3	1	1
4	1	0
5	0	0

X1	X2	X3
2	2	2
4	1	1
3	3	0
3	2	1
4	2	0
6	0	0
5	1	0

X1	X2	X3
3	3	1
4	2	1
2	2	3
4	3	0
5	1	1
5	2	0
6	1	0
7	0	0

X1	X2	X3
4	4	0
4	3	1
2	2	4
3	3	2
7	0	1
6	2	0
6	1	1
8	0	0
5	3	0
5	2	1

Πίνακας 8.6.2: Περιπτώσεις όπου οι αλγόριθμοι έχουν διαφορετικά αποτελέσματα

Ας ονομάσουμε την περιοχή όπου $X1+X2+X3 \leq 2$ ως ΑΣΠΡΗ ΖΩΝΗ, καθώς όλοι κερδίζουν.

Ας ονομάσουμε την περιοχή όπου $X1+X2+X3 \geq 9$ ως ΜΑΥΡΗ ΖΩΝΗ, καθώς όλοι χάνουν.

Ας ονομάσουμε την περιοχή όπου $5 \leq X1+X2+X3 \leq 8$, ως ΓΚΡΙΖΑ ΖΩΝΗ.

Στην ΓΚΡΙΖΑ ΖΩΝΗ εξαρτάται από το πόσο καλός είναι ο κάθε αλγόριθμος.

Είναι παραδεχτό ότι οποιοσδήποτε αλγόριθμος που επιτυγχάνει σε όλες τις περιπτώσεις της ΓΚΡΙΖΑΣ ΖΩΝΗΣ είναι ΒΕΛΤΙΣΤΟΣ.

Ακόμα ένα σημαντικό χαρακτηριστικό για κάποιο Αλγόριθμο που κερδίζει σε όλη την ΓΚΡΙΖΑ ΖΩΝΗ είναι το γεγονός ότι ,έχοντας 3 Παίχτες και 2 μηχανές χωρητικότητας ίση με 4 , ο αλγόριθμος αυτός είναι ΒΕΛΤΙΣΤΟΣ για ΚΑΘΕ ΚΑΤΑΝΟΜΗ.

Ας ελέγξουμε τον αλγόριθμο αυτό πως συμπεριφέρεται στην ΓΚΡΙΖΑ ΖΩΝΗ.

$$X1+X2+X3=5$$

X1	X2	X3	Επιτυγχάνει?
2	2	1	✓
2	1	2	✓
1	2	2	✓
3	2	0	✓
3	0	2	✓
0	3	2	✓
0	2	3	✓
2	3	0	✓
2	0	3	✓
3	1	1	✓
1	3	1	✓
1	1	3	✓
4	1	0	✓
4	0	1	✓
0	1	4	✓
0	4	1	✓
1	4	0	✓

Πίνακας 8.6.3

$$X1+X2+X3=6$$

X1	X2	X3	Επιτυγχάνει?
2	2	2	✓
4	1	1	✓
1	4	1	✓
1	1	4	✓
3	3	0	✓
3	0	3	✓
0	3	3	✓
3	2	1	✓
3	1	2	✓
2	1	3	✓
2	3	1	✓
1	2	3	✓
1	3	2	✓
4	2	0	✓
4	0	2	✓
0	2	4	✓
0	4	2	✓
2	4	0	✓
2	0	4	✓

Πίνακας 8.6.4

$$X1+X2+X3=7$$

X1	X2	X3	Επιτυγχάνει??
3	3	1	✓
3	1	3	✓
1	3	3	✓
4	2	1	✓
4	1	2	✓
1	2	4	✓
1	4	2	✓
2	4	1	✓
2	1	4	✓
2	2	3	✓
2	3	2	✓
3	2	2	✓
4	3	0	✓
4	0	3	✓
3	0	4	✓
3	4	0	✓
0	4	3	✓
0	3	4	✓

Πίνακας 8.6.5

$$X1+X2+X3=8$$

X1	X2	X3	Επιτυγχάνει??
4	4	0	✓
4	0	4	✓
0	4	4	✓
4	3	1	✓
4	1	3	✓
1	3	4	✓
1	4	3	✓
3	4	1	✓
3	1	4	✓
2	2	4	✓
2	4	2	✓
4	2	2	✓

Πίνακας 8.6.6

Παρατηρούμε ότι ο αλγόριθμος κερδίζει ΟΛΕΣ τις περιπτώσεις της γκρίζας ζώνης. Έτσι είναι Βέλτιστος για την περίπτωση αυτή , αλλά και Βέλτιστος για κάθε άλλη κατανομή με τα χαρακτηριστικά της Περίπτωσης αυτής (πχ 3 εργασίες και 2 μηχανές χωρητικότητας 4).

Κεφάλαιο 9

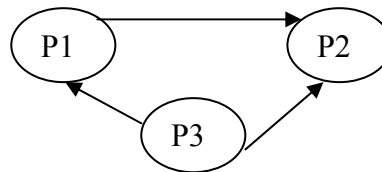
Περίπτωση 8 : Ο Παίχτης 1 γνωρίζει τον Παίχτη 3

Ο Παίχτης 2 γνωρίζει τον Παίχτη 1 και τον Παίχτη 3

9.1 Περιγραφή	68
9.2 Αλγόριθμος	69
9.3 Πειράματα	69
9.4 Ανάλυση Αλγορίθμου	70
9.5 Παρατηρήσεις	70
9.6 Βελτιστότητα Αλγορίθμου	70

9.1 Περιγραφή

Είμαστε στην περίπτωση όπου ο Παίχτης 2 γνωρίζει τον Παίχτη 1 και τον Παίχτη 3 ενώ ο Παίχτης 1 γνωρίζει μόνο τον Παίχτη 3. Ο Παίχτης 3 δεν έχει καμιά γνώση, όπως ακριβώς και το πιο κάτω σχήμα.



Σχήμα 9.1.1: Σχηματική αναπαράσταση Περίπτωσης 8

Η ιδέα για τον Αλγόριθμο που θα χρησιμοποιηθεί για την περίπτωση αυτή είναι ο Παίχτης 3 να πηγαίνει μόνος του, ο Παίχτης 1 βλέποντας τον Παίχτη 3 θα ελέγξει αν μπορεί να μπει μαζί του ή όχι και θα πράξει ανάλογα, και ο Παίχτης 2 θα παίξει βέλτιστα καθώς γνωρίζει τα πάντα.

9.2 Αλγόριθμος

```
P3(X3)=bin0
IF(X1+X3<=2M)
    P1(X1,X3)=bin0
ELSE
    P1(X1,X3)=bin1
IF(X1+X3<=2M)
    P2(X2,X1,X3)=bin1
ELSE
    IF (X1<=X3)
        P2(X2,X1,X3)=bin1
    ELSE
        P2(X2,X1,X3)=bin0
```

9.3 Πειράματα

Με την χρήση της συνάρτησης της MATLAB `poissrnd(μ)` που επιστρέφει ένα ακέραιο αριθμό με βάση την κατανομή Poisson και παράμετρο $\mu=2$, έτρεξα τον αλγόριθμο αυτό για 1 εκατομμύριο επαναλήψεις και κρατούσα σε 2 μετρητές τις περιπτώσεις όπου ο αλγόριθμος αποτύχανε και τις περιπτώσεις όπου επιτύχανε. Η διαδικασία αυτή επαναλήφθηκε αρκετές φορές. Μερικά από τα πειραματικά αποτελέσματα φαίνονται στον πιο κάτω Πίνακα.

Επανάληψη	Πιθανότητα Επιτυχίας (%)
1	76.4404
2	76.5689
3	76.3434
4	76.4231
5	76.4299
6	76.3544

Πίνακας 9.3.1: Πειραματική δειγματοληψία Περίπτωσης 8

9.4 Ανάλυση Αλγορίθμου

Ο αλγόριθμος αυτός δεν θα αναλυθεί καθώς θαδειχθεί πιο κάτω ότι και αυτός κερδίζει σε κάθε περιοχή της ΓΚΡΙΖΑΣ ΖΩΝΗΣ (βλέπε ΚΕΦ 8.6), όπως και ο αλγόριθμος της Περίπτωσης 7.

9.5 Παρατηρήσεις

Ο λόγος που υποψιαστήκαμε ότι ο αλγόριθμος αυτός έχει τα ίδια αποτελέσματα με τον αλγόριθμο της Περίπτωσης 7 είναι τα πολύ κοντινά πειραματικά αποτελέσματα των 2 περιπτώσεων. Και επειδή η συνάρτηση `poissrnd()` της MATLAB μας έδειξε μέχρι τώρα ότι είναι αρκετά αξιόπιστη καθώς όλες οι αναλύσεις των αλγορίθμων συμφωνούσαν με τα πειράματα που έγιναν, υποθέσαμε ότι δεν είναι τυχαίο το γεγονός ότι τα πειραματικά αποτελέσματα της Περίπτωσης 7 και 8 είναι πάρα πολύ κοντινά.

9.6 Βελτιστότητα Αλγορίθμου

Για κάθε περιοχή της ΓΚΡΙΖΑΣ ΖΩΝΗΣ θα ελέγξουμε την συμπεριφορά του αλγορίθμου μας, όπως ακριβώς έγινε και με τη Περίπτωση 7.

Ας ελέγξουμε λοιπόν τον αλγόριθμο αυτό πως συμπεριφέρεται στην ΓΚΡΙΖΑ ΖΩΝΗ.

$$X1+X2+X3=5$$

X1	X2	X3	Επιτυγχάνει?
2	2	1	✓
2	1	2	✓
1	2	2	✓
3	2	0	✓
3	0	2	✓
0	3	2	✓
0	2	3	✓
2	3	0	✓
2	0	3	✓
3	1	1	✓
1	3	1	✓
1	1	3	✓
4	1	0	✓
4	0	1	✓
0	1	4	✓
0	4	1	✓
1	4	0	✓

Πίνακας 9.6.1

$$X1+X2+X3=6$$

X1	X2	X3	Επιτυγχάνει?
2	2	2	✓
4	1	1	✓
1	4	1	✓
1	1	4	✓
3	3	0	✓
3	0	3	✓
0	3	3	✓
3	2	1	✓
3	1	2	✓
2	1	3	✓
2	3	1	✓
1	2	3	✓
1	3	2	✓
4	2	0	✓
4	0	2	✓
0	2	4	✓
0	4	2	✓
2	4	0	✓
2	0	4	✓

Πίνακας 9.6.2

$$X1+X2+X3=7$$

X1	X2	X3	Επιτυγχάνει??
3	3	1	✓
3	1	3	✓
1	3	3	✓
4	2	1	✓
4	1	2	✓
1	2	4	✓
1	4	2	✓
2	4	1	✓
2	1	4	✓
2	2	3	✓
2	3	2	✓
3	2	2	✓
4	3	0	✓
4	0	3	✓
3	0	4	✓
3	4	0	✓
0	4	3	✓
0	3	4	✓

Πίνακας 9.6.3

$$X1+X2+X3=8$$

X1	X2	X3	Επιτυγχάνει??
4	4	0	✓
4	0	4	✓
0	4	4	✓
4	3	1	✓
4	1	3	✓
1	3	4	✓
1	4	3	✓
3	4	1	✓
3	1	4	✓
2	2	4	✓
2	4	2	✓
4	2	2	✓

Πίνακας 9.6.4

Δηλαδή ο αλγόριθμος αυτός κερδίζει ακριβώς εκεί που κερδίζει και ο αλγόριθμος της Περίπτωσης 7 και χάνει ακριβώς εκεί που χάνει ο αλγόριθμος της Περίπτωσης 7.

$$P[\text{επιτυχίας Αλγορίθμου Περίπτωσης 8}] = P[\text{επιτυχίας Αλγορίθμου Περίπτωσης 7}] = 0.7642819 = 76.42819\%$$

Παρατηρούμε ότι ο αλγόριθμος κερδίζει ΟΛΕΣ τις περιπτώσεις της γκρίζας ζώνης. Έτσι είναι Βέλτιστος για την περίπτωση αυτή , αλλά και Βέλτιστος για κάθε άλλη κατανομή με τα χαρακτηριστικά της Περίπτωσης αυτής (πχ 3 εργασίες και 2 μηχανές χωρητικότητας 4).

Κεφάλαιο 10

Συμπεράσματα

10.1 Σύνοψη	73
10.2 Συμπεράσματα	73
10.3 Μελλοντική Δουλειά	74

10.1 Σύνοψη

Η εργασία αυτή ξεκίνησε με μια γενική αναφορά στα κατανεμημένα συστήματα και τους κατανεμημένους αλγόριθμους. Στη συνέχεια αναφερθήκαμε στο συγκεκριμένο μοντέλο της κατανεμημένης λήψης αποφάσεων το οποίο αποτελούσε και την βασική βιβλιογραφία [1] για την εργασία αυτή.

Τροποποιήσαμε το μοντέλο αυτό , αλλάζοντας την κατανομή που χρησιμοποιούσε καθώς και τις χωρητικότητες των δύο επεξεργαστών και δημιουργήθηκε έτσι ένα νέο μοντέλο.

Για κάθε περίπτωση επικοινωνίας (μελετήθηκαν συνολικά 8 περιπτώσεις επικοινωνίας) , επιλέξαμε ένα αλγόριθμο , τον αναλύσαμε ,βρήκαμε την πιθανότητα επιτυχίας του και δείξαμε κατά πόσο είναι βέλτιστος.

Για κάθε περίπτωση υπήρχαν κάποια νέα πράγματα , για παράδειγμα νέες γνώσεις και έτσι ο αλγόριθμος για κάθε περίπτωση έπρεπε να επιλεγεί προσεκτικά.

Κάθε αλγόριθμος που συνδυάζει τους αλγορίθμους των Παιχτών 1,2,3 είναι ένας Κατανεμημένος Αλγόριθμος.

10.2 Συμπεράσματα

Από αυτή την εργασία πάρθηκαν διάφορα συμπεράσματα. Για παράδειγμα για κάθε περίπτωση στο δικό μας μοντέλο κατανεμημένου συστήματος , πήραμε μεγαλύτερη

πιθανότητα επιτυχίας από το μοντέλο κατανεμημένου συστήματος το οποίο μελετήθηκε στην βασική βιβλιογραφία.

Ακόμα για τους αλγορίθμους των περιπτώσεων 7 και 8 μπορούμε να πούμε ότι είναι βέλτιστοι και οι 2 για ΚΑΘΕ κατανομή ,στο δικό μας μοντέλο κατανεμημένου συστήματος καθώς κερδίζουν σε κάθε περιοχή της «ΓΚΡΙΖΑΣ ΖΩΝΗΣ».

Είναι παραδεχτό το γεγονός ότι στο μοντέλο μας χρησιμοποιούσαμε διακριτή κατανομή ευκόλυνε την κατάσταση όσο αφορά τις βελτιστότητες των αλγορίθμων , την «ΓΚΡΙΖΑ ΖΩΝΗ» που χρησιμοποιήθηκε , την επιλογή καταωφλίου κ.α.

Όμως αν η κατανομή ήταν συνεχής , ποια θα ήταν η διαφορά όσον αφορά στα σημεία αυτά ; Θα υπήρχε κάποια διαφορά ή όχι ; Αυτό μπορεί να μελετηθεί εκτεταμένα και να παρθούν κάποια συμπεράσματα όσον αφορά τις διακριτές και τις συνεχείς κατανομές.

Μια ακόμα παρατήρηση που έχει να κάνει με άλλο θέμα , είναι ο generator τιμών με βάση την κατανομή Poisson που παρέχει η MATLAB, καθώς είναι πολύ αξιόπιστος , γιατί τα πειράματα που γίνονταν κάθε φορά συμφωνούσαν με την πραγματική ανάλυση του αλγορίθμου σε πολύ μεγάλο βαθμό.

10.3 Μελλοντική Δουλειά

Λόγω του γεγονότος ότι σε κάποιες περιπτώσεις χρειάζεται πάρα πολλής χρόνος για να αποφασίσει ο «ΕΠΑΛΗΘΕΥΤΗΣ ΒΕΛΤΙΣΤΟΤΗΤΑΣ» αν όντως ο δικός μας αλγόριθμος είναι ο βέλτιστος, δεν πήραμε κάποια απάντηση ακόμα . Υπάρχουν οι υποψίες για βελτιστότητα όμως πρέπει να επαληθευτούν.

Επιπλέον μπορεί να τροποποιηθεί ο αλγόριθμος «ΕΠΑΛΗΘΕΥΤΗΣ ΒΕΛΤΙΣΤΟΤΗΤΑΣ» , έτσι ώστε να μην επιστρέφει μόνο αν ο αλγόριθμος είναι ή όχι βέλτιστος , αλλά να επιστρέφει και την πιο δυνατή μάχη που έγινε μεταξύ 2 κατανεμημένων αλγορίθμων ή σε ταξινομημένη σειρά τους κατανεμημένους αλγόριθμους ανάλογα με το ποιους νικούν και από ποιους χάνουν.

Ακόμα μπορεί να γίνει και ανάλυση του μοντέλου αυτού με διαφορετική παράμετρο μ , έτσι ώστε να κατανοήσουμε πόσο σημαντική είναι η παράμετρος μ στην κατανομή Poisson και στο μοντέλο του κατανεμημένου συστήματος που μελετήθηκε.

Βιβλιογραφία

- [1] Christos H. Papadimitriou and Mihalis Yannakakis, “On the value of information in distributed decision making” , Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing, August 1991

- [2] M. Mitzenmacher and E. Upfal ,”Probability and Computing”, Randomized Algorithms and Probabilistic Analysis, 2005

Παράρτημα Α

Στο παράρτημα αυτό δίνονται τα πειράματα που έγιναν σε MATLAB για την κάθε μια από τις 8 περιπτώσεις. Από τα πειράματα πήραμε αποτελέσματα προσεγγιστικά ως προς την πραγματική πιθανότητα επιτυχίας . Τρέχουμε τον αλγόριθμο για 1000000 επαναλήψεις και κάθε φορά κρατούμε αν ξεχείλισε ή όχι κάποια μηχανή.

Η συνάρτηση `poissrnd(lambda)` επιστρέφει μια τιμή για κάθε εργασία με βάση την κατανομή Poisson.

Οι αλγόριθμοι για την κάθε Περίπτωση διαφέρουν μόνο στο κομμάτι όπου τρέχει ο αντίστοιχος αλγόριθμος.

Περίπτωση 1

```
%{  
the case of no communication  
%}  
lambda=2;  
numOfRunOvers=0;  
numOfNoRunOvers=0;  
repeats=1000000;  
for i=1:repeats  
    bin0=2*lambda;    %{initialize bin 0 size with 4%}  
    bin1=2*lambda;    %{initialize bin 1 size with 4%}  
    x1=poissrnd(lambda);%{generating a poisson variable named X1%}  
    x2=poissrnd(lambda);%{generating a poisson variable named X2%}  
    x3=poissrnd(lambda);%{generating a poisson variable named X3%}  
    disp(i);          %{display the counter i on the screen%}  
  
    %{//////////Algorithm Starts%}  
    if x1<=2  
        bin0=bin0-x1;  
    else  
        bin1=bin1-x1;  
    end
```

```

if x2<=2
    bin0=bin0-x2;
else
    bin1=bin1-x2;
end
if x3<=2
    bin0=bin0-x3;
else
    bin1=bin1-x3;
end
%{//////////Algorithm Ends%}
%{check if at least one the bins has been overflowed
%{and increase the appropriate counter by one%}
if((bin0>=0)&&(bin1>=0))
    numOfNoRunOvers=numOfNoRunOvers+1;
else
    numOfRunOvers=numOfRunOvers+1;
end
end%{for loop%}
%{Display on screen : %}
percent1=(numOfRunOvers*100)/repeats;
percent2=(numOfNoRunOvers*100)/repeats;
disp('Overflow');
disp(percent1);
disp('Success');
disp(percent2);

```

Περίπτωση 2

```
%{Player2 know x3 and Player1,3 have no knowledge %}
lambda=2;
numOfRunOvers=0;
numOfNoRunOvers=0;
repeats=1000000;
for i=1:repeats
    bin0=2*lambda;      %{initialize bin 0 size with 4%}
    bin1=2*lambda;      %{initialize bin 1 size with 4%}
    x1=poissrnd(lambda);%{generating a poisson variable named X1%}
    x2=poissrnd(lambda);%{generating a poisson variable named X2%}
    x3=poissrnd(lambda);%{generating a poisson variable named X3%}
    disp(i);             %{display the counter i on the screen%}

    %{//////////Algorithm Starts%}
    bin0=bin0-x3;
    bin1=bin1-x1;
    if((x2+x3)<=(2*lambda))
        bin0=bin0-x2;
    else
        bin1=bin1-x2;
    end
    %{//////////Algorithm Ends%}
    %{check if at least one the bins has been overflowed
    %{and increase the appropriate counter by one%}
    if((bin0>=0)&&(bin1>=0))
        numOfNoRunOvers=numOfNoRunOvers+1;
    else
        numOfRunOvers=numOfRunOvers+1;
    end
end %{for loop%}
%{Display on screen : %}
percent1=(numOfRunOvers*100)/repeats;
```

```
percent2=(numOfNoRunOvers*100)/repeats;  
disp('Overflow');  
disp(percent1);  
disp('Success');  
disp(percent2);
```

Περίπτωση 3

```
%{  
Player1 know x3 and Player2 know x3  
Player 3 has no knowledge  
%}  
lambda=2;  
numOfRunOvers=0;  
numOfNoRunOvers=0;  
repeats=1000000;  
for i=1:repeats  
    bin0=2*lambda;      %{initialize bin 0 size with 4%}  
    bin1=2*lambda;      %{initialize bin 1 size with 4%}  
    x1=poissrnd(lambda);%{generating a poisson variable named X1%}  
    x2=poissrnd(lambda);%{generating a poisson variable named X2%}  
    x3=poissrnd(lambda);%{generating a poisson variable named X3%}  
    disp(i);            %{display the counter i on the screen%}  
  
    %{//////////Algorithm Starts%}  
    bin0=bin0-x3;  
    if(x1+x3<=2*lambda)  
        bin0=bin0-x1;  
    else  
        bin1=bin1-x1;  
    end  
    if(x2+x3<=2*lambda)  
        bin0=bin0-x2;  
    else  
        bin1=bin1-x2;  
    end  
  
    %{//////////Algorithm Ends%}  
    %{check if at least one the bins has been overflowed  
    %{and increase the appropriate counter by one%}
```

```

if((bin0>=0)&&(bin1>=0))
    numOfNoRunOvers=numOfNoRunOvers+1;
else
    numOfRunOvers=numOfRunOvers+1;
end
end%{for loop%}
%{Display on screen : %}
percent1=(numOfRunOvers*100)/repeats;
percent2=(numOfNoRunOvers*100)/repeats;
disp('Overflow');
disp(percent1);
disp('Success');
disp(percent2);

```

Περίπτωση 4

```
%{  
Player3 knows Player1 and  
Player3 knows Player2  
Player1,2 have no knowledge  
%}  
lambda=2;  
numOfRunOvers=0;  
numOfNoRunOvers=0;  
repeats=1000000;  
for i=1:repeats  
    bin0=2*lambda;      %{initialize bin 0 size with 4%}  
    bin1=2*lambda;      %{initialize bin 1 size with 4%}  
    x1=poissrnd(lambda);%{generating a poisson variable named X1%}  
    x2=poissrnd(lambda);%{generating a poisson variable named X2%}  
    x3=poissrnd(lambda);%{generating a poisson variable named X3%}  
    disp(i);            %{display the counter i on the screen%}  
  
    %{//////////Algorithm Starts%}  
    if(x1<=2)  
        bin0=bin0-x1;  
    else  
        bin1=bin1-x1;  
    end  
    if(x2<=2)  
        bin0=bin0-x2;  
    else  
        bin1=bin1-x2;  
    end  
    if(x1<=2) && (x2<=2)  
        bin1=bin1-x3;  
    else  
        bin0=bin0-x3;
```

```

end
%{//////////Algorithm Ends%}

%{check if at least one the bins has been overflowed
%{and increase the appropriate counter by one%}
if((bin0>=0)&&(bin1>=0))
    numOfNoRunOvers=numOfNoRunOvers+1;
else
    numOfRunOvers=numOfRunOvers+1;
end
end
%{Display on screen : %}
percent1=(numOfRunOvers*100)/repeats;
percent2=(numOfNoRunOvers*100)/repeats;
disp('Overflow');
disp(percent1);
disp('Success');
disp(percent2);

```

Περίπτωση 5=Περίπτωση 6

```
%{  
the case that:  
P1 know x2 and  
P2 know x3  
P3 has no knowledge  
%}  
lambda=2;  
numOfRunOvers=0;  
numOfNoRunOvers=0;  
repeats=1000000;  
for i=1:repeats  
    bin0=2*lambda;      %{initialize bin 0 size with 4%}  
    bin1=2*lambda;      %{initialize bin 1 size with 4%}  
    x1=poissrnd(lambda);%{generating a poisson variable named X1%}  
    x2=poissrnd(lambda);%{generating a poisson variable named X2%}  
    x3=poissrnd(lambda);%{generating a poisson variable named X3%}  
    disp(i);            %{display the counter i on the screen%}  
  
    %{//////////Algorithm Starts%}  
    bin0=bin0-x3;  
    if(x2<=2 && x2+x3<=2*lambda)  
        bin0=bin0-x2;  
    else  
        bin1=bin1-x2;  
    end  
    if(x2>2 && x1+x2>2*lambda)  
        bin0=bin0-x1;  
    else  
        bin1=bin1-x1;  
    end  
    %{//////////Algorithm Ends%}
```

```

    %{check if at least one the bins has been overflowed
    %{and increase the appropriate counter by one%}
    if((bin0>=0)&&(bin1>=0))
        numOfNoRunOvers=numOfNoRunOvers+1;
    else
        numOfRunOvers=numOfRunOvers+1;
    end
end
%{Display on screen : %}
percent1=(numOfRunOvers*100)/repeats;
percent2=(numOfNoRunOvers*100)/repeats;
disp('Overflow');
disp(percent1);
disp('Success');
disp(percent2);

```

Περίπτωση 7

```
%{  
Player2 knows Player1 and  
Player1 knows Player2 and  
Player3 has no knowledge  
%}  
lambda=2;  
numOfRunOvers=0;  
numOfNoRunOvers=0;  
repeats=1000000;  
for i=1:repeats  
    bin0=2*lambda;      %{initialize bin 0 size with 4%}  
    bin1=2*lambda;      %{initialize bin 1 size with 4%}  
    x1=poissrnd(lambda);%{generating a poisson variable named X1%}  
    x2=poissrnd(lambda);%{generating a poisson variable named X2%}  
    x3=poissrnd(lambda);%{generating a poisson variable named X3%}  
    disp(i);            %{display the counter i on the screen%}  
  
    %{//////////Algorithm Starts%}  
    bin0=bin0-x3;  
    if((x1+x2)<=(2*lambda))  
        bin1=bin1-x2;  
        bin1=bin1-x1;  
    else  
        if(x1<=x2)  
            bin0=bin0-x1;  
            bin1=bin1-x2;  
        else  
            bin1=bin1-x1;  
            bin0=bin0-x2;  
        end  
    end  
end  
%{//////////Algorithm Ends%}
```

```

    %{check if at least one the bins has been overflowed
    %{and increase the appropriate counter by one%}
    if((bin0>=0)&&(bin1>=0))
        numOfNoRunOvers=numOfNoRunOvers+1;
    else
        numOfRunOvers=numOfRunOvers+1;
    end
end
%{Display on screen : %}
percent1=(numOfRunOvers*100)/repeats;
percent2=(numOfNoRunOvers*100)/repeats;
disp('Overflow');
disp(percent1);
disp('Success');
disp(percent2);

```

Παράρτημα Β

Στο παράρτημα αυτό δίνονται οι αναλύσεις των αλγορίθμων για κάθε περίπτωση, για εύρεση της πιθανότητας επιτυχίας σε MATLAB. Κάθε εξίσωση για την $P[\text{αποτυχίας}]$ την σπάζαμε σε μικρότερα μέρη. Κάθε μέρος A,B,C,.....,Z της ανάλυσης είναι το αντίστοιχο partA, partB, partC,, partZ στους πιο κάτω κώδικες για κάθε περίπτωση.

Περίπτωση 1

```
%{case of no communication %}
M=2;    REPEATS=100;
%{*****%}
%{part A  STARTS%}
r1=exp(-M)*(M^2)/factorial(2);
r1=r1^2;
r2=exp(-M)*(M^1)/factorial(1);
r4=3*(r1*r2);
r3=exp(-M)*(M^2)/factorial(2);
r3=r3^3;
partA=r4+r3;
disp('partA')
disp(partA);
%{part A  ENDS
%{*****%}
%{part B  STARTS%}
r1=0.0;r2=0.0;
for j=0 : 2
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
r1=r1^2;
for j=2*M+1 : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
r3=r1*r2;
```

```

partB=3*r3;
disp('partB')
disp(partB);
%{part B  ENDS
%{*****%}
%{part C  STARTS%}
r1=0.0;r2=0.0;
for j=0 : 2
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=3 : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
r2=r2^2;
r3=r1*r2;
partC=3*r3;
disp('partC');
disp(partC)
%{part C  ENDS
%{*****%}
%{part D  STARTS%}
r1=0.0;r2=0.0;r3=0.0;r4=0.0;r5=0.0;
for j=3 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
r1=r1^3;
partD=r1;
disp('partD');
disp(partD);
%{part D  ENDS*****%}
total=partA+partB+partC+partD;
sprintf('Overflow = %e',total)
sprintf('Success = %e',1.0-total)

```

Περίπτωση 2

```
%{
case 1:
Player2 know x3
Player 1 and 3 have no knowledge
%}
M=2;
REPEATS=100;
%{*****%}
%{part A STARTS%}
r1=0.0;
for j=2*M+1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
partA=r1*(2-r1);
disp('partA');
disp(partA);
%{part A ENDS%}
%{*****%}
%{part B STARTS%}
r2=0.0;r3=0.0;
for b=0 : REPEATS
    r1=exp(-M)*(M^b)/factorial(b);
    for j=2*M-b+1 : REPEATS
        if (j>=0)
            r2=r2+(exp(-M)*(M^j)/factorial(j));
        end
    end
    r2=r2^2;
    r3=r3+(r1*r2);
    r2=0.0;
end
partB=r3;
```

```

disp('partB');
disp(partB);
%{part B ENDS%}
%{*****%}
%{part C STARTS%}
r1=0.0;r2=0.0;r3=0.0;r4=0.0;r5=0.0;
for b=0 : REPEATS
    r1=exp(-M)*(M^b)/factorial(b);
    for j=2*M-b+1 : REPEATS
        if(j>=0)
            r2=r2+(exp(-M)*(M^j)/factorial(j));
        end
    end
    for j=2*M-b+1 : 2*M
        if(j>=0)
            r3=r3+(exp(-M)*(M^j)/factorial(j));
        end
    end
    for j=2*M+1 : REPEATS
        r4=r4+(exp(-M)*(M^j)/factorial(j));
    end
    r5=r5+(r1*r4*(r2+r3));
    r2=0.0;r3=0.0;r4=0.0;
end
partC=r5;
disp('partC');
disp(partC);
%{part C ENDS%}
%{*****%}
%{ printing the results on the screen%}
total=partA+partB-partC;
sprintf('Overflow = %e',total)
sprintf('Success = %e',1.0-total)

```

Περίπτωση 3

```
%{  
case:  
Player1 and Player2 know x3  
Player 3 has no knowledge  
%}  
M=2;  
REPEATS=100;  
%{*****%}  
%{part A  STARTS%}  
r1=0.0;  
for j=2*M+1 : REPEATS  
    r1=r1+(exp(-M)*(M^j)/factorial(j));  
end  
partA=r1;  
disp('partA');  
disp(partA)  
%{part A  ENDS  
%{*****%}  
%{part B  STARTS%}  
r2=0.0;r3=0.0;  
for c=0 : 3  
    r1=exp(-M)*(M^c)/factorial(c);  
    for a=1 : 2*M-c  
        for b=2*M-c-a+1 : 2*M-c  
            r2=r2+((exp(-M)*(M^a)/factorial(a))*(exp(-M)*(M^b)/factorial(b)));  
        end  
    end  
    r3=r3+(r1*r2);  
    r2=0.0;  
end  
partB=r3;
```

```

disp('partB');
disp(partB);
%{part B  ENDS
%{*****%}
%{part C and D  STARTS%}
r1=0.0;r2=0.0;
for j=2*M+1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for a=0 : 2*M
    for b=0 : 2*M-a
        r2=r2+((exp(-M)*(M^a)/factorial(a))*(exp(-M)*(M^b)/factorial(b)));
    end
end
r3=r1*r2;
partC=r3;
partD=r3;
disp('partC');
disp(partC);
disp('partD');
disp(partD);
%{part C and D  ENDS
%{*****%}
%{part E  STARTS%}
%{c=0
r1=0.0;
for j=2*M+1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
r1=r1^2;
r1=r1*(exp(-M)*(M^0)/factorial(0));
partE1=r1;
%{ c =1

```

```

r1=0.0;
for j=2*M : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
r1=r1^2;
r1=r1*(exp(-M)*(M^1)/factorial(1));
partE2=r1;
%{ c =2
r1=0.0;
for j=2*M-1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
r1=r1^2;
r1=r1*(exp(-M)*(M^2)/factorial(2));
partE3=r1;
%{c =3
r1=0.0;r2=0.0;r3=0.0;
for j=2 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=3 : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
r1=r1*r2;

for j=3 : REPEATS
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
r3=r3*(exp(-M)*(M^2)/factorial(2));
r4=r1+r3;
r4=r4*(exp(-M)*(M^3)/factorial(3));
partE4=r4;
%{ c=4

```

```

r1=0.0;r2=0.0;r3=0.0;r4=0.0;
for j=1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=4 : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
r1=r1*r2;

for j=4 : REPEATS
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
for j=1 : 3
    r4=r4+(exp(-M)*(M^j)/factorial(j));
end
r3=r3*r4;
r6=(exp(-M)*(M^2)/factorial(2))*(exp(-M)*(M^3)/factorial(3));
r7=(exp(-M)*(M^3)/factorial(3))*(exp(-M)*(M^3)/factorial(3));
r8=(exp(-M)*(M^3)/factorial(3))*(exp(-M)*(M^2)/factorial(2));
r5=r1+r3+r6+r7+r8;
r5=r5*(exp(-M)*(M^4)/factorial(4));
partE5=r5;
partE=partE1+partE2+partE3+partE4+partE5;
disp('partE');
disp(partE);
%{part E ENDS
%{*****%}
%{printing the results on the screen%}
total=partA+partB+partC+partD+partE;
sprintf('Overflow  = %e',total)
sprintf('Success   = %e',1.0-total)

```

Περίπτωση 4

```
%{
case that :
P3 knows X1 and X2
P2 and P1 have no knowledge
%}
M=2;
REPEATS=50;
THRESHOLD=2;
%{*****%}
%{part A  STARTS%}
r1=0.0;r2=0.0;r3=0.0;r4=0.0;r5=0.0;r6=0.0;r7=0.0;
for a=1 : THRESHOLD
    for b=2*M-a+1 : 2*M
        r1=r1+((exp(-M)*(M^a)/factorial(a))*(exp(-M)*(M^b)/factorial(b)));
    end
end
for j=0 : THRESHOLD
    r6=r6+(exp(-M)*(M^j)/factorial(j));
end
for j=2*M+1 : REPEATS
    r7=r7+(exp(-M)*(M^j)/factorial(j));
end
r1=r1+r6*r7;

for j=THRESHOLD+1 : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
r2=r2*r1;

for j=2*M+1 : REPEATS
    r3=r3+(exp(-M)*(M^j)/factorial(j));
```

```

end
r3=r3*r1;

for j=0 : THRESHOLD
    r4=r4+(exp(-M)*(M^j)/factorial(j));
end
for j=2*M+1 : REPEATS
    r5=r5+(exp(-M)*(M^j)/factorial(j));
end
r4=r4*r5;
partA=r4+r2-r3;
disp('partA')
disp(partA);
%{part A  ENDS%}
%{*****%}
%{part B  STARTS%}
partB=partA;
disp('partB')
disp(partB);
%{part B  ENDS%}
%{*****%}
%{part C  STARTS%}
r1=0.0;r2=0.0;r3=0.0;r4=0.0;r5=0.0;r6=0.0;r7=0.0;
for j=0 :THRESHOLD
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=0 :THRESHOLD
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
for j=2*M+1 :REPEATS
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
partC=r1*r2*r3;

```

```

disp('partC')
disp(partC);
%{part C  ENDS%}
%{*****%}
%{part D  STARTS%}
r1=0.0;
for j=THRESHOLD+1 :REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
r1=r1^2;
partD=r1;
disp('partD')
disp(partD);
%{part D  ENDS%}
%{*****%}
total=partA+partB+partC+partD;
sprintf('Overflow = %e',total)
sprintf('Success  = %e',1.0-total)

```

Περίπτωση5=Περίπτωση6

```
%{
case that
P2 knows P3 and P1 knows P2
and P3 has no knowledge
%}
M=2;
REPEATS=50;
%{*****%}
r1=0.0;
for j=2*M+1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
partA=r1;
disp('partA')
disp(partA);
%{*****%}
r1=0.0;r2=0.0;r3=0.0;r6=0.0;r7=0.0;
for j=3 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
r1=r1*(exp(-M)*(M^2)/factorial(2));
r1=r1*(exp(-M)*(M^3)/factorial(3));
for j=2*M : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
for j=1 : 2
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
r2=r2*r3;
r4=(exp(-M)*(M^3)/factorial(3))*(exp(-M)*(M^2)/factorial(2));
r5=(r2+r4)*(exp(-M)*(M^4)/factorial(4));
```

```

partB=r1+r5;
disp('partB');
disp(partB);
%{*****}%}
r1=0.0;r2=0.0;r3=0.0;r4=0.0;r5=0.0;r6=0.0;r7=0.0;
for j=2*M+1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=2*M+1 : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
for j=3 : 2*M
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
partC=(r1+r2*r3)*(exp(-M)*(M^0)/factorial(0));

r1=0.0;r2=0.0;r3=0.0;r4=0.0;r5=0.0;r6=0.0;r7=0.0;
for j=2*M+1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=2*M : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
for j=3 : 2*M
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
partC=partC+((r1+r2*r3)*(exp(-M)*(M^1)/factorial(1)));

r1=0.0;r2=0.0;r3=0.0;r4=0.0;r5=0.0;r6=0.0;r7=0.0;
for j=2*M+1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=3 : REPEATS

```

```

    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
for j=3 : 2*M
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
partC=partC+((r1+r2*r3)*(exp(-M)*(M^2)/factorial(2)));

r1=0.0;r2=0.0;r3=0.0;r4=0.0;r5=0.0;r6=0.0;r7=0.0;
for j=2*M+1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=2 : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
for j=3 : 2*M
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
partC=partC+((r1+r2*r3)*(exp(-M)*(M^3)/factorial(3)));

r1=0.0;r2=0.0;r3=0.0;r4=0.0;r5=0.0;r6=0.0;r7=0.0;
for j=2*M+1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=2 : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
for j=3 : 2*M
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
r4=(exp(-M)*(M^1)/factorial(1))*(exp(-M)*(M^4)/factorial(4));
partC=partC+((r1+r2*r3+r4)*(exp(-M)*(M^4)/factorial(4)));
disp('partC');
disp(partC);

```

```

%{*****}
r1=0.0;r2=0.0;r3=0.0;r4=0.0;r5=0.0;r6=0.0;r7=0.0;
for j=2*M+1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=0 : 2
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
partD=r1*r2*(exp(-M)*(M^0)/factorial(0));
partD=partD+(r1*r2*(exp(-M)*(M^1)/factorial(1)));
partD=partD+(r1*r2*(exp(-M)*(M^2)/factorial(2)));

r1=0.0;r2=0.0;r3=0.0;r4=0.0;r5=0.0;r6=0.0;r7=0.0;
for j=2*M+1 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=0 : 1
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
partD=partD+(r1*r2*(exp(-M)*(M^3)/factorial(3)));
partD=partD+(r1*(exp(-M)*(M^0)/factorial(0))*(exp(-M)*(M^4)/factorial(4)));
disp('partD');
disp(partD);
%{*****}
%{ printing the results on the screen%}
total=partA+partB+partC+partD;
sprintf('Overflow = %e',total)
sprintf('Success = %e',1.0-total)

```

Περίπτωση 7

```
%{
Player1 knows player2
player2 knows player1
%}
M=2;
REPEATS=100;
%{*****
%}
t1=(exp(-M)*(M^3)/factorial(3))*(exp(-M)*(M^4)/factorial(4));
s1=t1*(exp(-M)*(M^2)/factorial(2));

t2=t1+(exp(-M)*(M^2)/factorial(2))*(exp(-M)*(M^3)/factorial(3));
t2=t2+(exp(-M)*(M^2)/factorial(2))*(exp(-M)*(M^4)/factorial(4));
s2=t2*(exp(-M)*(M^3)/factorial(3));

t3=t2+(exp(-M)*(M^1)/factorial(1))*(exp(-M)*(M^4)/factorial(4));
s3=t3*(exp(-M)*(M^4)/factorial(4));

partA=s1+s2+s3;
disp('partA');
disp(partA);
%{*****
%}
t1=(exp(-M)*(M^4)/factorial(4))*(exp(-M)*(M^4)/factorial(4));
s1=t1*(exp(-M)*(M^1)/factorial(1));

t2=t1+(exp(-M)*(M^3)/factorial(3))*(exp(-M)*(M^3)/factorial(3));
t2=t2+(exp(-M)*(M^4)/factorial(4))*(exp(-M)*(M^3)/factorial(3));
s2=t2*(exp(-M)*(M^2)/factorial(2));

t3=t2+(exp(-M)*(M^3)/factorial(3))*(exp(-M)*(M^2)/factorial(2));
```

```

t3=t3+(exp(-M)*(M^4)/factorial(4))*(exp(-M)*(M^2)/factorial(2));
s3=t3*(exp(-M)*(M^3)/factorial(3));

t4=t3+(exp(-M)*(M^4)/factorial(4))*(exp(-M)*(M^1)/factorial(1));
s4=t4*(exp(-M)*(M^4)/factorial(4));
partB=s1+s2+s3+s4;
disp('partB');
disp(partB);
%{*****}
%}
r1=0.0;r2=0.0;
for j=5 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for a=0 : 4
    for b=0 : 4-a
        r2=r2+((exp(-M)*(M^a)/factorial(a))*(exp(-M)*(M^b)/factorial(b)));
    end
end

partC=r1*r2;
disp('partC');
disp(partC);
%{*****}
%}
r1=0.0;
for j=5 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
r2=(exp(-M)*(M^4)/factorial(4))*(exp(-M)*(M^1)/factorial(1));
r2=r2+((exp(-M)*(M^4)/factorial(4))*(exp(-M)*(M^2)/factorial(2)) );
r2=r2+((exp(-M)*(M^4)/factorial(4))*(exp(-M)*(M^3)/factorial(3)) );
r2=r2+((exp(-M)*(M^4)/factorial(4))*(exp(-M)*(M^4)/factorial(4)) );

```

```

r2=r2+((exp(-M)*(M^3)/factorial(3))*(exp(-M)*(M^2)/factorial(2)) );
r2=r2+((exp(-M)*(M^3)/factorial(3))*(exp(-M)*(M^3)/factorial(3)) );
r2=r2+((exp(-M)*(M^3)/factorial(3))*(exp(-M)*(M^4)/factorial(4)) );
r2=r2+((exp(-M)*(M^2)/factorial(2))*(exp(-M)*(M^3)/factorial(3)) );
r2=r2+((exp(-M)*(M^2)/factorial(2))*(exp(-M)*(M^4)/factorial(4)) );
r2=r2+((exp(-M)*(M^1)/factorial(1))*(exp(-M)*(M^4)/factorial(4)) );
partD=r1*r2;
disp('partD');
disp(partD);
%{*****}
%}
partE=partC;
partF=partD;
disp('partE');
disp(partE);
disp('partF');
disp(partF);
%{*****}
%}
%partGH=partG+partH
r1=0.0;r2=0.0;r3=0.0;
for j=0 : 4
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=5 : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
for j=5 : REPEATS
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
partGH=r1*r2*r3;
disp('partGH');
disp(partGH);

```

```

%{*****}
%}
%partIJKL=partI+partJ+partK+partL
r1=0.0;r2=0.0;r3=0.0;
for j=5 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=0 : 4
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
for j=0 : 4
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
partIJKL=r1*r2*r3;
disp('partIJKL');
disp(partIJKL);
%{*****}
%}
r1=0.0;r2=0.0;r3=0.0;
for j=5 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
for j=5 : REPEATS
    r2=r2+(exp(-M)*(M^j)/factorial(j));
end
for j=0 : 4
    r3=r3+(exp(-M)*(M^j)/factorial(j));
end
partM=r1*r2*r3;
disp('partM');
disp(partM);
%{*****}
%}

```

```

partN=partM;
disp('partN');
disp(partN);
%{*****
%}
%partOP=partO+partP
r1=0.0;r2=0.0;r3=0.0;
for j=5 : REPEATS
    r1=r1+(exp(-M)*(M^j)/factorial(j));
end
partOP=r1^3;
disp('partOP');
disp(partOP);
%{*****
%}
%{ printing the results on the screen%}
total=partA+partB+partC+partD+partE+partF+partGH+partIJKL;
total=total+(partM+partN+partOP);
sprintf('Overflow = %e',total)
sprintf('Success = %e',1.0-total)

```

Περίπτωση 8

Δεν χρειάστηκε ανάλυση καθώς δείχθηκε με άλλο τρόπο(βλέπε Κεφ9.4)

Παράρτημα Γ

Στο παράρτημα αυτό θα δοθεί ο αλγόριθμος «ΕΠΑΛΗΘΕΥΤΗΣ ΒΕΛΤΙΣΤΟΤΗΤΑΣ» στην γλώσσα προγραμματισμού C, για τις περιπτώσεις όπου βρήκαμε βελτιστότητα , δηλαδή ο αλγόριθμος έκανε όλες τις απαραίτητες συγκρίσεις και βρήκε τον αλγόριθμο μας Βέλτιστο. Οι είσοδοι στον αλγόριθμο αυτό δίνονται από αρχεία.

Περίπτωση 1

```
////////////////////////////////////
//case of no communication
////////////////////////////////////
//libraries
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
//constants
//global variables
int totalTimes=0;
//function prototypes
int compareAlgorithms();
int factorial(int num);
double findPoisson(int num);
/*****/
//function that returns the poisson probability
double findPoisson(int num)
{
    //this function returns the poisson possibility for P[X=num]
    double result;
    result=(double)exp(-
2)*(double)pow(2,num)/(double)factorial(num);
    return result;
}
/*****/
//function that returns the factorial of number num
int factorial(int num)
{
    int i;
    int fact=1;
    if(num==0)
        fact=1;
    else
    {
        for (i=1;i<=num;i++)
            fact=fact*i;
    }
    return fact;
}
/*****/
```

```

int compareAlgorithms()
{
    //this function compares "My algorithm" with every other
    algorithm
    //returns YES (1) if "my algorithm" wind every other algorithm OR
    NO(0)
    //this function reads all the possible algorithms for every
    player from a file

    int i,j,k,result;
    int table1[5];
    int table2[5];
    int table3[5];
    int myAlgorithmFailed;
    int otherAlgorithmFailed;
    int x1,x2,x3,bin0,bin1;
    double myWins;
    double otherWins;
    FILE *fp1;
    FILE *fp2;
    FILE *fp3;
    fp1=fopen("p1Algorithms.txt","r");
    int counter;
    if(fp1==NULL)
    {
        printf("Problem with file p1Algorithms.txt\n");
        exit(-1);
    }
    for(i=0;i<32;i++)
    {
        //read an algorithm for the Player1
        fscanf(fp1," %d %d %d %d
%d",&table1[0],&table1[1],&table1[2],&table1[3],&table1[4]
);
        fscanf(fp1,"\n");

        fp2=fopen("p2Algorithms.txt","r");
        if(fp2==NULL)
        {
            printf("Problem with file p2Algorithms.txt\n");
            exit(-1);
        }
        for(j=0;j<32;j++)
        {
            //read an algorithm for the Player2
            fscanf(fp2," %d %d %d %d
%d",&table2[0],&table2[1],&table2[2],&table2[3],&table2[4]);
            fscanf(fp2,"\n");

            fp3=fopen("p3Algorithms.txt","r");
            if(fp3==NULL)
            {
                printf("Problem with file
p3Algorithms.txt\n");
                exit(-1);
            }

            for(k=0;k<32;k++)
            {

```

```

//read an algorithm for the Player3
fscanf(fp3," %d %d %d %d
%d",&table3[0],&table3[1],&table3[2],&table3[3]
],&table3[4]);
fscanf(fp3,"\n");

//now we have a distributed Algorithm and we
//can compare it with "My Algorithm"
//for all possible values of X1,X2,X3
myWins=0.0;
otherWins=0.0;
for (x1=0;x1<=4;x1++)
{
    for (x2=0;x2<=4;x2++)
    {
        for (x3=0;x3<=4;x3++)
        {
            //"My algorithm"
            bin0=4;
            bin1=4;
            //P1
            if(x1<=2)
                bin0=bin0-x1;
            else
                bin1=bin1-x1;
            //P2
            if(x2<=2)
                bin0=bin0-x2;
            else
                bin1=bin1-x2;
            //P3
            if(x3<=2)
                bin0=bin0-x3;
            else
                bin1=bin1-x3;
            if(bin0>=0 && bin1>=0)
                myAlgorithmFailed=0;
            else
                myAlgorithmFailed=1;
            //"Other Algorithm"
            bin0=4;
            bin1=4;
            //P1
            if (table1[x1]==0)
                bin0=bin0-x1;
            else
                bin1=bin1-x1;
            //P2
            if (table2[x2]==0)
                bin0=bin0-x2;
            else
                bin1=bin1-x2;
            //P3
            if (table3[x3]==0)
                bin0=bin0-x3;
            else
                bin1=bin1-x3;

```

```

if(bin0>=0 && bin1>=0)

    otherAlgorithmFailed=0;
else

    otherAlgorithmFailed=1;

//compare the two Algorithms
//if both algorithms did not
//failed or both algorithms
// failed it does not matter
//when otherAlgorithm is
//better
if(otherAlgorithmFailed==0
&& myAlgorithmFailed==1)
{
    //printf("When other
    wins \n");

    //increase variable
    //OtherWins

    otherWins=otherW
ins+(findPoisson(x1)*f
indPoisson(x2)*findPoi
sson(x3));
    totalTimes++;
}
//when myAlgorithm is better
if(otherAlgorithmFailed==1
&& myAlgorithmFailed==0)
{
    //printf("When
    myAlgorithm wins\n");

    myWins=myWins+(findPoi
sson(x1)*findPoisson(x
2)*findPoisson(x3));
    totalTimes++;
}
//when both algorithms have
//the same result
if( (otherAlgorithmFailed==0
&&
myAlgorithmFailed==0)|| (othe
rAlgorithmFailed==1 &&
myAlgorithmFailed==1) )
{
    //printf("Same Result
    \n");
    totalTimes++;
}
} //end x3 loop
} //end x2 loop
} //end x3 loop

```

```

        //after all the x1,x2,x3 possible values
        //find which one is better
        if(myWins>=otherWins)
        {
            result=1;
            if(myWins==otherWins)
            {
                printf("Same Algorithm as ''My
                Algorithm''\n");
                printf("=====\n");
            }
        }
        else
        {
            result=0;
            return result;
        }
    }
    fclose(fp3);
}
fclose(fp2);
}

fclose(fp1);
return result;
}
/*****
int main()
{
    int isBest=compareAlgorithms();
    if (isBest==1)
        printf("is My Algorithm the Best ? ==>YES!!!\n");
    else
        printf("is My Algorithm the Best ? ==>NO!!!\n");

    printf("Total repeats in this case= %d\n",totalTimes);

    return;
}

```

Περίπτωση 4

```
////////////////////////////////////
//case that Player 3 knows X1 and X2
////////////////////////////////////
//libraries
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
//constants
//global variables
int totalTimes=0;
//function prototypes
int compareAlgorithms();
int factorial(int num);
double findPoisson(int num);
/*****
*****/
double findPoisson(int num)
{
    //this function returns the poisson possibility for P[X=num]
    double result;
    result=(double)exp(-
2)*(double)pow(2,num)/(double)factorial(num);
    return result;
}
/*****
*****/
int factorial(int num)
{
    int i;
    int fact=1;
    if(num==0)
        fact=1;
    else
    {
        for (i=1;i<=num;i++)
            fact=fact*i;
    }
    return fact;
}
/*****
*****/
int compareAlgorithms()
{
    //this function compares "My algorithm" with every other
algorithm
    //returns YES (1) if "my algorithm" wind every other algorithm OR
NO(0)
    //this function reads all the possible algorithms for every
player from a file
    int i,j,k,result;
    int table1[5];
    int table2[5];
    int table3[5];
    int myAlgorithmFailed;
    int otherAlgorithmFailed;
```

```

int x1,x2,x3,bin0,bin1;
double myWins;
double otherWins;
FILE *fp1;
FILE *fp2;
FILE *fp3;
fp1=fopen("p1Algorithms.txt","r");
int counter;
if(fp1==NULL)
{
    printf("Problem with file p1Algorithms.txt\n");
    exit(-1);
}
for(i=0;i<32;i++)
{
    //read an algorithm for Player1
    fscanf(fp1," %d %d %d %d %d",
    &table1[0],&table1[1],&table1[2],&table1[3],&table1[4]
    );
    fscanf(fp1,"\n");

    fp2=fopen("p2Algorithms.txt","r");
    if(fp2==NULL)
    {
        printf("Problem with file p2Algorithms.txt\n");
        exit(-1);
    }
    for(j=0;j<32;j++)
    {
        //read an algorithm for player2
        fscanf(fp2," %d %d %d %d %d",
        &table2[0],&table2[1],&table2[2],&table2[3],&table2[4]);
        fscanf(fp2,"\n");

        //p3 plays always optimally
        //now we have a distributed Algorithm and we can
        compare it with "My Algorithm"
        //for all possible values of X1,X2,X3
        myWins=0.0;
        otherWins=0.0;
        for (x1=0;x1<=4;x1++)
        {
            for (x2=0;x2<=4;x2++)
            {
                for (x3=0;x3<=4;x3++)
                {
                    //"My algorithm"
                    bin0=4;
                    bin1=4;
                    //P1
                    if(x1<=2)
                        bin0=bin0-x1;
                    else
                        bin1=bin1-x1;
                    //P2
                    if(x2<=2)
                        bin0=bin0-x2;
                    else

```

```

        bin1=bin1-x2;
//P3
if(x1<=2 && x2<=2)
    bin1=bin1-x3;
else
    bin0=bin0-x3;

if(bin0>=0 && bin1>=0)
    myAlgorithmFailed=0;
else
    myAlgorithmFailed=1;
// "Other Algorithm"
bin0=4;
bin1=4;
//P2
if (table2[x2]==0)
    bin0=bin0-x2;
else
    bin1=bin1-x2;
//P1
if (table1[x1]==0)
    bin0=bin0-x1;
else
    bin1=bin1-x1;
//P3 plays optimally
if(table2[x2]==0 && table1[x1]==0)
    bin1=bin1-x3;
if(table2[x2]==1 && table1[x1]==1)
    bin0=bin0-x3;
if(table2[x2]==0 && table1[x1]==1)
{
    if(x1<=x2)
        bin1=bin1-x3;
    else
        bin0=bin0-x3;
}
if(table2[x2]==1 && table1[x1]==0)
{
    if(x1<=x2)
        bin0=bin0-x3;
    else
        bin1=bin1-x3;
}
if(bin0>=0 && bin1>=0)
    otherAlgorithmFailed=0;
else
    otherAlgorithmFailed=1;

//compare the two Algorithms
//if both algorithms did not
failed or both algorithms failed it
does not matter
//when other algorithm is better
if(otherAlgorithmFailed==0 &&
myAlgorithmFailed==1)
{
    //printf("When other
wins\n");

```

```

//increase variable
//otherWins
        otherWins=otherWins+(findPoisson(x1)*findPoisson(x2)*findPoisson(x3));
totalTimes++;
}
//when myAlgorithm is better
if(otherAlgorithmFailed==1
&& myAlgorithmFailed==0)
{
        //printf("When
        //myAlgorithm wins
        \n");

        //increase variable

        //myWins

        myWins=myWins+
        (findPoisson(x1)*
        findPoisson(x2)
        *findPoisson(x3));
        totalTimes++;
}
//when both algorithms give
//the same result
if( (otherAlgorithmFailed==0
&&
myAlgorithmFailed==0)|| (otherAlgorithmFailed==1 &&
myAlgorithmFailed==1) )
{
        //printf("Same
        Result!!! \n");
        totalTimes++;
}
} //end loop x3
} //end loop x2
} //end loop x1
if(myWins>=otherWins)
{
        result=1;
        if(myWins==otherWins)
        {
                printf("Same Algorithm as 'My
                Algorithm'\n");
                printf("=====\n");
        }
}
else
{
        result=0;
        return result;
}

}

```

```

        fclose(fp2);
    }

    fclose(fp1);
    return result;
}
/*****
int main()
{

int isBest=compareAlgorithms();
if (isBest==1)
    printf("is My Algorithm the Best ? ====>YES!!!\n");
else
    printf("is My Algorithm the Best ? ====>NO!!!\n");

printf("Total repeats in this case= %d\n",totalTimes);

return;
}

```