

1^η Εργασία Τεχνητής Νοημοσύνης



24 ΝΟΕΜΒΡΙΟΥ

Ομαδική Εργασία 3 ατόμων

Έχει συνταχθεί από: Δικονιμάκη Χρυσούλα

3170039

Κωνσταντόπουλος Κωνσταντίνος

3170086

Γεράσιμος Λαπάκης

3170089



Υλοποίηση διεξαγωγής προγράμματος γυμνασίου με γενετικό αλγόριθμο

Περιγραφή αλγορίθμου:

Η ιδέα του γενετικού αλγορίθμου είναι η τυχαία παραγωγή η διαφορετικών τυχαίων προγραμμάτων(χρωμοσωμάτων) στην αρχή της εκτέλεσης και στην πορεία του χρόνου θα γίνονται διασταυρώσεις μεταξύ τυχαία επιλεγμένων χρωμοσωμάτων. Μετά την παραγωγή των τυχαίων χρωμοσωμάτων στην αρχή, το καθένα βαθμολογείται με βάση την ευρετική μας συνάρτηση (αναλύεται αργότερα).

Κρατάμε στη μνήμη ένα array στην κάθε επανάληψη (μια επανάληψη θεωρείται η διαδικασία συγκεκριμένου αριθμού διασταυρώσεων) και όσο είναι το score τόσο θα'ναι και τα αντίγραφα του αντίστοιχου χρωμοσώματος μέσα στο array. Η επιλογή καθ' ενός από τους δύο γονείς (γονέας θεωρείται ένα χρωμόσωμα) σε μία διασταύρωση γίνεται τυχαία μέσα από αυτό το array, επομένως όσα μεγαλύτερο score έχει, τόσο περισσότερα αντίγραφα θα έχει, άρα και μεγαλύτερη πιθανότητα να επιλεγθεί να γίνει γονέας.

Η διαδικασία αυτή συνεχίζεται μέχρις ότου να βρεθεί κάποιο χρωμόσωμα που να ξεπερνάει το κατώτατο επιθυμητό score ή μέχρι να εξαντληθεί ο αριθμός επαναλήψεων, δηλαδή διασταυρώσεων, που έχουμε εμείς ορίσει στην αρχή του προγράμματος, δηλαδή στην main.

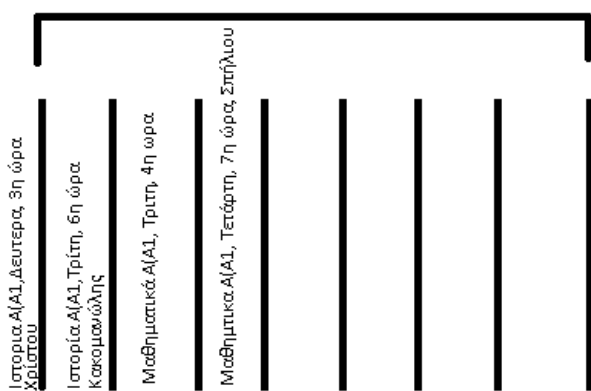
Πως έχουμε αναπαραστήσει τα δεδομένα μας:

Χρωμόσωμα:

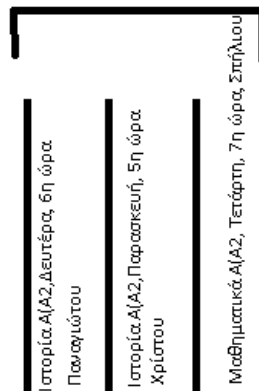
Το χρωμόσωμά, όπως έχουμε προαναφέρει, είναι μία σειρά από γονίδια. Το κάθε γονίδιο είναι ένα αντίγραφο μίας απαιτούμενης ώρας προς διδασκαλία για κάθε μάθημα. Για παράδειγμα, το μάθημα Ιστορίας της 1^{ης} Γυμνασίου πρέπει να διδαχθεί στο κάθε τμήμα 2 ώρες/βδομάδα. Έστω τώρα πως η κάθε τάξη Γυμνασίου έχει 3 τμήματα μαθητών, αυτό σημαίνει πως, αθροιστικά θα πρέπει το μάθημα αυτό να διδαχθεί $3 \times 2 = 6$ φορές. Στο χρωμόσωμα μας λοιπόν θα περιέχονται 6 γονίδια τα οποία αντιστοιχούν στις 6 απαιτούμενες ώρες διδασκαλίας της Ιστορίας 1^{ης} Γυμνασίου.

Σχηματικά:

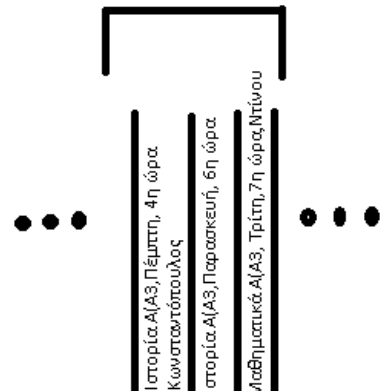
A1 ΤΑΞΗ



A2 ΤΑΞΗ



A3 ΤΑΞΗ



Γενικότερα η σχηματική απεικόνιση είναι η ακόλουθη:

Κάθε κελί αντιπροσωπεύει: (Τάξη Τμήμα, Μάθημα Τάξης, Ώρα Μαθήματος)
και περιέχει: (Ημέρα Ώρα Διεξαγωγής, Καθηγητής)

(Τρίτη 5η ώρα, Βλάχος)	(Α1, Μαθηματικά, 1η ώρα)
(Παρασκευή 2η ώρα, Δημάκης)	(Α1, Μαθηματικά, 2η ώρα)
(Πέμπτη 1η ώρα, Βλάχος)	(Α1, Μαθηματικά, 3η ώρα)
(Δευτέρα 1η ώρα, Παπαδημητρίου)	(Α1, Γλώσσα, 1η ώρα)
(Δευτέρα 5η ώρα, Νικολάου)	(Α1, Γλώσσα, 2η ώρα)
• • •	<div> <div>←</div> <div> <p>Όλα τα υπόλοιπα μαθήματα του Α1 (κάθε μάθημα εμφανίζεται τόσες φορές όσες και οι απαιτούμενες ώρες διδασκαλίας του)</p> </div> </div>
(Τετάρτη 1η ώρα, Δημάκης)	(Α2, Μαθηματικά, 1η ώρα)
• • •	

PrintSchedule:

Είναι ένας πίνακας 3-διάστατος(της μορφής [τμήμα][ώρα][μέρα]) ο οποίος παρουσιάζει τα δεδομένα του χρωμοσώματος σε μια αναγνώσιμη μορφή, ένα κανονικό δηλαδή πρόγραμμα. Κάθε χρωμόσωμα έχει τον δικό του πίνακα και χρησιμοποιείται σε συνδυασμό με την ευρετική.

Είσοδος:

Lessons.txt: Η δομή του αρχείου είναι η εξής:

Μοναδικό id μαθήματος-Όνομα μαθήματος-Τάξη-Ωρες διδασκαλίας

Teachers.txt: Η δομή του αρχείου είναι η εξής:

Id καθηγητή-όνομα καθηγητή-[id μαθημάτων που διδάσκει]-ημερήσιο-εβδομαδιαίο όριο

Ποιες κλάσεις χρησιμοποιήσαμε:

Reader: Η κλάση Reader είναι υπεύθυνη για την ανάγνωση των txt που περιέχουν πληροφορίες σχετικά με τους καθηγητές και τα μαθήματα. Έχει υλοποιηθεί με την χρήση της βιβλιοθήκης BufferedReader.

Main: Η κλάση main με τη βοήθεια της Reader, Genetic, Chromosome και Writer είναι υπεύθυνη για την διεκπεραίωση και την σύνδεση όλων των δεδομένων για την εκτέλεση του αλγορίθμου.

Genetic: Η κλάση Genetic χρησιμοποιείται για την διασταύρωση χρωμοσωμάτων και την παραγωγή νέων γενεών(reproduce) και την αντικατάσταση της υπάρχουσας γενιάς με μία πιθανότητα k το κάθε γονίδιο του παραγόμενου χρωμοσώματος να γίνει μετάλλαξη(mutate).

Chromosome: Η κλάση Chromosome περιέχει την ευρετική μας (calculateFitness) η οποία είναι ουσιαστικά χωρισμένη σε 4 περιορισμούς και 3 βελτιστοποιήσεις:
Περιορισμοί:

- 1) {Basic 1}: ο περιορισμός στον οποίο δεν επιτρέπουμε στο παραγόμενο πρόγραμμα να γίνεται κάθε χρονική στιγμή μάθημα από τον ίδιο καθηγητή(π.χ. ο καθηγητής X να μην διδάσκει την ώρα y και μέρα z σε 2 τμήματα ταυτόχρονα)

-
- 2) {Basic 2}: ο περιορισμός να μην ξεπερνάει ο κάθε καθηγητής το μέγιστο αριθμό ωρών ανά ημέρα και εβδομάδα που μπορεί να διδάξει.

Λιγότερο αυστηροί περιορισμοί:

- 1) {sum1}: ο περιορισμός του να μην υπάρχουν κενά(ως κενό θεωρούμε την κάθε ώρα στην οποία μεταξύ δύο διαδοχικών ωρών διδασκαλίας μεσολαβεί μία διδακτική ώρα στην οποία δεν διδάσκεται κάποιο μάθημα)
- 2){sum2}: ο περιορισμός στον οποίο δεν επιτρέπουμε έναν καθηγητή να διδάσκει πάνω από δύο συνεχόμενες ώρες

Βελτιστοποιήσεις (η διαδικασία με την οποία ελέγχονται αναλύονται αργότερα):

- 1) {sum3}: Ο ημερήσιος αριθμός ωρών διδασκαλίας κάθε τμήματος να είναι κατά το δυνατόν ομοιόμορφος όλες τις ημέρες
- 2) {sum4}: Οι ώρες διδασκαλίας κάθε μαθήματος σε ένα τμήμα να είναι κατά το δυνατόν ομοιόμορφα κατανεμημένες σε όλες τις ημέρες της εβδομάδας
- 3) {sum5}: Ο αριθμός ωρών διδασκαλίας ανά εβδομάδα να είναι κατά το δυνατόν ομοιόμορφος για όλους τους καθηγητές

Gene: Ένα χρωμόσωμα αποτελείται από μία σειρά γονιδίων(Gene) τα οποία έχουν αποθηκευμένα μέσα τους τα απαραίτητα δεδομένα, όπως ποια μέρα και ώρα διδάσκεται καθώς και σε ποιο τμήμα και τάξη, καθώς και το ποιος καθηγητής το διδάσκει.

Lesson: Είναι η κλάση στην οποία κρατάμε αποθηκευμένα τα δεδομένα του κάθε μαθήματος κατά την ανάγνωση του lessons.txt.

Teacher: Είναι η κλάση στην οποία κρατάμε αποθηκευμένα τα δεδομένα του κάθε Καθηγητή κατά την ανάγνωση του teachers.txt.

Writer: Είναι υπεύθυνος για την εγγραφή του τελικού αποτελέσματος σε ένα txt αρχείο

Πως λειτουργεί η ευρετική:

Περιορισμοί:

Τα παρακάτω υπολογισμοί συνεισφέρουν αρνητικά στο συνολικό κόστος. Οι 2 πρώτοι είναι αυστηροί περιορισμοί που δε πρέπει να παραβιάζονται για να υπάρξει το επιθυμητό πρόγραμμα. Οι δεύτεροι είναι καλό να υπάρχουν όσο το δυνατόν περισσότερο.

Basic1: Η 1^η βασική και προφανή συνθήκη που πρέπει να ελέγχεται είναι να μη συμπίπτουν ώρες καθηγητών μεταξύ τους. Δηλαδή ένας καθηγητής να μη καταλήξει να διδάσκει περισσότερες από 1 φορές μια συγκεκριμένη ώρα. Το κόστος της παραβίασης αυτής της συνθήκης υπολογίζεται με τη συνάρτηση `calculateSum2AndBasic1()`. Η ύπαρξη ενός συγκεκριμένου καθηγητή 3 φορές συγκεκριμένη ώρα συγκεκριμένης μέρας, θεωρητικά θεωρείται σημαντικότερη από την ύπαρξη ύπαρξη ενός συγκεκριμένου καθηγητή 2 φορές συγκεκριμένη ώρα συγκεκριμένης μέρας. Συνεπώς, ένα τέτοιο ενδεχόμενο βαθμολογείται χειρότερα.

Basic2: Η 2^η βασική και προφανή συνθήκη που πρέπει να ελέγχεται είναι να μη ξεπερνιούνται οι ημερήσιες και οι εβδομαδιαίες ώρες διδασκαλίας των καθηγητών. Η παράβλεψη αυτής της συνθήκης βαθμολογείται από τη συνάρτηση `basic2()`. Για κάθε παραβίαση κάθε καθηγητή είτε ανά ημέρα είτε ανά εβδομάδα αυξάνουμε το κόστος κατά 1.

Sum1: Ένας περιορισμός που πρέπει να ικανοποιείται όσο το δυνατόν περισσότερο είναι να μην υπάρχουν κενά στο πρόγραμμα των τμημάτων τμήματος. Έτσι προσθέτουμε στο κόστος κάθε κενό το οποίο βρίσκουμε. Προφανώς η ύπαρξη ενός κενού θεωρείται καλύτερη από την ύπαρξη δυο κενών. Με αυτή τη λογική γίνεται ο υπολογισμός του κόστους αυτού.

Sum2: Ένας επιπλέον περιορισμός που πρέπει να ικανοποιείται όσο το δυνατόν περισσότερο είναι να μην διδάσκει ένας καθηγητής περισσότερες από 2 ώρες. Έτσι αν για κάθε καθηγητή για κάθε επιπλέον ώρα που διδάσκει αυξάνουμε το κόστος.

Βελτιστοποιήσεις

Η 1^η και η 2^η βελτιστοποίηση χρησιμοποιούν την ίδια λογική αξιολόγησης.

Παράδειγμα για κατανόηση λογικής:

Έστω πως έχουμε το εξής πρόγραμμα και θέλουμε να ελέγξουμε πόσο καλά κατανεμημένες είναι οι ώρες διδασκαλίες μέσα στο πρόγραμμα. Αθροιστικά οι ώρες διδασκαλίας είναι 20. Το 'τέλειο' πρόγραμμα θα ήταν να γέμιζαν οι ώρες ανά επίπεδο:

	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
1η	1	1	1	1	1
2η	1	1	1	1	1
3η	1	1	1	1	1
4η	1	1	1	1	1
5η					
6η					
7η					

Το 'χειρότερο' θα ήταν να γέμιζαν ανά ημέρα:

	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
1η	1	1	1		
2η	1	1	1		
3η	1	1	1		
4η	1	1	1		
5η	1	1	1		
6η	1	1	1		
7η	1	1			

Θεωρούμε πως όπου έχει τον αριθμό 1 διδάσκεται κάποιο μάθημα.

Το πρόγραμμά μας το οποίο θέλουμε να βαθμολογήσουμε είναι έτσι:

	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
1η	Γλώσσα	Αρχαία	Γλώσσα	Οδύσσεια	Φυσική
2η	Μαθηματικά	Οδύσσεια	Χημεία	Αρχαία	Φυσική
3η	Ιστορία	Ιστορία	Βιολογία	Αρχαία	
4η	Εικαστικά	Θρησκευτικά	Μουσική	Μουσική	
5η	Γυμναστική			Μαθηματικά	
6η					
7η					

Ο τρόπος λοιπόν με τον οποίο βαθμολογούμε είναι ο εξής:

Συγκρίνουμε το πρόγραμμά μας με το 'τέλειο' για τις δεδομένες μας ώρες πρόγραμμα. Τρέχουμε μία ρουτίνα στο τέλειο πρόγραμμα η οποία βρίσκει το ανώτατο επίπεδο στο οποίο διδάσκεται κάποιο μάθημα, στην περίπτωση μας, το τελευταίο επίπεδο είναι η 4^η ώρα.

Στην αρχή αυτής της σύγκρισης αρχικοποιούμε μία μεταβλητή στην οποία θα κρατάμε το σκορ του προγράμματος. Στο σκορ αυτό αρχικά προσθέτουμε τη διαφορά του αθροίσματος των ωρών ανά επίπεδο, δηλαδή ανά ώρα:

	1η	2η	3η	4η	5η	6η	7η
Καλύτερη:	5	5	5	5	0	0	0
Τωρινή:	5	5	4	4	2	0	0
Σύνολο:	0	0	1	1	2	0	0

Μέχρι στιγμής έχουμε σκορ 4. Έπειτα θέλουμε να δούμε αυτές τις ώρες που απέχουν από την τέλεια κατάσταση, πόσο πολύ απέχουν ακριβώς από το τελευταίο επίπεδο που αναφέραμε πριν(4^η ώρα):

	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
1η	Γλώσσα	Αρχαία	Γλώσσα	Οδύσσεια	Φυσική
2η	Μαθηματικά	Οδύσσεια	Χημεία	Αρχαία	Φυσική
3η	Ιστορία	Ιστορία	Βιολογία	Αρχαία	
4η	Εικαστικά	Θρησκευτικά	Μουσική	Μουσική	
5η	Γυμναστική			Μαθηματικά	
6η					
7η					

Δηλαδή η Δευτέρα έχει 5 ώρες ενώ το τέλειο έχει 4. Σκανάροντας ώρα ώρα εφαρμόζεται η γενική βαθμολόγηση γίνεται με αυτόν τον τύπο: (ώρες τέλειου-ώρες τωρινού)*(διαφορά από τελευταίο επίπεδο+1)

Επομένως σύμφωνα με τον παραπάνω τύπο η γυμναστική και τα μαθηματικά απέχουν μία ώρα από το τελευταίο επίπεδο, το καθένα προσθέτει 2 στο σκορ. Άρα το συνολικό σκορ είναι 4+4=8.

Έπειτα συγκρίνουμε το 'καλύτερο' με το 'χειρότερο' πρόγραμμα εφαρμόζοντας την ίδια ακριβώς λογική και κρατάμε κάπου αποθηκευμένο το σκορ του χειρότερου προγράμματος. Για να έχουμε μια αυστηρή βαθμολόγηση βλέπουμε πόσο πολύ απέχει το τωρινό από το χειρότερο πρόγραμμα και εξάγοντας ένα τελικό ποσοστό:

$((\text{χειρότερο σκορ} - \text{τωρινό σκορ}) / \text{χειρότερο σκορ}) * \text{τέλειο σκορ}$

Το οποίο σημαίνει πως ένα τέλειο σκορ, θα είναι το τωρινό να είναι ίσο με το 0.

Η 3^η βελτιστοποίηση υπολογίζει ποια είναι η τέλεια κατανομή ωρών διδασκαλίας για όλους του καθηγητές και υπολογίζει στο τελικό πρόγραμμα πόσο απέχουν.

Ανάλυση συναρτήσεων γενετικού:

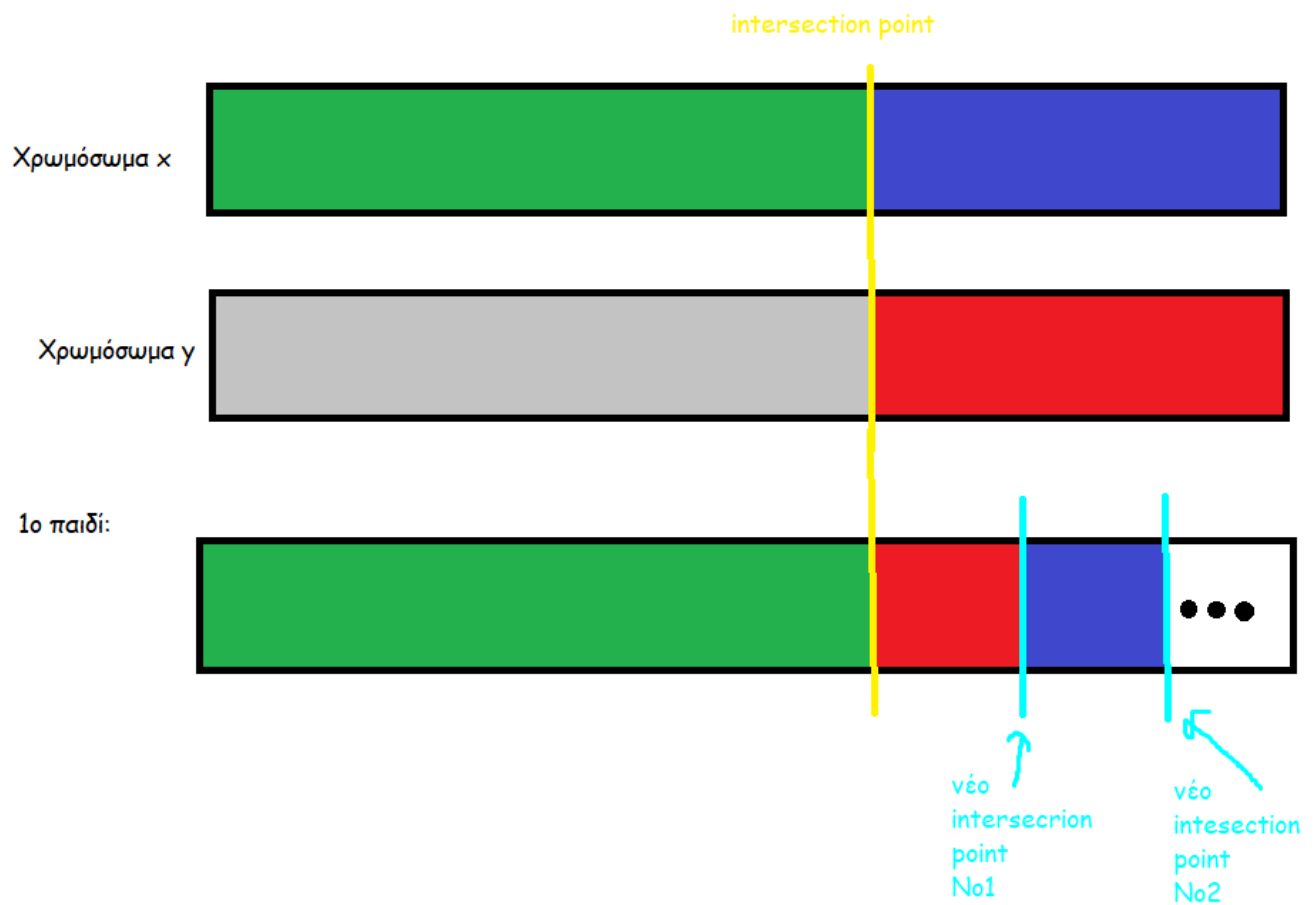
reproduce: Είναι το στάδιο κατά το οποίο δημιουργούμε 2 χρωμοσώματα με την αναπαραγωγή 2 τυχαίων χρωμοσωμάτων. Δέχεται σαν όρισμα τα 2 χρωμοσώματα προς γονιμοποίηση και επιστρέφει τα δύο νέα χρωμοσώματα. Για την δημιουργία των παιδιών εφαρμόζουμε όσα **έξυπνα κοψίματα** απαιτούνται. Η ιδέα προϋποθέτει ότι κάθε χρωμόσωμα το οποίο δίνεται ως είσοδος δεν έχει ποτέ επικαλύψεις ωρών. Αυτό εξασφαλίζεται κατά την κατασκευή του τυχαίου χρωμοσώματος και κατά τη γονιμοποίηση των 2 χρωμοσωμάτων.

Η γενική ιδέα του αλγορίθμου είναι η ακόλουθη: Επιλέγεται ένα τυχαίο σημείο(intersection point) κατά το οποίο θα γίνει το κόψιμο. Το σημείο αυτό οριοθετεί το βασικό κόψιμο. Επιλέγουμε να φιξάρουμε την μεγαλύτερη πλευρά του νέου χρωμοσώματος με όσα γονίδια έχει ο x (αντίστοιχα ο y). Μένει να ασχοληθούμε με την μικρότερη μεριά. (Θα μπορούσαμε να μην ασχοληθούμε με το ποια μεριά θα φιξάρουμε, ή να γίνεται τυχαία.)

Στην μικρότερη μεριά πρέπει να βάλουμε τα γονίδια που είχε στις αντίστοιχές θέσεις ο y (αντίστοιχα ο x). Όμως αυτό μπορεί να δημιουργήσει επικαλύψεις ωρών του τμήματος στο οποίο μέσα έγινε το κόψιμο. Σοβαρό λογικό λάθος καθώς είναι αδύνατο το ίδιο τμήμα να παρακολουθεί το διαφορετικά μαθήματα(ή διαφορετική ώρα του ίδιου μαθήματος) την ίδια ώρα. Έτσι, αν κατά την εισαγωγή γονιδίου y (αντίστοιχα x) διαπιστωθεί δημιουργία επικάλυψης ώρας επιλέγουμε να κρατήσουμε το αντίστοιχο γονίδιο του χρωμοσώματος x (αντίστοιχα y).

Όπως είναι λογικό, η απόφαση να κρατήσουμε το γονίδιο του χρωμοσώματος x δε δημιουργεί επικάλυψη καθώς στο γονίδιο x δε υπήρχε επικάλυψη ωρών. Επιπλέον, δεν γίνεται να επικαλύπτετε η νέα ώρα του x με προηγούμενη ώρα του y καθώς αυτό θα είχε διαπιστωθεί σε προηγούμενη επανάληψη.

Ουσιαστικά, σε αυτό το σημείο έχει δημιουργηθεί ένα έξυπνο κόψιμο γύρω από το στοιχείο το οποίο επιλέξαμε να πάρουμε. Δηλαδή έχουν δημιουργηθεί 2 επιπλέον intersection points. Σχηματική απεικόνιση ιδέας:



mutate:

Δέχεται σαν όρισμα την πιθανότητα αλλαγής κάθε γονιδίου. Έτσι, κάθε καθηγητής κάθε γονιδίου θα αλλάξει με την συγκεκριμένη πιθανότητα και θα επιλεγεί κατά προτίμηση ένας καθηγητής με τυχαίο τρόπο ο οποίος να πληρεί τις προϋποθέσεις. Δηλαδή να μπορεί να διδάξει τη συγκεκριμένη ώρα (να μη διδάσκει αλλού τη συγκεκριμένη ώρα) και προφανώς να μπορεί να διδάξει το συγκεκριμένο μάθημα.