

# Τεχνικές Βελτιστοποίησης Project

Κουκουλέτσου Αικατερίνη 10218

Φεβρουάριος 2023

## Θεωρητική Ανάλυση

Η εργασία αφορά την μελέτη γενετικών αλγορίθμων. Οι γενετικοί αλγορίθμοι στοχεύουν στην εύρεση βέλτιστων λύσεων σε προβλήματα τα οποία έχουν πολλούς παραμέτρους και άρα δεν μπορούν να εφαρμοστούν εύκολα οι άλλες κλασικές μέθοδοι. Ο βασικός μηχανισμός τους είναι εμπνευσμένος από την βιολογία και μάλιστα από την θεωρία της εξέλιξης του Δαρβίνου. Συγκεκριμένα, ακολουθώντας τον κανόνα της φυσικής επιλογής, οι οργανισμοί που είναι καλύτερα προσαρμοσμένοι στο περιβάλλον μεταβιβάζουν χαρακτηριστικά στους απογόνους σε αντίθεση με εκείνους που είναι λιγότερο προσαρμοσμένοι. Επιβιώνει δηλαδή ο ισχυρότερος, ο οποίος δημιουργεί και τους απογόνους. Η επόμενη γενιά αναπαράγεται με την ίδια λογική και η διαδικασία επαναλαμβάνεται έως ώτου να εντοπιστεί η βέλτιστη λύση.

Με τον όρο γενιά (*Generation*), ορίζεται ένας αρχικός πληθυσμός  $N$  πιθανών λύσεων. Οι λύσεις αυτές επιλέγονται τυχαία ε/ώ να ικανοποιούν κάθε φορά τις προδιαγραφές του προβλήματος. Η αναπαράσταση κάθε λύσης γίνεται με την βοήθεια χρωμοσωμάτων, δηλαδή χρησιμοποιούνται διανύσματα μεγέθους  $1 \times M$  για κάθε λύση (όπου  $M$  ο συνολικός αριθμός των παραμέτρων της λύσης). Στα χρωμοσώματα, η κωδικοποίηση μπορεί να γίνει δυαδικά (binary encoded), με πραγματικές τιμές (real value), με την σειρά (permutation) κλπ. Μόλις δημιουργηθεί η πρώτη γενιά με την βοήθεια των πράξεων της επιλογής (selection), διασταύρωσης (crossover) και μετάλλαξης (mutation) προκύπτει η επόμενη. Οι πράξεις αυτές, οι οποίες εξηγούνται αναλυτικά παρακάτω γίνονται με κριτήριο την μεγιστοποίηση της συνάρτησης καταλληλότητας (fitness function). Ο αλγόριθμος ακολουθεί αυτά τα βήματα επαναληπτικά και τερματίζει μόλις το  $|\frac{f(x_{k+1}) - f(x_k)}{f(x_k)}| < \varepsilon$  όπου ως ε ορίζουμε την ακρίβεια της λύσης.

## Μαθηματική Διατύπωση Προβλήματος

### Συνάρτηση καταλληλότητας

Στόχος είναι η ελαχιστοποίηση του χρόνου διάσχισης του δικτύου για συνολικό ρυθμό αυτοκινήτων  $V = 100$ . Επομένως, ορίζουμε συνάρτηση προς ελαχιστοποίηση την

$$f(x) = \sum_{i=1}^{17} x_i \cdot T_i$$

όπου  $i$  η διαδρομή μεταξύ δύο κόμβων,  $x_i$  ο ρυθμός κίνησης και  $T_i$  ο χρόνος κίνησης στον κόμβο  $i$ . Κάθε  $T_i$  υπολογίζεται από τον τύπο

$$T_i = t_i + a_i \cdot \frac{x_i}{1 - \frac{x_i}{c_i}}$$

Επειδή  $t_i$  είναι ο σταθερός χρόνος που απαιτείται για να κινηθούμε στον δρόμο  $i$  όταν η κίνηση είναι ασθενής, η τιμή του δεν επηρεάζει την ελαχιστοποίηση της συνάρτησης. Επειδή κατά κύριο λόγο, οι γενετικοί αλγόριθμοι είναι αλγόριθμοι μεγιστοποίησης τελικά ως συνάρτηση καταλληλότητας fitness function ορίζεται η  $\frac{1}{f(x)}$ .

### Περιορισμοί

Για κάθε διαδρομή ορίζεται ο μέγιστος ρυθμός διέλευσης  $c_i$ . Αν το  $x_i$  ξεπεράσει το  $c_i$  τότε ο χρόνος διάσχισης της διαδρομής γίνεται άπειρος και άρα η βελτιστοποίηση αποτυγχάνει. Επομένως πρέπει για κάθε  $i$ ,  $x_i < c_i$ . Επιπλέον, θεωρούμε ότι σε κόμβο το άθροισμα των εισερχόμενων ρυθμών ισούται με το άθροισμα των εξερχόμενων. Επομένως προκύπτει  $x_1 + x_2 + x_3 + x_4 = 100, x_1 = x_6 + x_5, x_2 = x_7 + x_8, x_4 = x_9 + x_{10}, x_3 + x_9 + x_8 = x_{11} + x_{12} + x_{13}, x_6 + x_7 + x_{13} = x_{14} + x_{15}, x_5 + x_{14} = x_{16}, x_{10} + x_{11} = x_{17}$  και  $x_{17} + x_{12} + x_{15} + x_{16} = 100$

### Χρωμόσωμα και κωδικοποίηση

Ως τρόπο κωδικοποίησης επιλέγουμε την κωδικοποίηση με πραγματικές τιμές (real value). Κάθε χρωμόσωμα για να περιλαμβάνει όλη την πληροφορία της λύσης θα πρέπει να έχει όλες τις τιμές  $x_i$ . Επομένως, στην υλοποίηση του κώδικα, κάθε χρωμόσωμα είναι ένα διάνυσμα με διαστάσεις 1x17, το οποίο έχει την μορφή  $A = [x_1 \ x_2 \ \dots \ x_{17}]$ .

## Μέρος Α

### Πρώτη Γενιά και Αξιολόγηση

Μετά από την δημιουργία της πρώτης γενιάς καλείται η συνάρτηση καταλληλότητας η οποία αξιολογεί τις πρώτες τυχαίες λύσεις. Η fitness function επιστρέφει μία βαθμολογία για κάθε χρωμόσωμα η οποία αντιπροσωπεύει την αποτελεσματικότητα της λύσης. Όσο μεγαλύτερη η τιμή που επιστρέφει η fitness function τόσο καλύτερο είναι και το αποτέλεσμα, δηλαδή τόσο πιο κοντά είναι η λύση στην βέλτιστη. Επειδή όμως προκειται για πρόβλημα πολλών παραμέτρων, μία κακή λύση δεν θα πρέπει να απορρίπτεται αυθαίρετα καθώς μπορεί να μην έχει καλό αποτέλεσμα συνολικά αλλά να περιλαμβάνει κάποια στοιχεία της βέλτιστης λύσης τα οποία θέλουμε να "επιβιώσουν" στις επόμενες γενιές. Αυτό βέβαια είναι μία πληροφορία που δεν μπορούμε να γνωρίζουμε. Για να ξεπεραστεί αυτό το πρόβλημα, η επιλογή των χρωμοσωμάτων που θα επιβιώσουν στην επόμενη γενιά γίνεται πιθανοκρατικά, με την βοήθεια της Ρουλέτας.

### Πράξη τυχαίας επιλογής - Ρουλέτα

Η ρουλέτα αφού ταξινομήσει τις βαθμολογίες των χρωμοσωμάτων σε αύξουσα σειρά, βρίσκει την αθροιστική πιθανότητα και κατασκευάζει ένα κυκλικό διάγραμμα. Με την τυχαία παραγωγή αριθμών στο διάστημα  $[0, 1]$  επιλέγεται το αντίστοιχο χρωμόσωμα και τελικά παράγεται η δεύτερη γενιά. Με αυτόν τον τρόπο αν μία λύση είναι καλή, καταλαμβάνει μεγαλύτερο μέρος του κυκλικού διαγράμματος και άρα έχει περισσότερες πιθανότητες να επιλεγεί ενώ αν μια λύση είναι κακή καταλαμβάνει μικρότερο τμήμα και άρα είναι πιο δύσκολο να επιβιώσει. Παρόλ'αυτα ο αλγόριθμος αποφασίζει τυχαία ποιά θα είναι τα επόμενα χρωμοσώματα και άρα καμία λύση δεν απορρίπτεται αυθαίρετα, το οποίο θα μπορούσε να οδηγήσει σε εγκλωβισμό σε τοπικό ελάχιστο.

### Πράξη Μετάλλαξης

Η πράξη της μετάλλαξης γίνεται με την βοήθεια της Gaussian κατανομής. Για κάθε χρωμόσωμα ξεχωριστά και για τυχαία επιλεγμένα πεδία του χρωμοσώματος αυτού, μεταβάλλεται η ποσότητα  $x_i$  κατά ένα  $\Delta x_i$ . Λόγω των περιορισμών του προβλήματος που θέλουν το άθροισμα των εισερχόμενων ρυθμών να είναι ίσο με το άθροισμα των εξερχόμενων για κάθε κόμβο, προκύπτει ότι μεταλλάξεις μπορούν να γίνουν μόνο στα  $x_1, x_2, x_3$  και  $x_4$ . Αυτό συμβαίνει γιατί, αν σε κάθε κόμβο πρέπει το άθροισμα των εισερχόμενων και των εξερχόμενων ρυθμών να είναι ίσο με μηδέν, κάθε μεταβολή που συμβαίνει σε  $x_i$  για  $i = 5, 6, \dots, 17$  είναι αποτέλεσμα μιας μεταβολής που έγινε στους αρχικούς κλάδους 1, 2, 3, 4. Με άλλα λόγια, μια μεταβολή π.χ. στο  $x_{15}$  συνεπάγεται μεταβολές στα 14, 13, 7 και 6 έτσι ώστε να εξακολουθεί να ισχύει  $x_{13} + x_7 + x_6 = x_{14} + x_{15}$  καθώς και στα  $x_{16}, x_{12}, x_{17}$  έτσι ώστε να εξακολουθεί να ισχύει  $x_{12} + x_{17} + x_{15} + x_{16} = 100$ . Οι μεταβολές αυτές με την σειρά τους προκαλούν μεταβολές στους υπόλοιπους κλάδους και τελικά στους  $x_1, x_2, x_3, x_4$ . Επομένως κάθε μεταβολή που συμβαίνει σε κάποιον κλάδο

είναι αποτέλεσμα της μεταβολής των κλάδων 1, 2, 3 ή 4.

Μελετώντας όλες τις πιθανές μεταλλάξεις που μπορούν να γίνουν στους κλάδους 1, 2, 3, 4, οι οποίες όπως είπαμε πριν καλύπτουν όλες τις πιθανές περιπτώσεις μετάλλαξης, προκύπτουν τα ακόλουθα αποτελέσματα. Αν γίνει μετάλλαξη σε έναν μόνο κλάδο π.χ.  $x_1 + \Delta x_1$  τότε για να ισχύει  $x_1 + x_2 + x_3 + x_4 = 100$  θα πρέπει ένας τουλάχιστον από τους υπόλοιπους κλάδους π.χ. ο κλάδος 2 να μεταβληθεί  $x_2 - \Delta x_1$ , έτσι ώστε το άθροισμα να παραμείνει σταθερό. Μπορεί όμως η μεταβολή αυτή να μοιραστεί σε 2 άλλους κλάδους π.χ.  $x_2 - \frac{k}{100} \cdot \Delta x_1$  και  $x_3 - \frac{1-k}{100} \cdot \Delta x_1$  ή μπορεί να μοιραστεί και στους τρεις.

Συγκεντρωτικά, οι περιπτώσεις για τον κλάδο 1 είναι οι ακόλουθες:

- α)μεταβολή του κλάδου 1 και του 2
  - β)μεταβολή του κλάδου 1 και του 3
  - γ)μεταβολή του κλάδου 1 και του 4
  - δ)μεταβολή του κλάδου 1 και του 2 και 3
  - ε)μεταβολή του κλάδου 1 και του 2 και 3 και 4
- αντίστοιχα προκύπτουν οι περιπτώσεις για τον κλάδο 2, 3 και 4.

Στην υλοποίηση του κώδικα επιλέγεται τυχαία ποιό από τα  $x_1, x_2, x_3, x_4$  θα αλλάξει και τυχαία σε πόσα από τα υπόλοιπα θα μοιραστεί η μεταβολή του. Επίσης, όταν η μεταβολή μοιράζεται σε 2 ή παραπάνω κλάδους το ποσοστό της μεταβολής που θα πάει σε κάθε κλάδο επιλέγεται και πάλι τυχαία. Μετά την μετάλλαξη που υλοποιείται στο αρχείο με όνομα *Mutation.m*, καλείται η *Split.m* η οποία αναλαμβάνει το μοίρασμα των  $x_i$  εκ νέου, έτσι ώστε να πληρούνται όλες οι προδιαγραφές. Τέλος, καλείται η *Check.m* η οποία ελέγχει άλλη μία φορά ότι η λύση που προέκυψε μετά την μετάλλαξη είναι μία έγκυρη λύση.<sup>1</sup>

---

<sup>1</sup>Για τον έλεγχο των συνθηκών, αντί να ελέγξω π.χ. αν το  $x_1 + x_2 + x_3 + x_4 == 100$  έλεγχα αν το  $x_1 + x_2 + x_3 + x_4 - 100 < \varepsilon$ , όπου  $\varepsilon = 10^{-9}$ , γιατί αλλιώς απέρριπτε λύσεις που ήταν έγκυρες λόγω του floating point error και δεν ήξερα πως αλλιώς να το διορθώσω.

## Πράξη Διασταύρωσης

Η πράξη διασταύρωσης που υλοποιείται στο αρχείο με όνομα *Crossover.m* μπορεί να γίνει με είτε εναλλάσσοντας τμήματα δύο χρωμοσωμάτων από ένα σημείο και μετά, είτε μεταβάλλοντας την τιμή, χρησιμοποιώντας τον μέσο όρο. Η πρώτη μέθοδος χρησιμοποιείται συνήθως όταν η κωδικοποίηση του χρωμοσώματος έχει γίνει δυαδικά. Στη συγκεκριμένη άσκηση, η χρήση αυτής της μεθόδου θα προκαλούσε πολλά προβλήματα γιατί όπως αναφέρθηκε και πριν, ακόμη και μια πολλή μικρή μεταβολή στην τιμή κάποιου  $x_i$  επηρεάζει όλο το δίκτυο. Αυτό σημαίνει πως θα κατέληγε να έχει την ίδια λειτουργία με την πράξη της μετάλλαξης και άρα θα έχανε την χρησιμότητα της, η οποία είναι να κάνει μικρές αλλαγές (tuning). Επιπλέον λόγω των περιορισμών, όλες σχεδόν η μεταβολές που θα γινόντουσαν με αυτόν τον τρόπο θα ήταν άκυρες.

Συμπερασματικά, λόγω του τρόπου κωδικοποίησης και των πολλών και αυστηρών περιορισμών που έχει το πρόβλημα επιλέχθηκε η δεύτερη μέθοδος. Ο αλγόριθμος της Διασταύρωσης παίρνει ως όρισμα δύο τυχαία χρωμοσώματα και δημιουργεί ένα νέο, στο οποίο κάθε στοιχείο  $x_i$  είναι ίσο με  $\frac{x_{1i} + x_{2i}}{2}$ .

## Μέρος Β

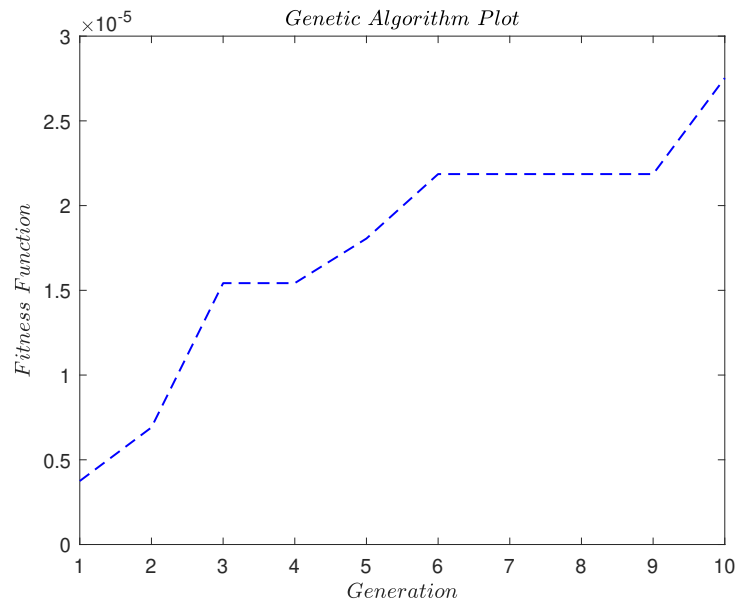
Στο αυτό το μέρος της άσκησης αλλάζει το  $V$  και παύει πια να θεωρείται σταθερό. Πλέον η τιμή του  $V$  που αντιπροσωπεύει το συνολικό άθροισμα των ρυθμών σε όλους τους κλάδους είναι μεταβλητή και μπορεί μάλιστα να μεταβάλλεται μέχρι  $\pm 15$

Ο τρόπος με τον οποίο προσεγγίστηκε η αλλαγή είναι θεωρώντας πως το πρόβλημα πλέον έχει 18 παραμέτρους. Δηλαδή, κάθε χρωμόσωμα είναι πάλι ένα διάνυσμα που αυτήν την φορά έχει μέγεθος  $1 \times 18$  και ως 18η παράμετρο έχει την τιμή του  $V$ .

Οι αλλαγές που χρειάστηκε να γίνουν είναι λίγες. Αρχικά, για την δημιουργία της πρώτης γενιάς, γίνεται τυχαία επιλογή για το  $V$  και έπειτα γίνεται το μοίρασμα σε όλες τις κατευθύνσεις όπως πριν. Επίσης, στην συνάρτηση της μετάλλαξης πλέον μπορεί να αλλάξει και η 18η παράμετρος και άρα να επιλεγεί  $V$  βέλτιστο, ενώ κατά τα άλλα ακολουθεί την λογική που αναφέρθηκε προηγουμένως. Στην διασταύρωση δεν χρειάστηκαν ιδιαίτερες αλλαγές καθώς είναι ανεξάρτητη του μεγέθους  $V$ . Με αυτές τις τροποποιήσεις προκύπτει καλύτερος ελάχιστος χρόνος και πάντα η τελική τιμή του  $V$  καταλήγει να είναι μικρότερη του 100, το οποίο είναι και το αναμενόμενο αποτέλεσμα.

## Διαγράμματα

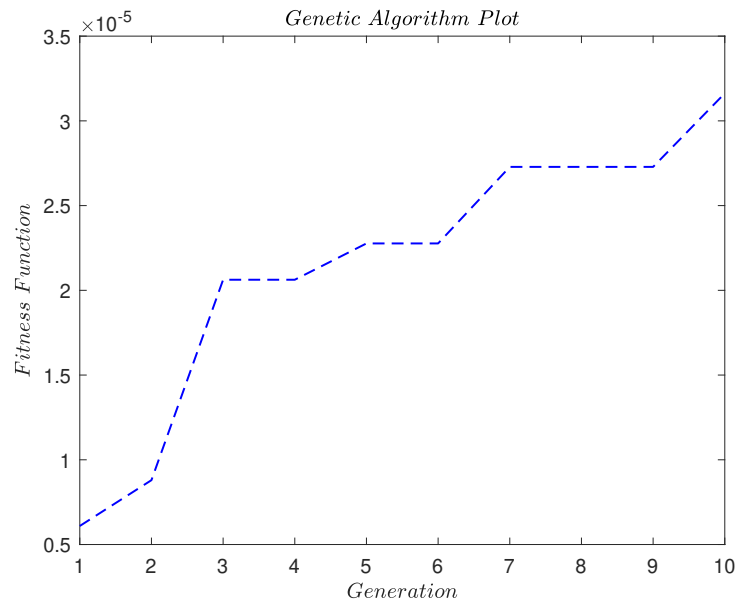
Καθώς τα δέκα χρωμοσώματα της πρώτης γενιάς, παράγονται τυχαία και άρα είναι διαφορετικά κάθε φορά που εκτελείται ο αλγόριθμος, τα διαγράμματα που παρατίθενται είναι ενδεικτικά.



```
>> GeneticAlgorithmMain
Number of iterations is 21
Final Chromosome is
 40.7518  12.1884  24.5211  22.5386  24.8519  15.8999  5.6524  6.5360  14.0526  8.4860  17.8221  15.5931  11.6946  12.4617  20.7653  37.3336  26.3081

Total time is 979.7326
This is a valid solution.
>>
```

Φαίνεται πως μεγιστοποιείται η συνάρτηση καταλληλότητας όσο περνάμε από γενιά σε γενιά. Από το command line φαίνεται το τελικό χρωμόσωμα και άρα η τελική λύση που προκύπτει. Φαίνεται επίσης ο χρόνος που απαιτείται. Μετά από πολλές επαναλήψεις προκύπτει ότι ο ελάχιστος χρόνος είναι κοντά στα 900s. Κάποιες φορές όμως, ο αλγόριθμος φαίνεται πως εγκλωβίζεται σε τοπικό ελάχιστο και προκύπτουν τιμές έως 1200s.



```
>> GeneticAlgorithmMainPart2
Number of iterations is 25
Final Chromosome is
Columns 1 through 17

    21.7255    10.4414    25.7907    28.5650     5.5660    16.1594     0.1704    10.2710    25.0138     3.5512     9.2915    18.2873    33.4966    17.7033    32.1232    23.2693    12.8427

Column 18

    86.5225

Total time is 931.5245
Value of V is 86.5225
This is a valid solution.
ξ >>
```

Φαίνεται πάλι η μεγιστοποίηση της fitness function με το πέρασμα των γενεών. Το 18ο στοιχείο στο διάνυσμα είναι το μέγεθος V και είναι ίσο με 86.5225, αναμενόμενο αποτέλεσμα αφού λιγότερη κίνηση συνεπάγεται λιγότερο χρόνο. Φαίνεται και πάλι το τελικό χρωμόσωμα και ο χρόνος που απαιτείται. Αυτήν την φορά ο αλγόριθμος δεν φαίνεται να αντιμετωπίζει προβλήματα όπως προηγουμένως αφού να διακυμάνεται σε τόσο μεγάλο εύρος τιμών από επανάληψη σε επανάληψη.