Αν. Καθηγητής Π. Λουρίδας

Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας

Οικονομικό Πανεπιστήμιο Αθηνών

Κοινωνική Αποστασιοποίηση

Στην εποχή του κορωνοϊού, σημαντικό ρόλο στην αντιμετώπιση της επιδημίας έχει η *κοινωνική αποστασιοποίηση* (social distancing). Αλλάζουμε τις συνήθειές μας και τον τρόπο με τον οποίο συναναστρεφόμαστε με τους συνανθρώπους μας ώστε να ελαχιστοποιήσουμε τις πιθανότητες μετάδοσης του ιού.

Ανάμεσα στους κανόνες που καλούνται οι πολίτες να εφαρμόσουν, κάποιοι σχετίζονται με την ελάχιστη απόσταση που πρέπει να διατηρούν μεταξύ τους. Έτσι, σε διάφορους χώρους πρέπει να λαμβάνεται πρόνοια ώστε οι παρευρισκόμενοι να είναι διασπαρμένοι ώστε να μην πλησιάζουν περισσότερο από την προτεινόμενη απόσταση.

Το πρόβλημα που τίθεται τότε είναι: αν έχουμε ένα συγκεκριμένο χώρο, πόσοι άνθρωποι μπορούν να βρίσκονται ταυτόχρονα μέσα στον χώρο αυτό; Και πού ακριβώς μπορούν να τοποθετηθούν αυτοί οι άνθρωποι;

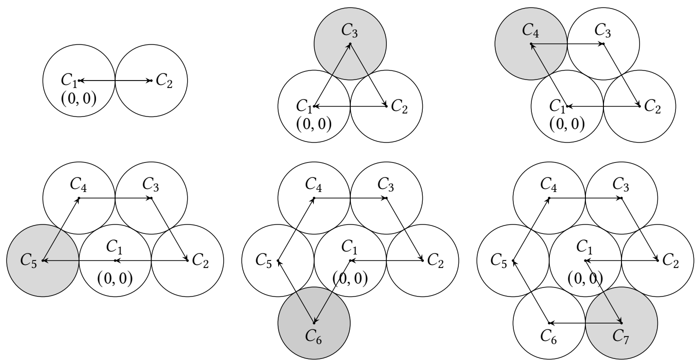
Μπορούμε να αναπαραστήσουμε το κάθε άτομο έναν κύκλο, ο οποίος έχει ακτίνα ίση με την ελάχιστη απόσταση που πρέπει να τηρείται. Τότε το πρόβλημα ανάγεται στην εισαγωγή κύκλων με την επιθυμητή διάμετρο μέσα στο γεωμετρικό σχήμα που περιγράφει τον χώρο που θέλουμε.

Για να βρούμε πού θα τοποθετηθούν αυτοί οι κύκλοι μπορούμε να εργαστούμε ως εξής.

1. Τοποθετούμε έναν κύκλο στο σημείο από το οποίο θέλουμε να ξεκινήσουμε τη διαδικασία της τοποθέτησης.
2. Τοποθετούμε έναν κύκλο εφαπτόμενο στον πρώτο, σημειώνοντας ότι ο ένας έπεται του άλλου.
3. Παρεμβάλουμε έναν κύκλο εφαπτόμενο στον τελευταίο κύκλο που βάλαμε και τον πρώτο.
4. Επιστρέφουμε στο βήμα 3.

Στην παρακάτω εικόνα μπορείτε να δείτε την εξέλιξη καθώς εισάγουμε τους πρώτους επτά κύκλους. Εδώ όμως σταματάμε, γιατί δεν μπορούμε να συνεχίσουμε με την παραπάνω μέθοδο. Αν προσπαθήσουμε να παρεμβάλουμε έναν κύκλο μεταξύ του C_1 και του C_7, θα πρέπει να φτιάξουμε έναν κύκλο ο οποίος πέφτει ακριβώς πάνω στον C_2, που έχουμε ήδη φτιάξει.

Μπορούμε όμως να παρατηρήσουμε το εξής. Τα κέντρα των κύκλων σχηματίζουν ένα κλειστό πολύγωνο. Αν θέλουμε να συνεχίσουμε τη διαδικασία, θα πρέπει να βγάλουμε από το πολύγωνο την κορυφή C_1 και να συνδέσουμε απ' ευθείας τις κορυφές



Σχήμα 1: Αρχικές Εισαγωγές

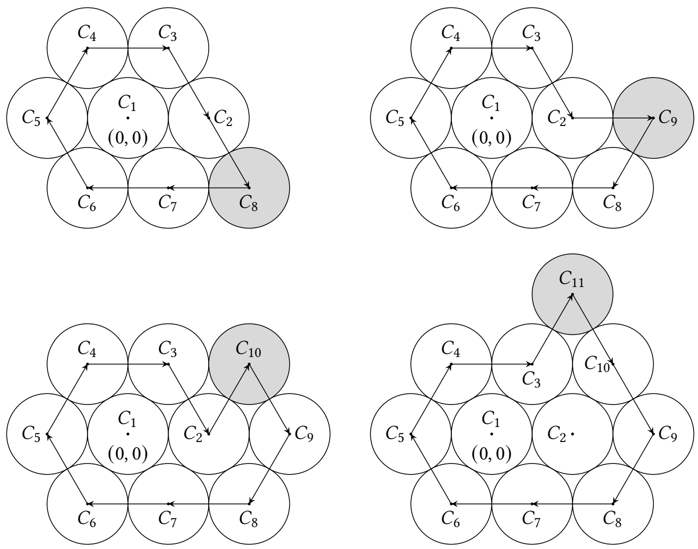
C_2 και C_7. Τότε μπορούμε να παρεμβάλουμε έναν εφαπτόμενο κύκλο στους C_2 και C_7, όπως φαίνεται στο πρώτο σχήμα της εικονας που ακολουθεί.

Αυτό λοιπόν μας δίνει μια ιδέα. Ονομάζουμε τους κύκλους στην περιφέρεια του σχήματος, πλάι στους οποίους μπορούμε να προσάψουμε εφαπτόμενους κύκλους, *μέτωπο*. Για να προσθέσουμε λοιπόν κύκλους, θα εργαζόμαστε ως εξής.

1. Εισάγουμε τους δύο πρώτους κύκλους (αυτοί αποτελούν ένα τετριμμένο μέτωπο).
2. Εντοπίζουμε τον κύκλο στο μέτωπο που είναι πιο κοντά στο σημείο εκκίνησης. Έστω ότι ο κύκλος αυτός είναι ο κύκλος C_m. Σε περίπτωση που έχουμε περισσότερους από έναν κύκλο με την ίδια ελάχιστη απόσταση, παίρνουμε τον αρχαιότερο.
3. Δοκιμάζουμε έναν κύκλο ανάμεσα στον C_m και σε αυτόν που τον ακολουθεί στο μέτωπο, C_n. Έστω ότι ο δοκιμαστικός αυτός κύκλος είναι ο C_i.
4. Αν ο C_i δεν τέμνει κανέναν κύκλο του μετώπου, τότε τον εισάγουμε και επιστρέφουμε στο βήμα 2.
5. Διαφορετικά, αφαιρούμε από το μέτωπο τους κύκλους που χρειάζεται και επιστρέφουμε στο βήμα 2.

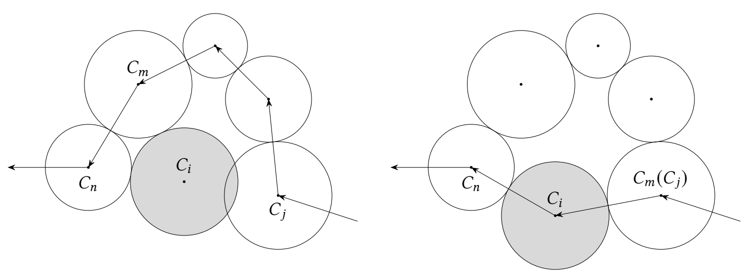
Με τον τρόπο αυτό βλέπουμε ότι εισάγουμε τους κύκλους C_9 και C_{10} χωρίς πρόβλημα. Μετά ο κοντινότερος στην εκκίνηση κύκλος του μετώπου είναι ο C_2 που τον ακολουθεί ο C_{10}. Αν δοκιμάσουμε να εισάγουμε κύκλο που να εφάπτεται στους C_2 και C_{10}, θα δούμε ότι αυτός πέφτει πάνω (άρα τέμνει) τον C_3. Ακολουθούμε το βήμα 5, αφαιρούμε τον C_2 από το μέτωπο άρα ο επόμενος του C_3 στο μέτωπο είναι ο C_{10}. Επιστρέφοντας στο βήμα 2 βρίσκουμε ότι ο κοντινότερος κύκλος του μετώπου στην εκκίνηση είναι πλέον ο C_3, οπότε προσθέτουμε τον C_{11}.

Ο αλγόριθμος που περιγράψαμε δεν είναι πλήρης, γιατί το βήμα 5 είναι ασαφές. Δεν έχουμε ορίσει ακριβώς ποιοι είναι οι κύκλοι οι οποίοι πρέπει να αφαιρέσουμε από το μέτωπο.



Σχήμα 2: Συνέχεια Εισαγωγών

Αν επιστρέψουμε στο παράδειγμά μας, είδαμε ότι αφού ο κύκλος που δοκιμάζουμε να βάλουμε εφαπτόμενο στους C_2 και C_{10} τέμνεται με τον C_3, αφαιρέσαμε τον C_2 από το μέτωπο. Γενικότερα, αν ο κύκλος C_i που δοκιμάζουμε να βάλουμε εφαπτόμενο στους C_m και C_n τέμνεται με έναν κύκλο C_j που προηγείται του C_m, αφαιρούμε από το μέτωπο τους κύκλους από τον επόμενο του C_j μέχρι τον προηγούμενο του C_n και ο C_m γίνεται ο C_j, όπως φαίνεται στο παρακάτω σχήμα:

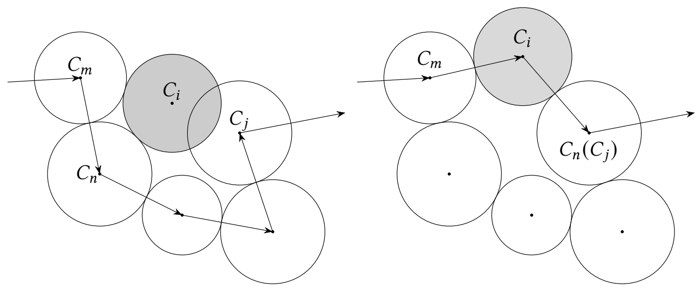


Σχήμα 3: Τομή Προηγείται του C_m

Υπάρχει και άλλο ενδεχόμενο, ο C_i που δοκιμάζουμε να βάλουμε εφαπτόμενο στους C_m και C_n να τέμνεται με έναν κύκλο C_j που έπεται του C_n. Τότε αφαιρούμε από το μέτωπο τους κύκλους από τον επόμενο του C_m μέχρι τον προηγούμενο του C_j και ο C_n γίνεται ο C_j, όπως φαίνεται στο επόμενο σχήμα.

Ο αλγόριθμός μας τότε μετασχηματίζεται ως εξής:

1. Εισάγουμε τους δύο πρώτους κύκλους (αυτοί αποτελούν ένα τετριμμένο μέ-



Σχήμα 4: Τομή Έπεται του C_n

τωπο).

2. Εντοπίζουμε τον κύκλο στο μέτωπο που είναι πιο κοντά στο σημείο εκκίνησης. Έστω ότι ο κύκλος αυτός είναι ο κύκλος C_m. Σε περίπτωση που έχουμε περισσότερους από έναν κύκλο με την ίδια ελάχιστη απόσταση, παίρνουμε τον αρχαιότερο.

3. Δοκιμάζουμε έναν κύκλο ανάμεσα στον C_m και σε αυτόν που τον ακολουθεί στο μέτωπο, C_n. Έστω ότι ο δοκιμαστικός αυτός κύκλος είναι ο C_i.

4. Αν ο C_i δεν τέμνει κανέναν κύκλο του μετώπου, τότε τον εισάγουμε στο μέτωπο και επιστρέφουμε στο βήμα 2.

5. Διαφορετικά:

- Αν ο κύκλος C_i που δοκιμάζουμε να βάλουμε εφαπτόμενο στους C_m και C_n τέμνεται με έναν κύκλο C_j που προηγείται του C_m (αν υπάρχουν παραπάνω από ένας, παίρνουμε τον κοντυνότερο στον C_m), αφαιρούμε από το μέτωπο τους κύκλους από τον επόμενο του C_j μέχρι τον προηγούμενο του C_n, θέτουμε τον C_m να είναι ο C_j, και επιστρέφουμε στο βήμα 3.

- Αν ο κύκλος C_i που δοκιμάζουμε να βάλουμε εφαπτόμενο στους C_m και C_n τέμνεται με έναν κύκλο C_j που έπεται του C_n (αν υπάρχουν παραπάνω από ένας, παίρνουμε τον κοντυνότερο στον C_n) αφαιρούμε από το μέτωπο τους κύκλους από τον επόμενο του C_m μέχρι τον προηγούμενο του C_j, θέτουμε τον C_n να είναι ο C_j, και επιστρέφουμε στο βήμα 3.

Ο αλγόριθμός μας τώρα δουλεύει και θα γεμίζει το χώρο, ξεκινώντας από το σημείο εκκίνησης. Δεν ελέγχει όμως πού πρέπει να σταματά η προσθήκη κύκλων επειδή έχουμε συναντήσει κάποιο όριο του χώρου που θέλουμε να γεμίσουμε. Για να το επιτύχουμε και αυτό, πρέπει να κάνουμε ακόμα μια προσαρμογή στον αλγόριθμό μας για να συμπεριλάβουμε τον σχετικό έλεγχο. Στη νέα εκδοχή του αλγορίθμου θα σημειώνουμε τους κύκλους του μετώπου που είναι *ζωντανοί*, δηλαδή αυτοί στους οποίους μπορούμε να προσθέσουμε εφαπτόμενο κύκλο χωρίς να φτάσουμε στα όρια του σχήματός μας. Κάθε κύκλος που μπαίνει στο μέτωπο είναι κατ' αρχήν ζωντανός.

1. Εισάγουμε τους δύο πρώτους κύκλους (αυτοί αποτελούν ένα τετριμμένο μέ-

τωπο).

2. Εντοπίζουμε τον ζωντανό κύκλο στο μέτωπο που είναι πιο κοντά στο σημείο εκκίνησης. Έστω ότι ο κύκλος αυτός είναι ο κύκλος C_m. Σε περίπτωση που έχουμε περισσότερους από έναν κύκλο με την ίδια ελάχιστη απόσταση, παίρνουμε τον αρχαιότερο.

3. Δοκιμάζουμε έναν κύκλο ανάμεσα στον C_m και σε αυτόν που τον ακολουθεί στο μέτωπο, C_n. Έστω ότι ο δοκιμαστικός αυτός κύκλος είναι ο C_i.

4. Αν ο C_i τέμνει κάποιον κύκλο του μετώπου:

- Αν ο κύκλος C_i που δοκιμάζουμε να βάλουμε εφαπτόμενο στους C_m και C_n τέμνεται με έναν κύκλο C_j που προηγείται του C_m (αν υπάρχουν παραπάνω από ένας, παίρνουμε τον κοντυνότερο στον C_m), αφαιρούμε από το μέτωπο τους κύκλους από τον επόμενο του C_j μέχρι τον προηγούμενο του C_n, θέτουμε τον C_m να είναι ο C_j, και επιστρέφουμε στο βήμα 3.

- Αν ο κύκλος C_i που δοκιμάζουμε να βάλουμε εφαπτόμενο στους C_m και C_n τέμνεται με έναν κύκλο C_j που έπεται του C_n (αν υπάρχουν παραπάνω από ένας, παίρνουμε τον κοντυνότερο στον C_n), αφαιρούμε από το μέτωπο τους κύκλους από τον επόμενο του C_m μέχρι τον προηγούμενο του C_j και επιστρέφουμε στο βήμα 3.

5. Στο σημείο αυτό έχουμε βρεί έναν κύκλο C_i που δεν τέμνει κύκλο του μετώπου. Ελέγχουμε λοιπόν αν τέμνει τα όρια του σχήματός μας. Αν ναι, τότε αλλάζουμε τον κύκλο C_m από ζωντανό σε πεθαμένο και επιστρέφουμε στο βήμα 2. Αλλιώς, εισάγουμε τον C_i στο μέτωπο και επιστρέφουμε στο βήμα 2.

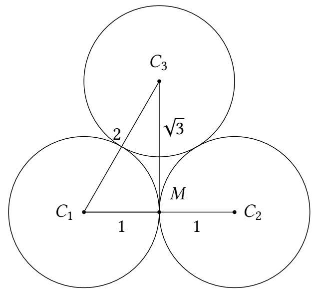
Η διαδικασία των βημάτων 2–5 επαναλαμβάνεται μέχρι να προσθέσουμε τον επιθυμητό αριθμών κύκλων ή να μην μπορούμε να προσθέσουμε άλλους κύκλους στο σχήμα, γιατί δεν έχει μείνει κανένας ζωντανός.

Στρογγυλοποιήσεις

Έστω ότι έχουμε τρεις κύκλους με ακτίνα 1. Ο C_1 έχει κέντρο στο $(0, 0)$, ο C_2 στο $(2, 0)$ και ο C_3 με κέντρο $(1, \sqrt{3})$ εφάπτεται στους C_1 και C_2. Πόσο απέχει το κέντρο του C_2 από το C_1 και πόσο απέχει το κέντρο του C_3 από το C_1; Και στις δύο περιπτώσεις ίση με 2, αφού όλες οι ακτίνες είναι ίδιες. Επιπλέον, από το Πυθαγόρειο θεώρημα, οι συντεταγμένες του C_3 είναι $(1, \sqrt{3})$.

Έστω τώρα ότι με βάση της συντεταγμένες των σημείων C_1 και C_3 θέλουμε να υπολογίσουμε εκ νέου την απόσταση C_1C_3. Αυτή, πάλι με βάση το Πυθαγόρειο θεώρημα, θα είναι ίση με $\sqrt{(\sqrt{3})^2 + 1^2}$. Αν όμως κάνουμε πράγματι την πράξη στον υπολογιστή, θα δούμε ότι το αποτέλεσμα είναι $1.999 \ldots \neq 2$.

Αυτό συμβαίνει γιατί οι πραγματικοί αριθμοί αποθηκεύονται με συγκεκριμένη ακρίβεια στον υπολογιστή, συνεπώς άρρητοι αριθμοί δεν αποθηκεύονται ακριβώς και πράξεις με άρρητους αριθμούς βγάζουν διαφορετικά αποτελέσματα από αυτά που θα περιμέναμε στη θεωρία. Αν στο πρόγραμμά μας προσπαθούσαμε να βρούμε τον



Σχήμα 5: Τρεις Εφαπτόμενοι Κύκλοι

κύκλο που είναι πιο κοντά στον C_1, το πρόγραμμά μας θα έβρισκε τον C_3, ενώ έχει την ίδια απόσταση με τον C_2. Για να το αποφύγουμε αυτό θα πρέπει στο πρόγραμμά μας να κάνουμε στρογγυλοποιήσεις όπου απαιτούνται.

Εύρεση Κοντινότερου Κύκλου στο Σημείο Εκκίνησης

Αν (x, y) οι συντεταγμένες του σημείου εκκίνησης, τότε για να βρούμε τον κοντινότερο κύκλο του μετώπου στο σημείο εκκίνησης, αρκεί να συγκρίνουμε τις αποστάσεις των κέντρων των κύκλων από το σημείο εκκίνησης. Για κάθε κύκλο C_m του μετώπου με κέντρο (m_x, m_y), η απόσταση είναι:

$$d = \sqrt{(m_x - x)^2 + (m_y - y)^2}$$

Αφού δεν μας ενδιαφέρει η τιμή της απόστασης, αλλά μόνο το ποια απόσταση είναι μικρότερη, μπορούμε να παραλείψουμε τη ρίζα και να χρησιμοποιήσουμε το:

$$l_2 = (m_x - x)^2 + (m_y - y)^2$$

Αν $(x, y) = (0, 0)$, τότε έχουμε:

$$l_2 = m_x^2 + m_y^2$$

Εύρεση Εφαπτόμενου Κύκλου

Αν έχουμε δύο κύκλους C_m και C_n και θέλουμε να βρούμε έναν τρίτο κύκλο C_k ακτίνας r που να εφάπτεται σε αυτούς τους δύο, εργαζόμαστε ως ακολούθως:

- Υπολογίζουμε την οριζόντια απόσταση d_x και την κάθετη απόσταση d_y των κέντρων των δύο κύκλων C_m και C_n. Αν (m_x, m_y) είναι το κέντρο του C_m και (n_x, n_y) είναι το κέντρο του C_n, έχουμε:

$$d_x = n_x - m_x$$

$$d_y = n_y - m_y$$

- Υπολογίζουμε την απόσταση μεταξύ των κέντρων των δύο κύκλων C_m και C_n:

$$d = \sqrt{{d_x}^2 + {d_y}^2}$$

- Αν r_m είναι η ακτίνα του C_m και r_n είναι η ακτίνα του c_n, υπολογίζουμε τις τιμές:

$$r_1 = r_m + r$$
$$r_2 = r_n + r$$

- Υπολογίζουμε τα:

$$\lambda = \frac{{r_1}^2 - {r_2}^2 + d^2}{2d^2}$$
$$\varepsilon = \sqrt{\frac{{r_1}^2}{d^2} - \lambda^2}$$

- Το κέντρο (k_x, k_y) του κύκλου C_k έχει τις συντεταγμένες:

$$k_x = m_x + \lambda d_x \mp \varepsilon d_y$$
$$k_y = m_u + \lambda d_y \pm \varepsilon d_x$$

- Παρατηρούμε ότι υπάρχουν δύο κέντρα. Εμείς θα χρησιμοποιούμε το πρώτο από αυτά (δηλαδή το $-\varepsilon d_y$ στο k_x και το $+\varepsilon d_x$ στο k_y). Τότε οι κύκλοι θα προστίθενται στο μέτωπο με φορά αντίθετη των δεικτών του ρολογιού, όπως στα παραδείγματά μας.

Αν στη διάθεσή μας έχουμε γεωμετρικά όργανα, χαρτί και μολύβι, τότε μπορούμε να βρούμε τους εφαπτόμενους κύκλους με κανόνα και διαβήτη. Στη σχήμα 5 μπορείτε να δείτε ότι αρκεί να βρούμε τα σημεία τομής των δύο κύκλων με κέντρα C_m και C_n και ακτίνα $r_m + r$ και $r_n + r$ αντίστοιχα. Οι μαθηματικοί τύπου που δώσαμε προκύπτουν από ανάλυση διανυσμάτων.

Απόσταση Κύκλου από Ευθύγραμμο Τμήμα

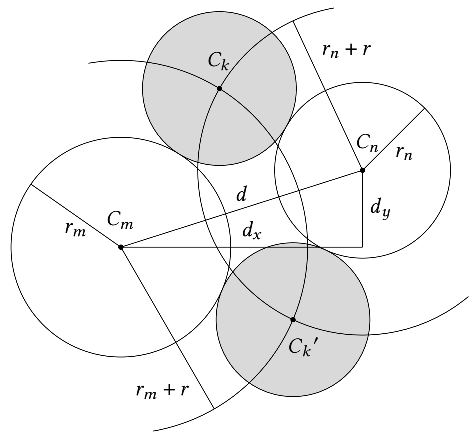
Αν έχουμε ένα ευθύγραμμο τμήμα μεταξύ των σημείων (u, v), τότε για να βρούμε την απόσταση ενός κύκλου από αυτό εκτελούμε τους παρακάτω υπολογισμούς:

- Υπολογίζουμε το τετράγωνο της απόστασης μεταξύ των δύο σημείων:

$$l_2 = (u_x - v_x)^2 + (u_y - v_y)^2$$

- Αν αυτή η απόσταση είναι ίση με το μηδέν, τότε τα δύο άκρα του ευθύγραμμου τμήματος συμπίπτουν οπότε η απόσταση του κύκλου είναι απλώς:

$$d = \sqrt{(u_x - c_x)^2 + (u_y - c_y)^2}$$



Σχήμα 6: Εύρεση Εφαπτόμενου Κύκλου

- Διαφορετικά, υπολογίζουμε το:

$$t = \frac{(c_x - u_x)(v_x - u_x) + (c_y - u_y)(v_y - u_y)}{l_2}$$

- Εξασφαλίζουμε ότι το t είναι μεταξύ 0 και 1:

$$t = \max(0, \min(1, t))$$

- Βρίσκουμε τη προβολή p του κέντρου του κύκλου πάνω στο ευθύγραμμο τμήμα:

$$p_x = u_x + t(v_x - u_x)$$
$$p_y = u_y + t(v_y - u_y)$$

- Η απόσταση είναι:

$$d = \sqrt{(p_x - c_x)^2 + (p_y - c_y)^2}$$

Απαιτήσεις Προγράμματος

Κάθε φοιτητής θα εργαστεί σε αποθετήριο στο GitHub. Για να αξιολογηθεί μια εργασία θα πρέπει να πληροί τις παρακάτω προϋποθέσεις:

- Για την υποβολή της εργασίας θα χρησιμοποιηθεί το ιδιωτικό αποθετήριο του φοιτητή που δημιουργήθηκε για τις ανάγκες του μαθήματος και του έχει αποδωθεί. Το αποθετήριο αυτό έχει όνομα του τύπου `username-algo-assignments`, όπου `username` είναι το όνομα του φοιτητή στο GitHub. Για παράδειγμα, το σχετικό αποθετήριο του διδάσκοντα θα ονομαζόταν `louridas-algo-assignments` και θα ήταν προσβάσιμο στο https://github.com/dmst-algorithms-course/louridas-algo-assignments. *Τυχόν άλλα αποθετήρια απλώς θα αγνοηθούν.*

- Μέσα στο αποθετήριο αυτό θα πρέπει να δημιουργηθεί ένας κατάλογος `assignment-2020-3`.
- Μέσα στον παραπάνω κατάλογο το πρόγραμμα θα πρέπει να αποθηκευτεί με το όνομα `social_distancing.py`.
- Δεν επιτρέπεται η χρήση έτοιμων βιβλιοθηκών γράφων ή τυχόν έτοιμων υλοποιήσεων των αλγορίθμων, ή τμημάτων αυτών, εκτός αν αναφέρεται ρητά ότι επιτρέπεται.
- Επιτρέπεται η χρήση δομών δεδομένων της Python όπως στοίβες, λεξικά, σύνολα, κ.λπ.
- Επιτρέπεται η χρήση της βιβλιοθήκης `math` και της βιβλιοθήκης `random`.
- Επιτρέπεται η χρήση της βιβλιοθήκης `argparse` ή της βιβλιοθήκης `sys` (συγκεκριμένα, της λίστας `sys.argv`) προκειμένου να διαβάσει το πρόγραμμα τις παραμέτρους εισόδου.
- Το πρόγραμμα θα πρέπει να είναι γραμμένο σε Python 3.

Το πρόγραμμα θα καλείται ως εξής (όπου `python` η κατάλληλη εντολή στο εκάστοτε σύστημα):

```
social_distancing.py [-items ITEMS]
                     [-r RADIUS]
                     [--min_radius MIN_RADIUS]
                     [--max_radius MAX_RADIUS]
                     [-b BOUNDARY]
                     [-s SEED]
                     output_file
```

Οι παράμετροι εμφανίζονται σε διαφορετικές γραμμές προς χάρη της τυπογραφίας. Η σημασία των παραμέτρων είναι η εξής:

- `-items ITEMS`, `--items ITEMS`: ο αριθμός των αντικειμένων (κύκλων) που θέλουμε να εισάγουμε. Αν δίνεται, το πρόγραμμα θα προσπαθεί να παράξει αυτόν τον αριθμό των κύκλων.
- `-r RADIUS`, `--radius RADIUS`: η ακτίνα των κύκλων. Αν δίνεται, όλοι οι κύκλοι θα έχουν αυτήν την ακτίνα.
- `--min_radius`: η ελάχιστη ακτίνα των κύκλων, αν δίνεται οι κύκλοι θα έχουν τυχαίες ακτίνες, με τουλάχιστον αυτό το μήκος. Η παράμετρος αυτή συνδυάζεται με την παράμετρο `--max_radius`.
- `--max_radius:`: η μέγιστη ακτίνα των κύκλων, αν δίνεται οι κύκλοι θα έχουν τυχαίες ακτίνες όχι μεγαλύτερες από αυτό το μήκος. Η παράμετρος αυτή συνδυάζεται με την παράμετρο `--min_radius`.
- `-b BOUNDARY`, `--b BOUNDARY`: τα όρια του σχήματος. Αν δίνεται, το πρόγραμμα θα εισάγει κύκλους μέσα στα όρια του σχήματος που δίνεται.

- `-s SEED`, `--s SEED`: αν δίνεται, η τιμή αρχικοποίησης (σπόρος) της γεννήτριας ψευδοτυχαίων αριθμών. Στην αρχή του προγράμματός σας, θα πρέπει να εκτελείται η εντολή `random.seed(SEED)`
- `output_file`: το αρχείο όπου αποθηκεύονται τα αποτελέσματα του προγράμματος, υποχρεωτική παράμετρος.

Όταν το πρόγραμμα τελειώνει, στην έξοδό θα τυπώνει μόνο έναν ακέραιο αριθμό, τον αριθμό των κύκλων που εισήγαγε. Στο αρχείο `output_file` θα αποθηκεύει τους κύκλους αυτούς, έναν σε κάθε γραμμή, με τη μορφή:

```
x y r
```

δηλαδή κάθε γραμμή θα αποτελείται από τρεις πραγματικούς αριθμούς, τη συντεταγμένη x του κέντρου του κύκλου, τη συντεταγμένη y του κέντρου του κύκλου, και την ακτίνα του r. Οι αριθμοί θα δίνονται με ακρίβεια εκατοστού.

Αν δίνονται τα όρια του σχήματος με την παράμετρο `-b BOUNDARY`, τα όρια αυτά θα τα παραθέτετε στο τέλος του αρχείου, μετά από όλους τους κύκλους, με γραμμές της μορφής:

```
x1 y1 x2 y2
```

δηλαδή τα δύο ζευγάρια συντεταγμένων που αντιστοιχούν σε κάθε ευθύγραμμο τμήμα.

Για την οπτικοποίηση των αποτελεσμάτων, μπορείτε να χρησιμοποιήσετε τα παρακάτω προγράμματα:

- `svg_draw.py` το οποίο καλείτε ως:

  ```
  python svg_draw.py input_file output_file
  ```

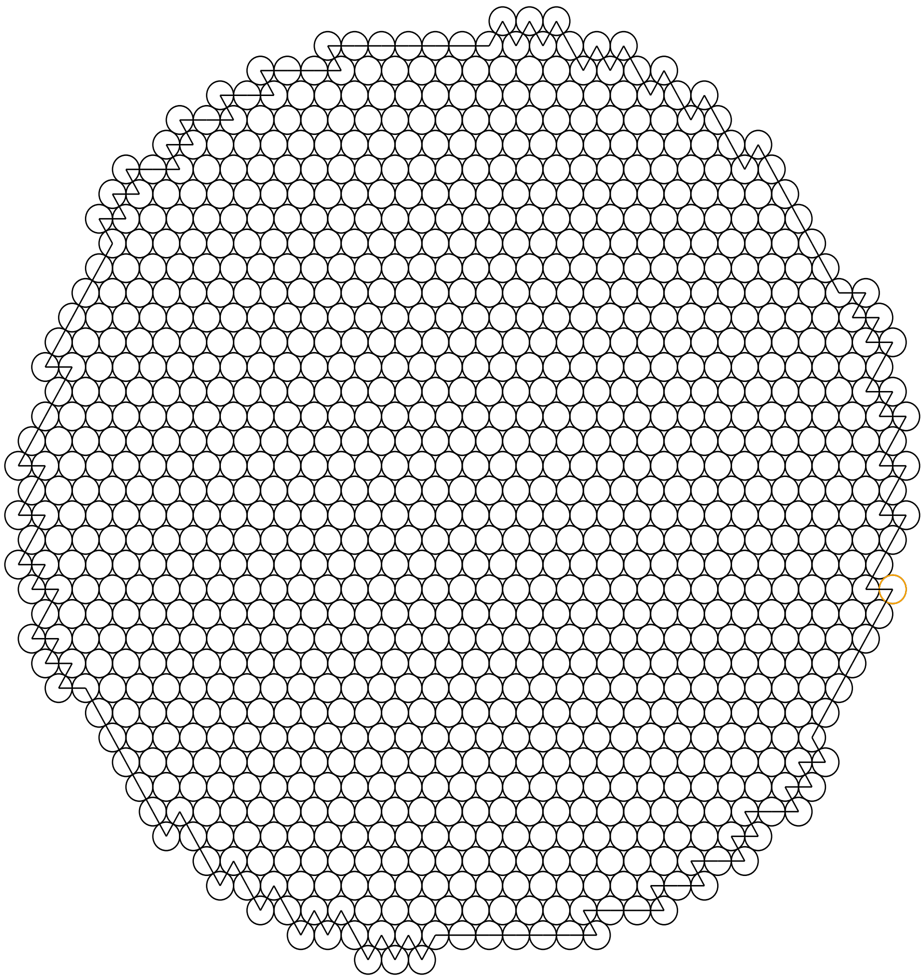
  όπου `input_file` είναι το αρχείο των αποτελεσμάτων σας και `output_file` είναι ένα αρχείο τύπου SVG. Για να λειτουργήσει, θα πρέπει να εγκαταστήσετε τη βιβλιοθήκη `svgwrite`.

- `mpl_draw.py` το οποίο καλείτε ως:

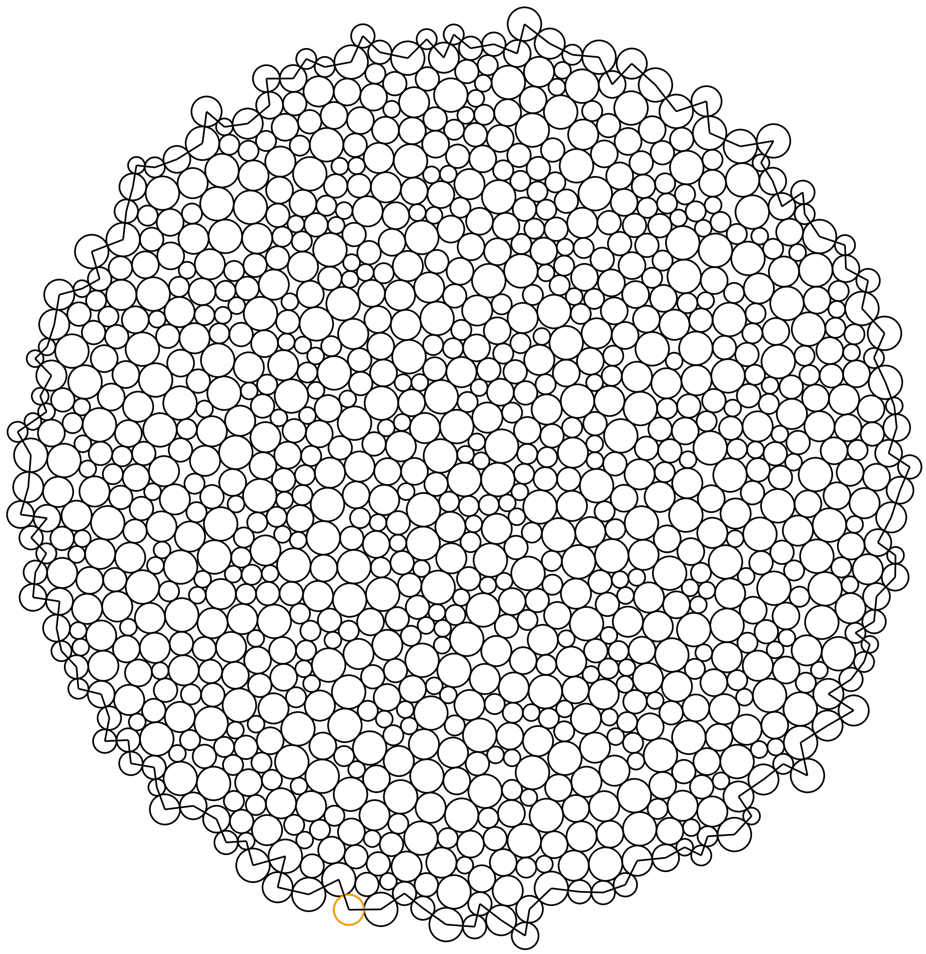
  ```
  python mpl_draw.py input_file output_file
  ```

  όπου `input_file` είναι το αρχείο των αποτελεσμάτων σας και `output_file` είναι ένα αρχείο, ο τύπος του οποίου εξαρτάται από την κατάληξή του. Για παράδειγμα, αν δώσετε ως `output_file` το `myfile.png` θα πάρετε αρχείο PNG, αν δώσετε ως `output_file` το `myfile.svg` θα πάρετε αρχείο SVG, αν δώσετε ως `output_file` το `myfile.pdf` θα πάρετε αρχείο PDF, κ.λπ. Για να λειτουργήσει, πρέπει να εγκαταστήσετε τη βιβλιοθήκη `matplotlib`.

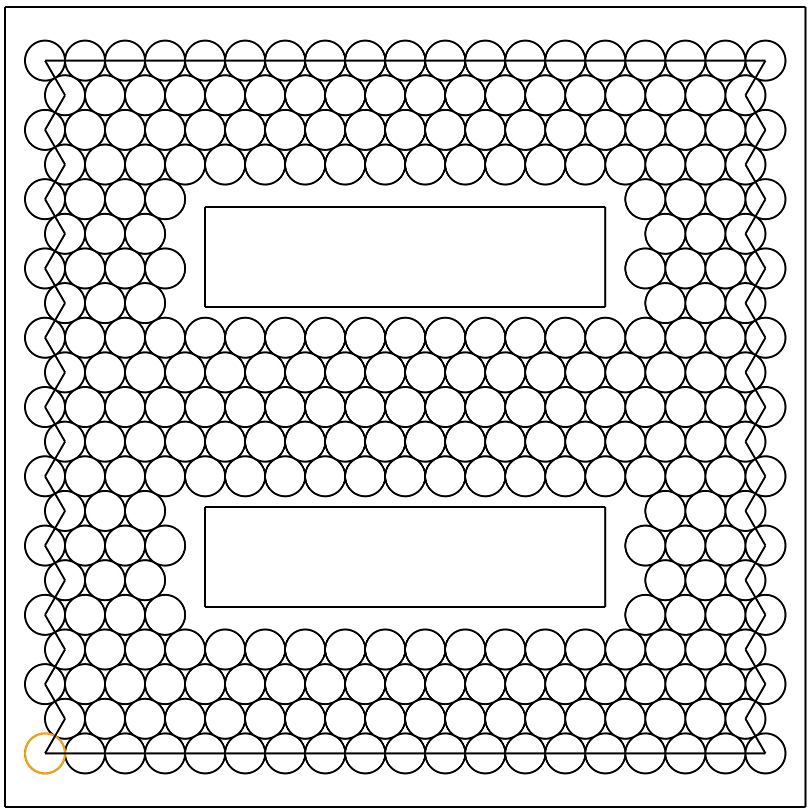
Παραδείγματα



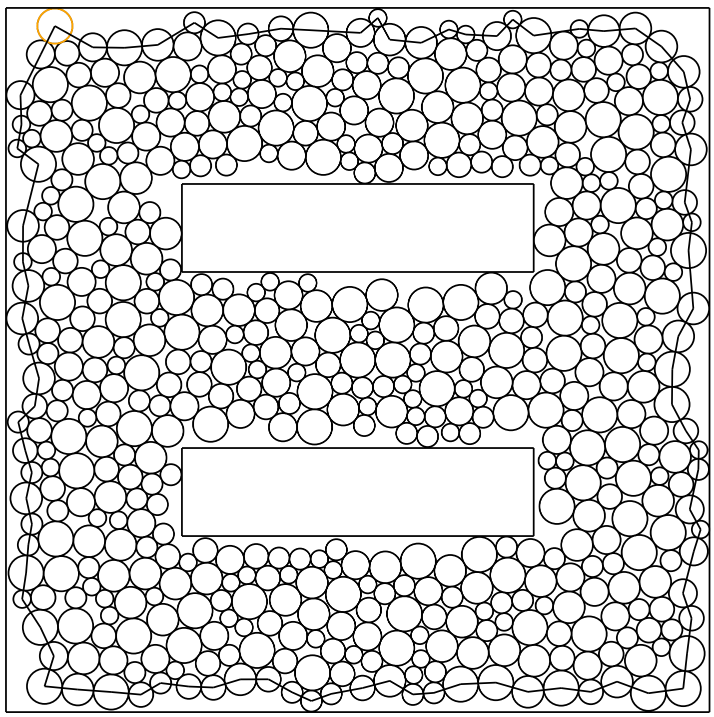
Σχήμα 7: Παράδειγμα με 1000 Ίσους Κύκλους



Σχήμα 8: Παράδειγμα με 1000 Τυχαίους Κύκλους



Σχήμα 9: Παράδειγμα με Ίσους Κύκλους σε Σχήμα



Σχήμα 10: Παράδειγμα με Τυχαίους Κύκλους σε Σχήμα