

28 / 4 / 2025

ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ

ΑΣΚΗΣΗ 1

**ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΣΑΚΙΔΙΟΥ
(KNAPSACK PROBLEM)**

ΕΡΓΑΣΤΗΡΙΑΚΗ ΕΡΓΑΣΙΑ

ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

2024-2025

Ηλίας Ξανθόπουλος 58545

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων.....	2
1. Εισαγωγή	3
2. Δημιουργία Αρχικού Πληθυσμού (<i>ksack_create_permutations.m</i>)	3
3. Εκτίμηση Καταλληλότητας (<i>ksack_fitness.m</i>)	4
4. Επιλογή Γονέων (<i>ksack_selection.m</i>)	4
5. Γενετική Διασταύρωση (<i>ksack_crossover_permutation.m</i>)	5
6. Γενετικές Μεταλλάξεις (<i>ksack_mutate_permutation.m</i>)	5
7. Απεικόνιση προόδου και αποτελεσμάτων (<i>ksack_plot.m</i>)	5
8. Κύριο πρόγραμμα (<i>ksack_ga.m</i>)	6
9. Σχολιασμός αποτελεσμάτων.....	6

Το παρόν report όπως και οι κώδικες της υλοποίησης είναι ανεβασμένα στο GitHub στο repository: https://github.com/iliaxant/GA_knapsack_problem

1. Εισαγωγή

Η άσκηση που επιλύεται με την χρήση Γενετικού Αλγορίθμου (ΓΑ) είναι η “*Άσκηση 1. Το πρόβλημα του σακιδίου (knapsack problem)*”. Η υλοποίηση αποτελείται από το δεδομένο *.xls* αρχείο *knapsack.xls*, το οποίο περιέχει τις προδιαγραφές του προβλήματος (π.χ. αξίες και βάρη), και από εφτά *.m* αρχεία MATLAB, εκ των οποίων το *ksack_ga.m* αποτελεί το κύριο script, ενώ τα υπόλοιπα περιλαμβάνουν τις βοηθητικές/custom συναρτήσεις. Στα επόμενα κεφάλαια αναλύεται ποιοτικά το περιεχόμενο των προαναφερθέντων αρχείων MATLAB περιγράφοντας έτσι ταυτόχρονα και την στρατηγική επίλυσης του προβλήματος με ΓΑ.

Σχόλιο: Στα αρχεία MATLAB των βοηθητικών συναρτήσεων μπορεί να περιλαμβάνονται μπλοκ κώδικα τοποθετημένα μέσα σε σχόλια. Αυτά αποτελούν διαφορετικές τακτικές επίλυσης οι οποίες αν και υλοποιήθηκαν και εξετάστηκαν, εν τέλει δεν αξιοποιήθηκαν στον τελικό (βέλτιστο) αλγόριθμο. Για πρακτικούς λόγους αυτοί οι τρόποι αντιμετώπισης δεν αναλύονται στα παρακάτω κεφάλαια αλλά υπάρχουν στα *.m* αρχεία για οποιονδήποτε ενδιαφερόμενο.

2. Δημιουργία

Αρχικού

Πληθυσμού

(*ksack_create_permutations.m*)

Η συνάρτηση αυτή αναλαμβάνει τον ορισμό της δομής των χρωμοσωμάτων και την δημιουργία ενός αρχικού πληθυσμού.

Μια πιθανή λύση του προβλήματος του σακιδίου είναι ένας συνδυασμός από ένα δεδομένο πλήθος αντικειμένων. Επομένως, καθεμία από αυτές τις λύσεις, δηλαδή τα χρωμοσώματα, μπορεί να κωδικοποιηθεί από ένα bitstring με μήκος τόσο, όσο το πλήθος των πιθανών αντικειμένων (στην συγκεκριμένη περίπτωση 100). Κάθε bit αυτής της ροής αντιπροσωπεύει ένα από τα πιθανά αντικείμενα και παίρνει την τιμή “1” όταν αυτό τοποθετείται μέσα στο σάκο, και την τιμή “0” όταν αυτό μένει εκτός του σάκου. Για παράδειγμα, μια πιθανή δομή είναι η *[1 0 0 0 1 1 0 ... 1 0 0 0 1 0 0]*.

Η τιμή του κάθε χρωμοσώματος του αρχικού πληθυσμού προκύπτει με εντελώς τυχαίο τρόπο, με το κάθε bit να έχει 50% πιθανότητα να αποκτήσει την τιμή “1” και 50% την τιμή “0”. Αυτή η τεχνική έχει ως συνέπεια καθεμία από τις αρχικές λύσεις να είναι “υπέρβαρα” (στατιστικά απίθανο να μην είναι), δηλαδή να αντιστοιχεί σε βάρος μεγαλύτερο από την αντοχή του σακιδίου. Ωστόσο αυτό είναι επιθυμητό, γιατί έτσι αποφεύγεται η πρόωρη σύγκλιση και επιτρέπεται η σταδιακή βελτίωση των λύσεων μέχρι να αποκτήσουν τα επιθυμητά όρια βάρους.

3. Εκτίμηση Καταλληλότητας (*ksack_fitness.m*)

Η συνάρτηση αυτή αναλαμβάνει την ανάθεση ενός σκορ σε χρωμοσώματα ανάλογα με την καταλληλότητα τους, δηλαδή με το πόσο ορθά επιλύουν το πρόβλημα.

Η καλύτερη λύση του προβλήματος έχει την μεγαλύτερη δυνατή συνολική αξία αντικειμένων, ενώ ταυτόχρονα υπακούει στο όριο βάρους του σάκου. Είναι απαγορευτικό το τελικό χρωμόσωμα να είναι “υπέρβαρο”, αλλά σίγουρα δεν είναι και απαραίτητο να εκμεταλλεύεται πλήρως την διαθέσιμη χωρητικότητα, εφόσον το επιτρέπουν τα διαθέσιμα αντικείμενα. Επομένως μια λύση είναι καλύτερη από τις άλλες, όταν έχει μεγαλύτερη αξία και ταυτόχρονα το βάρος της είναι χαμηλότερο από το μέγιστο επιτρεπτό βάρος του σάκου.

Για την επίτευξη τέτοια βαθμολόγησης ορίζεται η συνάρτηση fitness:

$$Fitness = \begin{cases} \frac{1}{total_value}, & \text{για } total_weight \leq max_weight \\ 100 * \frac{(total_weight - max_weight)}{total_value}, & \text{για } total_weight > max_weight \end{cases}$$

Η συνάρτηση αυτή αντιστοιχεί σε καλό χρωμόσωμα όταν είναι ελάχιστη. Είναι σχεδιασμένη έτσι ώστε να τιμωρεί βαριά τις “υπέρβαρες” λύσεις πολλαπλασιάζοντας με έναν συντελεστή που αυξάνει σημαντικά το σκορ. Ένα overweight χρωμόσωμα έχει χαμηλότερο fitness value από ένα άλλο “υπέρβαρο” όταν το βάρος του είναι πιο κοντά στην μέγιστη αντοχή του σάκου (με την υπόθεση ότι έχουν την ίδια αξία). Οι λύσεις κάτω από το μέγιστο βάρος επιβραβεύονται με χαμηλότερη τιμή fitness, αφού απουσιάζει ο συντελεστής penalty και το σκορ τους εξαρτάται μόνο από την αξία των αντικειμένων μέσα στο σάκο.

4. Επιλογή Γονέων (*ksack_selection.m*)

Η συνάρτηση αυτή αναλαμβάνει την επιλογή χρωμοσωμάτων σε κάθε γενιά που γίνονται γονείς και συμμετέχουν στην διαδικασία διασταύρωσης και μετάλλαξης.

Ως γονείς πρέπει κατά βάση να επιλέγονται τα χρωμοσώματα με την καλύτερη τιμή fitness, γιατί η αναπαραγωγή τους συνήθως οδηγεί σε επίσης καλές λύσεις. Ωστόσο, κάποιες φορές είναι αποτελεσματικότερο να αναπαράγονται και κάποιες μη ιδανικές λύσεις, διότι μέσω αυτών μπορεί να γεννηθούν ικανοποιητικοί απόγονοι που διαφορετικά δεν θα μπορούσαν να εμφανιστούν από τα τρέχοντα καλύτερα χρωμοσώματα.

Για αυτό λοιπόν, για την επιλογή γονέων, επιλέχθηκε η τεχνική της “ρουλέτας” η οποία προσδίδει μια πιθανότητα αναπαραγωγής σε κάθε χρωμόσωμα. Όσο μεγαλύτερο είναι το σκορ της κάθε λύσης, τόσο μεγαλύτερη και η πιθανότητα επιλογή της και το άθροισμα όλων αυτών των πιθανοτήτων είναι ίσο με την μονάδα. Έπειτα με τυχαίο

τρόπο, που ομοιάζει με μία “ρουλέτα” που περιστρέφεται, επιλέγονται επαναληπτικά οι γονείς μέχρι να επιλεγεί ένα συγκεκριμένος αριθμός τους.

5. Γενετική Διασταύρωση (*ksack_crossover_permutation.m*)

Η συνάρτηση αυτή αναλαμβάνει την αναπαραγωγή επιπλέον λύσεων αξιοποιώντας αυτούσια κομμάτια λύσεων της τρέχουσας γενιάς.

Στην συγκεκριμένη εφαρμογή, η συνάρτηση διασταύρωσης παίρνει κάθε φορά ένα ζευγάρι από τους γονείς που έχουν επιλεγεί να συμμετάσχουν στην διασταύρωση και σπάει τον καθένα σε δύο κομμάτια επιλέγοντας το ίδιο τυχαίο σημείο τομής και στους δύο. Έπειτα χρησιμοποιεί αυτούσια το πρώτο κομμάτι από το πρώτο χρωμόσωμα και το δεύτερο από το δεύτερο για να παράξει μία νέα λύση-παιδί.

6. Γενετικές Μεταλλάξεις (*ksack_mutate_permutation.m*)

Η συνάρτηση αυτή αναλαμβάνει την τοπική αλλαγή τιμών των χρωμοσωμάτων για την επίτευξη ποικιλομορφίας και την “απόδραση” από τα τοπικά ελάχιστα.

Στο συγκεκριμένο πρόβλημα, η μετάλλαξη αντιστοιχεί σε αντιστροφή μεμονωμένων bit. Η αντιστροφή γίνεται με τυχαίο τρόπο, αποδίδοντας σε κάθε στοιχείο του χρωμοσώματος μία πιθανότητα μετάλλαξης *mutationRate*, η τιμή της οποίας ορίζεται στο κύριο πρόγραμμα *ksack_ga.m*.

7. Απεικόνιση προόδου και αποτελεσμάτων (*ksack_plot.m*)

Αυτή η συνάρτηση αναλαμβάνει την εκτύπωση δύο γραφημάτων για την κατανόηση της συμπεριφοράς και της αποτελεσματικότητας του γενετικού αλγορίθμου.

Το πρώτο προβάλλεται και ενημερώνεται καθόλη την διάρκεια εκτέλεσης του αλγορίθμου και απεικονίζει το καλύτερο, το χειρότερο και το μέσο fitness value με το πέρασμα των γενεών. Χρησιμοποιείται κυρίως για να παρατηρείται η συγκλίνση του ΓΑ σε ένα ελάχιστο σκορ και της ταχύτητας σύγκλισης προς αυτή την ιδανική λύση.

Το δεύτερο γράφημα είναι ένα heatmap που παράγεται στο τέλος της εκτέλεσης του ΓΑ και προβάλλει τον καλύτερο συνδυασμό αντικειμένων, δηλαδή το καλύτερο χρωμόσωμα, της κάθε γενιάς. Σκοπός του σχεδιαγράμματος αυτού είναι η οπτικοποίηση της πορείας του αλγορίθμου προς την βέλτιστη λύση.

8. Κύριο πρόγραμμα (*ksack_ga.m*)

Το *ksack_ga.m* αποτελεί τον εγκέφαλο της υλοποίησης και χωρίζεται σε 4 μέρη:

a. Εξαγωγή δεδομένων και αρχικοποιήσεις:

Η ενότητα αυτή αρχικοποιεί την γεννήτρια τυχαίων αριθμών (για την αναπαραγωγή συγκεκριμένων αποτελεσμάτων ή τυχαίων), εξάγει τα χρήσιμα δεδομένα του αρχείου *knapsack.xls* και τυπώνει σημαντικές πληροφορίες για το πρόβλημα.

b. Ρύθμιση γενετικού αλγορίθμου:

Το μέρος αυτό είναι υπεύθυνο για τον ρύθμιση του ΓΑ και αποτελεί το σημαντικότερο κομμάτι του κώδικα. Σε αυτό ορίζεται ότι οι συναρτήσεις που θα χρησιμοποιηθούν για τις χαρακτηριστικές λειτουργίες του αλγορίθμου είναι αυτές που αναλύθηκαν στα προηγούμενα κεφάλαια, αλλά και καθορίζονται οι τιμές διαφόρων παραμέτρων που αφορούν την εκτέλεση, όπως είναι για παράδειγμα ο μέγιστος αριθμός γενεών. Αυτές οι παράμετροι ρυθμίστηκαν εμπειρικά και δεν χρήζουν περαιτέρω εξήγηση, αλλά δύο από αυτές αξίζουν να σχολιαστούν.

Το *mutationRate* έχει οριστεί να είναι πάρα πολύ μικρό (5%), γιατί οποιαδήποτε αντιστροφή bit επηρεάζει σημαντικά το συνολικό βάρος του σάκου και συνεπώς το *fitness score*. Επομένως, πρέπει οι αλλαγές αυτές να γίνονται όχι τόσο συχνά, έτσι ώστε το βάρος να μεταβάλλεται σταδιακά.

Παρομοίως το *crossover_fraction*, δηλαδή το ποσοστό των γονέων που θα διασταυρωθούν αντί να μεταλλαχθούν, είναι ρυθμισμένο στο σχετικά χαμηλό ποσοστό των 20%, προκειμένου να δίνεται μεγαλύτερη βαρύτητα στις σταδιακές ρυθμίσεις βάρους της μετάλλαξης και όχι τόσο στις απότομες αλλαγές της διασταύρωσης.

c. Εκτέλεση γενετικού αλγορίθμου:

Σε αυτό το block εκτελείται ο ΓΑ, με βάση τις ρυθμίσεις του προηγούμενου τμήματος, όπως ακριβώς αναλύθηκε στα προηγούμενα κεφάλαια. Στο τέλος της κάθε εκτέλεσης, εμφανίζονται λεπτομέρειες που αφορούν το συγκεκριμένο run, όπως είναι η συνθήκη τερματισμού που συναντήθηκε, ο χρόνος εκτέλεσης κτλ.

d. Λεπτομέρειες βέλτιστης λύσης:

Με το τελευταίο κομμάτι του κύριου προγράμματος τυπώνονται όλες οι λεπτομέρειες της βέλτιστης λύσης που υπολόγισε στο προηγούμενο βήμα ο ΓΑ, όπως είναι τα βάρη και οι αξίες των επιλεγμένων αντικειμένων.

9. Σχολιασμός αποτελεσμάτων

Εκτελώντας πολλές φορές τον κώδικα του αρχείου *ksack_ga.m* γίνεται φανερό ότι ο συγκεκριμένος ΓΑ συγκλίνει πολύ κοντά στην βέλτιστη λύση, με το τελικό χρωμόσωμα να μην ξεπερνάει ποτέ την αντοχή βάρους του σάκου. Μάλιστα σε ένα αρκετά ικανοποιητικό ποσοστό εκτελέσεων, η τελική λύση αντιστοιχεί στην βέλτιστη αξία των 1173, ενώ οι μη βέλτιστες κυμαίνονται λίγο πιο κάτω από αυτήν. Η αδυναμία του

αλγορίθμου να μην συγκλίνει πάντοτε στην βέλτιστη λύση είναι αναμενόμενη, καθώς τα βάρη και οι αξίες όλων των διαθέσιμων αντικειμένων είναι πολύ κοντά μεταξύ τους, με αποτέλεσμα στο τέλος να είναι πολύ δύσκολο να επιτευχθεί μία καλύτερη λύση. Αξιίζει όμως να σημειωθεί ότι όλα αυτά ισχύουν έχοντας χρησιμοποιήσει τα προτεινόμενα στο αρχείο πλήθη γενεών και συνθηκών τερματισμού, τα οποία έχουν επιλεγεί έτσι ώστε ο χρόνος εκτέλεσης να είναι σχετικά μικρός. Εάν κάποιος επιλέξει να τρέξει τον αλγόριθμο για πολύ περισσότερο χρόνο και με πολύ πιο αυστηρές συνθήκες τερματισμού, είναι πολύ πιθανόν να υπάρξει πάντοτε σύγκλιση προς τον καλύτερο συνδυασμό αντικειμένων, απλώς όχι σε σύντομο χρονικό διάστημα.