ΠΛΗ417 – Τεχνητή Νοημοσύνη
1η Προγραμματιστική εργασία – Μέρος Β

LAB41744570

Μανώλης Πετράκος ΑΜ 2014030009

Για την υλοποίηση του δεύτερου μέρους έχουν δημιουργηθεί τρία αρχεία. Στο πρώτο (WHPP.java) υπάρχει η main και ο σκελετός του γενετικού αλγορίθμου, στο δεύτερο (Population.java) υλοποιείται η κλάση του πληθυσμού με τις μεθόδους επιλογής χρωμοσωμάτων και τερματισμού, και στο τρίτο (Chromosome.java) υλοποιείται η κλάση των χρωμοσωμάτων με τις μεθόδους ελέγχου των constraints, διασταύρωσης και μετάλλαξης. Έχουν κατασκευαστεί περισσότερες μέθοδοι από όσες ζητούνται. Στην αναφορά επεξηγούνται κυρίως αυτές που χρησιμοποιήθηκαν για τις πειραματικές μετρήσεις. Τα κομμάτια του γενετικού αλγορίθμου έχουν κατασκευαστεί με γενικό τρόπο, ώστε να μπορούν να χρησιμοποιηθούν με οποιαδήποτε άλλα, και η υλοποίηση κάθε μεθόδου είναι ανεξάρτητη από αυτές που χρησιμοποιείται μαζί. Για παράδειγμα, οι μέθοδοι επιλογής γονέων μπορούν να επιστρέψουν οποιοδήποτε αριθμό γονέων, ενώ όλες οι μέθοδοι ζευγαρώματος που υλοποιήθηκαν χρειάζονται ακριβώς δύο.

Επεξήγηση προσέγγισης

Δημιουργία αρχικού πληθυσμού:

Ο αρχικός πληθυσμός δημιουργείται εισάγοντας του feasible τυχαία χρωμοσώματα. Η πρώτη υλοποίηση για την δημιουργία ενός χρωμοσώματος είναι να κατασκευαστεί με τυχαίες τιμές. Σχεδόν όλα από αυτά που θα κατασκευαστούν με αυτόν τον τρόπο, δεν είναι feasible λόγο των αυστηρών hard constraints. Για αυτό, χρησιμοποιείται η μέθοδος *FeasibleRandomInit*. Κατασκευάζει πρώτα ένα ντετερμινιστικό πίνακα που καλύπτει ακριβώς όλες τις βάρδιες και μετά ανακατεύει τα στοιχεία των στηλών του. Έτσι το πλήθος κάθε βάρδιας ανά μέρα παραμένει ίδιο και δεν παραβιάζονται τα hard constraints. Δηλαδή, η τιμή κάθε θέσης του πίνακα είναι τυχαία, αλλά το πλήθος κάθε τιμής ανά στήλη είναι ντετερμινιστικό. Με αυτό τον τρόπο μπορούν να δημιουργηθούν όλα τα πιθανά feasible χρωμοσώματα και ο πληθυσμός έχει υψηλό diversity.

Συνάρτηση καταλληλότητας:

Για την αξιολόγηση των χρωμοσωμάτων χρησιμοποιείται η μέθοδος *CalculateScore* η οποία υπολογίζει το score τους ανάλογα τα soft constraint που παραβιάζουν. Κάθε constraint μπορεί να ελέγχεται μια φορά για όλους τους εργαζόμενους, πολλές φορές ανά εργαζόμενο ή μία φορά ανά εργαζόμενο.

Στην πρώτη περίπτωση, αν ένα constraint ελέγχεται μία φορά για όλο τον πίνακα, ένα χρωμόσωμα που το παραβιάζει για έναν εργαζόμενο έχει το ίδιο score με ένα άλλο που το παραβιάζει για όλους τους εργαζόμενους. Το πρώτο είναι πιο κοντά σε μια καλή λύση, αλλά έχουν την ίδια πιθανότητα να παράξουν απογόνους. Επίσης, το διάστημα των τιμών που μπορεί να πάρει το score είναι αρκετά μικρό, με αποτέλεσμα να γίνεται ακόμα πιο δύσκολη η επιλογή των κατάλληλων χρωμοσωμάτων.

Στην δεύτερη περίπτωση, αν παραβιαστεί πολλές φορές ένα constraint σε έναν εργαζόμενο, μπορεί να εκπροσωπηθεί δυσανάλογα στο τελικό σκορ. Ένα χρωμόσωμα που χρειάζεται αλλαγές μόνο σε μια γραμμή μπορεί να έχει χειρότερο σκορ από ένα άλλο που χρειάζεται αλλαγές σε πολλές γραμμές.

Στην τελευταία περίπτωση, όπου κάθε constraint υπολογίζεται μια φορά ανά εργαζόμενο, το σκορ έχει αρκετό range για να ξεχωρίζουν τα χρωμοσώματα, αλλά δεν υπερεκπροσωπούνται ακραίες σειρές γονιδίων. Για αυτούς τους λόγους, επιλέχθηκε αυτή η μέθοδος αξιολόγησης των χρωμοσωμάτων.

Επιλογή χρωμοσωμάτων:

Η επιλογή των χρωμοσωμάτων για αναπαραγωγή γίνεται με τον αλγόριθμο του tournament selection. Το μέγεθος του tournament ορίζεται από την πιθανότητα επιλογής ενός χρωμοσώματος επί το μέγεθος του πληθυσμού. Το tournament γεμίζει με τυχαία χρωμοσώματα και επιλέγεται αυτό με το καλύτερο σκορ. Αυτή η διαδικασία συνεχίζεται μέχρι να επιλεχθούν όσα χρωμοσώματα χρειάζονται. Ένα χρωμόσωμα δεν μπορεί να πάρει πάνω από μια θέση σε κάθε tournament ή να επιλεχθεί πάνω από μια φορά για την ίδια αναπαραγωγή, αλλά είναι δυνατόν να λάβει θέση σε πολλά tournament ή να επιλεχθεί για πολλές διαφορετικές αναπαραγωγές. Τέλος, επιλέχθηκε αυτός ο αλγόριθμος γιατί μπορεί εύκολα να λειτουργήσει τυχαιοκρατικά ή ελιτιστικά αλλάζοντας την πιθανότητα επιλογής.

Διασταύρωση χρωμοσωμάτων:

Η πρώτη μέθοδος διασταύρωσης είναι η *MeritCollumnCrosover*. Οι στήλες γονιδίων του παραγόμενου χρωμοσώματος, μέχρι το crossover point είναι αντίγραφα από αυτές του πρώτου γονέα, ενώ οι υπόλοιπες από αυτές του δεύτερου. Η μέθοδος είναι προκατειλημμένη με γραμμική κατανομή υπέρ του γονέα με το καλύτερο σκορ. Αν η σχετική διαφορά τους είναι μικρή (>~5%) έχουν την ίδια συμμετοχή ενώ αν είναι πολύ μεγάλη (>~95%) το παραγόμενο χρωμόσωμα είναι αντίγραφο του καλύτερου γονέα. Η γενική ιδέα της μεθόδου είναι να εξαλείψει πλήρως χρωμοσώματα με πολύ κακό σκορ καθώς και να υπάρχει ελιτισμός υπέρ των καλύτερων.

Η δεύτερη μέθοδος διασταύρωσης είναι η *RandomColumnCrossing*. Το νέο χρωμόσωμα κατασκευάζεται επιλέγοντας τυχαία από τις αντίστοιχες στήλες των γονέων. Η πιθανότητα επιλογής από τον καλύτερο γονέα είναι μεταβαλλόμενη και η μέθοδος μπορεί να γίνει προκατειλημμένη υπέρ ή κατά του, ανάλογα την μεταβλητή pcross που δέχεται.

Και οι δύο μέθοδοι εγγυούνται ότι το νέο χρωμόσωμα είναι feasible καθώς είναι κατασκευασμένο από στήλες feasible χρωμοσωμάτων και δεν μεταβάλουν τα πλήθη των στοιχείων τους. Έγινε δοκιμή με

μια μέθοδο διασταύρωσης που δεν είναι βάση στηλών (UniformCrossing), αλλά κάτω από το ένα εκατομμυριοστό των παραγόμενων χρωμοσωμάτων ήταν feasible. Λόγο αυστηρότητας των hard-constraints, δεν υπάρχει ιδιαίτερο νόημα να χρησιμοποιηθούν μέθοδοι που δεν βασίζονται σε πράξεις στηλών.

Μετάλλαξη χρωμοσωμάτων:

Η πρώτη μέθοδος μετάλλαξης είναι η *ColumnInversionMutation*, η οποία αναποδογυρίζει κομμάτια στηλών γονιδίων. Ο αριθμός των γονιδίων που θα μεταλλαχτούν ορίζεται από την πιθανότητα μετάλλαξης ενός γονιδίου (pmut) επί τον αριθμό γονιδίων και σε μία μετάλλαξη μπορούν να αναποδογυρίσουν πολλά διαφορετικά κομμάτια στηλών. Ο αλγόριθμος επιλέγει τυχαία ποια γονίδια θα μεταλλάξει και εγγυάται feasibility, καθώς μεταβάλλει μόνο την θέση των βαρδιών μέσα στην μέρα και δεν επηρεάζει τα πλήθη τους.

Η δεύτερη μέθοδος μετάλλαξης είναι η *SwapMutation*, η οποία ανταλλάζει τις τιμές δύο θέσεων μέσα στην ίδια στήλη γονιδίων. Τα γονίδια που μεταλλάζονται είναι τυχαία και ο αριθμός τους είναι ίσως με την πιθανότητα μετάλλαξης ενός γονιδίου (pmut) επί τον αριθμό γονιδίων. Ο αλγόριθμος εγγυάται feasibility καθώς τα πλήθη των βαρδιών είναι σταθερά. Δεν υπάρχει νόημα να γίνει ανταλλαγή μεταξύ γονιδίων διαφορετικών στηλών, καθώς είτε θα έχουν την ίδια τιμή και δεν θα υπάρξει κάποια ουσιαστική αλλαγή, είτε θα αλλάξει το πλήθος κάποιας βάρδιας σε μια μέρα και το παραγόμενο χρωμόσωμα θα είναι μη feasible.

Κριτήρια τερματισμού:

Τα κριτήρια τερματισμού είναι δύο:

- Βρέθηκε το βέλτιστο χρωμόσωμα με σκορ 0.
- Δημιουργήθηκαν γενιές ίσο με το όριο τους.

Αρχικά υλοποιήθηκαν δύο ακόμα κριτήρια αλλά απορρίφθηκαν. Το πρώτο είναι να σταματάει ο αλγόριθμος, εάν το σκορ του καλύτερου χρωμοσώματος ή ο μέσος όρος γενιάς είχε πολύ μικρή διαφορά από την γενιά με αριθμό 1/10 αυτής. Για παράδειγμα, η γενιά 100 συγκρίνεται με την γενιά 10, η γενιά 1000 με την γενιά 100 κλπ. Τότε, εμφανίζεται το πρόβλημα ότι μια γενιά μπορεί να έχει χειρότερα στατιστικά από τις γύρω τις, επειδή φεύγει από τοπικό ελάχιστο, και να ενεργοποιήσει το κριτήριο ενώ υπάρχει περιθώριο για βελτίωση. Το δεύτερο κριτήριο είναι να σταματάει ο αλγόριθμος, εάν για το τελευταίο 10% των γενεών δεν υπάρχει σημαντική μεταβολή των στατιστικών τους. Αυτό το κριτήριο σπάνια ενεργοποιείται καθώς μπορούν να υπάρξουν απότομες μεταβολές όταν ο αλγόριθμος προσπαθεί να φύγει από τοπικά ελάχιστα. Τα παραπάνω προβλήματα γίνονται ιδιαίτερα αισθητά όταν υπάρχει υψηλή πιθανότητα μετάλλαξης, άρα και μεγάλες διαφορές μεταξύ γενεών. Τα κριτήρια πρέπει να είναι εξίσου αποτελεσματικά ανεξαρτήτου παραμέτρων και για αυτό απορρίφθηκαν.

Πειραματική Διαδικασία

Επιλέχθηκαν τα παρακάτω σετ τιμών:

Set	pop	$iter_{max}$	p_{sel}	p_{cross}	p_{mut}
A	1500	500	0.022	0.5	0.028
B	2000	300	0.04	0.6	0.05

Το πρώτο σετ είναι σχετικά τυχαιοκρατικό, καθώς με την μικρή πιθανότητα επιλογής στο tournament συμμετέχουν μόλις 33 χρωμοσώματα και κατά την αναπαραγωγή δεν έχει bias υπέρ των καλύτερων. Αν μειωθεί παραπάνω η πιθανότητα επιλογής, ο αλγόριθμος γίνεται πολύ τυχαίος, με αποτέλεσμα να μην συγκλίνει πάντα σε καλύτερη λύση. Τέλος, έχει μικρή πιθανότητα μετάλλαξης που αν μειωθεί και άλλο ο αλγόριθμος συγκλίνει γρήγορα σε τοπικό ελάχιστο. Βασίζεται στην τυχαιότητα για την διατήρηση της ποικιλομορφίας.

Αντίθετα, το δεύτερο σετ είναι πιο ελιτιστικό καθώς έχει μεγαλύτερη πιθανότητα επιλογής, με 80 χρωμοσώματα να λαμβάνουν μέρος σε ένα tournament. Επίσης, κατά την αναπαραγωγή, είναι biased υπέρ των καλύτερων. Προσπαθεί να διατηρήσει ποικιλομορφία μέσω μεγαλύτερου μεγέθους πληθυσμού και πιθανότητα μετάλλαξης. Λόγο του ελιτισμού, ο αλγόριθμος συγκλίνει γρηγορότερα σε ένα τοπικό ελάχιστο και χρειάζεται λιγότερες επαναλήψεις. Σε περίπτωση που αυξηθούν και άλλο οι παράμετροι της επιλογής και της αναπαραγωγής, ο αλγόριθμος γίνεται υπερβολικά ελιτίστικος με αποτέλεσμα να φτάνει σε τοπικό ελάχιστο μέσα σε λίγες γενιές. Επίσης, αν αυξηθεί περισσότερο η πιθανότητα μετάλλαξης, εμφανίζονται μεγάλες διαφορές από γενιά σε γενιά και ο αλγόριθμος δεν μπορεί να κατασταλάξει σε ένα ελάχιστο.

Τα παραπάνω σετ επιλέχθηκαν γιατί παρουσιάζουν τα δύο άκρα της λειτουργίας του αλγόριθμου, με το πρώτο λειτουργεί τυχαιοκρατικά και με το δεύτερο ελιτιστικά.

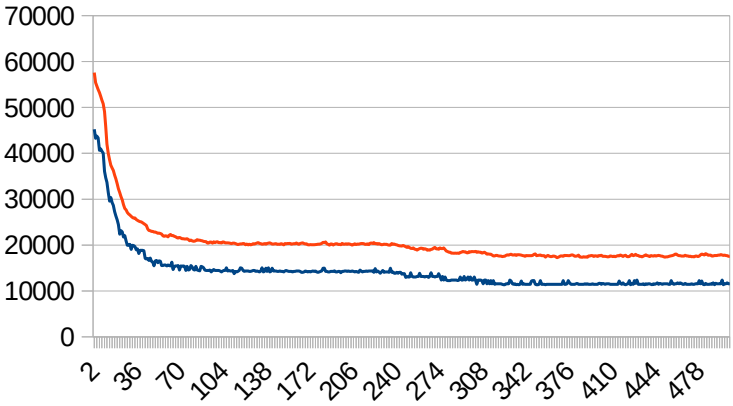
Μετρήσεις:
Μετά από μια σειρά μετρήσεων παρήχθησαν τα παρακάτω αποτελέσματα. Στην πρώτη γραμμή αναγράφεται ο συνδυασμός μεθόδων και στις υπόλοιπες το καλύτερο σκορ και ο μέσος όρος κάθε μίας.

Set	Random Inversion		Random Swap		Merit Inversion		Merit Swap	
A	11639	17767	11063	18289	11801	17991	11485	17687
B	14766	23418	19943	29038	15526	23940	19745	27576

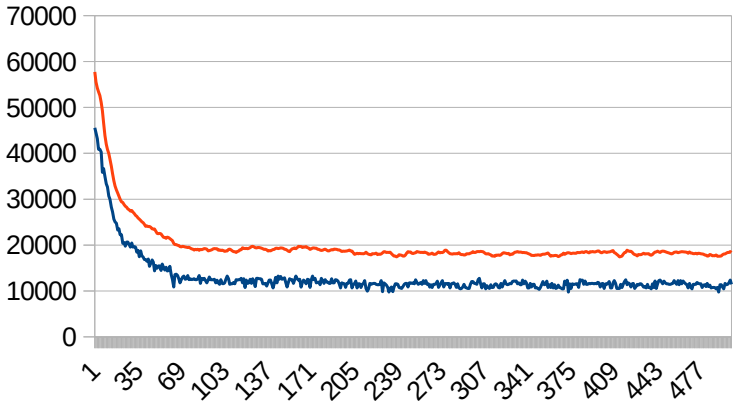
Καλύτερο σκορ αρχικού πληθυσμού: ~45000
Μέσος Όρος αρχικού πληθυσμού: ~57500
Ανεξάρτητα Υλοποίησης.

Στα παρακάτω διαγράμματα φαίνεται η εξέλιξη του μέσου όρου (κόκκινο) και του καλύτερου σκορ (μπλε) ανά γενιά για κάθε συνδυασμό σετ και μεθόδων.

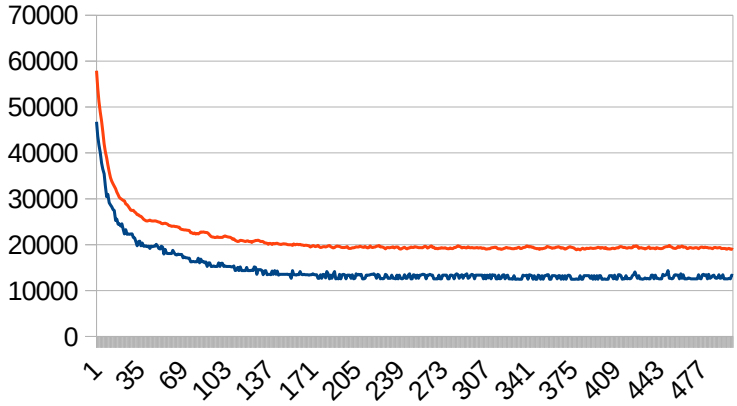
A / Random / Inversion



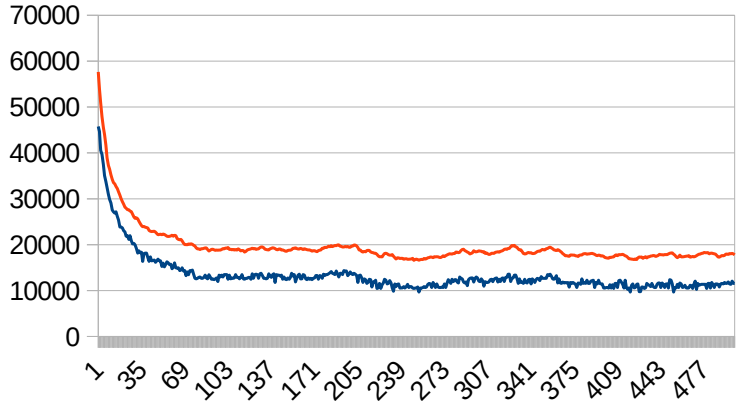
A / Random / Swap



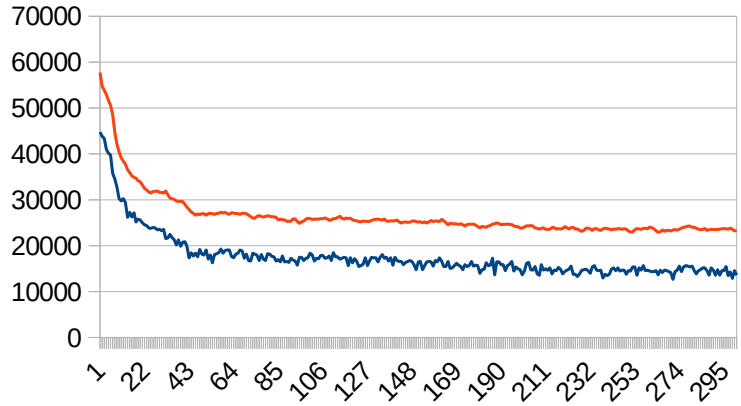
A / Merit / Inversion



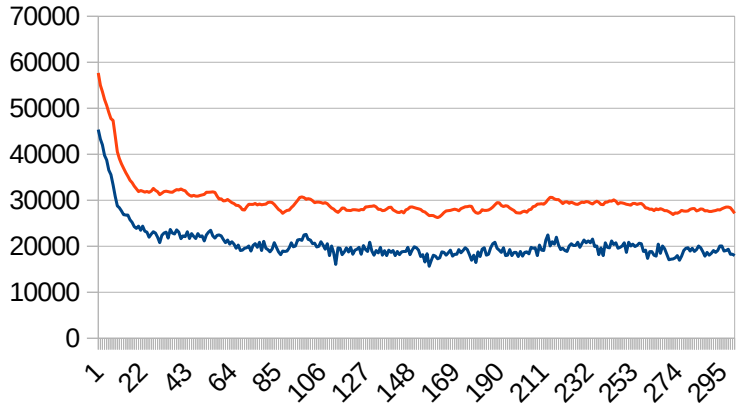
A / Merit / Swap



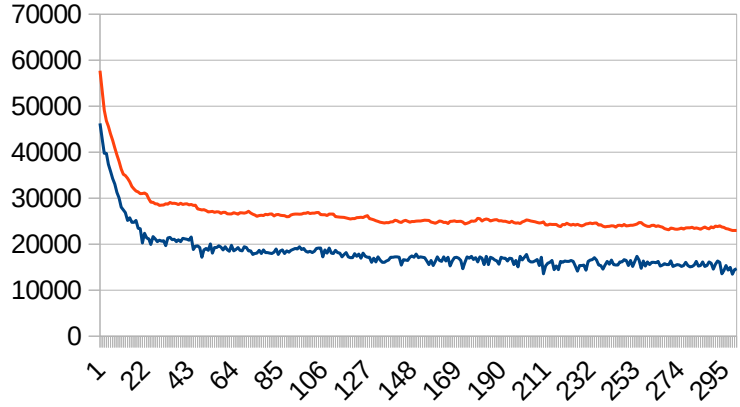
B / Random / Inversion



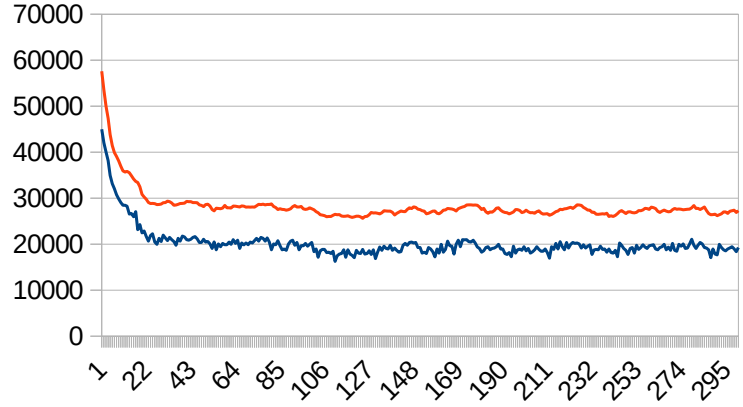
B / Random / Swap



B / Merit / Inversion



B / Merit / Swap



Συμπεράσματα:

Από τις μετρήσεις παράγονται τα παρακάτω συμπεράσματα:

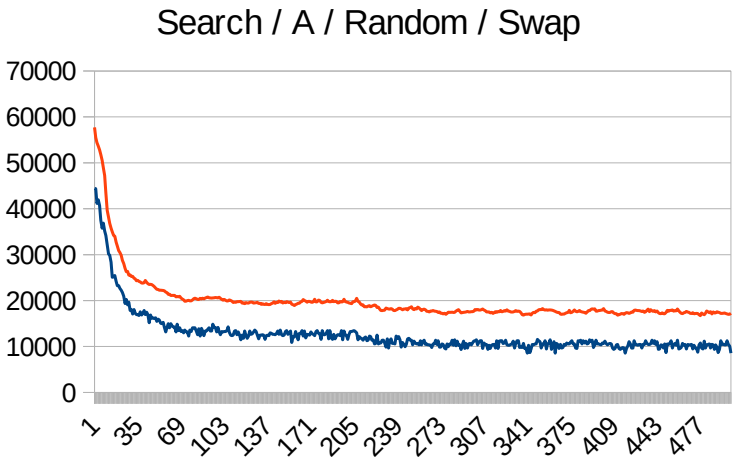
- Όπως ήταν αναμενόμενο, με το ελιτιστικό σετ παραμέτρων, ο αλγόριθμος χρειάζεται λιγότερες επαναλήψεις για να συγκλίνει.
- Επίσης, είναι αρκετά πιο αποδοτικός με το τυχαιοκρατικό σετ με όλους τους συνδυασμούς μεθόδων. Η διαφορά μεταξύ των σετ μεγιστοποιείται με τον συνδυασμό *RandomColumnCrossing*/ *InversionMutation* όπου για το Α εμφανίζεται το καλύτερο σκορ, ενώ για το Β το χειρότερο.
- Η μετάλλαξη swap προκαλεί μεγαλύτερες διαφορές ανά γενιά. Σε αυτό πιθανώς ευθύνεται ότι έχει μεγαλύτερο εύρος παραγόμενων χρωμοσωμάτων από την μετάλλαξη column inversion. Αυτή, όταν μεταφέρει μια τιμή, είναι πολύ πιθανό να μεταφέρει και τις γειτονικές της, με αποτέλεσμα πιο “προβλέψιμη” και ομαλή. Αυτό δεν ισχύει στην swap, οι μετακινήσεις είναι ανεξάρτητες από τις υπόλοιπες και σαν αποτέλεσμα είναι πιο volatile.
- Η μέθοδος *MeritColumnCrossing* τείνει να έχει χειρότερα αποτελέσματα και να συγκλίνει πιο νωρίς, ανεξάρτητα από την παράμετρο διασταύρωσης, καθώς είναι εκ κατασκευής biased. Θα μπορούσε να βελτιωθεί αν είχε παραπάνω από ένα crossing points ή αν λάμβανε υπόψιν την παράμετρο διασταύρωσης στην γραμμικότητας της.
- Τέλος, στο σετ Β υπάρχει μεγαλύτερη διαφορά μεταξύ μέσου όρου και καλύτερου σκορ, δηλαδή έχει υψηλότερη διασπορά. Για αυτό ευθύνεται η μεγάλη πιθανότητα μετάλλαξης.
- Με όλους τους συνδυασμούς δεν εξαφανίζεται η ποικιλομορφία, καθώς ο μέσος όρος δεν συγκλίνει στην καλύτερη λύση.

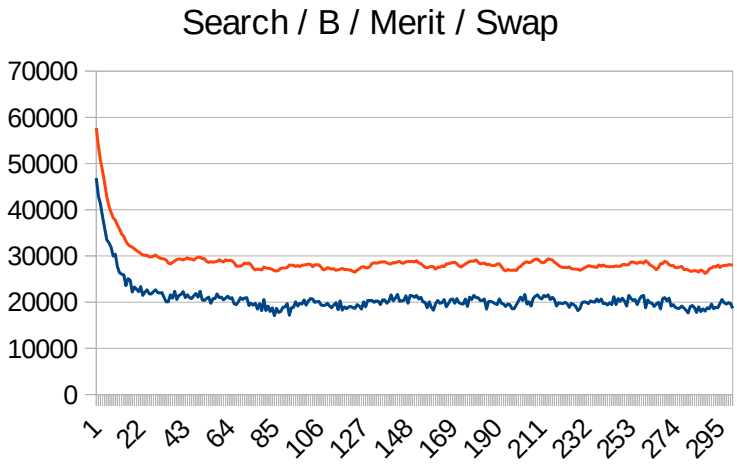
Μέθοδος Τοπικής Αναζήτησης

Ο γενετικός αλγόριθμος μπορεί να επεκταθεί σε memetic αλγόριθμο χρησιμοποιώντας μια μέθοδο τοπικής αναζήτησης μετά το στάδιο της μετάλαξης. Για αυτό δημιουργήθηκε η μέθοδος *LocalSearch* όπου υλοποιεί στοχαστική αναζήτηση στο πίνακα των γονιδίων. Προσπαθεί να ανταλλάξει την τιμή ενός γονιδίου με την τιμή όλων των επόμενων γονιδίων και κρατάει τις αλλαγές που καταλήγουν σε καλύτερο σκορ. Σκοπός της αναζήτησης είναι να ξεφύγει ο αλγόριθμος από τοπικά ελάχιστα και όχι να βρει την βέλτιστη λύση. Για αυτό, οι ανταλλαγές γίνονται μόνο προς τα μπρος και δεν λαμβάνει υπόψιν τους προηγούμενους γείτονες.

Για τον έλεγχο της αναζήτησης εκτελέστηκε με πιθανότητα 0.003 μαζί με τους συνδυασμούς A / Random / Inversion και B / Merit / Swap. Εμφανίστηκαν τα παρακάτω αποτελέσματα:

Search / A / Random / Inversion		Search / B / Merit / Swap	
8566	17263	17869	27487





Από τις παραπάνω μετρήσεις φαίνεται ότι η μέθοδος τοπικής αναζήτησης μπορεί να λειτουργήσει ευνοϊκά στην εύρεση καλύτερης λύσης. Σε όλες τις μετρήσεις χωρίς αναζήτηση, δεν εμφανίστηκε σκορ κάτω τον 9000. Αντίθετα, στην πρώτη μέτρηση με αναζήτηση, αυτό εμφανίστηκε χωρίς να μειωθεί ο μέσος όρος, δηλαδή το εύρος τιμών.

Με τον δεύτερο συνδυασμό δεν εμφανίζεται ουσιώδης βελτίωση. Η μεγάλη πιθανότητα μετάλλαξης και το υψηλό επίπεδο ελιτισμού φαίνεται να υπερκαλύπτουν οποιαδήποτε θετική αλλαγή προκαλεί η αναζήτηση.

Πηγές

Σκελετός γενετικού αλγορίθμου

Μέθοδοι επιλογής

Memetic Algorith