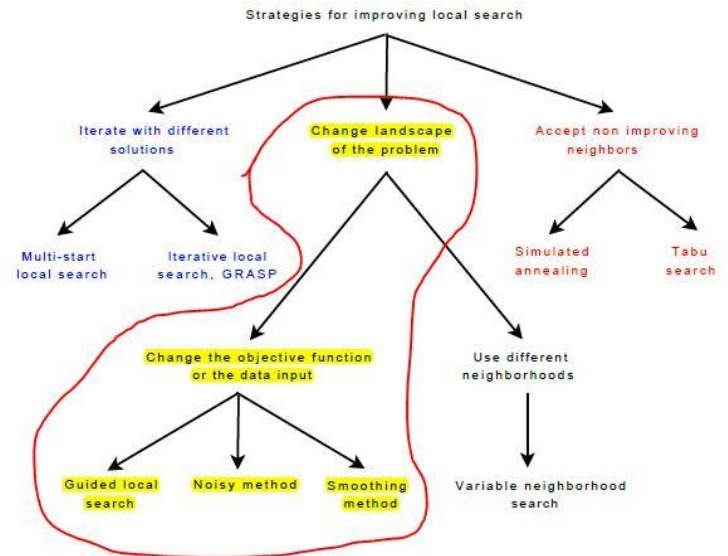


Διάλεξη 11

Τοπική Αναζήτηση και Βελτίωση Λύσεων

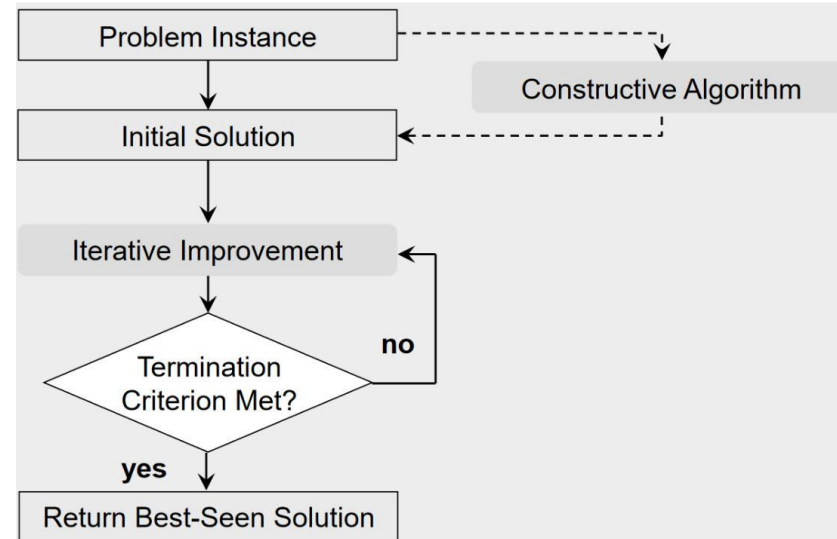
1. Από την Κατασκευή στη Βελτίωση

- Στη Διάλεξη 10:
 - κατασκευάσαμε αρχικές λύσεις
 - χρησιμοποιήσαμε απλές επιλογές
 - δημιουργήσαμε λύσεις βήμα προς βήμα
- Όμως:
 - μία αρχική λύση δεν είναι πάντα αρκετά καλή
- Το νέο ερώτημα:
 - πώς βελτιώνουμε μία υπάρχουσα λύση;



2. Τι Θα Έκανε Ένα Λογισμικό

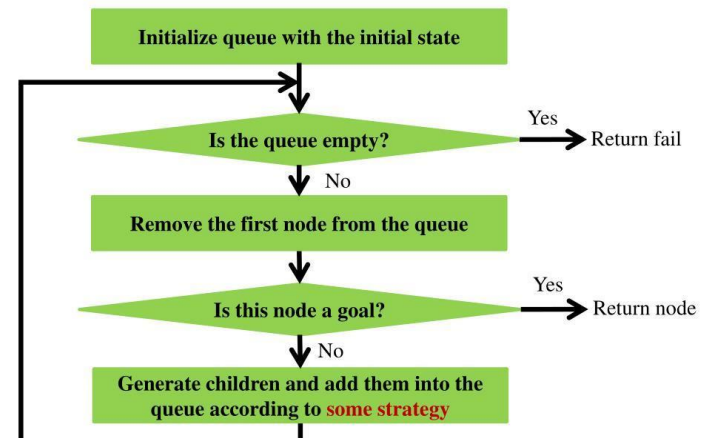
- Αν υλοποιούσαμε τη μέθοδο σε λογισμικό:
 - το πρόγραμμα δεν θα “μαντεύει” τη λύση
- Θα χρειαζόταν:
 - μία αρχική εφικτή λύση
 - έναν τρόπο υπολογισμού της ποιότητας
 - έναν τρόπο δημιουργίας μικρών αλλαγών
 - έναν κανόνα αποδοχής ή απόρριψης



3. Η Βασική Ιδέα της Τοπικής Αναζήτησης

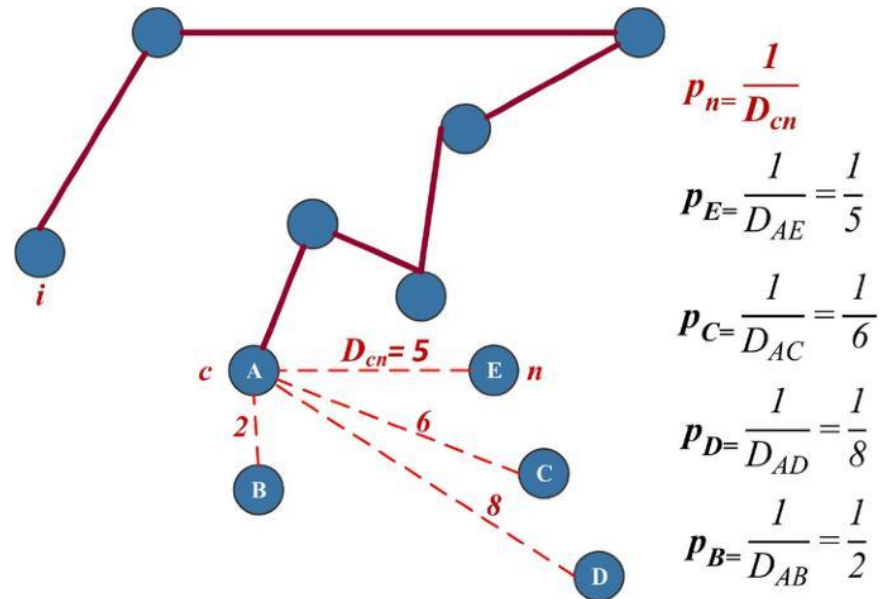
- Τοπική αναζήτηση:
 - μέθοδος βελτίωσης μιας υπάρχουσας λύσης
- Ξεκινάμε με:
 - μία αρχική λύση S
- Δημιουργούμε:
 - μία κοντινή λύση S'
- Αν η νέα λύση είναι καλύτερη:
 - αντικαθιστά την παλιά

Flowchart of search algorithms



4. Βασικός Βρόχος Τοπικής Αναζήτησης

- Αρχική λύση: S
- Υπολογισμός ποιότητας: $f(S)$
- Δημιουργία γειτονικής λύσης: S'
- Υπολογισμός ποιότητας: $f(S')$
- Αν $f(S')$ είναι καλύτερο:
 - $S \leftarrow S'$
- Διαφορετικά:
 - η αλλαγή απορρίπτεται



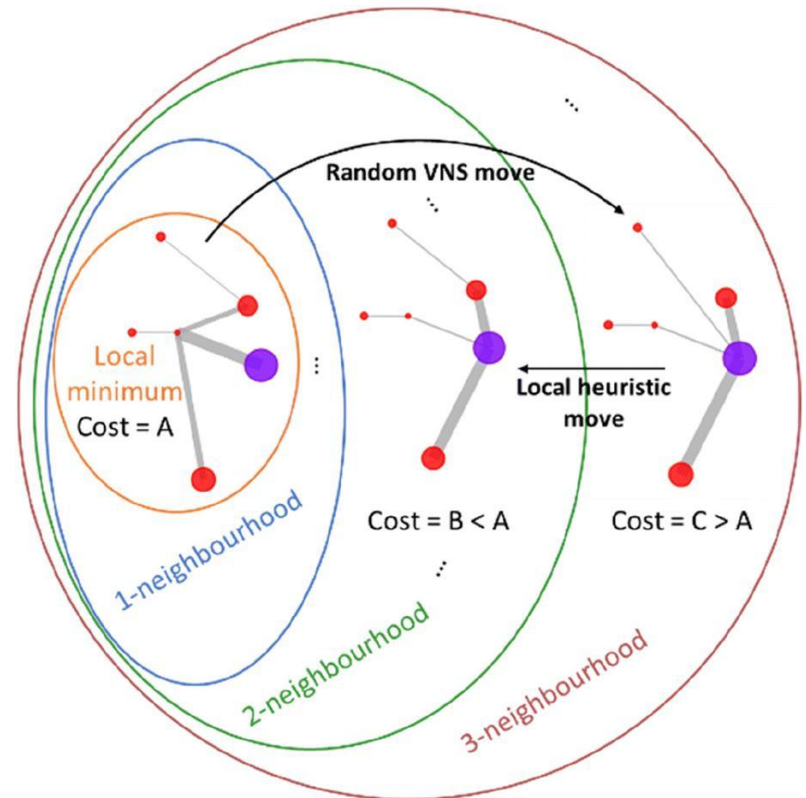
5. Τι Είναι η Ποιότητα μιας Λύσης

- Η ποιότητα εξαρτάται από το πρόβλημα
- Σε πρόβλημα διαδρομής:
 - ποιότητα = συνολικό μήκος ή κόστος
- Σε πρόβλημα έργων:
 - ποιότητα = συνολικό όφελος
- Σε πρόβλημα παραγωγής:
 - ποιότητα = κέρδος ή κόστος παραγωγής

	Desirability	Viability	Feasibility
Scoring criteria	Expected CSAT increase	Yearly cost reduction (in \$)	Development time (in hours)
Scoring guide	1 = 0-10% 2 = 10-20% 3 = 20-30% 4 = 30-40% 5 = 40%+	1 = 0-100k 2 = 100k-200k 3 = 200k-400k 4 = 400k-800k 5 = 800k+	1 = 400 Hours + 2 = 200 - 400 Hours 3 = 100 - 200 Hours 4 = 50 - 100 Hours 5 = 0- 50 Hours

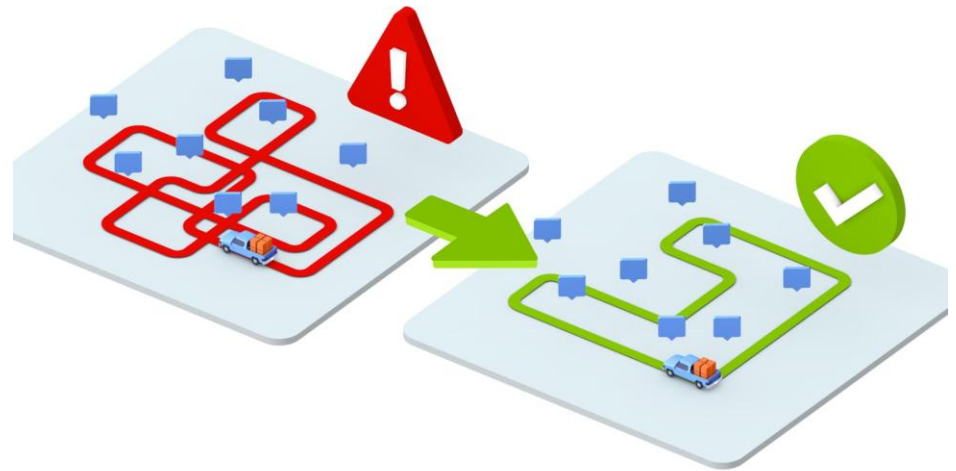
6. Η Έννοια της Γειτονικής Λύσης

- Γειτονική λύση:
 - λύση που προκύπτει από μικρή αλλαγή
- Παραδείγματα αλλαγών:
 - ανταλλαγή δύο κόμβων
 - αντικατάσταση έργου
 - αλλαγή σύνδεσης δικτύου
- Η αναζήτηση:
 - κινείται από λύση σε λύση



7. Παράδειγμα Διαδρομής

- Αρχική διαδρομή:
- – $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$
- Πιθανή αλλαγή:
- – ανταλλαγή των B και C
- Νέα διαδρομή:
- – $A \rightarrow C \rightarrow B \rightarrow D \rightarrow E$



8. Υπολογιστικό Παράδειγμα Διαδρομής

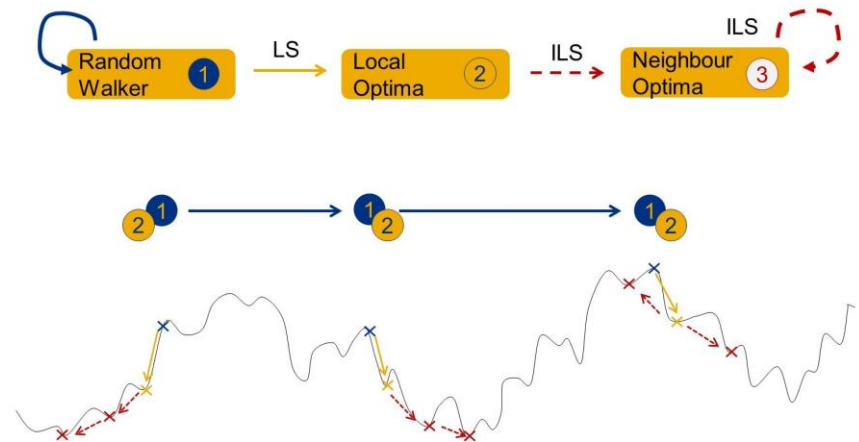
- Αρχική διαδρομή:
- – κόστος 120
- Μετά από ανταλλαγή δύο κόμβων:
- – νέο κόστος 105
- Εφόσον $105 < 120$:
- – η νέα διαδρομή γίνεται αποδεκτή



9. Επόμενη Επανάληψη

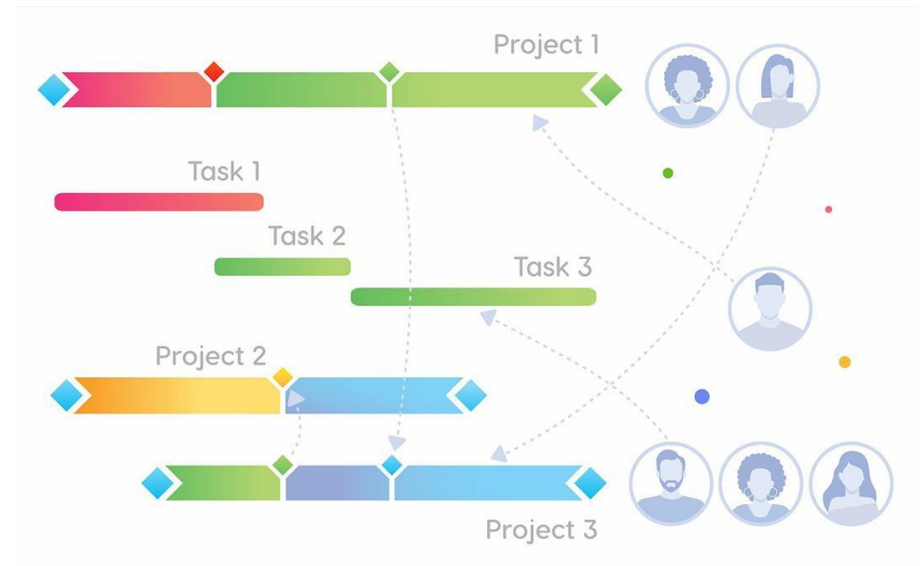
- Η νέα λύση:
 - γίνεται πλέον η τρέχουσα λύση
- Το πρόγραμμα:
 - δημιουργεί νέα μικρή αλλαγή
 - υπολογίζει νέο κόστος
 - συγκρίνει ξανά

Exploring the search space



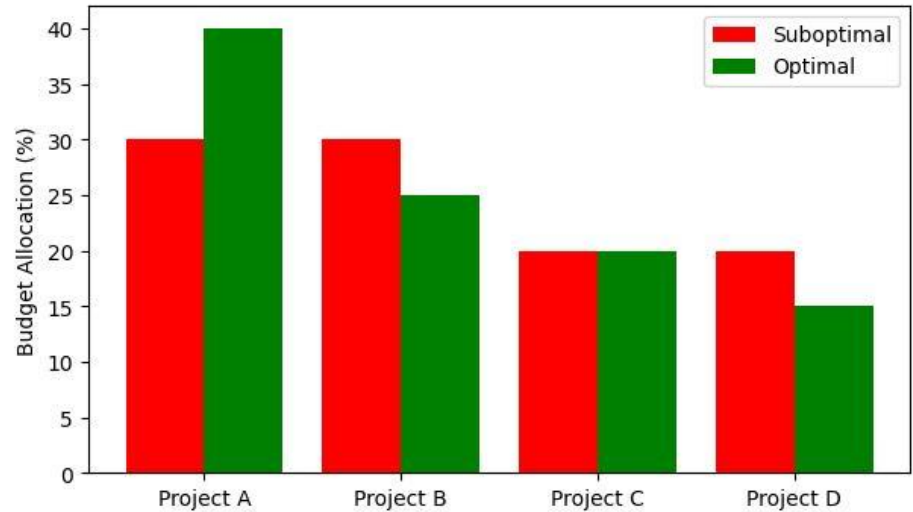
10. Παράδειγμα σε Επιλογή Έργων

- Διαθέσιμος προϋπολογισμός: 50
- Αρχική λύση:
 - έργα A και B
- Συνολικό κόστος: 45
- Συνολικό όφελος: 140



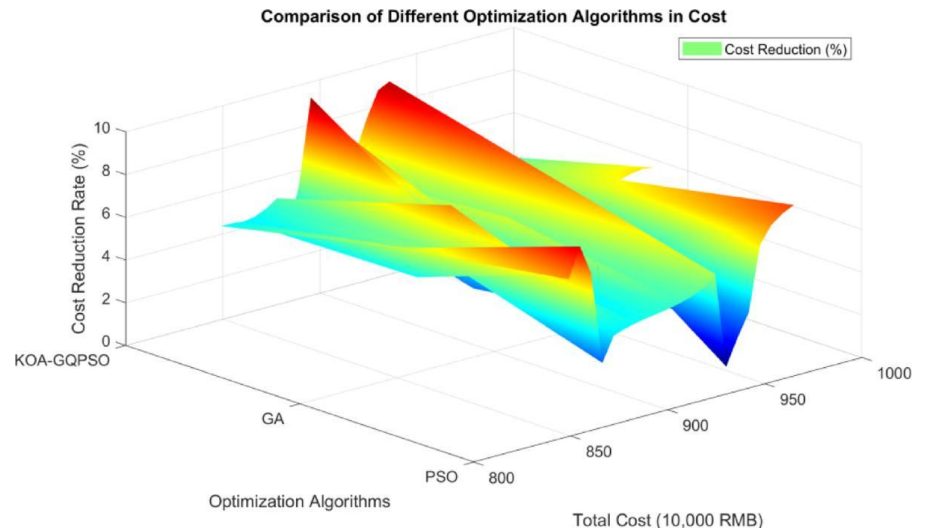
11. Μικρή Αλλαγή στη Λύση

- Πιθανή αλλαγή:
- – αφαιρούμε το έργο Β
- – προσθέτουμε το έργο Γ
- Νέα λύση:
- – έργα Α και Γ
- Συνολικό όφελος: 155



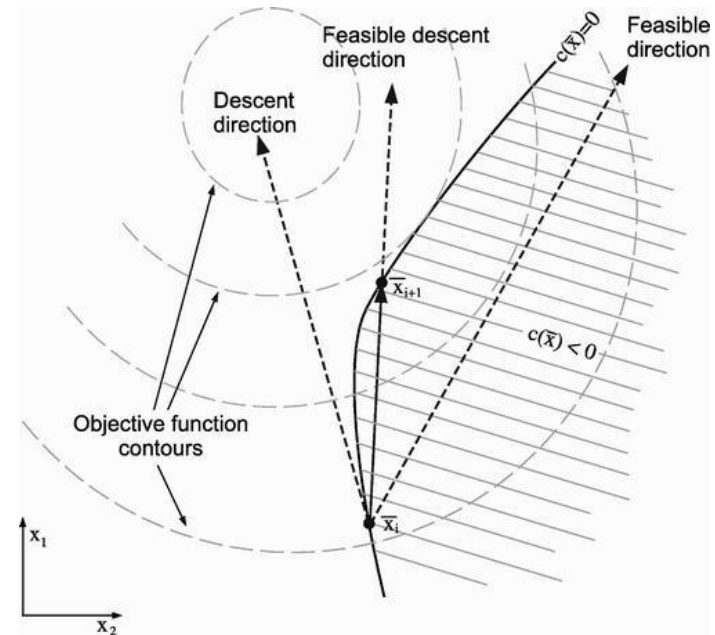
12. Αποδοχή Νέας Λύσης

- Παλιά λύση:
- – όφελος 140
- Νέα λύση:
- – όφελος 155
- Εφόσον $155 > 140$:
- – η νέα λύση γίνεται αποδεκτή



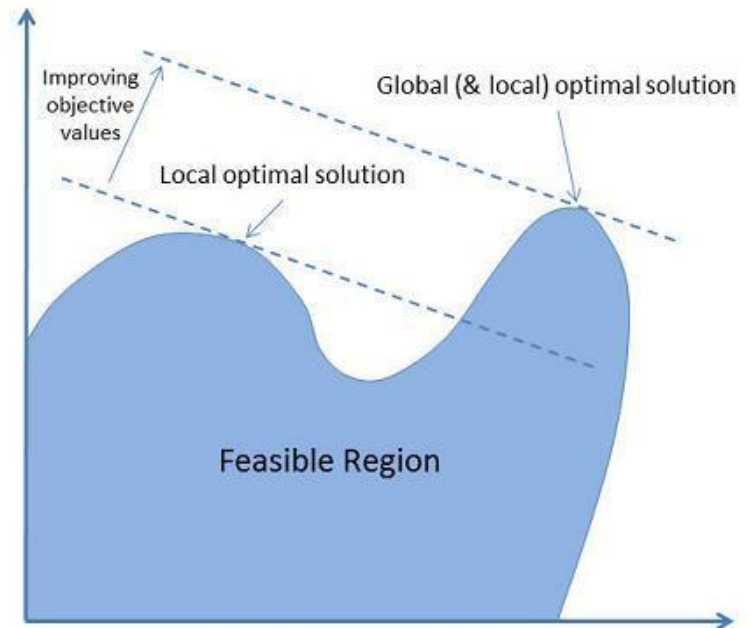
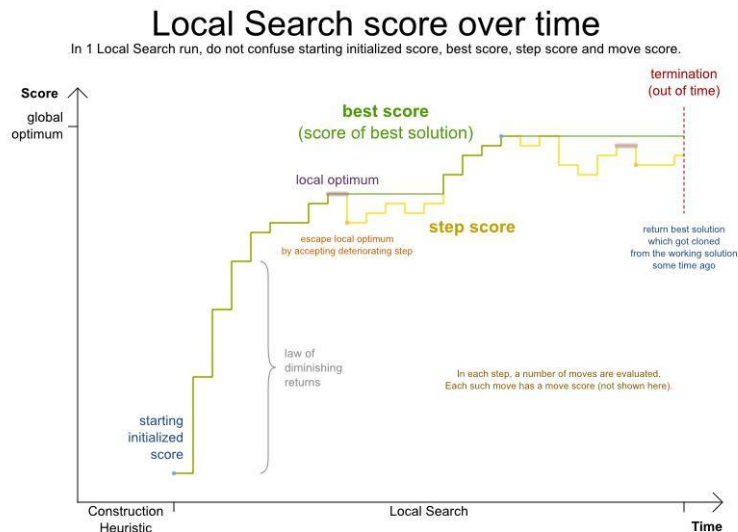
13. Κανόνας Αποδοχής

- Για πρόβλημα ελαχιστοποίησης:
 - αποδεχόμαστε λύση με μικρότερο κόστος
- Για πρόβλημα μεγιστοποίησης:
 - αποδεχόμαστε λύση με μεγαλύτερο όφελος
- Αν η νέα λύση δεν είναι καλύτερη:
 - απορρίπτεται



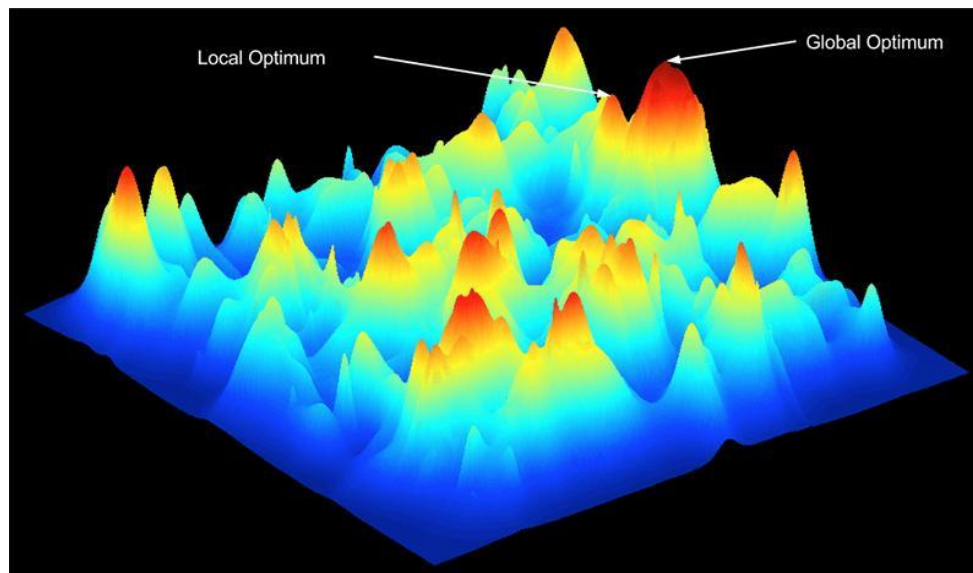
14. Κριτήριο Τερματισμού

- Η διαδικασία δεν μπορεί να συνεχίζεται για πάντα
- Παραδείγματα τερματισμού:
 - δεν υπάρχει καλύτερη γειτονική λύση
 - έγιναν αρκετές επαναλήψεις
 - πέρασε συγκεκριμένος χρόνος



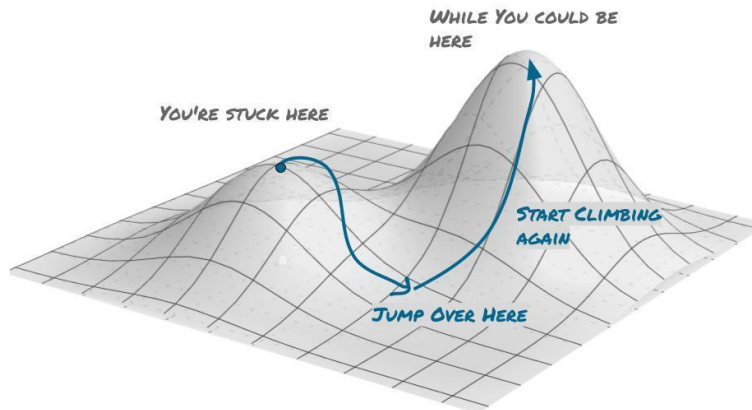
15. Τοπικό Βέλτιστο

- Μία λύση είναι τοπικά βέλτιστη όταν:
 - – καμία μικρή αλλαγή δεν τη βελτιώνει
- Όμως:
 - – μπορεί να υπάρχει καλύτερη λύση πιο μακριά
- Τοπικό βέλτιστο σημαίνει:
 - – καλή λύση μόνο στη γειτονιά της



16. Παράδειγμα Τοπικού Βέλτιστου

- Τρέχουσα λύση:
- – κόστος 80
- Όλες οι μικρές αλλαγές:
- – δίνουν κόστος 82, 85 ή 90
- Όμως:
- – μπορεί να υπάρχει άλλη λύση με κόστος 60



17. Γιατί Αυτό Είναι Χρήσιμο

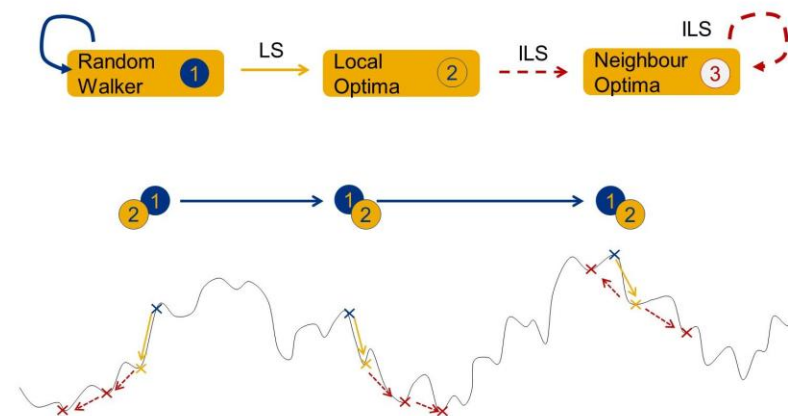
- Η τοπική αναζήτηση:
 - δεν απαιτεί εξέταση όλων των λύσεων
- Μπορεί:
 - να ξεκινήσει από οποιαδήποτε λύση
 - να τη βελτιώσει σταδιακά
 - να εφαρμοστεί σε μεγάλα προβλήματα

Combinatorial Optimization vs. Heuristics			
	Goal	Outcome	Time Required
Combinatorial Optimization lightningbolt	Finds optimal solution to a problem	Can guarantee the optimal solution (or how close it is to the optimal solution)	Works to find the best solution possible in minutes; does not need to be monitored or tailored as it works
Heuristic Scheduling Other Vendors	Generally finds non-optimal solutions to problems	Cannot guarantee the optimal solution or how close it is to the optimal solution	Usually runs fast

18. Πρακτική Λογική Υλοποίησης

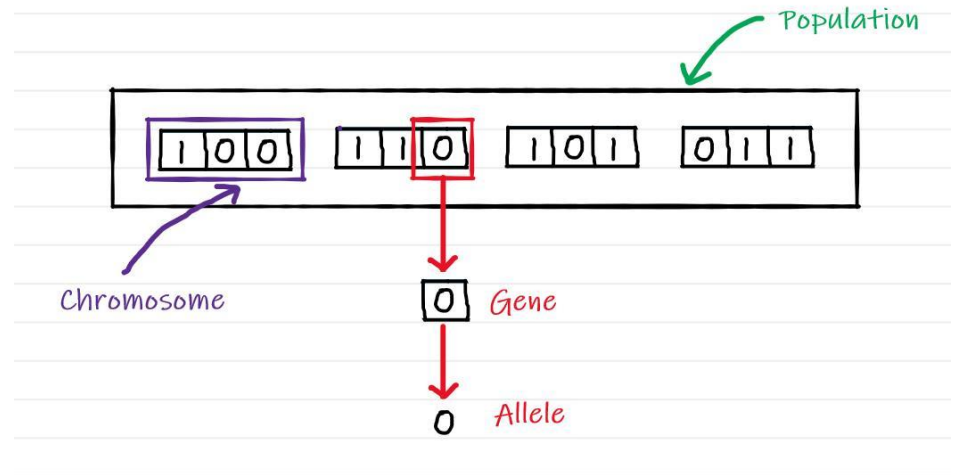
- Ένα πρόγραμμα χρειάζεται:
- 1. Αναπαράσταση λύσης
- 2. Συνάρτηση αξιολόγησης
- 3. Δημιουργία γειτονικών λύσεων
- 4. Κανόνα αποδοχής

Exploring the search space



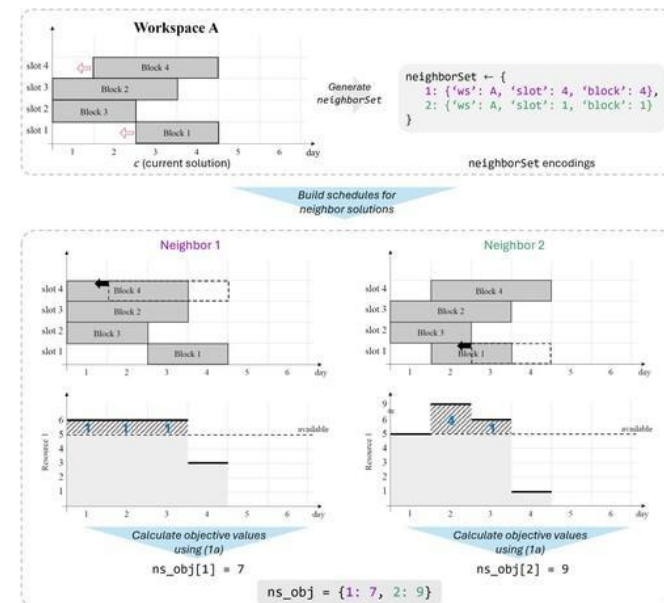
19. Αναπαράσταση Λύσης

- Σε πρόβλημα διαδρομής:
- – η λύση είναι σειρά κόμβων
- Παράδειγμα:
- – [A, B, C, D, E]
- Σε πρόβλημα έργων:
- – [1, 0, 1, 0, 1]



20. Δημιουργία Γειτονικών Λύσεων

- Το πρόγραμμα μπορεί να δημιουργεί λύσεις με:
 - – ανταλλαγή στοιχείων
 - – αντικατάσταση επιλογής
 - – μετακίνηση στοιχείου
 - – αλλαγή δυαδικής απόφασης



22. Έλεγχος Εφικτότητας

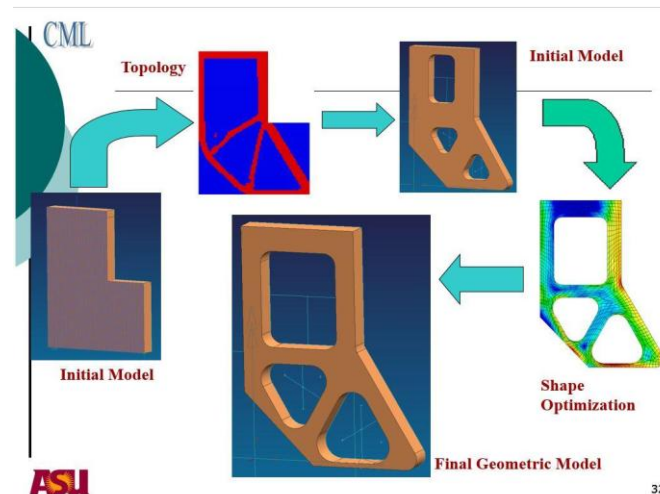
- Κάθε νέα λύση πρέπει να ελέγχεται
- Σε πρόβλημα έργων:
 - δεν πρέπει να ξεπερνά τον προϋπολογισμό
- Σε πρόβλημα διαδρομής:
 - πρέπει να αποτελεί έγκυρη διαδρομή

23. Απλός Ψευδοκώδικας

- Ξεκίνα με αρχική λύση S
- Όσο υπάρχει βελτίωση:
 - δημιούργησε γειτονική λύση S'
 - έλεγξε αν είναι εφικτή
 - αν είναι καλύτερη, θέσε $S = S'$

24. Απόφαση Σχεδίασης και Αναζήτηση

- Σε προβλήματα σχεδίασης:
 - υπάρχουν πολλές πιθανές λύσεις
- Η τοπική αναζήτηση:
 - επιτρέπει συστηματική βελτίωση
- Η σχεδίαση:
 - γίνεται διαδικασία δοκιμής και προσαρμογής



25. Κεντρικό Μήνυμα Διάλεξης

- Η επίλυση δεν τελειώνει στην αρχική λύση
- Μία λύση μπορεί:
 - να αξιολογηθεί
 - να τροποποιηθεί
 - να βελτιωθεί
- Η τοπική αναζήτηση:
 - μετατρέπει τη βελτιστοποίηση σε επαναληπτική διαδικασία

