



Πολυτεχνείο
Κρήτης

Σχολή Μηχανικών Παραγωγής και Διοίκησης

*Εισαγωγή στην Τεχνητή
Νοημοσύνη*

Αλγόριθμοι τοπικής αναζήτησης Μεθευρετικοί Αλγόριθμοι

Τσαφαράκης Στέλιος

Δύσκολα προβλήματα βελτιστοποίησης

- Η επίλυση ενός προβλήματος βελτιστοποίησης & ιδιαίτερα συνδυαστικής (combinatorial) βελτιστοποίησης γίνεται ολοένα & δυσκολότερη όσο αυξάνει το μέγεθος του προβλήματος.
- Πολλές φορές το να προσπαθούμε να βρούμε την ολικά βέλτιστη (global optimal) λύση σε λογικό χρόνο είναι πρακτικά αδύνατο.
- Π.χ. τα **μη πολυωνυμικού χρόνου δύσκολα (NP-Hard)** αποτελούν μια κατηγορία προβλημάτων όπου σημαίνει όχι μόνο δεν είναι γνωστός κάποιος εφικτός αλγόριθμος για την επίλυση του προβλήματος αλλά είναι απίθανο να υπάρχει τέτοιος αλγόριθμος
- Για να επιλυθούν προβλήματα αυτής της μορφής συχνά καταφεύγουμε σε διαφορετικές τεχνικές που μας οδηγούν σε μια σχεδόν βέλτιστη (near-optimal), αλλά ικανοποιητική λύση.
- Μια λύση ενός ευρετικού αλγορίθμου γίνεται αποδεκτή αν ικανοποιεί κάποια κριτήρια όπως η ποιότητα της λύσης (η απόκλιση της από τη βέλτιστη), η ευκολία απόκτησης μιας λύσης κ.α.

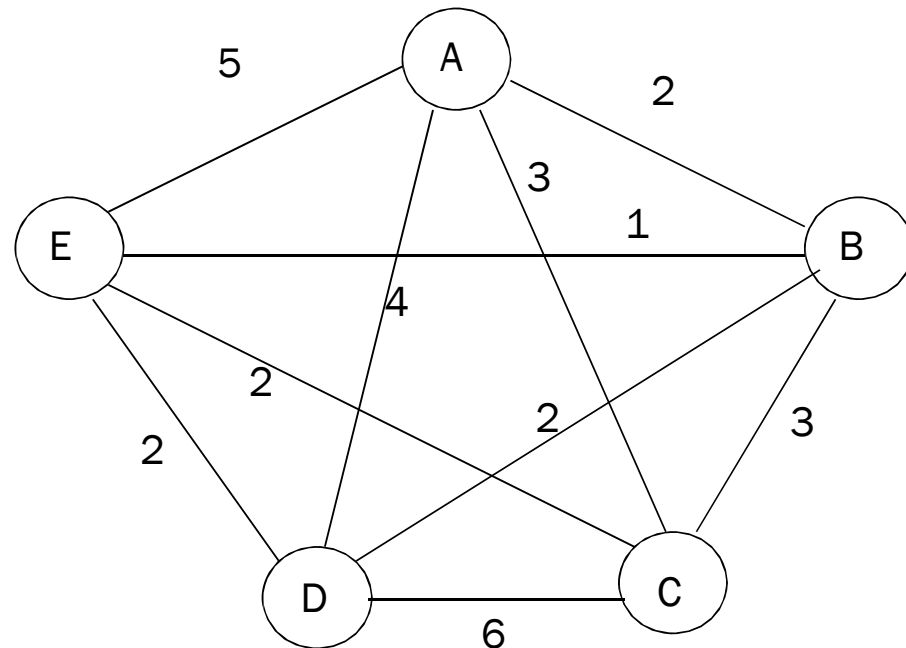
Το πρόβλημα του Πλανόδιου Πωλητή – Traveling Salesman Problem (TSP)

- Θέλω να περάσω από κάθε πόλη ακριβώς 1 φορά & να γυρίσω σε αυτή που ξεκίνησα.
- Γνωρίζω τις αποστάσεις μεταξύ των πόλεων.
- Στόχος: να βρω τη συντομότερη διαδρομή (πρόβλημα ελαχιστοποίησης):

$$\min f = d_{12} + d_{23} + \dots$$

όπου d_{ij} η απόσταση μεταξύ των πόλεων i και j

πόλη εκκίνησης: A



μικρότερη διαδρομή: ABDECA
μήκος(ABDECA) = 2+2+2+2+3=11

Εξαντλητική Αναζήτηση

- Υπολόγισε όλα τα μονοπάτια και διάλεξε το συντομότερο
 - Μήκος(ABCDEA) = $2 + 3 + 6 + 2 + 5 = 18$
 - Μήκος(ACBDEA) = $3 + 3 + 2 + 2 + 5 = 15$
 -
 - Μήκος(ABDECA) = $2 + 2 + 2 + 2 + 3 = 11$
 - Μήκος(AEBCDA) = $5 + 1 + 3 + 6 + 4 = 19$
- Υπάρχουν συνολικά 24 διαφορετικά μονοπάτια (πιθανές λύσεις του προβλήματος)
- Για n πόλεις υπάρχουν $(n-1)!$ μονοπάτια =>
15 πόλεις υπάρχουν 1.307.674.368.000 μονοπάτια
- Η πολυπλοκότητα του προβλήματος αυξάνει εκθετικά (NP-hard)
- Ακόμα & ισχυρός Η/Υ θα χρειαστεί χρόνια για να απαριθμήσει όλες τις λύσεις & θα εξαντληθεί η μνήμη του.

Αναπαράσταση λύσης

- Κρίσιμο παράγοντα για την αποτελεσματικότητα του αλγορίθμου αποτελεί το είδος της αναπαράστασης (κωδικοποίησης) της λύσης βάσει των απαιτήσεων του προβλήματος
- Συνεχή προβλήματα αναπαρίστανται με πραγματικές τιμές:

3,8 5,2 -2,9 0,4

- Διακριτά προβλήματα 2 τιμών (δυαδικά) αναπαρίστανται με δυαδικές τιμές:

1 0 0 1 1 0 1 0

- Ακέραια προβλήματα αναπαρίστανται με ακέραιες τιμές:

9 2 1 8 3 6

- Για προβλήματα διαδρομών χρησιμοποιούμε αναπαράσταση μονοπατιού:

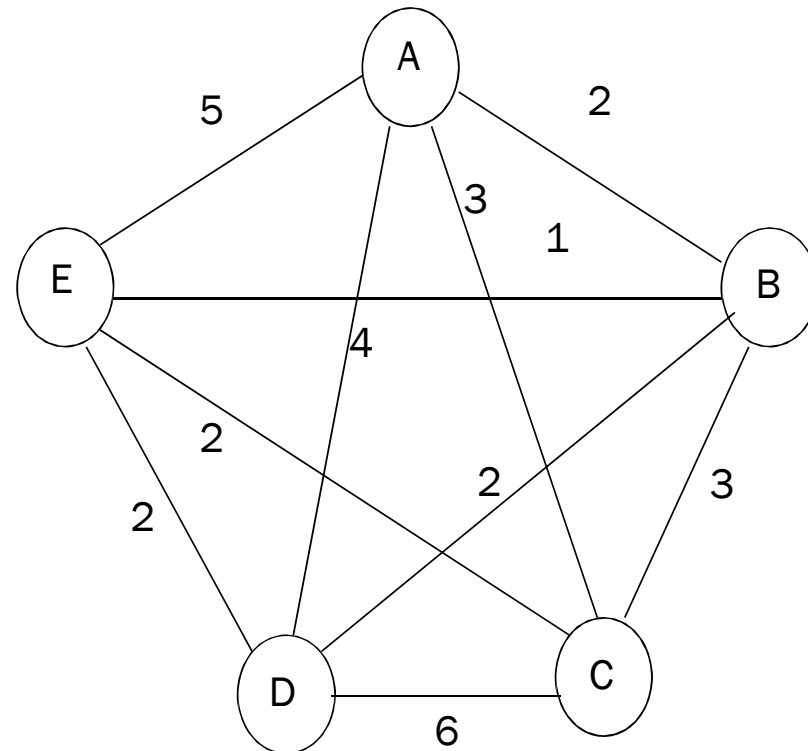
6 7 4 5 9 2 8 1 3

Αναπαράσταση του TSP

- Μια λύση αναπαρίσταται από ένα διάνυσμα με τη σειρά επίσκεψης των πόλεων, π.χ:
A C E D B A ή 1 3 5 4 2 1
- Για τον ευκολότερο υπολογισμό της αντικειμενικής συνάρτησης δημιουργούμε πίνακα με τις αποστάσεις:

	A	B	C	D	E
A	-	2	3	4	5
B	2	-	3	2	1
C	3	3	-	6	2
D	4	2	6	-	2
E	5	1	2	2	-

πόλη εκκίνησης: A



Δυαδική κωδικοποίηση του TSP

- Έστω d_{ij} η απόσταση (μήκος διαδρομής) μεταξύ των πόλεων i και j .
- Έστω επίσης η δυαδική μεταβλητή απόφασης:

$$x_{ij} = \begin{cases} 1, & \text{εαν η διαδρομη (ακμη) περιλαμβανεται στη λυση} \\ 0, & \text{αλλιως} \end{cases}$$

Η αντικειμενική συνάρτηση γίνεται (n ο αριθμός των πόλεων):

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}, \quad i \neq j \quad (1)$$

$$\text{υπό } \sum_{j=1}^n x_{ij} = 2, \quad \forall i = 1, 2, \dots, n \quad (2)$$

$$\text{με } x_{ij} \in \{0, 1\}, \quad \forall i, \forall j, i \neq j$$

Ο περιορισμός (2) εξασφαλίζει ότι σε κάθε κόμβο (πόλη) υπάρχει ακριβώς 1 ακμή που εισέρχεται & 1 που εξέρχεται

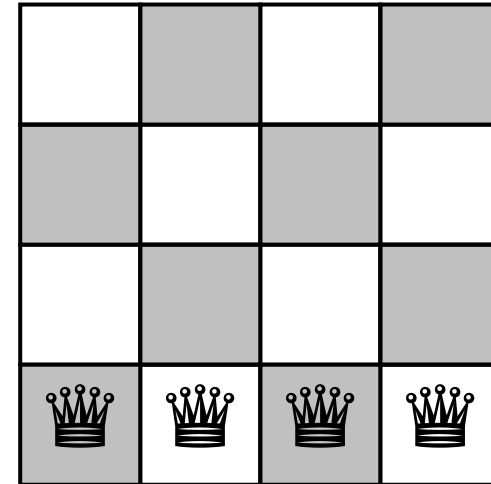
d_{ij}	1	2	3	4	5
1	-	2	3	4	5
2	2	-	3	2	1
3	3	3	-	6	2
4	4	2	6	-	2
5	5	1	2	2	-

Αλγόριθμοι τοπικής αναζήτησης

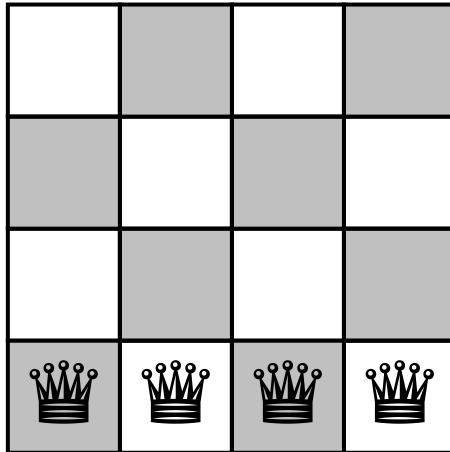
- Σε πολλά προβλήματα η διαδρομή προς τον στόχο δεν έχει σημασία
- Π.χ. στις 8 βασίλισσες σημασία έχει μόνο η τελική διάταξη & όχι η σειρά με την οποία προστίθενται
- Οι αλγόριθμοι τοπικής αναζήτησης (local search) λειτουργούν χρησιμοποιώντας μόνο 1 τρέχουσα κατάσταση (αντί για πολλές διαδρομές).
- Γενικά μετακινούνται μόνο σε γειτονικές καταστάσεις
- Κανονικά οι διαδρομές που ακολουθούνται από την αναζήτηση δεν διατηρούνται στη μνήμη
- Είναι χρήσιμοι σε προβλήματα βελτιστοποίησης που έχουν στόχο την εύρεση της καλύτερης κατάστασης σύμφωνα με μια αντικειμενική συνάρτηση (objective function)
- Πλεονεκτήματα αλγορίθμων τοπικής αναζήτησης:
 - Χρησιμοποιούν πολύ λίγη μνήμη
 - Μπορούν συχνά να βρίσκουν λογικές λύσεις σε μεγάλους ή & άπειρους (συνεχείς) χώρους καταστάσεων

Παράδειγμα με 4 βασίλισσες (1/2)

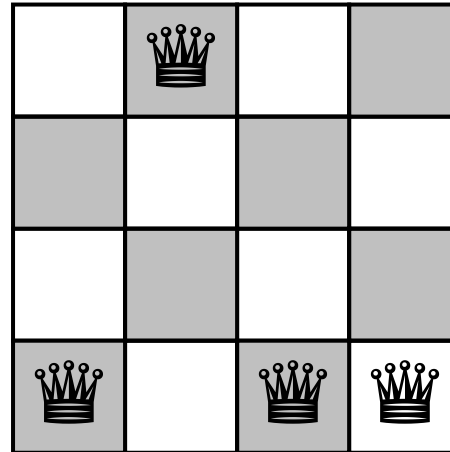
- Σε κάθε στήλη θα υπάρχει μία μόνο βασίλισσα.
- Ξεκινάμε με όλες τις βασίλισσες στην κάτω γραμμή.
- Σε κάθε βήμα μπορούμε να μετακινήσουμε μια βασίλισσα σέ μια άλλη θέση στη στήλη της, άρα οι δυνατές κινήσεις είναι $4 \times 3 = 12$.
- Χρησιμοποιούμε ως ευρετική συνάρτηση το πόσες απειλές υπάρχουν κάθε φορά (όλες οι απειλές είναι διπλές, εμείς τις μετράμε ως μία απειλή κάθε φορά).



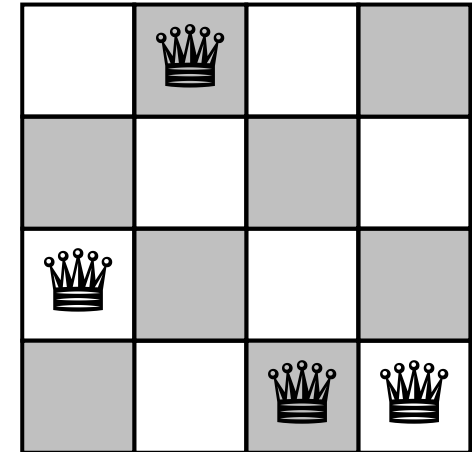
Παράδειγμα με 4 βασίλισσες (2/2)



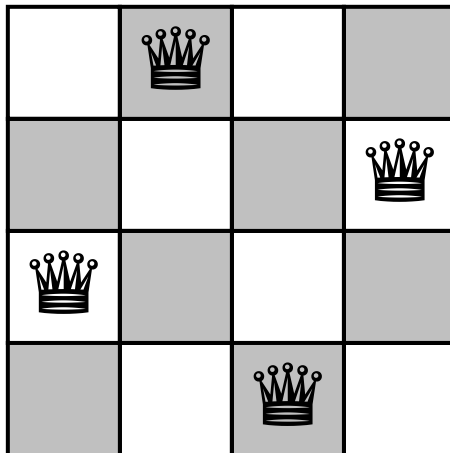
6



3



1

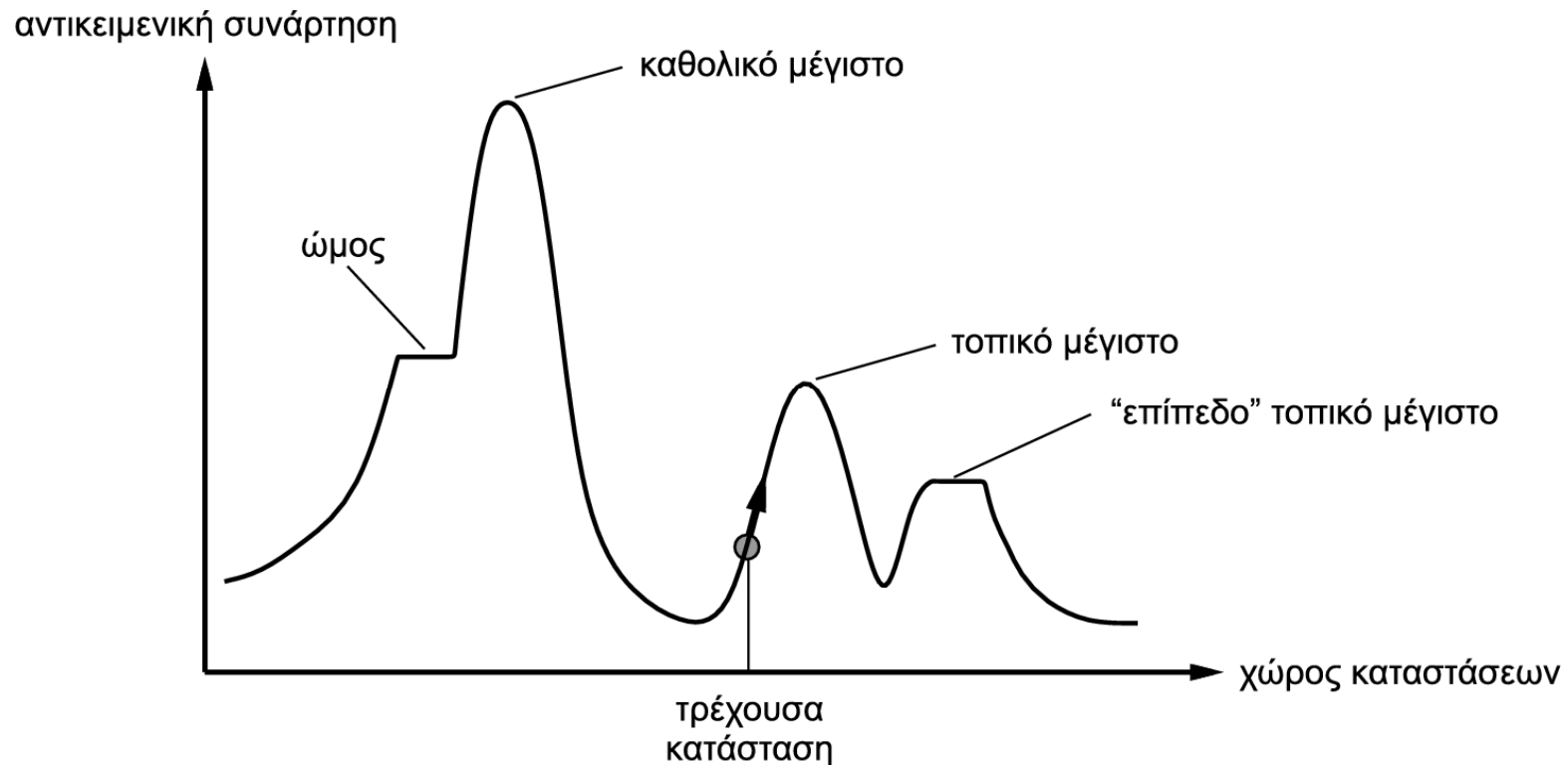


0

- Στο συγκεκριμένο παράδειγμα είμασταν τυχεροί, μιας και όλες οι επιλογές μας βγήκαν σωστές.

Αναρρίχηση λόφων (hill climbing)

- Μετακινείται συνεχώς στην καλύτερη γειτονική κατάσταση.
 - Συνήθως διατύπωση με πλήρεις καταστάσεις
- Δεν κοιτάζει πιο πέρα από τους άμεσους γείτονες της τρέχουσας κατάστασης:
 - «σαν να προσπαθεί κανείς να βρει την κορυφή του Έβερεστ μέσα σε πυκνή ομίχλη ενώ πάσχει από αμνησία»



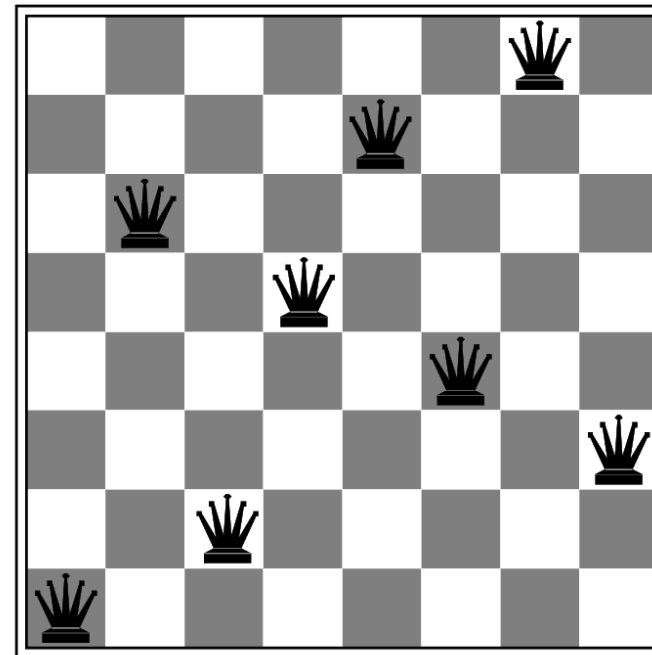
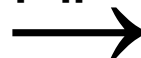
Αναρρίχηση λόφων στις βασίλισσες

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♚	13	16	13	16
♚	14	17	15	♚	14	16	16
17	♚	16	18	15	♚	15	♚
18	14	♚	15	15	14	♚	16
14	14	13	17	12	14	12	18

Τρέχουσα τιμή $h=17$

(α)

5 βήματα



Τοπικό ελάχιστο με $h=1$

(β)

Μεθευρετικοί αλγόριθμοι

- Οι μεθευρετικοί αλγόριθμοι είναι μέθοδοι επίλυσης που συνδυάζουν διαδικασίες τοπικής αναζήτησης & ανωτέρου επιπέδου στρατηγικές για να αντιμετωπίσουν τα προβλήματα των αλγορίθμων τοπικής αναζήτησης:
 - Εγκλωβισμός σε τοπικά βέλτιστα.
 - Εξάρτηση από επιλογή αρχικής λύσης.
- Βρίσκουν εφαρμογή σε πολλά δύσκολα προβλήματα βελτιστοποίησης στον πραγματικό κόσμο.
- Τα τελευταία χρόνια έχουν αναπτυχθεί διάφορες κατηγορίες μεθευρετικών αλγορίθμων (βασισμένοι σε μια ή σε πληθυσμό λύσεων, με ή χωρίς μνήμη, εμπνευσμένοι από τη φύση)

Προσομοιωμένη Ανόπτηση - Simulated Annealing (SA)

- Η στρατηγική του αλγόριθμου προέρχεται από την προσομοίωση της ανόπτησης των υλικών: ένα στερεό υλικό θερμαίνεται μέχρι το σημείο τήξης του & στη συνέχεια ψύχεται αργά.
- Όταν η ψύξη σταματήσει τότε το υλικό πέφτει σε στάδιο χαμηλότερης ενέργειας.
- Αν η αρχική θερμοκρασία δεν είναι αρκετά υψηλή ή αν επιλεγεί μία στρατηγική απότομης μείωσής της τότε μπορεί να παρουσιασθούν ατέλειες στο σύστημα: το στερεό δεν θα μπορέσει να έρθει σε κατάσταση θερμικής ισορροπίας σε καμία θερμοκρασία.
- Ο αλγόριθμος της προσομοιωμένης ανόπτησης προσομοιώνει τις αλλαγές στην ενέργεια ενός συστήματος που υπόκειται σε μία διαδικασία ψύξης μέχρι να συγκλίνει σε ένα σημείο ισορροπίας.

Προσομοιωμένη ανόπτηση

- Προ-επιλέγει τυχαία μια κίνηση.
- $\Delta E = f(\text{next}) - f(\text{current})$
- Αν αυτή βελτιώνει τα πράγματα ($\Delta E > 0$), γίνεται αποδεκτή.
- Αν τα χειροτερεύει ($\Delta E < 0$), γίνεται αποδεκτή με πιθανότητα $P = e^{\Delta E/T}$:
 - $\Delta E < 0$, η χειροτέρευση
 - $T > 0$, «θερμοκρασία»
- Η θερμοκρασία μειώνεται με την πάροδο του χρόνου:
 - Στην αρχή της αναζήτησης γίνονται ευκολότερα αποδεκτές οι κινήσεις χειροτέρευσης (global search – escape from local optima)
 - Όσο προχωρά η αναζήτηση οι κινήσεις χειροτέρευσης γίνονται δυσκολότερα αποδεκτές (local search)

Ψευδοκώδικας SA

- Δημιούργησε (τυχαία) μια αρχική εφικτή λύση s_0
- Καθόρισε μια αρχική τιμή t_0 για τη θερμοκρασία
- Καθόρισε μια συνάρτηση μείωσης της θερμοκρασίας $a(t_k)$
- Επανάλαβε
 - Επανάλαβε
 - Δημιούργησε μια καινούρια λύση s' στη γειτονιά της s_0
 - $\Delta = f(s') - f(s_0)$
 - Εάν $\Delta < 0$ (για πρόβλημα ελαχιστοποίησης)
 - Τότε θέσε $s_0 = s'$
 - Αλλιώς ($\Delta \geq 0$)
 - Δημιούργησε ένα τυχαίο αριθμό x ομοιόμορφα κατανομημένο στο $(0,1)$
 - Εάν $x < e^{-\frac{\Delta}{t}}$ τότε θέσε $s_0 = s'$
 - Μέχρι να ολοκληρωθεί ο αριθμός επαναλήψεων στη θερμοκρασία t_k
 - $t_{k+1} = a(t_k)$ (ανανέωση τιμής θερμοκρασίας)
 - Μέχρι να ικανοποιηθεί κάποια συνθήκη τερματισμού
 - Επέστρεψε την καλύτερη λύση