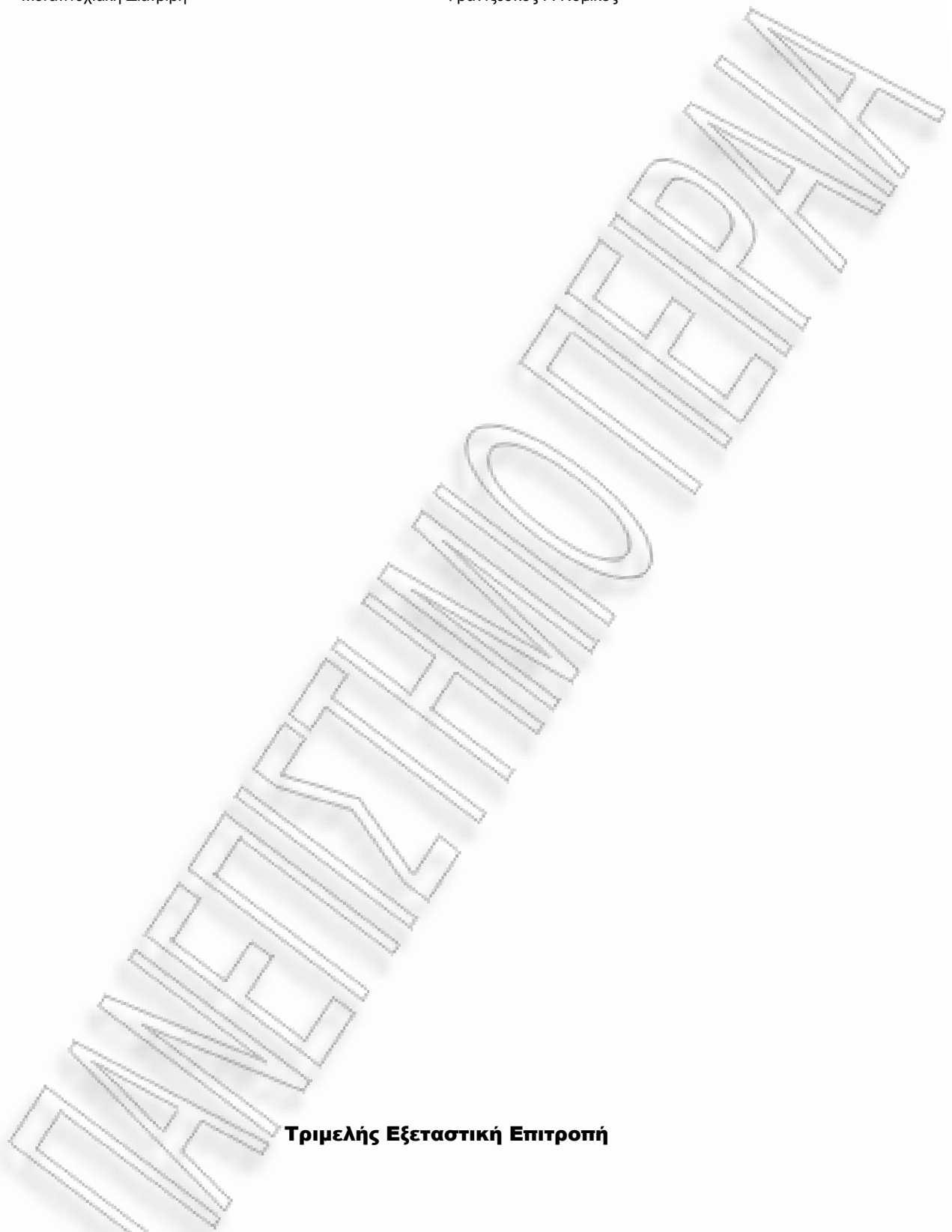




Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Πληροφορική»

**Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	<b>Αλγόριθμοι Αποικίας Μυρμηγκιών</b>
Όνοματεπώνυμο Φοιτητή	<b>Φραντζέσκος Νομικός</b>
Πατρώνυμο	<b>Γεώργιος</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ/ 07049</b>
Επιβλέπων	<b>Άγγελος Πικράκης</b>



**Τριμελής Εξεταστική Επιτροπή**

Ευάγγελος Φούντας  
Καθηγητής

Γεώργιος Τσιχριντζής  
Καθηγητής

Άγγελος Πικράκης  
Λέκτορας

## Αλγόριθμοι Αποικίας Μυρμηγκιών

### Περίληψη.

Το Κεφάλαιο 1, παρουσιάζει μία περίληψη της βελτιστοποίησης αποικίας μυρμηγκιών (ACO) - μία μεταεвриστική εμπνευσμένη από τη συμπεριφορά πραγματικών μυρμηγκιών, μία μέθοδος επίλυσης δύσκολων προβλημάτων συνδυαστικής βελτιστοποίησης

Το Κεφάλαιο 2, παρουσιάζει ένα τρόπο να εκτείνουμε τον ACO σε συνεχή χωρία χωρίς να χρειαστεί να κάνουμε καμία κρίσιμη αλλαγή στη δομή του. Δηλώνουμε ότι ο ACO επεκτείνεται σε συνεχή χωρία από τον  $ACO_R$ .

Το Κεφάλαιο 3, παρουσιάζει στην ενότητα 3.1 έναν αλγόριθμο εμπνευσμένο από τα μυρμηγκία για βελτιστοποίηση σε συνεχείς χώρους αναζήτησης που βασίζεται στη παραγωγή τυχαίων διανυσμάτων με πολυμεταβλητή Γκαουσιανή συνάρτηση πυκνότητας πιθανότητας. Ονομάζεται MACACO και συγκρίνεται με το Συνεχές Σύστημα Μυρμηγκιακής Αποικίας (CACS) και τη Βελτιστοποίηση Αποικίας Μυρμηγκιών στον  $R^n$  ( $ACO_R$ ).

Στην ενότητα 3.2 περιγράφεται ένα σύστημα συνάθροισης φερομόνης (APS), το οποίο είναι μία επέκταση του ACO για συνεχή χωρία, χρησιμοποιώντας τη συλλογική συμπεριφορά των ατόμων που επικοινωνούν χρησιμοποιώντας φερομόνες συνάθροισης. Το APS δοκιμάστηκε σε αρκετές συναρτήσεις ελέγχου. Τα αποτελέσματα δείχνουν ότι το APS μπορεί να λύσει προβλήματα βελτιστοποίησης πραγματικών παραμέτρων αρκετά καλά.

Το Κεφάλαιο 4, παρουσιάζει στην ενότητα 4.1 κάποιες ιδιότητες σύγκλισης για μία κλάση από αλγορίθμους βελτιστοποίησης αποικίας μυρμηγκιών. Αποδεικνύεται επίσης, ότι μετά την εύρεση μίας βέλτιστης λύσης, χρειάζεται ένας πεπερασμένος αριθμός από επαναλήψεις για να αυξηθούν τα ίχνη φερομόνης που σχετίζονται με την εύρεση βέλτιστης λύσης περισσότερο από κάθε άλλο ίχνη φερομόνης

Στην ενότητα 4.2 παρουσιάζεται μία ανάλυση σύγκλισης του ACO σε παραπλανητικά προβλήματα. και αποδεικνύεται ότι ο ACO μπορεί να επιτύχει σύγκλιση προσβασιμότητας αλλά όχι ασυμπτωτική σύγκλιση για μία κλάση από πρώτης τάξης παραπλανητικά προβλήματα (FODS) χωρίς να υποθέσουμε μία ελάχιστη φερομόνη σε κάθε επανάληψη.

### Abstract.

Chapter 1 presents a summary of the ant colony optimization (ACO) – a metaheuristic inspired from the behavior of real ants, a method of solving difficult problems of combinatorial optimization.

Chapter 2 analyzes a way to expand ACO in continuous domains without having to do any critical change to its structure. We denote that ACO is extended to continuous domains from  $ACO_R$ .

Chapter 3 demonstrates in section 3.1 an algorithm inspired from the ants for optimization in continuous search spaces based on the production of random vectors with multivariable Gaussian probability density function. It is called MACACO and is compared with the Continuous Ant Colony System (CACS) and the Ant Colony Optimization on  $R^n$  ( $ACO_R$ ).

In section 3.2 is described an aggregation pheromone system (APS), that is an expansion of ACO for continuous domains, using the collective behavior of the individuals that communicate utilizing aggregation pheromones. APS is tested on several test functions. Results indicate that APS could solve efficiently real parameter optimization problems.

Chapter 4 presents in section 4.1 some convergence properties for a class of ant colony optimization algorithms. Furthermore, it is proved that after an optimal solution has been found, it demanded a finite number of iterations to grow higher the pheromone trails that are associated with the search of the optimum solution more than any other pheromone trail.

In section 4.2 is presented a convergence analysis of ACO to deceptive problems and it is established that ACO can achieve reachability convergence but not asymptotic convergence for a class of first order deceptive problems (FODS) without the assumption of a minimum pheromone at each iteration of the algorithm.

## ΠΕΡΙΕΧΟΜΕΝΑ

### Κεφάλαιο 1

#### Βελτιστοποίηση Αποικίας Μυρμηγκιών

1.1	Από τη Βιολογία στους Αλγόριθμους	7
1.1.1	Μυρμήγκια	7
1.1.2	Αλγόριθμοι	8
1.2	Η Μεταεριστική της Βελτιστοποίησης Αποικίας Μυρμηγκιών.	7
1.2.1	Παράδειγμα: Το Πρόβλημα του Περιοδεύοντα Πωλητή	12
1.3	Κύριες Παραλλαγές του ACO	13
1.3.1	Το Σύστημα Μυρμηγκιών (AS)	13
1.3.2	MAX-MIN Σύστημα Μυρμηγκιών	14
1.3.3	Σύστημα Αποικίας Μυρμηγκιών	15

### Κεφάλαιο 2

#### Ο ACO για Συνεχή Χωρία

2.1	Ο Αλγόριθμος	17
2.1.1	Δομή Αρχείου, αρχικοποίηση, και ενημέρωση	18
2.1.2	Πιθανοθεωρητική κατασκευή λύσης	19
2.2	Τοποθέτηση του $ACO_{\square}$	20
2.2.1	Ο $ACO_{\square}$ και Άλλοι Swarm-Based Αλγόριθμοι	21
2.2.2	Ο $ACO_{\square}$ και οι Εξελικτικοί Αλγόριθμοι	22
2.3	Πειραματική Ρύθμιση και Αποτελέσματα	23

### Κεφάλαιο 3

#### Διάφοροι τύποι αλγορίθμων για συνεχή προβλήματα

3.1	Πολυμεταβλητή Βελτιστοποίηση Αποικίας Μυρμηγκιών σε Συνεχείς Χώρους Αναζήτησης	25
3.1.1	Εισαγωγή	25

3.1.2. Σύστημα Μυρμηγκιών (AS) και Βελτιστοποίηση Αποικίας Μυρμηγκιών (ACO)	26
Βελτιστοποίηση Αποικίας Μυρμηγκιών (ACO)	26
Προσαρμόζοντας τον αλγόριθμο ACO για να αντιμετωπίσει Συνεχείς Χώρους Αναζήτησης	27
Αλγόριθμος CACS	28
Ο αλγόριθμος ACO <sub>R</sub>	28
3.1.3. MACACO : Πολυμεταβλητός αλγόριθμος Αποικίας Μυρμηγκιών για συνεχή βελτιστοποίηση	29
3.1.4. Αποτίμηση Απόδοσης	31
Πειραματικά Αποτελέσματα	32
3.1.5. Σχόλια	34
3.2 Βελτιστοποίηση Αποικίας Μυρμηγκιών για Συνεχή Χωρία με Μεταφορά Συνάθροισης Φερομονών	35
3.2.1. Εισαγωγή	35
3.2.2. Το Σύστημα Συνάθροισης Φερομόνης	36
Μία Σύντομη Περίληψη του ACO	36
Το Βασικό Μοντέλο του Συστήματος Συνάθροισης Φερομόνης	36
Τεχνική Δειγματοληψίας	38
3.2.3. Πειραματική Μελέτη	38
Πειραματική Μεθοδολογία	38
Αποτελέσματα	39
3.2.4. Συμπεράσματα	41

## Κεφάλαιο 4

### Μια Εισαγωγή Στη Σύγκλιση Των Αλγορίθμων ACO

4.1 Μία Σύντομη Απόδειξη Σύγκλισης για μία Κλάση από Αλγορίθμους Βελτιστοποίησης Αποικίας μυρμηγκίων	42
4.1.1 Εισαγωγή	42
4.1.2 Το Πρόβλημα Και Ο Αλγόριθμος	42

4.1.3	Απόδειξη Σύγκλισης	44
4.1.4	Σχόλια	47
4.1.5	ACO Αλγόριθμοι Και Σύγκλιση	49
4.1.6	Συμπέρασμα	52
4.2	Σύγκλιση Βελτιστοποίησης Αποικίας μυρμηγκιών σε Πρώτης-Τάξης Παραπλανητικά Συστήματα (first-order deceptive system)	52
4.2.1	Εισαγωγή	52
4.2.2	Βελτιστοποίηση Αποικίας μυρμηγκιών και Παραπλανητικά Προβλήματα	53
4.2.3	Ανάλυση Σύγκλισης του ACO στο FODS	55
	Το πρόβλημα n-bit παγίδας	55
	Ασυμπτωτική Σύγκλιση του ACO σε ένα πρόβλημα n-bit παγίδας	56
	Σύγκλιση προσβασιμότητας στου ACO σε ένα πρόβλημα n-bit παγίδας	58
4.2.4	Πειραματικά Αποτελέσματα	60
4.2.5	Συμπεράσματα	60
	Αναφορές	62

## Κεφάλαιο 1

### Βελτιστοποίηση Αποικίας Μυρμηγκιών

Αυτό το κεφάλαιο παρουσιάζει μία περίληψη της βελτιστοποίησης αποικίας μυρμηγκιών (ACO) - μία μετα-ευριστική εμπνευσμένη από τη συμπεριφορά πραγματικών μυρμηγκιών. Η Βελτιστοποίηση Αποικίας Μυρμηγκιών (ACO) προτάθηκε από τον Dorigo και συναδέλφους [Dorigo και άλλοι, 1991] ως μία μέθοδος επίλυσης δύσκολων προβλημάτων συνδυαστικής βελτιστοποίησης.

Οι ACO αλγόριθμοι μπορούν να θεωρηθούν μέρος της νοημοσύνης σμηνών, το ερευνητικό πεδίο που μελετάει αλγορίθμους εμπνευσμένους από τη παρατήρηση της συμπεριφοράς των σμηνών. Οι αλγόριθμοι νοημοσύνης σμηνών επινοήθηκαν από απλά χαρακτηριστικά που συνεργάζονται διαμέσου αυτό-οργάνωσης, χωρίς καμία μορφή κεντρικού ελέγχου πάνω από τα μέλη του σμήνους. Μία λεπτομερής περίληψη των αρχών αυτό-οργάνωσης προωθούμενη από αυτούς τους αλγορίθμους, όπως επίσης και προβλήματα από τη βιολογία, μπορούν να βρεθούν στο [Camazine και άλλοι, 2003]. Πολλοί αλγόριθμοι νοημοσύνης σμηνών έχουν προταθεί στη βιβλιογραφία. Για μία περίληψη στο τομέα της νοημοσύνης σμηνών, παραπέμπουμε στο [Bonabeau και άλλοι, 1999].

#### 1.1 Από τη Βιολογία στους Αλγορίθμους

Η βελτιστοποίηση αποικίας μυρμηγκιών εμπνεύστηκε από τη παρατήρηση της συμπεριφοράς πραγματικών μυρμηγκιών. Σε αυτό το τμήμα, παρουσιάζουμε ένα αριθμό από παρατηρήσεις που προέκυψαν από πειράματα με πραγματικά μυρμηγκία, και δείχνουμε πώς οι παρατηρήσεις ενέπνευσαν τη δημιουργία του μετα-ευριστικού ACO.

##### 1.1.1 Μυρμηγκία

Ένας από τους πρώτους ερευνητές που εξέτασε τη κοινωνική συμπεριφορά των εντόμων ήταν ο Γάλλος εντομολόγος Pierre-Paul Grassé. Στις δεκαετίες του 40 και 50 του 20ου αιώνα, παρατηρούσε τη συμπεριφορά των τερμιτών - συγκεκριμένα, τα είδη *Bellicositermes natalensis* και *Cubitermes*. Ανακάλυψε [Grassé, 1946] ότι αυτά τα έντομα είναι ικανά να αντιδρούν σε αυτό που εκείνος ονόμασε «εκφραστικό ερέθισμα», σήματα που ενεργοποιούν μια γενετικά κωδικοποιημένη αντίδραση. Παρατήρησε [Grassé, 1959] ότι τα αποτελέσματα αυτών των αντιδράσεων μπορούν να δράσουν ως νέα ερεθίσματα και για τα έντομα που τα παρήγαγαν αλλά και για τα άλλα έντομα στην αποικία. Ο Grassé χρησιμοποίησε τον όρο στιγμαργία (*stigmergy*) [Grassé, 1959] για να περιγράψει αυτό τον ιδιαίτερο τύπο έμμεσης επικοινωνίας στην οποία "οι εργάτες διεγείρονται από την απόδοση που πέτυχαν".

Τα δύο κύρια χαρακτηριστικά της στιγμαργίας που τη διαφοροποιούν από τις άλλες έννοιες της επικοινωνίας είναι:

- η φυσική, μη συμβολική φύση της πληροφορίας που αφήνεται από τα επικοινωνούντα μυρμηγκία και αντιστοιχεί σε μία τροποποίηση των καταστάσεων του φυσικού περιβάλλοντος που επισκεπτόνται τα μυρμηγκία και
- η τοπική φύση της αφημένης πληροφορίας, η οποία μπορεί να προσεγγιστεί μόνο από εκείνα τα έντομα που επισκέπτονται το χώρο που αφήθηκε (ή της άμεσης γειτονιάς του).

Παραδείγματα της στιγμαργίας μπορούν να παρατηρηθούν στις αποικίες των μυρμηγκιών. Σε πολλά είδη μυρμηγκιών, τα μυρμηγκία πηγαίνουν κι έρχονται από μία πηγή τροφής αποθέτοντας στο έδαφος μία ουσία που ονομάζεται φερομόνη. Άλλα μυρμηγκία είναι ικανά να μυρίσουν αυτή τη φερομόνη, και η παρουσία της επιδράει στην επιλογή του μονοπατιού - δηλαδή, τείνουν να ακολουθήσουν ισχυρές συγκεντρώσεις φερομόνης. Η φερομόνη που εναποτέθηκε στο έδαφος σχηματίζει ένα ίχνος φερομόνης, και επιτρέπει στα μυρμηγκία να βρίσκουν καλές πηγές τροφής που έχουν προηγουμένως αναγνωριστεί από άλλα μυρμηγκία.

Ορισμένοι ερευνητές ανακάλυψαν πειραματικά αυτή την εναπόθεση φερομόνης και τη ακόλουθουμένη συμπεριφορά για να τη κατανοήσουν καλύτερα και να είναι ικανοί να την ποσοτικοποιούν. Ο Deneubourg και άλλοι [Deneubourg και άλλοι, 1990] έκαναν ένα πείραμα επονομαζόμενο "πείραμα δυαδικής γέφυρας". Χρησιμοποίησαν μυρμηγκία *Linepithema humile* (γνωστά και ως Αργεντίνικα μυρμηγκία). Η φωλιά των μυρμηγκιών ήταν συνδεδεμένη με μία πηγή τροφής από δύο γέφυρες ίσου μήκους. Τα μυρμηγκία μπορούσαν ελεύθερα να επιλέξουν ποιά διαδρομή να χρησιμοποιήσουν όταν ψάχνουν για φαγητό και το φέρνουν πίσω στη φωλιά. Η συμπεριφορά τους παρατηρήθηκε για μία χρονική περίοδο.

Σε αυτό το πείραμα, αρχικά δεν υπάρχει φερομόνη στις δύο γέφυρες. Τα μυρμηγκία ξεκινούν να εξερευνούν το περιβάλλον της φωλιάς και τελικά διασχίζουν μία από τις γέφυρες και φτάνουν τη πηγή της τροφής. Όταν περπατούν από τη πηγή της τροφής προς τα πίσω, τα μυρμηγκία εναποθέτουν φερομόνη στη γέφυρα που χρησιμοποιούν. Αρχικά, κάθε μυρμηγκί επιλέγει τυχαία κάθε μία από τις γέφυρες. Παρόλα αυτά, εξαιτίας τυχαίων διακυμάνσεων, μετά από μερικό χρόνο θα υπάρχει περισσότερη φερομόνη αποθηκευμένη σε μία από τις δύο γέφυρες παρά στην άλλη. Επειδή τα μυρμηγκία τείνουν να ακολουθούν το ισχυρότερο ίχνος φερομόνης, η γέφυρα που έχει περισσότερη φερομόνη θα ελκύει περισσότερα μυρμηγκία. Αυτό διαδοχικά κάνει το ίχνος φερομόνης να μεγαλώνει ισχυρότερα, μέχρι η αποικία των μυρμηγκιών να συγκλίνει άμεσα στη χρήση της ίδιας γέφυρας.

Σε επίπεδο αποικίας αυτή η συμπεριφορά, βασισμένη στην αυτοκατάλυση, δηλαδή, στην προώθηση θετικής ανάδρασης, μπορεί να χρησιμοποιηθεί από τα μυρμηγκία στην εύρεση του συντομότερου μονοπατιού μεταξύ μία πηγής τροφής και της φωλιάς τους. Αυτό περιγράφηκε σε ένα άλλο πείραμα που διεξήχθη από τον Goss και άλλους [Goss και άλλοι, 1989], στο οποίο οι δύο γέφυρες δεν ήταν ίδιου μήκους: η μία ήταν σημαντικά μακρύτερη από την άλλη. Σε αυτή τη περίπτωση επειδή οι δυο κλάδοι μοιάζουν τα μυρμηγκία διαλέγουν τυχαία έναν από τους δύο κλάδους. Ενας δεύτερος μηχανισμός παίζει σημαντικό ρόλο: τα μυρμηγκία που επέλεξαν κατά τύχη τη συντομότερη γέφυρα ήταν επίσης τα πρώτα που έφτασαν στη τροφή και όταν επέστρεφαν στη φωλιά επέλεξαν τη συντομότερη γέφυρα με μεγαλύτερη πιθανότητα αφού αυτή είχε ένα ισχυρότερο ίχνος φερομόνης. Συνεπώς, τα μυρμηγκία-χάρης στο μηχανισμό ακολουθώ και αποθηκεύω φερομόνη-γρήγορα συγκλιναν στη χρήση της συντομότερης γέφυρας.

Στο επόμενο τμήμα εξηγούμε πώς αυτά τα αποτελέσματα και ευρήματα χρησιμοποιήθηκαν για να αναπτυχθούν οι αλγόριθμοι βελτιστοποίησης.

### 1.1.2 Αλγόριθμοι

Παρακινούμενοι από τα ενδιαφέροντα αποτελέσματα των πειραμάτων που περιγράφονται στο προηγούμενο τμήμα, ο Goss και άλλοι [Goss και άλλοι, 1989] ανέπτυξαν ένα μοντέλο για να εξηγήσουν τη συμπεριφορά που παρατηρήθηκε στο πείραμα της δυαδικής γέφυρας. Ας υποθέσουμε ότι μετά από  $t$  χρονικές μονάδες από την αρχή του πειράματος,  $m_1$  μυρμηγκία χρησιμοποίησαν τη πρώτη γέφυρα και  $m_2$  την δεύτερη, η πιθανότητα  $p_1$  να επιλέξει το  $(m+1)$ -οστό μυρμηγκί τη πρώτη γέφυρα μπορεί να δοθεί από:

$$p_1(m+1) = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h} \quad (1.1)$$

όπου οι παράμετροι  $k$  και  $h$  χρειάστηκαν για να ρυθμίσουμε το μοντέλο στα πειραματικά δεδομένα. Η πιθανότητα το  $(m+1)$ -οστό μυρμηγκί να επιλέξει τη δεύτερη γέφυρα είναι

$$p_2(m+1) = 1 - p_1(m+1)$$

Οι προσομοιώσεις του Monte Carlo, που έτρεξαν για να δοκιμάσουν πώς το μοντέλο ανταποκρίνεται στα πραγματικά δεδομένα [Pasteels και άλλοι, 1987], έδειξαν πολύ καλή προσαρμογή για  $k \approx 20$  και  $h \approx 2$ .

Αυτό το βασικό μοντέλο, που εξηγεί τη συμπεριφορά των πραγματικών μυρμηγκιών, μπορεί να χρησιμοποιηθεί ως μία έμπνευση για τη σχεδίαση τεχνητών μυρμηγκιών που λύνουν προβλήματα βελτιστοποίησης με ένα παρόμοιο τρόπο. Στο παραπάνω περιγραφόμενο παράδειγμα συμπεριφοράς μυρμηγκιών στην αναζήτηση τροφής, η στιγμεργιακή (stigmergic)



επικοινωνία συμβαίνει μέσω της φερομόνης που τα μυρμήγκια εναποθέτουν στο έδαφος. Αναλογικά, στα τεχνητά μυρμήγκια το επίπεδο φερομόνης μπορεί να προσομοιωθεί από κατάλληλες μεταβλητές φερομόνης που συνδέονται με τις καταστάσεις προβλήματος ενώ κατασκευάζουν λύσεις στο πρόβλημα βελτιστοποίησης. Επιπλέον, σύμφωνα με το μοντέλο στιγμεργικής επικοινωνίας, τα τεχνητά μυρμήγκια θα μπορούσαν να έχουν μόνο τοπική πρόσβαση σε αυτές της μεταβλητές φερομόνης.

Επομένως, τα κύρια χαρακτηριστικά στιγμεργίας που αναφέρονται στα προηγούμενα τμήματα μπορούν να επεκταθούν σε τεχνητούς αντιπροσώπους από:

- αντιστοιχίζοντας μεταβλητές καταστάσεων στα διάφορα προβλήματα καταστάσεων και
- δίνοντας στους αντιπροσώπους μόνο τοπική πρόσβαση σε αυτές τις μεταβλητές

Μία άλλη σημαντική όψη της συμπεριφοράς αναζήτησης τροφής των πραγματικών μυρμηγκιών που ενδέχεται να χρησιμοποιηθεί από τα τεχνητά μυρμήγκια είναι η σύζευξη μεταξύ του μηχανισμού αυτοκατάλυσης και του έμμεσου υπολογισμού των λύσεων. Με τον έμμεσο υπολογισμό λύσης, εννοούμε το γεγονός ότι συντομότερα μονοπάτια (που αντιστοιχούν σε λύσεις χαμηλότερου κόστους στη περίπτωση τεχνητών μυρμηγκιών) ολοκληρώνονται νωρίτερα από τα μακρύτερα, και συνεπώς λαμβάνουν ενίσχυση φερομόνης γρηγορότερα. Έμμεσος υπολογισμός λύσης συνδεδεμένος με αυτοκατάλυση μπορεί να είναι πολύ αποτελεσματικός: όσο πιο συντομότερο το μονοπάτι, πιο συντομα αποθηκεύεται η φερομόνη, και περισσότερα μυρμήγκια χρησιμοποιούν το συντομότερο μονοπάτι. Εάν χρησιμοποιηθεί κατάλληλα, μπορεί να είναι ένας ισχυρός μηχανισμός σε βασισμένους στο πληθυσμό αλγορίθμους βελτιστοποίησης (π.χ., σε εξελικτικούς αλγορίθμους [Holland, 1975 · Fogel, 1995] Η αυτοκατάλυση υλοποιήθηκε από τον μηχανισμό επιλογής/αναπαραγωγής).

Η στιγμεργία μαζί με τον έμμεσο υπολογισμό λύσης και την αυτοκαταλυτική συμπεριφορά, έδωσαν ανάπτυξη στον ACO. Η βασική ιδέα του ACO ακολουθεί από πολύ κοντά την βιολογική έμπνευση. Επομένως, υπάρχουν πολλές ομοιότητες μεταξύ πραγματικών και τεχνητών μυρμηγκιών. Και οι πραγματικές και οι τεχνητές αποικίες μυρμηγκιών συγκροτήθηκαν από πληθυσμό ατόμων που δουλεύουν μαζί για να επιτύχουν ένα συγκεκριμένο σκοπό. Μία αποικία είναι ένας πληθυσμός από απλούς, ανεξάρτητους, ασύγχρονους αντιπροσώπους που συνεργάζονται για να βρουν μία καλή λύση στο πρόβλημα. Στη περίπτωση των πραγματικών μυρμηγκιών, το πρόβλημα είναι να βρουν το φαγητό, ενώ στη περίπτωση των τεχνητών μυρμηγκιών, είναι να βρουν μία καλή λύση σε ένα δοσμένο πρόβλημα βελτιστοποίησης. Ένα μόνο μυρμήγκι (είτε ένα πραγματικό είτε ένα τεχνητό μυρμήγκι) είναι ικανό να βρει μία λύση στο πρόβλημα του, αλλά μόνο η συνεργασία μεταξύ πολλών ατόμων μέσω της στιγμεργίας τα καθιστά ικανά να βρουν καλές λύσεις.

Στη περίπτωση των πραγματικών μυρμηγκιών, εναποθέτουν και αντιδρούν σε μία χημική ουσία επονομαζόμενη φερομόνη. Τα πραγματικά μυρμήγκια, απλά την **εναποθέτουν** στο έδαφος καθώς περπατάνε. Τα τεχνητά μυρμήγκια ζούνε σε ένα τεχνητό κόσμο, γι'αυτό το λόγο τροποποιούν μόνο αριθμητικές τιμές (ονομάστηκαν αναλογικά τεχνητές φερομόνες) συνδεδεμένες με τις διαφορετικές καταστάσεις των προβλημάτων. Μία ακολουθία από τιμές φερομόνης συνδεδεμένη με τις καταστάσεις των προβλημάτων ονομάζεται τεχνητό ίχνος φερομόνης. Στον ACO, τα τεχνητά ίχνη φερομόνης είναι οι μοναδικές έννοιες επικοινωνίας μεταξύ των μυρμηγκιών. Ένας μηχανισμός ανάλογος με την εξάτμιση της φυσικής φερομόνης στις πραγματικές αποικίες μυρμηγκιών επιτρέπει στα τεχνητά μυρμήγκια να *ξεχνούν* τη πέρασμένη ιστορία και να εστιάζουν σε νέες ερευνητικές κατευθύνσεις.

Όπως τα πραγματικά μυρμήγκια, τα τεχνητά μυρμήγκια μπορούν να δημιουργήσουν τις λύσεις τους διαδοχικά μετακινούμενα από τη μία κατάσταση προβλήματος στην άλλη. Τα πραγματικά μυρμήγκια απλά περπατούν, επιλέγοντας την κατεύθυνση βασισμένη σε τοπικές συγκεντρώσεις φερομόνης και την τακτική στοχαστικών αποφάσεων. Τα τεχνητά μυρμήγκια επίσης δημιουργούν λύσεις βήμα προς βήμα, κινούμενα μέσω διαθέσιμων καταστάσεων των προβλημάτων και παίρνοντας στοχαστικές αποφάσεις σε κάθε βήμα.

Υπάρχουν παρόλα αυτά σημαντικές διαφορές μεταξύ πραγματικών και τεχνητών μυρμηγκιών:

- Τα τεχνητά μυρμήγκια ζούνε σε ένα διακριτό κόσμο - κινούνται διαδοχικά διαμέσου ενός πεπερασμένου συνόλου από τις καταστάσεις του προβλήματος.

- Η ενημέρωση φερομόνης (δηλαδή η φερομόνη που αποθηκεύεται και εξατμίζεται) δεν ολοκληρώνεται με τον ίδιο ακριβώς τρόπο στα τεχνητά και στα πραγματικά μυρμηγκία. Μερικές φορές η αναβάθμιση φερομόνης γίνεται μόνο από μερικά από τα τεχνητά μυρμηγκία, και συχνά μόνο αφού μια λύση έχει κατασκευαστεί.
- Μερικές υλοποιήσεις τεχνητών μυρμηγκιών χρησιμοποιούν συμπληρωματικούς μηχανισμούς που δεν υπάρχουν στη περίπτωση των πραγματικών μυρμηγκιών. Για παράδειγμα έχουν όραση, δυνατότητα τοπικής αναζήτησης, αντιστροφή θέσης, κ.τ.λ.

## 1.2 Η Μεταευριστική της Βελτιστοποίησης Αποικίας Μυρμηγκιών.

Η βελτιστοποίηση αποικίας μυρμηγκιών (ACO) τυποποιήθηκε μέσα σε μία μεταευριστική συνδυαστικής βελτιστοποίησης από τον Dorigo και άλλους [Dorigo και Di Caro, 1999 · Dorigo και άλλοι, 1999 · Dorigo και Stützle, 2004] και από τότε χρησιμοποιήθηκε για να καταπιαστεί με συνδυαστικά προβλήματα βελτιστοποίησης.

Δοθέντος ενός προβλήματος συνδυαστικής βελτιστοποίησης (COP), το πρώτο βήμα για την εφαρμογή του ACO στη λύση του αποτελείται από τον ορισμό ενός ικανοποιητικού μοντέλου. Αυτό τότε χρησιμοποιείται για να οριστεί η κεντρική συνιστώσα του ACO: το μοντέλο φερομόνης.

Πρώτα, μία μεταβλητή στιγμιαίας απόφασης  $X_i = v_i^j$  (δηλαδή η μεταβλητή  $X_i$  με μία τιμή  $v_i^j$  εκχωρημένη από το πεδίο της  $D_i$ ), ονομάζεται συνιστώσα λύσης και δηλώνεται με  $c_{ij}$ . Το σύνολο όλων των δυνατών συνιστωσών δηλώνεται με  $C$ . Μία παράμετρος ίχνους φερομόνης  $T_{ij}$  συνδέεται τότε με κάθε συνιστώσα  $c_{ij}$ . Το σύνολο όλων των παραμέτρων ίχνους φερομόνης δηλώνεται με  $T$ . Η τιμή της παραμέτρου ίχνους φερομόνης  $T_{ij}$  δηλώνεται με  $\tau_{ij}$  (και ονομάζεται τιμή φερομόνης. Οι τιμές της φερομόνης είναι γενικά μία συνάρτηση του αλγορίθμου επανάληψης  $t$ :  $\tau_{ij} = \tau_{ij}(t)$ ). Αυτή η τιμή φερομόνης χρησιμοποιείται και ενημερώνεται από τον ACO αλγόριθμο κατά τη διάρκεια της έρευνας. Αυτό επιτρέπει τη μοντελοποίηση της κατανομής πιθανότητας για τις διαφορές συνιστώσες της λύσης.

Στον ACO, τα τεχνητά μυρμηγκία κατασκευάζουν μία λύση σε ένα πρόβλημα συνδυαστικής βελτιστοποίησης διερχόμενα τον επονομαζόμενο κατασκευαστικό γράφο,  $G_c(V, E)$ . Ο πλήρως συνδεδεμένος κατασκευαστικός γράφος αποτελείται από ένα σύνολο από κορυφές  $V$  και ένα σύνολο από ακμές  $E$ . Το σύνολο των συνιστωσών  $C$  ενδέχεται να συνδέεται είτε με το σύνολο των κορυφών  $V$  του γράφου  $G_c$ , είτε με το σύνολο των ακμών  $E$ . Τα μυρμηγκία κινούνται από κορυφή σε κορυφή κατά μήκος των ακμών του γράφου, κατασκευάζοντας αυξητικά μία *μερική λύση*. Επιπλέον, τα μυρμηγκία αποθηκεύουν μία προκαθορισμένη ποσότητα φερομόνης στις συνιστώσες, που είναι, είτε οι κορυφές είτε οι ακμές που διέρχονται. Η ποσότητα  $\Delta_T$  της αποθηκευμένης φερομόνης μπορεί να εξαρτάται από τη ποιότητα της λύσης που βρέθηκε. Διαδοχικά μυρμηγκία εκμεταλλεύονται τη πληροφορία φερομόνης ως οδηγό για περισσότερο αποδοτικές περιοχές στο χώρο αναζήτησης.

Η ACO μεταευριστική δείχνεται στον Αλγόριθμο 1. Αποτελείται από ένα βήμα αρχικοποίησης και ένα βρόχο πάνω από τρεις αλγοριθμικές συνιστώσες. Μία μόνο επανάληψη του βρόχου αποτελείται από κατασκευαστικές λύσεις όλων των μυρμηγκιών, την (προαιρετική) βελτίωση τους με τη χρήση ενός αλγορίθμου τοπικής αναζήτησης, και μία ενημέρωση των φερομονών. Ακολουθώντας, εξηγούμε αυτές τις τρεις αλγοριθμικές συνιστώσες πιο λεπτομερώς.

### Αλγόριθμος 1 Μεταευριστική βελτιστοποίησης αποικίας μυρμηγκιών

---

```

Ανέθεσε παραμέτρους, αρχικοποίησε ίχνη φερομόνης
while συνθήκες τερματισμού δεν ικανοποιούνται do
Κατασκεύασε Μυρμηγκικές Λύσεις
Εφάρμοσε Τοπική Αναζήτηση {προαιρετικό}
Ενημέρωσε Φερομόνες
end while

```

---

**Κατασκεύασε Μυρμηγκικές Λύσεις:** Ένα σύνολο από  $m$  τεχνητά μυρμηγκία κατασκευάζουν λύσεις από τα στοιχεία ενός πεπερασμένου συνόλου από διαθέσιμες συνιστώσες λύσεων  $C = \{c_{ij}\}$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, |D_i|$ . Μία κατασκευή λύσης ξεκινάει από μία κενή μερική λύση  $s^p = \emptyset$ .

Τότε, σε κάθε κατασκευαστικό βήμα, η τρέχουσα μερική λύση  $s^p$  επεκτείνεται προσθέτοντας μία εφικτή συνιστώσα λύσης από το σύνολο των εφικτών περιοχών  $N(s^p) \subseteq C$ . Η διαδικασία κατασκευής λύσεων μπορεί να θεωρηθεί ως ένα μονοπάτι του κατασκευαστικού γράφου  $G_C = (V, E)$ . Τα επιτρεπόμενα μονοπάτια στο  $G_C$  είναι επι του παρόντος έμμεσα ορισμένα από ένα μηχανισμό κατασκευής λύσης που ορίζει το σύνολο  $N(s^p)$  ως προς μία μερική λύση  $s^p$ .

Η επιλογή μίας συνιστώσας λύσης από το  $N(s^p)$  γίνεται πιθανοθεωρητικά σε κάθε κατασκευαστικό βήμα. Οι ακριβείς κανόνες για την πιθανοθεωρητική επιλογή των συνιστωσών της λύσης ποικίλει ανάμεσα σε διαφορετικές παραλλαγές του ACO. Ο καλύτερα γνωστός κανόνας είναι εκείνος του Συστήματος Μυρμηγκίων (AS) [Dorigo και άλλοι, 1996]:

$$p(c_{ij} | s^p) = \frac{\tau_{ij}^a \cdot \eta(c_{ij})^b}{\sum_{c_{il} \in N(s^p)} \tau_{il}^a \cdot \eta(c_{il})^b}, \quad \forall c_{ij} \in N(s^p) \quad (1.2)$$

όπου  $\tau_{ij}$  είναι η τιμή της φερομόνης που συνδέεται με τη συνιστώσα  $c_{ij}$ , και  $\eta(\cdot)$  είναι μία συνάρτηση που αναθέτει σε κάθε κατασκευαστικό βήμα μία ευριστική τιμή σε κάθε εφικτή συνιστώσα λύσης  $c_{ij} \in N(s^p)$ . Οι τιμές που δίνονται από αυτή τη συνάρτηση ονομάζονται *ευριστική πληροφορία*. Επιπλέον, τα  $a$  και  $b$  είναι θετικές παράμετροι, των οποίων οι τιμές καθορίζουν τη σχετική σημασία της φερομόνης έναντι της ευριστικής πληροφορίας. Η Εξίσωση 1.2 είναι μία γενίκευση της Εξίσωσης 1.1 που παρουσιάζεται στο τμήμα 1.1: Η τυποποίηση του ACO ακολουθεί από κοντά την βιολογική έμπνευση.

**Εφαρμοσε Τοπική Αναζήτηση:** Μόλις οι λύσεις έχουν κατασκευαστεί, και πριν την ενημέρωση των φερομονών, συχνά ενδέχεται να απαιτούνται μερικές προαιρετικές ενέργειες. Αυτές συχνά ονομάζονται *κακές ενέργειες (daemon actions)*, και μπορούν να χρησιμοποιηθούν για να υλοποιήσουν προβλήματα συγκεκριμένων ενεργειών, που δεν μπορούν να πραγματοποιηθούν από ξεχωριστά μυρμηγκία. Οι πιο χρησιμοποιημένες κακές ενέργειες αποτελούνται από την εφαρμογή τοπικής αναζήτησης στις κατασκευασμένες λύσεις: οι τοπικά βελτιστοποιημένες λύσεις τότε χρησιμοποιούνται για να επιλεχτεί ποιές φερομόνες θα ενημερωθούν.

**Ενημέρωσε Φερομόνες:** Ο στόχος της ενημέρωσης φερομόνης είναι να αυξηθούν οι τιμές φερομόνης που αντιστοιχούν στις καλές και υποσχόμενες λύσεις, και να μειώσουμε αυτές που αντιστοιχούν στις κακές. Συνήθως, αυτό επιτυγχάνεται (i) μειώνοντας όλες τις τιμές της φερομόνης διαμέσου της *εξάτμισης φερομόνης*, και (ii) με την αύξηση των επιπέδων φερομόνης που αντιστοιχούν στο επιλεγμένο σύνολο από καλές λύσεις  $S_{upd}$ :

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \rho \cdot \sum_{s \in S_{upd} | c_{ij} \in s} F(s) \quad (1.3)$$

όπου  $S_{upd}$  είναι το σύνολο των λύσεων που χρησιμοποιήθηκαν για την ενημέρωση,  $\rho \in (0, 1]$  είναι μία παράμετρος που ονομάζεται ρυθμός εξάτμισης, και  $F: S \rightarrow \mathbb{R}_0^+$  είναι μία συνάρτηση τέτοια ώστε  $f(s) < f(s') \Rightarrow F(s) \geq F(s')$ ,  $\forall s \neq s' \in S$ . Η  $F(\cdot)$  ονομάζεται *συνάρτηση προσαρμογής*.

Η εξάτμιση φερομόνης χρειάζεται για να αποφύγουμε μία πάρα πολύ γρήγορη σύγκλιση του αλγορίθμου. Υλοποιεί μία χρήσιμη μορφή *λησμονιάς*, ευνοώντας την εξερεύνηση νέων περιοχών στον χώρο αναζήτησης. Διαφορετικοί ACO αλγόριθμοι, όπως για παράδειγμα το Σύστημα Αποικίας Μυρμηγκιών (ACS) [Dorigo και Gambardella, 1997] ή το Σύστημα Μυρμηγκιών *MAX-MIN (MMAS)* [Stützle και Hoos, 2000] διαφέρουν στον τρόπο που ενημερώνουν τη φερομόνη.

Επικαιροποιήσεις του κανόνα ενημέρωσης που παρουσιάζονται στην Εξίσωση 1.3 λαμβάνονται από τις διάφορες προδιαγραφές του  $S_{upd}$ , που σε πολλές περιπτώσεις είναι ένα υποσύνολο του  $S_{iter} \cup \{s_{bs}\}$ , όπου  $S_{iter}$  είναι το σύνολο από τις λύσεις που κατασκευάστηκαν στην τρέχουσα επανάληψη, και  $s_{bs}$  είναι η *καλύτερη μέχρι τώρα λύση*, που είναι, η καλύτερη λύση που βρέθηκε

από τη πρώτη αλγοριθμική επανάληψη. Ένα πολύ γνωστό παράδειγμα είναι ο AS-κανόνας ενημέρωσης, που είναι, ο κανόνας ενημέρωσης του Συστήματος Μυρμηγκιών [Dorigo και άλλοι, 1996], όπου:

$$S_{\text{upd}} \leftarrow S_{\text{iter}} \quad (1.4)$$

Ένα παράδειγμα κανόνα ενημέρωσης φερομόνης που χρησιμοποιήθηκε συχνότερα στην πράξη είναι ο IB-κανόνας ενημέρωσης (όπου IB σημαίνει *καλύτερη-επανάληψη*):

$$S_{\text{upd}} \leftarrow \arg \max_{s \in S_{\text{iter}}} F(s). \quad (1.5)$$

Ο IB κανόνας εισάγει μία ισχυρότερη μεροληψία σε σχέση με τις καλές λύσεις που βρέθηκαν με τον AS-κανόνα ενημέρωσης. Αν και αυτό αυξάνει τη ταχύτητα με την οποία βρίσκονται οι καλές λύσεις, επίσης αυξάνει την πιθανότητα πρόωρης σύγκλισης. Μία ακόμα ισχυρότερη μεροληψία εισάγεται από τον BS-κανόνα ενημέρωσης, όπου ο BS αναφέρεται στη χρήση της καλύτερης λύσης  $s_{\text{bs}}$ . Σε αυτή τη περίπτωση, το  $S_{\text{upd}}$  προκαθορίζεται στο  $\{s_{\text{bs}}\}$ . Στη πράξη, οι ACO αλγόριθμοι που χρησιμοποιούν παραλλαγές από τους IB ή BS-κανόνες ενημέρωσης και που επιπρόσθετα περιέχουν μηχανισμούς αποφυγής πρόωρης σύγκλισης, επιτυγχάνουν καλύτερα αποτελέσματα από εκείνους που χρησιμοποιούν τον AS-κανόνα ενημέρωσης.

### 1.2.1 Παράδειγμα: Το Πρόβλημα του Περιοδεύοντα Πωλητή

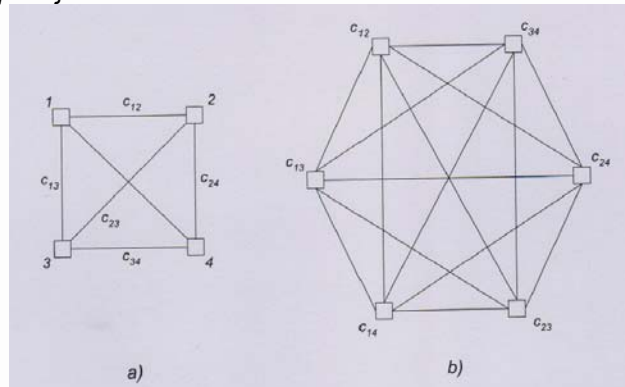
Ένας από τους καλύτερους τρόπους να επεξηγήσουμε πώς λειτουργεί ο ACO μεταευριστικός, είναι μέσω της εφαρμογής του στο πρόβλημα του περιοδεύοντα πωλητή (TSP). Το πρόβλημα TSP αποτελείται από ένα σύνολο από τοποθεσίες (πόλεις) και έναν περιοδεύοντα πωλητή που πρέπει να επισκεφτεί όλες τις τοποθεσίες μία και μόνο φορά. Οι αποστάσεις μεταξύ των τοποθεσιών δίνονται και ο στόχος είναι να βρεθεί μία Χαμιλτονιανή διαδρομή με το ελάχιστο μήκος. Το πρόβλημα έχει αποδειχτεί ότι είναι NP hard [Lawler και άλλοι, 1985].

Η εφαρμογή του ACO στο TSP εφαρμόζεται ακριβώς. Οι κινήσεις μεταξύ των τοποθεσιών γίνονται οι συνιστώσες των λύσεων-δηλαδή, η κίνηση από τη πόλη  $i$  στη πόλη  $j$  γίνεται μία συνιστώσα λύσης  $c_{ij} \equiv c_{ji}$ . Ο κατασκευαστικός γράφος  $G_C = (V, E)$  ορίζεται αντιστοιχίζοντας το σύνολο των τοποθεσιών με το σύνολο  $V$  των κορυφών του γράφου. Επειδή μπορούμε να κινηθούμε από τη μία πόλη στην άλλη, ο κατασκευαστικός γράφος είναι πλήρως συνεκτικός και ο αριθμός των κορυφών είναι ίσος με τον αριθμό των τοποθεσιών που ορίζονται από το πρόβλημα. Επιπροσθέτως, τα μήκη των ακμών μεταξύ των κορυφών είναι ανάλογα με τις αποστάσεις μεταξύ των τοποθεσιών που αναπαριστώνται από τις κορυφές. Η φερομόνη αντιστοιχεί με το σύνολο  $E$  των ακμών του γράφου. Ένα παράδειγμα του προκύπτοντα κατασκευαστικού γράφου  $G_C$  παρουσιάζεται στην εικόνα 1.1α.

Τα μυρμηγκία κατασκευάζουν τις λύσεις ως ακολούθως. Κάθε μυρμηγκι ξεκινάει από μία τυχαία επιλεγμένη τοποθεσία (κορυφή του γράφου  $G_C$ ). Τότε, σε κάθε κατασκευαστικό βήμα κινείται κατα μήκος των ακμών του γράφου. Κάθε μυρμηγκι κρατάει στην μνήμη του το μονοπάτι διαμέσου του γράφου, και σε διαδοχικά βήματα επιλέγει απο τις ακμές που δεν οδηγούν στις κορυφές που ήδη επισκέφθηκε. Ένα μυρμηγκι έχει κατασκευάσει μία λύση μόλις έχει επισκεφθεί όλες τις κορυφές του γράφου. Σε κάθε κατασκευαστικό βήμα ένα μυρμηγκι επιλέγει πιθανοθεωρητικά την ακμή που θα επιλέξει μεταξύ των διαθέσιμων (εκείνων που οδηγούν στις μη επισκεφθείσες κορυφές). Ο ακριβής κανόνας εξαρτάται από την υλοποίηση, ένα παράδειγμα είναι η Εξίσωση 1.2. Μόλις όλα τα μυρμηγκια έχουν τελειώσει τη διαδρομή τους, η φερομόνη στις ακμές ενημερώνεται σύμφωνα με τις πιθανές υλοποιήσεις της Εξίσωσης 1.3. Ο ACO έδειξε ότι εκτελείται σχεδόν καλά στο TSP [Stützle και Dorigo, 1999].

Αξίζει να σημειωθεί ότι είναι επίσης πιθανό να αντιστοιχιστεί το σύνολο των συνιστωσών της λύσης του TSP (ή κάθε άλλου προβλήματος συνδυαστικής βελτιστοποίησης) με το σύνολο των κορυφών  $V$  παρά με το σύνολο των ακμών  $E$  του κατασκευαστικού γράφου  $G_C$ . Για το TSP, αυτό θα σήμαινε αντιστοιχισμα των κινήσεων μεταξύ των τοποθεσιών με το σύνολο  $V$  των κορυφών του κατασκευαστικού γράφου, και των τοποθεσιών με το σύνολο  $E$  των ακμών. Το αντίστοιχο παράδειγμα κατασκευαστικού γράφου για ένα TSP 4 πόλεων παρουσιάζεται στην εικόνα 1.1β. Όταν χρησιμοποιείται αυτή η προσέγγιση, η διαδικασία κατασκευής της λύσης των μυρμηγκιών θα πρέπει επίσης να τροποποιηθεί κατάλληλα: τα μυρμηγκια θα πρέπει να

κινηθούν από τη μία κορυφή στην άλλη του κατασκευαστικού γράφου επιλέγοντας με αυτό το τρόπο τις *συνδέσεις μεταξύ των πόλεων*.



**Εικόνα 1.1:** Παράδειγμα κατασκευαστικών γράφων για ένα TSP τεσσάρων πόλεων. α) όταν οι συνιστώσες συνδέονται με τις ακμές τους γράφου, β) όταν οι συνιστώσες συνδέονται με τις κορυφές τους γράφου.

Είναι σημαντικό να σημειωθεί ότι και οι δύο τρόποι ορισμού του κατασκευαστικού γράφου είναι σωστοί και οι δύο μπορούν να χρησιμοποιηθούν στη πράξη. Αναλογα με το πρόβλημα, ο ένας ενδέχεται να είναι πιο αποτελεσματικός από τον άλλο. Για παράδειγμα, στο Πρόβλημα Προγραμματισμού Πανεπιστημιακών Μαθημάτων (UTCP) ο δεύτερος δείχνει καλύτερα προσαρμοσμένος [Socha Και άλλοι, 2002].

### 1.3 Κύριες Παραλλαγές του ACO

Αρκετές παραλλαγές του ACO προτάθηκαν στη βιβλιογραφία. Παρουσιάζουμε τα κύρια χαρακτηριστικά των πιο επιτυχημένων μαζί με μερικές από τις εφαρμογές τους.

Παρακάτω παρουσιάζουμε το *Σύστημα Μυρμηγκιών* (ant system)-τη πρώτη υλοποίηση ενός ACO αλγορίθμου- και στη συνέχεια το Σύστημα Μυρμηγκιών MAX-MIN και το Σύστημα Αποικίας Μυρμηγκιών. Για να επεξηγήσουμε καθαρά τις διαφορές μεταξύ αυτών, χρησιμοποιούμε το πρόβλημα του περιοδεύοντα πωλητή, όπως περιγράφεται στο Τμήμα 1.2.1.

#### 1.3.1 Το Σύστημα Μυρμηγκιών (AS)

Το *Σύστημα Μυρμηγκιών* (AS) ήταν ο πρώτος ACO αλγόριθμος που προτάθηκε στη βιβλιογραφία [Dorigo και άλλοι, 1991 · Dorigo και άλλοι, 1996]. Το κύριο χαρακτηριστικό του είναι ότι οι τιμές της φερομόνης ενημερώνονται από όλα τα μυρμηγκία που ολοκλήρωσαν τη διαδρομή. Η ενημέρωση φερομόνης  $\tau_{ij}$ , για την ακμή που συνδέει τις πόλεις  $i$  και  $j$ , πραγματοποιήθηκε ως εξής:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (1.6)$$

όπου  $\rho$  είναι ο ρυθμός εξατμίσης,  $m$  είναι ο αριθμός των μυρμηγκιών, και  $\Delta\tau_{ij}^k$  είναι η ποσότητα φερομόνης ανά μονάδα μήκους στην ακμή  $(i, j)$  από το  $k$ -οστό μυρμηγκί:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{εάν το μυρμηγκί } k \text{ χρησιμοποίησε την ακμή } (i, j) \text{ στη διαδρομή του,} \\ 0 & \text{αλλιώς,} \end{cases} \quad (1.7)$$

όπου  $Q$  είναι μία σταθερά, και  $L_k$  είναι το μήκος της διαδρομής του  $k$ -οστού μυρμηγκιού.

Όταν κατασκευάζουν τις λύσεις, τα μυρμηγκία στον AS διασχίζουν ένα κατασκευαστικό γράφο και παίρνουν πιθανοθεωρητικές αποφάσεις σε κάθε κορυφή. Η πιθανότητα μετάβασης  $p_{ij}^k$  του  $k$ -οστού μυρμηγκιού που κινείται από τη πόλη  $i$  στη πόλη  $j$  δίνεται από:

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^a \cdot \eta_{ij}^\beta}{\sum_{l \in \text{επιτρεπόμενα}_k} \tau_{il}^a \cdot \tau_{lj}^\beta} & \text{εάν } j \in \text{επιτρεπόμενα}_k, \\ 0 & \text{αλλιώς,} \end{cases} \quad (1.8)$$

όπου το  $\text{επιτρεπόμενα}_k$  είναι η λίστα με τις πόλεις που δεν έχει επισκεφτεί το  $k$ -οστό μυρμήγκι, και  $a$  και  $\beta$  είναι παράμετροι που ελέγχουν τη σχετική σημασία της φερομόνης έναντι της ευριστικής πληροφορίας  $\eta_{ij}$  που δίνεται από:

$$\eta_{ij} = \frac{1}{d_{ij}}, \quad (1.9)$$

όπου  $d_{ij}$  είναι το μήκος της ακμής  $(i, j)$ .

Αρκετές υλοποιήσεις του αλγορίθμου AS έχουν εφαρμοστεί σε διάφορα συνδυαστικά προβλήματα βελτιστοποίησης. Η πρώτη και πιο γνωστή είναι η εφαρμογή στο TSP [Dorigo και άλλοι, 1991]. Παρόλα αυτά, ο AS χρησιμοποιήθηκε επίσης επιτυχώς για την αντιμετώπιση και άλλων συνδυαστικών προβλημάτων. Ο AS-QAP [Maniezzo και άλλοι, 1994 · Maniezzo και Colonna, 1999] αλγόριθμος χρησιμοποιήθηκε για την αντιμετώπιση του προβλήματος ανάθεσης τετραγώνων (QAP), ο AS-VRP [Bullheimer και άλλοι, 1998, 1999] για τα προβλήματα δρομολογησης οχημάτων (VRP).

### 1.3.2 MAX-MIN Σύστημα Μυρμηγκιών

Το MAX-MIN Σύστημα Μυρμηγκιών (MMAS) είναι μία βελτίωση πάνω στη αρχική ιδέα του Συστήματος Μυρμηγκιών. Ο MMAS προτάθηκε από τους Stützle και Hoos [Stützle και Hoos, 2000] και εισάγει τις ακόλουθες δύο αλλαγές:

- μόνο το καλύτερο μυρμήγκι μπορεί να ενημερώσει τα ίχνη φερομόνης, και
- οι ελάχιστες και οι μέγιστες τιμές της φερομόνης είναι περιορισμένες.

Η εξίσωση 1.6 παίρνει από τώρα την ακόλουθη νέα μορφή:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{\text{best}}, \quad (1.10)$$

όπου  $\Delta\tau_{ij}^{\text{best}}$  είναι ο κανόνας ενημέρωσης φερομόνης ορισμένος από:

$$\Delta\tau_{ij}^{\text{best}} = \begin{cases} \frac{Q}{L_{\text{best}}} & \text{εάν το καλύτερο μυρμήγκι} \\ & \text{χρησιμοποίησε την ακμή } (i, j) \text{ στη διαδρομή του,} \\ 0 & \text{αλλιώς} \end{cases} \quad (1.11)$$

Το  $L_{\text{best}}$  είναι το μήκος της διαδρομής του καλύτερου μυρμηγκιού. Αυτό ενδέχεται να είναι (υπόκειται στην απόφαση του σχεδιαστή του αλγορίθμου) είτε η καλύτερη διαδρομή που βρέθηκε στη τρέχουσα επανάληψη-καλύτερη-επανάληψη,  $L_{\text{ib}}$ -είτε η καλύτερη λύση που βρέθηκε από την αρχή του αλγορίθμου-καλύτερη μέχρι τώρα,  $L_{\text{bs}}$ -ή ένας συνδυασμός των δύο.

Για τα όρια της ελάχιστης και μέγιστης επιτρεπόμενης τιμής φερομόνης, αντίστοιχα  $\tau_{\min}$  και  $\tau_{\max}$ , οι Stützle και Hoos προτείνουν ότι πρέπει να επιλεχθούν εμπειρικά, αναλογα με το πρόβλημα. Η μέγιστη τιμή  $\tau_{\max}$  μπορεί να υπολογιστεί αναλυτικά υπό τον όρο ότι το μήκος της διαδρομής του βέλτιστου μυρμηγκιού είναι γνωστό. Στη περίπτωση του TSP, το  $\tau_{\max}$  δίνεται από:

$$\tau_{\max} = \frac{1}{\rho} \cdot \frac{1}{L^*}, \quad (1.12)$$

όπου  $L^*$  είναι το μήκος της βέλτιστης διαδρομής. Η ελάχιστη τιμή φερομόνης  $\tau_{\min}$  πρέπει να επιλεγεί με προσοχή καθώς έχει μία κάπως ισχυρότερη επίδραση στην απόδοση του αλγορίθμου. Παρουσιάζουν μία αναλυτική προσέγγιση για την εύρεση αυτής της τιμής βασισμένη στη πιθανότητα  $p_{\text{best}}$  που ένα μυρμήγκι κατασκευάζει τη καλύτερη διαδρομή μέχρι τώρα. Αυτό

γίνεται ως εξής. Πρώτα, υποθέτουμε ότι σε κάθε κατασκευαστικό βήμα ένα μυρμήγκι έχει ένα σταθερό αριθμό από διαθέσιμες επιλογές  $k$ . Συνεπώς, η πιθανότητα ένα μυρμήγκι να παρει τη σωστή απόφαση (δηλαδή, την απόφαση που ανήκει στην ακολουθία αποφάσεων που οδηγούν στην κατασκευή της καλύτερης λύσης που έχει βρεθεί μέχρι τώρα) σε κάθε από τα  $n$  βήματα δίνεται από το  $p_{dec} = \frac{1}{k} \sqrt[n]{p_{best}}$ . Ο αναλυτικός τύπος που προτείνουν για την εύρεση του  $\tau_{min}$  είναι:

$$\tau_{min} = \frac{\tau_{max} \cdot (1 - p_{dec})}{k \cdot p_{dec}} \quad (1.13)$$

Για περισσότερες λεπτομέρειες στο πώς να επιλεχθούν τα  $\tau_{max}$  και  $\tau_{min}$ , αναφερόμαστε στους [Stützle και Hoos, 2000]. Είναι σημαντικό να αναφέρουμε εδώ ότι έχει δειχθεί [Socha και άλλοι, 2002] ότι για κάποια προβλήματα η επιλογή μίας κατάλληλης τιμής του  $\tau_{min}$  είναι πιο εύκολο να δοθεί πειραματικά παρά αναλυτικά.

Η διαδικασία της ενημέρωσης φερομόνης στον MMAS αποφασίζεται από την επαλήθευση ότι οι τιμές της φερομόνης είναι μέσα στα απαιτούμενα όρια:

$$\tau_{ij} = \begin{cases} \tau_{max} & \text{εάν } \tau_{ij} > \tau_{max}, \\ \tau_{min} & \text{εάν } \tau_{ij} < \tau_{min}. \end{cases} \quad (1.14)$$

Το MMAS παρέχει μίας σημαντική βελτίωση στην απόδοση του Συστήματος Μυρμηγκιών. Ενώ οι πρώτες υλοποιήσεις επικεντρώθηκαν στο TSP [Stützle και Hoos, 2000], αργότερα εφαρμόστηκε σε πολλά άλλα προβλήματα συνδυαστικής βελτιστοποίησης όπως είναι το πρόβλημα αναθεσης τετραγώνων QAP [Stützle και Hoos, 1998] ή το πρόβλημα προγράμματος των πανεπιστημιακών μαθημάτων (UTCP) [Socha και άλλοι, 2002].

### 1.3.3 Σύστημα Αποικίας Μυρμηγκιών

Μία άλλη βελτίωση πάνω στο πρωτότυπο Σύστημα Μυρμηγκιών (AS) είναι το Σύστημα Αποικίας Μυρμηγκιών (ACS) εισηγμένο από τους Gambardella και Dorigo [Gambardella και Dorigo, 1996 · Dorigo και Gambardella, 1997]. Η πιο ενδιαφέρουσα συνεισφορά του ACS είναι η εισαγωγή μίας τοπικής ενημέρωσης φερομόνης επιπλέον στην ενημέρωση φερομόνης που πραγματοποιήθηκε στο τέλος της κατασκευαστικής διαδικασίας (εδώ ονομάζεται αδρανής (offline) ενημέρωση φερομόνης).

Η τοπική ενημέρωση φερομόνης πραγματοποιείται από όλα τα μυρμήγκια μετά από κάθε κατασκευαστικό βήμα. Κάθε μυρμήγκι την εφαρμόζει μόνο στην τελευταία ακμή που διέσχισε:

$$\tau_{ij} = (1 - \phi) \cdot \tau_{ij} + \phi \cdot \tau_0 \quad (1.15)$$

όπου  $\phi \in (0, 1]$  είναι ο συντελεστής μείωσης φερομόνης, και  $\tau_0$  είναι η αρχική τιμή της φερομόνης.

Ο βασικός στόχος της τοπικής ενημέρωσης είναι να διαφοροποιεί την αναζήτηση που πραγματοποιήθηκε από διαδοχικά μυρμήγκια κατά τη διάρκεια μίας επανάληψης. Στη πραγματικότητα, μειώνοντας τη συγκέντρωση φερομόνης στις ακμές που περάστηκαν κατά τη διάρκεια μία επανάληψης ενθαρρύνει τα επόμενα μυρμήγκια να επιλέξουν άλλες ακμές και γι'αυτό το λόγο να παράγουν διαφορετικές λύσεις. Αυτό τον κάνει λιγότερο κατάλληλο διότι αρκετά μυρμήγκια παράγουν ίδιες λύσεις κατά τη διάρκεια μιας επανάληψης.

Η αδρανής (offline) ενημέρωση φερομόνης, όμοια με το MMAS, εφαρμόζεται στο τέλος κάθε επανάληψης από ένα μόνο μυρμήγκι (εκείνο που βρήκε τη καλύτερη λύση στην επανάληψη). Παρ όλα αυτά, ο τύπος ενημέρωσης είναι λίγο διαφορετικός:

$$\tau_{ij} \leftarrow \begin{cases} (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij} & \text{εάν η ακμή } (i, j) \text{ ανήκει στο } T_{best} \\ \tau_{ij} & \text{αλλιώς,} \end{cases} \quad (1.16)$$

και στη περίπτωση του TSP,  $\Delta\tau_{ij} = \frac{1}{L_{best}}$ .

Μία άλλη σημαντική διαφορά μεταξύ του AS και του ACS είναι στον κανόνα απόφασης που χρησιμοποιείται από τα μυρμήγκια κατά τη διάρκεια της κατασκευαστικής διαδικασίας. Τα μυρμήγκια στον ACS χρησιμοποιούν τον επονομαζόμενο ψευδοτυχαίο αναλογικό κανόνα: η πιθανότητα για ένα μυρμήγκι να κινηθεί από τη πόλη  $i$  στη πόλη  $j$  εξαρτάται από μία τυχαία

μεταβλητή  $q$  ομοιόμορφα κατανομημένη πάνω στο  $[0,1]$ , και μία παράμετρο  $q_0$ . Αν  $q \leq q_0$ , τότε  $j = \arg \max_{i \in N(e^p)} \{\tau_{ii}^\beta\}$ , αλλιώς χρησιμοποιείται η Εξίσωση (1.8).

Ο ACS αναπτύχθηκε αρχικά για το πρόβλημα του περιοδεύοντα πωλητή [Gambardena και Dorigo, 1996 · Dorigo και Gambardella, 1997], αλλά χρησιμοποιήθηκε αργότερα για την αντιμετώπιση ποικίλων συνδυαστικών προβλημάτων, περιλαμβάνοντας τη δρομολόγηση οχήματος [Bianchi και άλλοι, 2004] και το πρόβλημα προγράμματος των πανεπιστημιακών μαθημάτων (UTCP) [Socha και άλλοι, 2003].



## Κεφάλαιο 2

### Ο ACO για Συνεχή Χωρία

Η συνδυαστική βελτιστοποίηση-όπως και το όνομα υποδηλώνει –ασχολείται με την εύρεση βέλτιστων *συνδυασμών* ή *μεταθέσεων* στις διαθέσιμες συνιστώσες των προβλημάτων. Γι' αυτό το λόγο, απαιτείται το πρόβλημα να είναι διαμερισμένο σε ένα πεπερασμένο σύνολο από συνιστώσες, και ο αλγόριθμος συνδυαστικής βελτιστοποίησης προσπαθεί να βρει το βέλτιστο συνδυασμό ή μετάθεση. Πολλά προβλήματα του πραγματικού κόσμου μπορούν να εκφράσουν ως προβλήματα συνδυαστικής βελτιστοποίησης (COPs) με ακριβή τρόπο. Υπάρχει όμως μία σημαντική ομάδα προβλημάτων τα οποία δεν είναι σ' αυτή τη περίπτωση: η ομάδα των προβλημάτων βελτιστοποίησης που απαιτούν επιλογή τιμών για συνεχείς μεταβλητές. Τέτοια προβλήματα μπορούν να αντιμετωπιστούν με έναν αλγόριθμο συνδυαστικής βελτιστοποίησης μόνο όταν τα συνεχή πεδία τιμών των επιτρεπόμενων τιμών μετατρέπουν σε πεπερασμένα σύνολα. Αυτό δεν είναι πάντα βολικό, ειδικά όταν το αρχικό δυνατό πεδίο είναι ευρύ, και η ανάλυση που απαιτείται είναι πολύ μεγάλη. Σε αυτές τις περιπτώσεις, οι αλγόριθμοι που μπορούν να χειριστούν συνεχείς μεταβλητές συνήθως εκτελούνται καλύτερα.

Αυτό το κεφάλαιο παρουσιάζει ένα τρόπο για να εφαρμοστεί αποτελεσματικά ο ACO - ένας αλγόριθμος αρχικά ανεπτυγμένος για να αντιμετωπίσει τα COPs σε συνεχή προβλήματα βελτιστοποίησης.

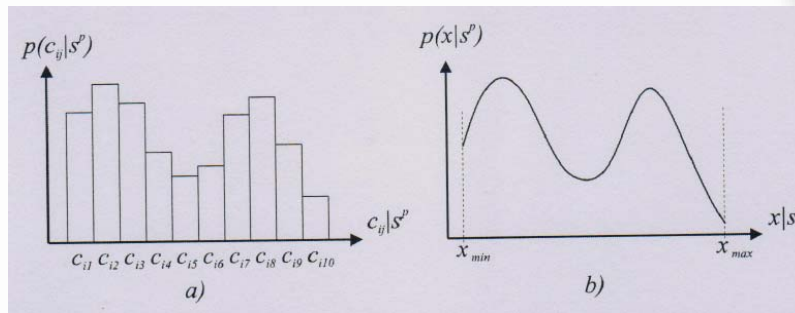
Από την αρχή του ACO ως εργαλείο συνδυαστικής βελτιστοποίησης, οι ερευνητές προσπάθησαν να τον χρησιμοποιήσουν επίσης για την αντιμετώπιση συνεχών προβλημάτων. Παρόλα αυτά, η εφαρμογή του μεταευσριστικού μηχανισμού του ACO σε συνεχή χωρία δεν ήταν προφανής. Οι διαφορετικές μέθοδοι που προτάθηκαν συχνά παρέκλιναν από τη αρχική τυποποίηση του ACO.

Εδώ παρουσιάζεται ένας τρόπος να εκτείνουμε τον ACO σε συνεχή χωρία χωρίς να χρειαστεί να κάνουμε καμία κρίσιμη αλλαγή στη δομή του. Για να το δηλώσουμε αυτό, δηλώνουμε ότι ο ACO επεκτάθηκε σε συνεχή χωρία από τον  $ACO_{\square}$ . Στόχος είναι η παρουσίαση της βασικής ιδέας της εφαρμογής του  $ACO_{\square}$  σε συνεχή χωρία όπως επίσης μία υλοποίηση του που εκτελείται καλά σε προβλήματα δοκιμασίας επιδόσεων. Για να έχουμε μία καλή αντίληψη της απόδοσης του  $ACO_{\square}$ , τον συγκρίνουμε όχι μόνο με άλλες σχετικές με μυρμηγκία μεθόδους, αλλά επίσης και με άλλες μεταευσριστικές που χρησιμοποιήθηκαν για συνεχή βελτιστοποίηση.

#### 2.1 Ο Αλγόριθμος

Η κυρία ιδέα στο τρόπο με τον οποίο δουλεύει ο ACO είναι η αυξητική κατασκευή των λύσεων βασισμένη στην μεροληπτική πιθανοθεωρητική επιλογή των συνιστωσών των λύσεων. Στον ACO που εφαρμόστηκε σε προβλήματα συνδυαστικής βελτιστοποίησης, το σύνολο των διαθέσιμων συνιστωσών λύσης ορίστηκε από τη τυποποίηση του προβλήματος. Σε κάθε κατασκευαστικό βήμα, τα μυρμηγκία κάνουν μία πιθανοθεωρητική επιλογή της συνιστώσας λύσης  $c_i$  από το σύνολο  $N(s^p)$  των διαθέσιμων συνιστωσών σύμφωνα με την εξίσωση 1.2. Οι πιθανότητες που συνδέονται με τα στοιχεία του συνόλου  $N(s^p)$  κάνουν μία *διακριτή κατανομή πιθανότητας* (Εικόνα 2.1a) απ' όπου ένα μυρμηγκι δειγματίζει για να επιλέξει μία συνιστώσα που θα προστεθεί στην τρέχουσα μερική λύση  $s^p$ .

Η θεμελιώδης ιδέα που διέπει τον  $ACO_{\square}$  είναι η αλλαγή από τη χρήση *διακριτής* κατανομής πιθανότητας στη χρήση μίας *συνεχούς*, που είναι, μία *συνάρτηση πυκνότητας πιθανότητας* (PDF) (Εικόνα 2.1b). Στον  $ACO_{\square}$ , αντί να επιλέξουμε μία συνιστώσα  $c_{ij} \in N(s^p)$  σύμφωνα με την Εξίσωση 1.2, ένα μυρμηγκι δειγματίζει απο μία συνάρτηση πυκνότητας πιθανότητας



**Εικόνα 2.1:** (α) Διακριτή κατανομή πιθανότητας  $P_a(c_{ij} | s^p)$  ενός πεπερασμένου συνόλου  $\{c_{i1}, \dots, c_{i10}\} \in N(s^p)$  διαθέσιμων συνιστωσών. (β) Συνεχή συνάρτηση πυκνότητας πιθανότητας  $P_c(x | s^p)$  με πιθανό εύρος  $x \in [x_{\min}, x_{\max}]$ . Ο  $y$  άξονας και στις δύο γραφικές παραστάσεις δείχνει τη πιθανότητα  $p..$

Η ACO μεταευριστική βρίσκει προσεγγιστικές λύσεις σε ένα πρόβλημα βελτιστοποίησης επαναλαμβάνοντας τα ακόλουθα δύο βήματα:

1. Οι υποψήφιες λύσεις κατασκευάστηκαν με ένα πιθανοθεωρητικό τρόπο χρησιμοποιώντας μία κατανομή πιθανότητας πάνω στο χώρο αναζήτησης.
2. Οι υποψήφιες λύσεις χρησιμοποιούνται για τη τροποποίηση της κατανομής πιθανότητας με ένα τρόπο που μεροληπτεί στη μελλοντική δειγματοληψία έναντι των υψηλής ποιότητας λύσεων.

Οι ACO αλγόριθμοι για προβλήματα συνδυαστικής βελτιστοποίησης κάνουν χρήση ενός μοντέλου φερομόνης για να κατασκευάσουν πιθανοθεωρητικά λύσεις. Ένα μοντέλο φερομόνης είναι ένα σύνολο από τις ονομαζόμενες *παραμέτρους ίχνους φερομόνης*. Οι αριθμητικές τιμές αυτών των παραμέτρων ίχνους φερομόνης (που είναι, οι τιμές της φερομόνης) αντανακλούν την εμπειρία αναζήτησης του αλγορίθμου. Χρησιμοποιούνται για να προκαταλάβουν τη κατασκευή της λύσης στη διάρκεια του χρόνου έναντι των χωρίων του χώρου αναζήτησης που περιέχουν υψηλής ποιότητας λύσεις.

Στα ACO συνδυαστικά προβλήματα, οι τιμές της φερομόνης συνδέθηκαν με ένα πεπερασμένο σύνολο από διακριτές τιμές που σχετίζονται με τις απόφασεις που κάνουν τα μυρμήγκια. Αυτό επιτρέπει να αναπαριστώνται οι τιμές της φερομόνης στη μορφή του πίνακα φερομόνης. Αυτό δεν είναι δυνατό στη συνεχή περίπτωση, αφού ο αριθμός των δυνατών τιμών δεν είναι πεπερασμένος. Γι'αυτό το λόγο, ο  $ACO_{\square}$  χρησιμοποιεί ένα *αρχείο λύσης* ως τρόπο περιγραφής της κατανομής φερομόνης πάνω στο χώρο αναζήτησης. Το αρχείο λυσης περιέχει ένα αριθμό από πλήρεις λύσεις στο πρόβλημα. Ενώ το μοντέλο φερομόνης σε συνδυαστική βελτιστοποίηση μπορεί να ιδωθεί ως μία έμμεση (implicit) μνήμη του ιστορικού αναζήτησης, ένα αρχείο λύσης είναι μία άμμεση μνήμη.

Η βασική ροή του  $ACO_{\square}$  αλγορίθμου είναι ως εξής. Ως πρώτο βήμα, το αρχείο λύσης παίρνει αρχικές τιμές. Τότε, σε κάθε επανάληψη, ένας αριθμός από λύσεις κατασκευάζεται πιθανοθεωρητικά από τα μυρμήγκια. Αυτές οι λύσεις ενδέχεται να βελτιωθούν από ένα μηχανισμό βελτίωσης (για παράδειγμα, τοπική αναζήτηση). Τελικά, το αρχείο λύσης ενημερώνεται με τις παραγόμενες λύσεις. Ακολούθως περιγράφουμε τις συνιστώσες του  $ACO_{\square}$  με λεπτομέρειες.

### 2.1.1 Δομή Αρχείου, αρχικοποίηση, και ενημέρωση

Ο  $ACO_{\square}$  κρατάει ιστορικό της διαδικασίας αναζήτησης αποθηκεύοντας λύσεις σε ένα *αρχείο λύσης*  $T$  διάστασης  $|T|=k$ . Δοθέντος ενός  $n$ -διάστατου συνεχούς προβλήματος βελτιστοποίησης και  $k$  λύσεων, ο  $ACO_{\square}$  αποθηκεύει στο  $T$  τις τιμές των λύσεων των  $n$  μεταβλητών και τη τιμή των αντικειμενικών συναρτήσεων. Η τιμή της  $i$ -οστής μεταβλητής της  $j$ -οστής λύσης ακολούθως δηλώνεται ως  $s_j^i$ . Η εικόνα 2.2 δείχνει τη δομή του αρχείου λύσης.

		1	2	...	i	...	n		
$s_1$	1	$s_1^1$	$s_1^2$	...	$s_1^i$	...	$s_1^n$	$f(s_1)$	$\omega_1$
$s_2$	2	$s_2^1$	$s_2^2$	...	$s_2^i$	...	$s_2^n$	$f(s_2)$	$\omega_2$
...	...	...	...	...	...	...	...	...	...
$s_j$	j	$s_j^1$	$s_j^2$	...	$s_j^i$	...	$s_j^n$	$f(s_j)$	$\omega_j$
...	...	...	...	...	...	...	...	...	...
$s_k$	k	$s_k^1$	$s_k^2$	...	$s_k^i$	...	$s_k^n$	$f(s_k)$	$\omega_k$

**Εικόνα 2.2:** Η δομή του αρχείου λύσης. Οι λύσεις στο αρχείο ταξινομήθηκαν σύμφωνα με τη ποιότητα τους (δηλαδή, τη τιμή της αντικειμενικής συνάρτησης  $f(s)$ ), γι'αυτό το λόγο η θέση μιας λύσης στο αρχείο πάντα αντιστοιχεί στη τάξη της.

Πριν από το ξεκίνημα του αλγορίθμου, το αρχείο αρχικοποιείται με  $k$  τυχαίες λύσεις. Σε κάθε αλγοριθμική επανάληψη, πρώτα, ένα σύνολο από  $m$  λύσεις παράγεται από τα μυρμήγκια και προστίθεται σε εκείνες του  $T$ . Από αυτό το σύνολο των  $k+m$  λύσεων, οι  $m$  χειρότερες αφαιρούνται. Οι υπόλοιπες  $k$  λύσεις ταξινομούνται σύμφωνα με τη ποιότητα τους (δηλαδή, τη τιμή της αντικειμενικής συνάρτησης) και ταξινομήθηκαν στο νέο  $T$ . Με αυτό το τρόπο, η διαδικασία αναζήτησης είναι προκατειλημμένη έναντι των καλύτερων λύσεων που βρέθηκαν κατά τη διάρκεια της αναζήτησης. Οι λύσεις στο αρχείο πάντα διατηρούνται ταξινομημένες ως προς την ποιότητα τους (δηλαδή, τις τιμές της αντικειμενικής συνάρτησης), έτσι ώστε η καλύτερη λύση να είναι στη κορυφή.

### 2.1.2 Πιθανοθεωρητική κατασκευή λύσης

Η κατασκευή των νέων λύσεων από τα μυρμήγκια πραγματοποιείται με έναν αυξητικό τρόπο, μεταβλητή προς μεταβλητή. Πρώτα, ένα μυρμήγκι επιλέγει πιθανοθεωρητικά μία από τις λύσεις στο αρχείο. Η πιθανότητα επιλογής της  $j$  λύσης δίνεται από:

$$p_j = \frac{\omega_j}{\sum_{r=1}^k \omega_r}, \quad (2.1)$$

όπου  $\omega_j$  είναι το βάρος συνδεδεμένο με τη λύση  $j$ . Το βάρος ενδέχεται να υπολογιστηκε χρησιμοποιώντας διάφορους τύπους ανάλογα με το πρόβλημα που αντιμετωπίζουμε. Στη συνέχεια εδώ χρησιμοποιούμε την Γκαουσιανή συνάρτηση  $g(\mu, \sigma) = g(1, qk)$ :

$$\omega_j = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(j-1)^2}{2q^2k^2}}, \quad (2.2)$$

όπου  $q$  είναι μία παράμετρος του αλγορίθμου και  $k$  είναι το μέγεθος του αρχείου. Ο μέσος της Γκαουσιανής συνάρτησης ορίστηκε σε 1, έτσι ώστε η καλύτερη λύση να έχει το μεγαλύτερο βάρος.

Η επιλογή της Γκαουσιανής συνάρτησης παρακινήθηκε από την ευελιξία της και τα μη γραμμικότητα της. Χάρη στη μη γραμμικότητα της, επιτρέπει έναν ευέλικτο έλεγχο πάνω στα βάρη. Είναι δυνατό να δώσουμε μεγαλύτερη πιθανότητα σε μερικές ηγετικές λύσεις, μειώνοντας σημαντικά τη πιθανότητα των υπολοίπων.

Τα μυρμήγκια αντιμετωπίζουν κάθε μεταβλητή  $i=1, \dots, n$  του προβλήματος ξεχωριστά. Παίρνει τη τιμή  $s_i^j$  της μεταβλητής  $i$  της επιλεγμένης  $j$  οστής λύσης και δειγματίζει στη γειτονία της. Αυτό γίνεται χρησιμοποιώντας μία συνάρτηση πυκνότητας πιθανότητας (PDF). Ξανά, όπως στη περίπτωση επιλογής βαρών, πολλές διαφορετικές συναρτήσεις ενδέχεται να χρησιμοποιηθούν. Μία PDF  $P(x)$  πρέπει παρόλα αυτά να ικανοποιεί τη συνθήκη:

$$\int_{-\infty}^{\infty} P(x)dx = 1 \quad (2.3)$$

Εδώ, όπως και σε άλλες παρουσιάσεις του  $ACO_{\square}$  [Socha και Dorigo, 2008], χρησιμοποιείται ως PDF η Γκαουσιανή συνάρτηση:

$$P(x) = g(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (2.4)$$

Η συνάρτηση έχει δύο παραμέτρους που πρέπει να οριστούν:  $\mu$ , και  $\sigma$ . Όταν λαμβάνουμε υπόψη τη μεταβλητή  $i$  της λύσης  $j$ , αντιστοιχίζουμε  $\mu \leftarrow s_j^i$ . Ακόμη, αντιστοιχίζουμε το  $\sigma$ :

$$\sigma \leftarrow \xi \sum_{r=1}^k \frac{|s_r^i - s_j^i|}{k-1}. \quad (2.5)$$

που είναι η μέση απόσταση μεταξύ της  $i$  οστής μεταβλητής της λύσης  $s_j$  και των  $i$  οστών μεταβλητών των άλλων λύσεων στο αρχείο, πολλαπλασιασμένο με μία παράμετρο  $\xi$ . Η παράμετρος  $\xi$  έχει μία επίδραση όμοια με εκείνη τους ρυθμούς εξάτμισης φερομόνης στον ACO. Όσο μεγαλύτερη είναι η τιμή του  $\xi$ , τόσο μικρότερη είναι η ταχύτητα σύγκλισης του αλγορίθμου. Ενώ ο ρυθμός εξάτμισης της φερομόνης στον ACO επιδρά στη μακροχρόνια μνήμη -δηλαδή οι χειρότερες λύσεις ξεχνιούνται γρηγορότερα- το  $\xi$  στον  $ACO_{\square}$  επιδρά στο τρόπο όπου χρησιμοποιείται η μακροχρόνια μνήμη - δηλαδή, οι νέες λύσεις θεωρήθηκαν *πιο κοντά* σε γνωστές καλές λύσεις.

Όλη αυτή η διαδικασία επαναλαμβάνεται για κάθε διάσταση  $i=1, \dots, n$  με τη σειρά του κάθε ενός από τα  $m$  μυρμηγκία.

## 2.2 Τοποθέτηση του $ACO_{\square}$

Ο  $ACO_{\square}$  είναι μέρος μίας σχετικά μεγάλης οικογένειας από αλγορίθμους για συνεχή βελτιστοποίηση. Σε αυτό το τμήμα, δίνουμε μία σύντομη περίληψη και συζητάμε, πώς ο  $ACO_{\square}$  μπορεί να τοποθετηθεί σε σχέση με τους άλλους.

Για συνεχή προβλήματα βελτιστοποίησης, ένα πλήθος από μεθόδους έχει προταθεί στη βιβλιογραφία. Αυτές περιέχουν μερικές σχετικές με μυρμηγκια μεθόδους [Bilchev και Parmee, 1995 · Monmarche και άλλοι, 2000 · Dréo και Siarry, 2002], καθώς και μερικές πιο γενικές εμπνευσμένες από τη μέθοδο των σιμίων όπως η Βελτιστοποίηση Σμήνους Σωματιδίων [Kennedy και Eberhart, 1995]. Υπάρχουν επίσης πολλές άλλες μεταεвриστικές που αναπτύχθηκαν αρχικά για συνδυαστική βελτιστοποίηση και αργότερα υιοθετήθηκαν στη συνεχή περίπτωση.

Επιπλέον, υπάρχουν και άλλες μέθοδοι που-όμοια με τον ACO-αμμεσα χρησιμοποιούν κάποια έννοια εκτίμησης κατανομής πιθανότητας. Πολλοί από αυτούς τους αλγορίθμους αναπαράχθηκαν από τη γενική κλάση των Εξελικτικών Αλγορίθμων (EAs). Τα παραδείγματα περιλαμβάνουν Εξελικτικές Στρατηγικές (ES) [Schwefel, 1981 · Ostermeier και άλλοι, 1994 · Hansen και Ostermeier, 2001], Επαναληπτικό Αλγόριθμο Εκτίμησης Πυκνότητας (IDEA) [Bosman και Thierens, 2000], ή το Μικτό Αλγόριθμο Μπαεσιανής Βελτιστοποίησης (MBOA) [Ocenasek και Schwarz, 2002]. Μερικές από αυτές, όμοια με τον ACO, χρησιμοποιήθηκαν αρχικά για συνδυαστική βελτιστοποίηση, και μόνο αργότερα υιοθετήθηκαν για να αντιμετωπίζουν συνεχή χωρία.

Επίσης με όλους τους αλγόριθμους που αναφέρθηκαν μέχρι τώρα, υπάρχουν και πολλοί αλγόριθμοι για συνεχή βελτιστοποίηση βασισμένοι στη κλίση. Είναι γρήγοροι, αλλά έχουν κάποιες αναγκαίες προϋποθέσεις. Είναι ικανοί να βρουν γρήγορα ένα τοπικό ελάχιστο, αλλά απαιτούν η συνάρτηση βελτιστοποίησης να είναι συνεχής και διαφορίσιμη. Παραδείγματα τέτοιων αλγορίθμων περιλαμβάνουν τη μέθοδο του Newton [Ralston και Rabinowitz, 1978], ή τον αλγόριθμο οπισθοδιάδοσης [Rumelhart και άλλοι, 1986] που χρησιμοποιήθηκαν για την εκπαίδευση τεχνητών νευρωνικών δικτύων. Η χρησιμότητα των βασισμένων στη κλίση αλγορίθμων είναι περιορισμένη εξαιτίας των αναγκαίων προϋποθέσεων που αναφέρθηκαν παραπάνω. Ο  $ACO_{\square}$  καθώς και όλοι οι άλλοι αλγόριθμοι για συνεχή βελτιστοποίηση που αναφέρθηκαν, δεν έχουν τέτοιους περιορισμούς, πράγμα το οποίο τους κάνει πιο γενικούς.

Τελευταίες, αλλά όχι λιγότερο σημαντικές, υπάρχουν άμεσες μέθοδοι αναζήτησης, όπως είναι η μέθοδος simplex, [Nelder και Mead, 1965], ή η μέθοδος του Powell [Powell, 1964]. Είναι γρήγορες και αποτελεσματικές στις περισσότερες περιπτώσεις, αλλά επιτρέπουν να βρουν μόνο το κοντινότερο τοπικό βέλτιστο. Είναι παρόλα αυτά ενδιαφέρουσες ως μια μέθοδος συμπληρωματική στον ACO ή άλλες μεταεριστικές. Δηλαδή, ενδέχεται να χρησιμοποιήθηκαν ως ρουτίνες τοπικής αναζήτησης για τις μεταεριστικές.

### 2.2.1 Ο ACO και Άλλοι Swarm-Based Αλγόριθμοι

Ο κύριος τύπος των βασισμένων σε σμηνη αλγορίθμων που θα να αναφέρουμε σε αυτό το τμήμα, είναι οι σχετικοί με μυρμηγκία αλγόριθμοι. Ένας αλγόριθμος βασισμένος στα σμηνη που δεν είναι σχετισμένος με μυρμηγκία θα αναφερθεί στο τέλος τους τμήματος.

Υπήρξαν προηγούμενες προσπάθειες εφαρμογής σχετικών με μυρμηγκία αλγορίθμων σε συνεχή χωρία. Κάποιες προσπάθειες, ήταν περισσότερο επιτυχείς από κάποιες άλλες, αλλά καμία από αυτές δεν ήταν μία επέκταση του ACO σε ένα συνεχές χωρίο. Αντιθέτως, ήταν νέοι αλγόριθμοι που επίσης άντλησαν την αρχική τους έμπνευση από τη συμπεριφορά των μυρμηγκιών. Στις παρακατω παραγράφους, παρουσιάζουμε σύντομα αυτούς τους αλγορίθμους και δείχνουμε πώς διαφέρουν από τον ACO.

Μία από τις πρώτες προσπάθειες να εφαρμοστεί ένας σχετισμένος με μυρμηγκία αλγόριθμος σε συνεχή προβλήματα βελτιστοποίησης ήταν ο Συνεχής ACO (CACO) [Bilchev και Parmee, 1995]. Στον CACO το μυρμηγκί αρχίζει από ένα σημείο, που ονομάστηκε φωλιά, τοποθετημένο κάπου στο χώρο αναζήτησης. Οι καλές λύσεις που βρέθηκαν αποθηκεύθηκαν ως ένα σύνολο από διανύσματα, προερχόμενα από τη φωλιά. Τα μυρμηγκία σε κάθε επανάληψη του αλγορίθμου επιλέγουν πιθανοθεωρητικά ένα από τα διανύσματα. Τότε συνεχίζουν την αναζήτηση από το τελικό σημείο του επιλεγμένου διανύσματος κάνοντας κάποιες τυχαίες κινήσεις από εκεί. Τα διανύσματα ενημερώνονται με τα καλύτερα αποτελέσματα που βρέθηκαν. Αν και οι συγγραφείς του CACO ισχυρίζονται ότι άντλησαν έμπνευση από τη πρωτότυπη τυποποίηση του ACO, υπάρχουν σημαντικές διαφορές. Εκείνοι εισάγουν την έννοια της *φωλιάς*, η οποία δεν υπάρχει στον μεταεριστικό ACO. Επίσης, ο CACO δεν εκτελεί μία αυξητική κατασκευή λύσεων, που είναι ένα από τα κύρια χαρακτηριστικά του ACO μεταεριστικού. Ο CACO ως εκ τούτου δεν πληρεί τις προϋποθέσεις για να είναι μία επέκταση του ACO.

Μία άλλη βασισμένη στα μυρμηγκία προσέγγιση στη συνεχή βελτιστοποίηση είναι ο αλγόριθμος API [Monmarché και άλλοι, 2000]. Ο API δεν ισχυρίζεται ότι είναι βασισμένος στον ACO μεταεριστικό. Τα μυρμηγκία εκτελούν την αναζήτηση τους ανεξάρτητα αλλά ξεκινώντας από την ίδια φωλιά (η φωλιά μετακινείται περιοδικά). Τα μυρμηγκία χρησιμοποιούν μόνο παράλληλο τρέξιμο, ένα είδος στρατηγικής στρατολόγησης. Είναι ο μόνος γνωστός αλγόριθμος μεταξύ των σχετικών με μυρμηγκία αλγορίθμων που εκδόθηκαν μέχρι τώρα και επιτρέπουν την αντιμετώπιση διακριτών και συνεχών προβλημάτων βελτιστοποίησης.

Η τρίτη βασισμένη στα μυρμηγκία προσέγγιση σε συνεχή βελτιστοποίηση είναι η Συνεχώς Αλληλεπιδρούσα Αποικία Μυρμηγκιών (CIAC) [Dréo και Siarry, 2002]. Ο CIAC χρησιμοποιεί δύο είδη επικοινωνίας μεταξύ των μυρμηγκιών: στιγμεργική πληροφορία (σημεία φερομόνης που τοποθετήθηκαν στο χώρο αναζήτησης) και άμεση επικοινωνία μεταξύ των μυρμηγκιών. Τα μυρμηγκία κινούνται μέσω του χώρου αναζήτησης ελκόμενα από τη φερομόνη που είναι απλωμένη σε διάφορα σημεία, και κατευθύνονται από κάποια άμεση πληροφορία μεταξύ των μυρμηγκιών. Αν και επίσης ο CIAC ισχυρίζεται ότι αντλεί την πρωτότυπη του έμπνευση από τον ACO, οι διαφορές είναι πολλές: υπάρχει μία απευθείας επικοινωνία μεταξύ των μυρμηγκιών και μη αυξητική κατασκευή των λύσεων. Όπως και ο CACO, έτσι και ο CIAC δεν πιστοποιείται ως μία επέκταση του ACO.

Τελικά, όπως αναφέρθηκε στην αρχή του τμήματος, υπάρχει ένας γνωστός βασισμένος στα σμηνη αλγόριθμος για συνεχή βελτιστοποίηση που δεν είναι σχετισμένος με τα μυρμηγκία. Ονομάζεται Βελτιστοποίηση Σωματιδίου Σμήνους (PSO) [Kennedy και Eberhart, 1995]. Ο PSO δουλεύει με ένα πληθυσμό από σωματίδια. Αυτά τα σωματίδια κινούνται στο χώρο αναζήτησης με μία συγκεκριμένη ταχύτητα. Η τιμή και η κατεύθυνση του διανύσματος ταχύτητας αλλάζουν σύμφωνα με τους προσελκυτές του χώρου αναζήτησης. Κάθε σωματίδιο αντιδρά σε δύο

τέτοιους προσελκυτές. Ο ένας είναι η καλύτερη τιμή που έχει βρεθεί από το σωματίδιο, και ο άλλος είναι η καλύτερη τιμή που έχει βρεθεί καθολικά. Ο PSO έχει δείχθει πειραματικά ότι εκτελείται καλά σε πολλά συνεχή προβλήματα βελτιστοποίησης.

### 2.2.2 Ο $ACO_{\square}$ και οι Εξελικτικοί Αλγόριθμοι

Ο  $ACO_{\square}$  για συνδυαστική βελτιστοποίηση είναι όμοιος με τους Εξελικτικούς Αλγόριθμους (EAs) από πολλές απόψεις. Αμφότεροι οι  $ACO_{\square}$  και EAs χρησιμοποιούν κάποια έννοια κατανομής πιθανότητας για να βρουν υποσχόμενες περιοχές στο χώρο αναζήτησης. Αυτή η ομοιότητα διατηρήθηκε όταν συγκρίνοντας τον  $ACO_{\square}$  με τους EAs τον αναπτύξαμε σε συνεχή χωρία.

Ο Kern και οι άλλοι [Kern και άλλοι, 2004] παρουσιάζουν αρκετούς εξελικτικούς αλγόριθμους για συνεχή βελτιστοποίηση-από απλούς μέχρι πολύ προηγμένους. Τους παρουσιάζουμε σύντομα.

Το σύνολο των αλγορίθμων που συγκρίθηκαν από τον Kern περιέχει τρεις εκδοχές των Εξελικτικών Στρατηγικών (ES), και δύο άλλους αλγορίθμους-το Μικτό Αλγόριθμο Μπαεσιανής Εκτιμητικής (MBOA), και τον Αλγόριθμο Επαναληπτικής Εκτίμησης Πυκνότητας (IDEA). Ο απλούστερος αλγόριθμος που χρησιμοποιήθηκε σε αυτή τη σύγκριση είναι (1+1) ES [Kern και άλλοι, 2004]. Είναι ένας απλός ES με ένα γεννήτορα που παράγει ένα απόγονο ανά επανάληψη. Μόνο τα άτομα που παρουσιάζουν υψηλή ποιότητα λύσης διατηρούνται. Ο επόμενος ES που περιλήφθηκε στην σύγκριση είναι η Εξελικτική Στρατηγική με Συσσωρευτική Προσαρμογή Μεγέθους Βήματος (CSA-ES) [Ostermeier και άλλοι, 1994 · Kern και άλλοι, 2004]. Υιοθετεί το καθολικό μέγεθος βήματος χρησιμοποιώντας το μονοπάτι που διασχίζεται από το γεννήτορα πληθυσμού πάνω από ένα αριθμό γενεών. Ο τρίτος ES που θεωρήθηκε είναι ο CMA-ES. ES με Υιοθέτηση Πίνακα Συνδιακύμανσης [Hansen και Ostermeier, 2001 · Kern και άλλοι, 2004]. Είναι μία επέκτασή του CSA-ES, με υιοθέτηση αποτυχαιοποίησης του πίνακα συνδιακύμανσης.

Ο πρώτος από τους δύο αλγορίθμους που δεν είναι ES είναι IDEA, προτάθηκε από τους Bosman και Thierens [Bosman και Thierens, 2000]. Επισημοποιεί τους Αλγόριθμους Εκτίμησης Κατανομής (EDA) σε συνεχή χωρία. Για να εκτιμηθεί η κατανομή του πληθυσμού γεννητορων, ο IDEA εκμεταλλεύεται το γεγονός ότι κάθε πολυμεταβλητή κατανομή πιθανότητας μπορεί να γραφτεί ως μία *δεσμευμένη παραγοντοποίηση (conditional factorization)* της ακόλουθης μορφής:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | x_{i+1}, x_{i+2}, \dots, x_n). \text{ Το πιθανοθεωρητικό μοντέλο του πληθυσμού γεννητόρων}$$

ανακατασκευάζεται σε κάθε γενιά.

Ο τελευταίος αλγόριθμος σ' αυτή τη σύγκριση είναι ο MBOA [Ocenasek και Schwarz, 2002]. Είναι ένα Μπαεσιανό δίκτυο με τοπικές δομές στη μορφή δέντρων αποφάσεων που συλλαμβάνει τις αμοιβαίες εξαρτήσεις μεταξύ των γεννητόρων. Ο πρώτος EDA που απασχόλησε το Μπαεσιανό μοντέλο με δέντρα αποφάσεων ήταν ο ιεραρχικός Αλγόριθμος Μπαεσιανής Εκτιμητικής (hBOA) [Pelikan και άλλοι, 2000]. Ο MBOA είναι μία επέκταση του hBOA από δυαδικά σε συνεχή χωρία. Στη πραγματικότητα, ο MBOA είναι ικανός να αντιμετωπίσει διακριτές και συνεχείς μεταβλητές ταυτόχρονα, όπως και ο  $ACO_{\square}$ .

Κάθε ένας από αυτούς τους αλγορίθμους είναι διαφορετικός, αλλά όλοι χρησιμοποιούν κάποιο τρόπο εκμάθησης και μοντελοποίησης άμμεσων κατανομών πιθανοτήτων. Υπάρχουν δύο τρόποι που αυτοί οι αλγόριθμοι χρησιμοποιούν τις κατανομές πιθανότητας. Όλες οι εκδόσεις του ES αυξητικά ενημερώνουν τις κατανομές πιθανότητας σε κάθε επανάληψη. Σε αντίθεση, ο IDEA και ο MBOA σε κάθε χρόνο τις κατασκευάζουν ολοκληρωτικά. Ο  $ACO_{\square}$  δρα εν προκειμένω ακόμα διαφορετικά. Όμοια με τον ACO για συνδυαστική βελτιστοποίηση, στον  $ACO_{\square}$  τα μυρμηγκία χρησιμοποιούν σε κάθε κατασκευαστικό βήμα μία διαφορετική, δυναμικά κατασκευασμένη κατανομή πιθανότητας. Αυτή η κατανομή εξαρτάται από τα προηγούμενα κατασκευαστικά βήματα και ενδέχεται να είναι διαφορετική για το κάθε μυρμηγκί. Συνεπώς, η  $ACO_{\square}$  προσέγγιση είναι πιο κοντά σε αυτό που κάνουν οι IDEA και MBOA, αλλά είναι από τις πιο λεπτομερείς- μερικές δυναμικά δημιουργημένες κατανομές πιθανότητας που χρησιμοποιήθηκαν κατά τη διάρκεια μίας επανάληψης.

Όλοι οι αλγόριθμοι εκμάθησης πιθανότητας του Kern και των άλλων χρησιμοποιούν κάποια μορφή της Γκαουσιανής συνάρτησης για τη μοντελοποίηση των κατανομών πιθανότητας. Οι (1+1)ES και CSA-ES χρησιμοποιούν ισότροπες Γκαουσιανές κατανομές. Ο IDEA χρησιμοποιεί μία τυχαία Γκαουσιανή κατανομή. Ο MBOA χρησιμοποιεί μία ιδέα κάπως όμοια με εκείνη που χρησιμοποιήθηκε από τον  $ACO_{\square}$ -Γκαουσιανή κεντρική κατανομή, αλλά ορισμένη σε διαμέρησεις του χώρου αναζήτησης.

### 2.3 Πειραματική Ρύθμιση και Αποτελέσματα

Σε αυτό το τμήμα, παρουσιάζουμε τη πειραματική ρύθμιση για τον υπολογισμό της επίδοσης του  $ACO_{\square}$  και των ληφθέντων αποτελεσμάτων. Για να έχουμε μία επισκόπηση της επίδοσης του  $ACO_{\square}$  σε σύγκριση με τις άλλες μεθόδους για συνεχή βελτιστοποίηση, χρησιμοποιούμε τις τυπικές συναρτήσεις δοκιμασίας επιδόσεων που χρησιμοποιήθηκαν στη βιβλιογραφία για τη παρουσίαση της απόδοσης διαφόρων μεθόδων και αλγορίθμων.

Προφανώς, δεν είναι πρακτικό να συγκρίνουμε τον  $ACO_{\square}$  με κάθε μέθοδο που χρησιμοποιήθηκε για συνεχή βελτιστοποίηση στο παρελθόν. Γι'αυτό το λόγο, αποφασίσαμε να περιορίσουμε τη σύγκριση μας σε άλλες μεταευστικές που χρησιμοποιήθηκαν για αυτό το σκοπό. Αποφασίσαμε να χωρίσουμε αυτές τις μεταευστικές σε τρεις ομάδες, βασισμένες στην ομοιότητα τους με τον  $ACO_{\square}$  :

- Μέθοδοι εκμάθησης πιθανότητας-μέθοδοι που ρητά μοντελοποιούν και δειγματίζουν κατανομές πιθανοτήτων.
- Συσχετισμένες με μυρμηγκία μέθοδοι - μέθοδοι που ισχυρίζονται ότι αντλούν την έμπνευση τους από τη συμπεριφορά των μυρμηγκιών.
- Άλλες μεταευστικές αρχικά ανεπτυγμένες για συνδυαστική βελτιστοποίηση και αργότερα υιοθετημένων σε συνεχή χωρία.

Χρησιμοποιήσαμε μία ελαφρώς διαφορετική πειραματική μέθοδος για κάθε σύγκριση για να κάνουμε τα αποτελέσματα που πήραμε από τον  $ACO_{\square}$  συγκρισιμα όσο το δυνατόν με εκείνα που ελήφθησαν στις άλλες μεθόδους.

Είναι σημαντικό να τονίσουμε ότι-σε αντίθεση με τη συνδυαστική βελτιστοποίηση-η σύγκριση των αλγορίθμων για συνεχή βελτιστοποίηση δεν γίνεται συνήθως βασισμένη στο χρόνο της κεντρικής μονάδας επεξεργασίας. Στη περίπτωση της συνδυαστικής βελτιστοποίησης, για κάθε αλγόριθμο δίνεται συνήθως η ίδια ποσότητα του χρόνου κεντρικής μονάδας επεξεργασίας και τα αποτελέσματα που λαμβάνονται μέσα σ αυτό το χρόνο συγκρίνονται. Αυτό κάνει τη σύγκριση των διαφορετικών αλγορίθμων περίπλοκη, αφού ο χρόνος της κεντρικής μονάδας επεξεργασίας εξαρτάται σημαντικά από τη προγραμματιστική γλώσσα που χρησιμοποιήθηκε, το μεταγλωττιστή, τις ικανότητες του προγραμματιστή, και απο τις μηχανές που χρησιμοποιήθηκαν για την εκτέλεση των πειραμάτων. Γι'αυτό το λόγο, στη περίπτωση συνδυαστικής βελτιστοποίησης συνίσταται νέα υλοποίηση όλων των αλγορίθμων που χρησιμοποιούνται για να γίνει η σύγκριση δίκαιη. Αυτό όμως δεν εγγυάται μία εξ' ολοκλήρου δίκαιη σύγκριση, αφού είναι δύσκολο να εξασφαλίσουμε την ίδια βελτισση του κώδικα όλων των υλοποιημένων αλγορίθμων.

Σε αντίθεση, η μεγάλη πλειοψηφία των επιστημονικών δημοσιεύσεων σε αλγορίθμους συνεχούς βελτιστοποίησης χρησιμοποιεί ως κριτήριο σύγκρισης τον αριθμό των υπολογισμών *συνάρτησης* που χρειάστηκαν για να επιτευχθεί μία προκαθορισμένη ποιότητα λύσης [Kern και άλλοι, 2004 · Bilchev και Parmee, 1995 · Monmarché και άλλοι, 2000 · Dréo και Siarry, 2004]. Μία τέτοια προσέγγιση δίνει αρκετά βασικά πλεονεκτήματα: Λύνει το πρόβλημα των αλγορίθμων που υλοποιήθηκαν χρησιμοποιώντας διαφορετικές προγραμματιστικές γλώσσες, 'δεν σχετίζεται με τη βελτισση του κώδικα απο το προγραμματιστή (ή το μεταγλωττιστή που χρησιμοποιήθηκε) και επιτρέπει την εύκολη σύγκριση των αποτελεσμάτων σε διαφορετικές μηχανές. Το μειονέκτημα αυτής τη προσέγγισης είναι ότι δεν λαμβάνει υπόψη τη χρονική πολυπλοκότητα των αλγορίθμων που συγκρίθηκαν. Παρόλα αυτά, εν όψει των άλλων

πολυάριθμων μειονεκτημάτων της χρήσης του χρόνου της κεντρικής μονάδας επεξεργασίας ως κριτήριο, είναι μία αποδεκτή μεθοδολογία, και εδώ την υιοθετούμε. Επίσης, στη περίπτωση προβλημάτων συνεχούς βελτιστοποίησης, συνήθως η πλειονότητα του χρόνου εκτέλεσης του αλγορίθμου σπαταλιέται στον υπολογισμό της αντικειμενικής συνάρτησης, έτσι γενικά η μέτρηση του αριθμού των υπολογισμών της συνάρτησης είναι ένας καλός τρόπος για να προσεγγίσουμε το χρόνο που απαιτήθηκε για την εκτέλεση του αλγορίθμου.

Η χρήση του αριθμού των υπολογισμών συνάρτησης ως ένα κριτήριο επιτρέπει να εκτελέσουμε τα πειράματα μόνο με τον  $ACO_{\square}$  και να συγκρίνουμε τα αποτελέσματα που ελήφθησαν με εκείνα που βρέθηκαν στη βιβλιογραφία. Επιπροσθέτως, για να εξασφαλίσουμε μία δίκαιη σύγκριση, αναπαράγουμε προσεκτικά τη πειραματική ρύθμιση (συγκεκριμένα: αρχικοποίηση διαστήματος, μεθοδολογία ρύθμισης παραμέτρου, και συνθήκη τερματισμού) που χρησιμοποιήθηκε από τους ανταγωνιζόμενους αλγορίθμους.



## Κεφάλαιο 3

### Διάφοροι τύποι αλγορίθμων για συνεχή προβλήματα

#### 3.1 Πολυμεταβλητή Βελτιστοποίηση Αποικίας Μυρμηγκιών σε Συνεχείς Χώρους Αναζήτησης

Αυτή η ενότητα εισάγει έναν εμπνευσμένο απο μυρμηγκια- αλγόριθμο για βελτιστοποίηση σε συνεχείς χώρους αναζήτησης που βασίζεται στη παραγωγή τυχαίων διανυσμάτων με πολυμεταβλητή Γκαουσιανή συνάρτηση πυκνότητας πιθανότητας. Η προτεινόμενη προσέγγιση ονομάζεται MACACO- Πολυμεταβλητός Αλγόριθμος Αποικίας Μυρμηγκιών για συνεχή Βελτιστοποίηση- και είναι ικανός να προσαρμόσει ταυτόχρονα όλες τις διαστάσεις μίας τυχαίας κατανομής που χρησιμοποιείται για να δημιουργήσει τα νέα χαρακτηριστικά κάθε επανάληψης. Για να αναλύσουμε την αποτελεσματικότητα αναζήτησης του MACACO, η προσέγγιση συγκρίθηκε με το Συνεχές Σύστημα Αποικίας Μυρμηγκιών (CACS) και τη γνωστή ως Βελτιστοποίηση Αποικίας Μυρμηγκιών στον  $R^n$  ( $ACO_R$ ). Η συγκριτική ανάλυση δείχνει ότι ο MACACO υπερτερεί έναντι των (CACS) και ( $ACO_R$ ) στις περισσότερες περιπτώσεις ως προς τη ποιότητα της τελικής λύσης.

##### 3.1.1 Εισαγωγή

Το 1995, οι Bilchev και Parmee πρώτοι πρότειναν μία υιοθέτηση της Βελτιστοποίησης Αποικίας Μυρμηγκιών (ACO) σε συνεχείς χώρους αναζήτησης [1], το οποίο ονομάστηκε CACO (Συνεχές Βελτιστοποίηση Αποικίας Μυρμηγκιών). Η πρόταση τους αρχικοποιεί μία φωλιά σε ένα δοθέν σημείο του χώρου αναζήτησης και παράγει τυχαία διανύσματα που αντιστοιχούν στις κατευθύνσεις που θα ακολουθηθούν από κάθε μυρμηγκί στην αναζήτηση του για καλύτερες λύσεις. Εάν ένα μυρμηγκί πετύχει σε μία τέτοια αναζήτηση, το διάνυσμα κατεύθυνσης που επιλέχθηκε αναβαθμίζεται. Στο [11], προτάθηκε μία άλλη εμπνευσμένη απο τα μυρμηγκια-προσέγγιση για συνεχή βελτιστοποίηση, επονομαζόμενη API (από το μυρμηγκί *Pachycondyla apicalis*). Σε αυτή την προσέγγιση, κάθε μυρμηγκί ψάχνει ανεξάρτητα για μία λύση, αν και όλα ξεκινάνε από το ίδιο σημείο (τη φωλιά). Αυτός ο αλγόριθμος χρησιμοποιεί επίσης στρατηγική στρατολόγησης για να βελτιώσει την αναζήτηση. Αργότερα, το 2002, οι Dreco και Siarry δημιούργησαν τον CIAC (Συνεχή Αλληλεπιδρούσα Αποικία Μυρμηγκιών) αλγόριθμο, ο οποίος χρησιμοποιεί κάποια σημεία στο χώρο αναζήτησης στα οποία τα μυρμηγκία έλκονται και ένα προκαθορισμένο βαθμό άμεσης επικοινωνίας μεταξύ των μυρμηγκιών για να βελτιώσει την εξερεύνηση.

Οι Pourtakdoust και Nobahari πρότειναν το 2004 μια άλλη προσέγγιση για αυτό το πρόβλημα-ο επονομαζόμενος αλγόριθμος CACS (Συνεχές Σύστημα Αποικίας Μυρμηγκιών). Στην προσέγγισή τους, η διακριτή πιθανοθεωρητική συνάρτηση φερομόνης αντικαθίσταται από μία Γκαουσιανή (κανονική) πιθανοθεωρητική συνάρτηση πυκνότητας ( $pdf$ ) της οποίας οι παράμετροι του μέσου και της διασποράς είναι δυναμικά προσαρμοσμένες από τα μυρμηγκία σε κάθε επανάληψη.

Τελικά, το 2006, οι Socha και Dorigo [14] πρότειναν μία άλλη προσέγγιση, εμπνευσμένη από το PB-ACO (*Population Based ACO*) στο οποίο οι λύσεις χτίζονται σύμφωνα με ένα αρχείο των  $n$  καλύτερων λύσεων που βρέθηκαν μέχρι τώρα. Αυτός ο αλγόριθμος επίσης χρησιμοποιεί την ίδια ιδέα του CACS [12], σε συνεργασία με μία Γκαουσιανή συνάρτηση πυκνότητας πιθανότητας της φερομόνης, και πήρε το όνομα  $ACO_R$  (Βελτιστοποίηση Αποικίας Μυρμηγκιών στον  $R^n$ ). Στο [14], οι Socha και Dorigo συγκρίναν την επίδοση του  $ACO_R$  με των CACO, API και

CIAC, μαζί με κάποιες εξελικτικές προσεγγίσεις, και ο ACO<sub>R</sub> οδήγησε στα βέλτιστα αποτελέσματα στις περισσότερες περιπτώσεις.

Μπορεί να συναχθεί από τα συγκριτικά πειραματικά αποτελέσματα τα οποία φαίνονται στο [14] ότι υπάρχει χώρος για βελτίωση της επίδοσης που επιτυγχάνεται από τους ACO αλγόριθμους για συνεχή βελτιστοποίηση. Μία μέχρι τώρα ανεξερεύνητη δυνατότητα είναι να θεωρήσουμε την ύπαρξη της εξάρτησης μεταξύ μεταβλητών: όλοι οι αλγόριθμοι που είναι βασισμένοι στην αποικια μυρμηγκιών μέχρι τώρα αντιμετωπίζουν αυτές σαν να είναι αμοιβαίως ανεξάρτητες, το οποίο μπορεί να μειώσει την ικανότητα του χρησιμοποιούμενου μηχανισμού δειγματοληψίας.

Προς αυτή την κατεύθυνση, προτείνεται μία νέα βασισμένη στα μυρμηγκια- τεχνική που χρησιμοποιεί την πληροφορία που περιέχεται στον πίνακα συνδιακύμανσης (covariance matrix) του πληθυσμού των μυρμηγκιών για να δημιουργήσουμε μία πολυμεταβλητή Γκαουσιανή τυχαία συνάρτηση κατανομής πιθανότητας pdf και να χρησιμοποιηθεί για την μετατόπιση των μυρμηγκιών κατά μήκος του συνεχούς χώρου αναζήτησης. Με αυτόν το τρόπο, σκοπεύουμε να απεικονίσουμε και να εξερευνήσουμε το χώρο αναζήτησης περισσότερο αποτελεσματικά. Αυτή η πρόταση εφαρμόστηκε σε αρκετά γνωστά προβλήματα και τα ληφθέντα αποτελέσματα συγκρίθηκαν με αυτά των αλγορίθμων CACS και ACO<sub>R</sub>. Όπως θα φανεί, οι πολυμεταβλητοί μηχανισμοί κατανομής βελτιώνουν τη ποιότητα λύσης και την εξερεύνηση του χώρου αναζήτησης, και, συνεπώς, οδηγούν σε μία σημαντική βελτίωση της επίδοσης σε σύγκριση με τους αλγορίθμους CACS και ACO<sub>R</sub>.

### 3.1.2. Σύστημα Μυρμηγκιών (AS) και Βελτιστοποίηση Αποικίας Μυρμηγκιών (ACO)

Η Βελτιστοποίηση Αποικίας Μυρμηγκιών (ACO) [4] ήταν ο πρώτος βασισμένος στα μυρμηγκια αλγόριθμος που εφαρμόζεται για να λύνει προβλήματα συνδυαστικής βελτιστοποίησης. Περισσότερο από μία δεκαετία μετά από αυτή τη πρωτοβουλία, αρκετές διαφορετικές εκδόσεις, βελτιώσεις και εφαρμογές παρουσιάστηκαν (Dorigo και Stutzle, de Franca και άλλοι). Σε αυτό το τμήμα, η πρωτότυπη πρόταση του Dorigo θα αναθεωρηθεί, μαζί με κάποιες προσαρμογές που παρουσιάζονται στην βιβλιογραφία για να λύσει προβλήματα βελτιστοποίησης σε συνεχείς χώρους αναζήτησης.

#### Βελτιστοποίηση Αποικίας Μυρμηγκιών (ACO)

Ο ACO [4] είναι εννοιολογικά απλός, όπως περιγράφεται στον αλγόριθμο 1.

##### Αλγόριθμος 1 Βασικός ACO Αλγόριθμος

```
while iteration < maximumNumberOfIterations do
  for each ant do
    build_solution();
    update_pheromone();
  end for
end while
```

Στο κλασικό ACO αλγόριθμο που παρουσιάζεται στον Αλγόριθμο 1, το συνδυαστικό πρόβλημα που θα λυθεί είναι κατανοητό με όρους ενός γράφου, και τα μυρμηγκια τοποθετούνται στους κόμβους του. Μία λύση σε αυτό το πρόβλημα συνεπώς αντιστοιχεί σε μία σειρά από ακμές που επιλέγονται σε μία ένα προς ένα βάση ενός δοσμένου μυρμηγκιού. Όταν έχει κατασκευαστεί μια πλήρης λύση, αξιολογείται, και οι ακμές που ανήκουν σε αυτή έχουν τη συγκέντρωση φερομόνης αυξημένη αναλογικά με το σχετικό ποιοτικό δείκτη. Η συγκέντρωση της φερομόνης σε κάθε ακμή επηρεάζει τη πιθανότητα της επιλογής της από τα μυρμηγκια στη μετέπειτα γενεά του αλγόριθμου. Συνεπώς, οι ακμές που ανήκουν σε υψηλής ποιότητας λύσεις θα τείνουν να έχουν μεγαλύτερη πιθανότητα να επιλεγούν κατά τη διάρκεια της κατασκευής μίας νέας λύσης του προβλήματος. Ανάλογα με τη τυποποίηση του προβλήματος, ο τρόπος που οι υποψήφιας λύσεις παρουσιάζονται ενδέχεται να αναγκάσει τη φερομόνη να συνδεθεί με τους κόμβους, και όχι με τις ακμές του γράφου.

Στον Αλγόριθμο 1, η διαδικασία build solution() κατασκευάζει μία λύση για το πρόβλημα βασισμένη στο ίχνος της φερομόνης και στις προαιρετικές πληροφορίες ευριστικής η οποία δεν θα συμπεριλαμβάνεται στην πρότυπη τυποποίηση που περιγράφεται σε αυτό το υποτόμημα. Κάθε μυρμηγκί  $k$  διασχίζει μια ακμή ανά επανάληψη  $t$  και σε κάθε ακμή, η τοπική πληροφορία σχετικά με τη φερομόνη της,  $\tau_{ij}$  χρησιμοποιείται από το μυρμηγκί έτσι ώστε αυτό να μπορεί πιθανοθεωρητικά να αποφασίσει τον επόμενο κόμβο στον οποίο θα κινηθεί, σύμφωνα με τον ακόλουθο κανόνα:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}}{\sum_{j \in J^k} \tau_{ij}} & \text{if } j \in J^k \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

όπου  $\tau_{ij}(t)$  είναι το επίπεδο φερομόνης για την ακμή  $(i, j)$ , δηλαδή η ακμή που συνδέει τους κόμβους  $i$  και  $j$ ,  $J^k$  είναι η λίστα από τους κόμβους που πρέπει ακόμα να επισκεφτεί το μυρμηγκί  $k$ , και ο κόμβος  $i$  είναι ο τρέχων κόμβος του μυρμηγκιού  $k$ .

Στην διαδικασία update pheromone(), το επίπεδο φερομόνης της ακμής  $(i, j)$  έχει ενημερωθεί σύμφωνα με την εξίσωση (2):

$$\tau_{ij} \leftarrow \rho \cdot \tau_{ij} + \Delta\tau_{ij}, \quad (2)$$

όπου  $\rho \in (0, 1]$  είναι ο ρυθμός διάσπασης της φερομόνης και  $\Delta\tau_{ij}$  είναι η αύξηση του επιπέδου φερομόνης. Στα προβλήματα ελαχιστοποίησης η αύξηση της φερομόνης δίνεται από :

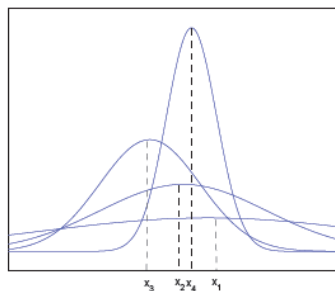
$$\Delta\tau_{ij} = \begin{cases} \frac{1}{f(S)} & \text{if } (i, j) \in S \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

όπου  $S$  είναι η λύση η οποία χρησιμοποιείται για να ενημερωθεί το μονοπάτι και  $f(S)$  είναι η συνάρτηση που αντανάκλα το κόστος της λύσης, δηλαδή, όσο χαμηλότερο  $f(S)$  τόσο καλύτερη ποιότητα της πρότασης  $S$ .

### Προσαρμόζοντας τον αλγόριθμο ACO για να αντιμετωπίσει Συνεχείς Χώρους Αναζήτησης

Προκειμένου να προσαρμοστεί ο αλγόριθμος ACO για να αντιμετωπίσει συνεχείς χώρους αναζήτησης, είναι απαραίτητο να τροποποιήσουμε το πιθανοθεωρητικό μοντέλο που χρησιμοποιήθηκε στην διαδικασία δειγματοληψίας νέων υποψήφιων λύσεων (δοσμένων στην Εξίσωση 1) έτσι ώστε να παραχθούν λύσεις ως διανύσματα παραμέτρων πραγματικών τιμών. Για να το κάνουμε αυτό, η φερομόνη θα πρέπει να ορίσει μία συνάρτηση πυκνότητας πιθανότητας που προσδιορίζει την ευκαιρία επιλογής συγκεκριμένων χωρίων του χώρου αναζήτησης όταν δημιουργούνται διανύσματα τυχαίων λύσεων. Η τυχαιότητα τείνει να είναι ανάλογη με την σχετική ποιότητα τέτοιων χωρίων όπως παρατηρήθηκε στις προηγούμενες επαναλήψεις του αλγορίθμου.

Γενικά, υιοθετήθηκε μία Γκαουσιανή συνάρτηση πυκνότητας πιθανότητας (pdf) (Γράφημα 1), αφού εύκολα δειγματίστηκε από μεθόδους όπως τη Box-Muller [2] και τον αλγόριθμο Ziggurat [10].



Εικόνα 1 : Τέσσερις Γκαουσιανές συναρτήσεις πυκνότητας πιθανότητας (επικεντρωμένες στα  $x_1, x_2, x_3$  και  $x_4$ ) περικλείοντας τέσσερα διαφορετικά χωρία του προβλήματος.

Παρόλα αυτά, έχει το μειονέκτημα να περιλαμβάνει μόνο ένα υποσχόμενο χωρίο του χώρου αναζήτησης, αφού αυτό έχει ένα μόνο μέγιστο Το άνοιγμα (διασπορά) των Γκαουσιανών, όπως φαίνεται στην Εικόνα 1, καθορίζει το βαθμό διασποράς των νέων τυχαίων λύσεων γύρω από τα αντίστοιχα κέντρα. Συνεπώς, στους ACO αλγόριθμους για τη βελτιστοποίηση σε συνεχείς χώρους αναζήτησης, κάθε νέο διάλυμα λύσης δημιουργείται με δειγματοληψία Γκαουσιανών τυχαίων μεταβλητών, μία κάθε φορά, για κάθε διάσταση του προβλήματος. Οι διαφορές μεταξύ των διαφόρων προσεγγίσεων που αναφέρονται στην βιβλιογραφία βρίσκονται στο πώς αυτές αναβαθμίζουν το κέντρο και τη διασπορά των Γκαουσιανών συναρτήσεων πυκνότητας πιθανότητας. Στο επόμενο τμήμα, θα δείξουμε πώς οι CACS [12] και ACO<sub>R</sub> [14] ασχολούνται με τέτοιες παραμέτρους.

### Αλγόριθμος CACS

Ο αλγόριθμος CACS παρουσιάστηκε πρώτα στο [12], στο οποίο προτάθηκε ένα συνεχές μοντέλο φερομόνης αποτελούμενο από μία Γκαουσιανή συνάρτηση κατανομής πιθανότητας επικεντρωμένη στη καλύτερη λύση που έχει βρεθεί μέχρι τώρα. Η διασπορά μίας τέτοιας συνάρτησης κατανομής πιθανότητας ξεκινάει με μία τιμή τρεις φορές μεγαλύτερη από το εύρος της κάθε μεταβλητής του προβλήματος (δηλαδή  $3(x_{\max} - x_{\min})$ ), και, κατά τη διάρκεια της εξέλιξης του αλγορίθμου, τροποποιήθηκε σύμφωνα με τον σταθμισμένο μέσο όρο της απόστασης μεταξύ κάθε ατόμου στο πληθυσμό και στη καλύτερη λύση που έχει βρεθεί μέχρι τώρα, όπως φαίνεται στην εξίσωση (4), όπου  $K$  είναι ο αριθμός των μυρμηγκιών που είναι ορισμένα στον αλγόριθμο.

$$\sigma^2 = \frac{\sum_{j=1}^K \frac{1}{|f_j - f_{\min}|} (x_j - x_{\min})^2}{\sum_{j=1}^K \frac{1}{|f_j - f_{\min}|}}. \quad (4)$$

Τα κύρια πλεονεκτήματα αυτού του αλγορίθμου είναι ότι απαιτεί τον καθορισμό μίας μόνο παραμέτρου (ο αριθμός των μυρμηγκιών στον πληθυσμό) και έχει έναν πολύ απλό μηχανισμό για να παράγει τα μυρμηγκία της επόμενης γενιάς. Από την άλλη μεριά, ένα μειονέκτημα είναι ότι αυτός διερευνά μόνο ένα υποσχόμενο χωρίο του προβλήματος κάθε φορά, το οποίο σημαίνει ότι ο αλγόριθμος τείνει να συγκεντρώσει τις Γκαουσιανές συναρτήσεις κατανομής πιθανότητας γύρω από ένα τοπικό βέλτιστο πολύ γρήγορα, που οδηγεί συνεπώς σε μία πρόωρη σύγκλιση.

### Ο αλγόριθμος ACO<sub>R</sub>

Ο αλγόριθμος ACO<sub>R</sub> [14] επινοήθηκε για να ασχοληθεί με την προαναφερθείσα πρόωρη σύγκλιση μειονέκτημα των προτάσεων που χρησιμοποιούν μία απλή Γκαουσιανή συνάρτηση. Αυτός ο αλγόριθμος αποτελείται από ένα αρχείο που κατέχει τις  $k$  καλύτερες λύσεις που βρέθηκαν μέχρι τώρα, όπου κάθε λύση αντιπρόσωπεύεται στο κέντρο μίας διαφορετικής Γκαουσιανής συνάρτησης πυκνότητας πιθανότητας. Επιπλέον, αυτό το αρχείο χρησιμοποιείται για να υπολογιστεί η διασπορά της κάθε κατανομής, έτσι ώστε ολόκληρη η διαδικασία να μπορεί να περιγραφεί ως ακολούθως.

Αρχικά, ολόκληρο το αρχείο είναι τυχαία δημιουργημένο (χρησιμοποιώντας μία ομοιόμορφη κατανομή), και τα παραγόμενα άτομα κατατάσσονται σε φθίνουσα σειρά καταλληλότητας (από καλύτερο στο χειρότερο). Τότε, η κύρια επανάληψη ξεκινάει από τη πρώτη αποδίδοντας μία λύση στο αρχείο στο κάθε μυρμηγκί του προβλήματος, με πιθανότητα ανάλογη με το βάρος της  $l$ -οστής λύσης  $\omega_l$  του αρχείου (Εξίσωση 5) :

$$\omega_l = \frac{1}{qK\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2K^2}}, \quad (5)$$

όπου  $K$  είναι ο αριθμός των μυρμηγκιών,  $l$  είναι η τάξη της λύσης στο αρχείο, και  $q$  είναι μία μεταβλητή επανομαζόμενη τοπικότητα της διαδικασίας αναζήτησης. Ο ρόλος της μεταβλητής  $q$  είναι να εξισορροπήσει την εκμετάλλευση και την εξερεύνηση. Το κέντρο κάθε Γκαουσιανής

РАМЕТЪМО РЕПАА

Συνεπώς, σε μία προσπάθεια να εξερευνήσουμε τη μεγάλη ευελιξία που παρέχεται από την πολυμεταβλητή κατανομή, προτείνουμε ένα πολυμεταβλητό αλγόριθμο Αποκίας Μυρμηγκίων για συνεχή βελτιστοποίηση, δηλωμένο ως MACACO. Για να το κανουμε, δοσμένου ενός πίνακα συνδιακύμανσης  $\Sigma$  και κέντρου  $\bar{\mu}$ , έχουμε επιλέξει να δημιουργήσουμε τα διανύσματα λύσεις σύμφωνα με τη τεχνική που βασίζεται στους μετασχηματισμούς περιγραφόμενη στο [9], η οποία αποτελείται από τρία απλά βήματα :

- Πρώτα δημιουργούμε ένα διάνυσμα  $\bar{x} \in \mathbf{R}^n$  με μέσους μίας Γκαουσιανής κατανομής με ανεξάρτητες μεταβλητές (δηλαδή  $\bar{x}_i = N(0,1)$ ,  $i = 1, \dots, \dim$ ), έτσι ώστε το  $x$  να υπακούει την κατανομή  $N(\bar{0}, I)$ , όπου  $I$  ο ταυτοτικός πίνακας.
- Έπειτα υπολογίζουμε το πίνακα  $\Phi$  των κανονικοποιημένων ιδιοδανυσμάτων (όπου κάθε ιδιοδιάνυσμα αντιστοιχεί σε κάθε στήλη του  $\Phi$ ) του πίνακα συνδιακύμανσης  $\Sigma$ , και δημιουργεί ένα διαγώνιο πίνακα  $\Lambda$  του οποίου οι τιμές είναι οι αντίστοιχες ιδιοτιμές. Έστω

$$Q = \Lambda^{-\frac{1}{2}} \Phi \quad (7)$$

- Τελικά το διάνυσμα  $\bar{x}$  μετασχηματίζεται από την εξίσωση 8 έτσι ώστε το  $\bar{y}$  να υπακούει στη κατανομή  $N(\bar{\mu}, \Sigma)$ :

$$\bar{y} = Q\bar{x} + \bar{\mu} \quad (8)$$

Είναι σημαντικό να παρατηρήσουμε ότι, ενώ η διαδικασία της εκτέλεσης της απαιτούμενης ιδιοδιάσπασης ενδέχεται να είναι δαπανηρή, η ίδια η ουσία του τρόπου λειτουργίας του αλγορίθμου MACACO επιτρέπει και η ενημέρωση συνδιακύμανσης και ο υπολογισμός του πίνακα μετασχηματισμού  $Q$  να μπορούν να διεξαχθούν μόνο μία φορά ανά παραγωγή.

Αφού ο MACACO λειτουργεί με ένα πίνακα συνδιακύμανσης (αντί για ένα απλό διάνυσμα διασποράς όπως και οι άλλοι δύο), το βήμα ενημέρωσης για κάθε μυρμηγκί πρέπει να γίνεται με ένα τρόπο όπου η νέα κατανομή των λύσεων τείνει στο υποσχεμένο χωρίς υποθετικής υποδεικνυόμενο από τις υποψήφιες λύσεις στην προηγούμενη επανάληψη. Αυτό μπορεί να ολοκληρωθεί απλά επαναυπολογίζοντας το πίνακα χρησιμοποιώντας μόλις το 70% (εμπειρική τιμή) των καλύτερων λύσεων που βρέθηκαν στην τρέχουσα επανάληψη. Με το τρόπο αυτό, η παραγόμενη κατανομή θα διακόψει τη λειτουργία μέχρι η συνδιακύμανση να γίνει σχεδόν 0 και επικεντρωμένη στο καλύτερο τοπικό βέλτιστο που έχει βρεθεί μέχρι τώρα. Δυστυχώς, αυτή η διαδικασία αναβάθμισης έχει ένα μειονέκτημα, αφού αυτή απαιτεί ένα μεγάλο αριθμό από υποψήφιες λύσεις για να γίνει μία καλή δειγματοληψία.

Εμπειρικά, παρατηρήθηκε ότι αυτή η διαδικασία αναβάθμισης δουλεύει καλύτερα κάθε φορά που η αρχική κατανομή είναι επικεντρωμένη κοντά στο ολικό βέλτιστο. Αφού αυτό το βέλτιστο είναι άγνωστο πρακτικά, ο MACACO χωρίστηκε σε δύο βήματα, το πρώτο αφιερώθηκε στο να βρούμε ένα καλό αρχικό διάνυσμα κέντρου και το δεύτερο σχεδιάστηκε για να βελτιώσει τοπικά το χωρίο που αναγνωρίστηκε στο προηγούμενο βήμα. Ο ψευδοκώδικας του προτεινόμενου αλγορίθμου είναι στον Αλγόριθμο 2.

Στον Αλγόριθμο 2, η συνάρτηση `generate_initial_ants()` δημιουργεί  $n_{\text{μυρμηγκία}}$  υποψήφιες λύσεις με ομοιόμορφη κατανομή  $U(0, \text{εύρος})$ , όπου το εύρος είναι το πεδίο ορισμού του προβλήματος. Αυτή η συνάρτηση υπολογίζει επίσης τον αρχικό πίνακα συνδιακύμανσης και το πίνακα μετασχηματισμού  $Q$ . Κατά τη διάρκεια της Φάσης 1, οι λύσεις κατασκευάστηκαν σύμφωνα με την εξίσωση 8, χρησιμοποιώντας την καλύτερη υποψήφια λύση που βρέθηκε μέχρι τώρα ως το διάνυσμα κέντρου  $\bar{\mu}$  (αυτό έγινε στη διαδικασία `build_solution_meanbest()`). Στη Φάση 2, η διαδικασία `build_solution()` θεωρεί ως κέντρο της κατανομής το μέσο του προσφάτως δημιουργημένου πληθυσμού ατόμων, για να λάβουμε το μετασχηματισμό της πολυμεταβλητής κατανομής. Η διαδικασία `update_pheromone()` δουλεύει παίρνοντας το 70% των καλύτερων λύσεων που βρέθηκαν στην τρέχουσα επανάληψη και θεωρώντας αυτά τα άτομα στον επαναυπολογισμό του πίνακα συνδιακύμανσης, όπως περιγράφηκε παραπάνω.

Επιπλέον, αυτό επαναυπολογίζει τον πίνακα μετασχηματισμού  $Q$  για την επόμενη επανάληψη. Όταν η καλύτερη λύση σταματάει να βελτιώνεται για ένα δοσμένο αριθμό επαναλήψεων

**Algorithm 2** MACACO Algorithm.

---

```

generate_initial_ants();
phase = 1;
while iteration < maximumNumberOfIterations do
  for each ant do
    if phase == 1 then
      build_solution_meanbest();
    else
      build_solution();
    end if
  end for
  update_pheromone();
  if best solution is no more improving then
    phase = 2;
    restart_covar();
  end if
end while

```

---

(εμπειρικά σύνολο από 10 επαναλήψεις), η συνδιακύμανση επαναυπολογίζεται με δειγματοληψία  $n_{\text{μυρμηγκιών}}$  διανυσμάτων με κατανομή  $U(\text{ant-best}, \text{εύρος})$ , όπου ant-best είναι η καλύτερη λύση που έχει βρεθεί μέχρι τώρα. Και εάν ο αλγόριθμος δουλεύει ακόμα στη Φάση 1, γυρίζει στη Φάση 2.

**3.1.4. Αποτίμηση Απόδοσης**

Στο πειραματικό τμήμα αυτής της εργασίας, θα συγκρίνουμε την πρόταση μας (τον αλγόριθμο MACACO) με δύο από τους υπάρχοντες συνεχούς χωρίου ACO αλγορίθμους : τον CACS [12] και τον  $ACO_R$ . Η λογική της επιλογής μας είναι διπλή: i) στο [14], έχει γίνει ήδη μία εκτενής σύγκριση μεταξύ του  $ACO_R$  και άλλων βασισμένων στον ACO προσεγγίσεων, αλλά χωρίς τον CACS και ii) η πλειοψηφία των προσεγγίσεων που δεν αναφέρθηκαν στο [14] είναι πολύ όμοιες με τον CACS.

Για να εκτιμήσουμε την απόδοση των τριών αλγορίθμων, επιλέξαμε έξι πολύ γνωστά προβλήματα αναφοράς από τη βιβλιογραφία (δοσμένη στο Πίνακα 1) και υπολογίσαμε το μέσο και την τυπική απόκλιση της φόρμας των καλύτερων χαρακτηριστικών (τιμών συναρτήσεων) που βρέθηκαν από κάθε αλγόριθμο σε 30 ανεξάρτητες εκτελέσεις.

**Πίνακας 1 : Προβλήματα αναφοράς που υιοθετήθηκαν σε αυτή την εργασία.**

Name	Problem	Dimension (n)	Range
Sphere	$f(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]^n$
Rosenbrock	$f(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	30	$[-30, 30]^n$
Rastrigin	$f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$	30	$[-5.12, 5.12]^n$
Griewank	$f(x) = 1 + 0.00025 \sum_{i=1}^n x_i - \prod_{i=1}^n \cos \frac{x_i}{\sqrt{i}}$	30	$[-500, 500]^n$
Schwefel	$f(x) = -\sum_{i=1}^n x_i \sin \sqrt{ x_i }$	30	$[-500, 500]^n$
Salomon	$f(x) = 1 - \cos(2\pi \sqrt{\sum_{i=1}^n x_i^2}) + 0.1 \sqrt{\sum_{i=1}^n x_i^2}$	30	$[-100, 100]^n$

Οι παράμετροι του  $ACO_R$  ορίστηκαν όπως προτάθηκαν στη πρωτότυπη επιστημονική δημοσίευση [14], βλέπε Πίνακα 2. Ο αριθμός των επαναλήψεων δεν επιλέχθηκε εκ των προτέρων ορίστηκε, χάριν αμεροληψίας, ότι σε κάθε αλγόριθμο θα πρέπει να επιτρέπεται ένα μέγιστο από  $10^6$  υπολογισμούς συναρτήσεων για όλα τα προβλήματα αναφοράς που έχουν μελετηθεί εδώ, εκτός από τη συνάρτηση του Rosenbrock, η οποία θα μπορούσε να αποτιμηθεί μέχρι  $6 \times 10^6$  φορές.

Ένας πληθυσμός από 100 μυρμηγκία ορίστηκε και για τους δύο αλγορίθμους CACS και MACACO. Είναι σημαντικό να τονίσουμε εδώ ότι, αν και αυτοί οι αλγόριθμοι είχαν πολύ

περισσότερα μυρμηγκία από τον  $ACO_R$ , ο τελευταίος εκτελέστηκε για ένα μικρότερο συνολικό αριθμό επαναλήψεων για να διατηρήσει τον ίδιο συνολικό αριθμό από υπολογισμούς συναρτήσεων.

**Πίνακας 2 : Προτεινόμενες παράμετροι για το  $ACO_R$  .**

Parameter	Value
Number of Ants	2
Speed of Convergence ( $\xi$ )	0.85
Locality of the Search Process ( $q$ )	$10^{-4}$
Archive Size	50

Επίσης, για να συμπεράνουμε ότι ακόμα και αν συντηρούμε ένα μεγάλο αριθμό από μυρμηγκία όταν συγκρίνουμε με τον  $ACO_R$ , θα δειχτεί, διαμέσου κάποιων πειραμάτων, ότι αυξάνοντας αυτή τη τιμή ενδέχεται ακόμα και να χειροτερεύσει τον μεσο ορό των αποτελεσμάτων του. Θα έπρεπε να σημειωθεί ότι οι υιοθετημένες παράμετροι για τους  $ACO_R$  αλγόριθμους ήταν οι βέλτιστες τιμές που αναφέρθηκαν από τους αντίστοιχους συγγραφείς σε παρόμοια πειράματα. Όλες οι προσομοιώσεις που πραγματοποιήθηκαν σε αυτή την εργασία έγιναν σε έναν AMD Athlon64 3500+, 2.2 GHz, 1GB μνήμη RAM τρέχοντας Windows XP Professional SP2 και όλοι οι αλγόριθμοι προγραμματίστηκαν σε C++ και μεταγλωττίστηκαν με GCC 3.4.2. Στο ακόλουθο υποπλήγμα, εμείς θα περιγράψουμε τα ληφθέντα πειραματικά αποτελέσματα και θα συζητήσουμε την απόδοση του κάθε αλγορίθμου.

### Πειραματικά Αποτελέσματα

Για κάθε πρόβλημα αναφοράς που θεωρήθηκε σε αυτή τη εργασία, έχουμε, στο Πίνακα 3, το μέσο και τη τυπική απόκλιση που παράχθηκαν από το κάθε αλγόριθμο μετά από 30 ανεξάρτητες εκτελέσεις.

Από το Πίνακα 3, συγκρίνοντας αρχικά τους  $ACO_R$  και CACS, μπορεί να γίνει ορατό ότι ο αλγόριθμος CACS, ήταν ικανός να υπερτερήσει του  $ACO_R$  στις πέντε από τις έξι συναρτήσεις, ενώ είναι ελαφρα χειρότερος από τον  $ACO_R$  στην συνάρτηση Schwefel, μολονότι παρουσιάζει μία μικρότερη διασπορά για αυτή τη συνάρτηση.

**Πίνακας 3 : Τιμές μέσης τιμής και τυπικής απόκλισης για τα καλύτερα χαρακτηριστικά ληφθέντα από τους  $ACO_R$ , CACS και MACACO , για τα πέντε προβλήματα αναφοράς που μελετήθηκαν σε αυτή την επιστημονική δημοσίευση (και είναι δοσμένα στο Πίνακα 1). Τα καλύτερα αποτελέσματα είναι τονισμένα με μαύρο χρώμα.**

	$ACO_R$	CACS	MACACO
Sphere	0	0	0
Rosenbrock	29.93 ± 35.14	1.23 ± 1.91	7.48 ± 11.68
Rastrigin	101.65 ± 21.01	57.17 ± 14.14	0.00058 ± 0.00009
Griewank	0.09 ± 0.180	0.02 ± 0.02	0
Schwefel	-8703.26 ± 721.53	-8934.57 ± 633.78	-12569.49 ± 0.01
Salomon	3.05 ± 1.43	0.33 ± 0.05	0.15 ± 0.05

Θεωρώντας τώρα τα αποτελέσματα που πάρθηκαν με τον αλγόριθμο MACACO, μπορεί να γίνει ορατό ότι επιτεύχθηκε μία σπουδαία συνολική βελτίωση, στις τιμές του μέσου και της διασποράς για όλα τα προβλήματα αναφοράς, εκτός από του Rosenbrock.

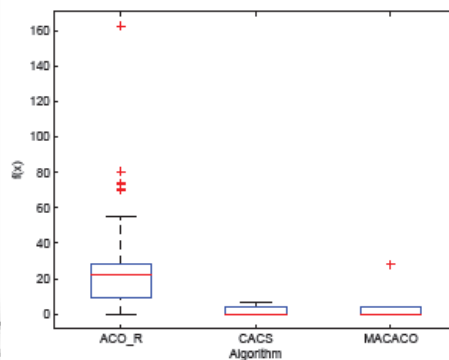
Για να επιβεβαιώσουμε ότι οι αλγόριθμοι MACACO και CACS δεν εκμεταλλεύονται απλά τον μεγάλο αριθμό από μυρμηγκία για να βρουν καλές λύσεις ως μία απλή δειγματοληπτική μέθοδο, τα προβλήματα στο Πίνακα 4 δοκιμάστηκαν στον  $ACO_R$  αλλάζοντας τον αριθμό των μυρμηγκιών σε 1000 και ένα ανάλογο αριθμό επαναλήψεων. Όπως μπορεί να γίνει ορατό στο Πίνακα 4, κάποια από τα παρουσιαζόμενα αποτελέσματα του  $ACO_R$  είναι χειρότερα από αυτά που ελήφθησαν με τις προτεινόμενες παραμέτρους.



**Πίνακας 4 : Αποτελέσματα ληφθέντα από το ACO<sub>R</sub> χρησιμοποιώντας χίλια μυρμηγκία σε κάθε πρόβλημα ελέγχου.**

	ACO <sub>R</sub>
Sphere	0
Rosenbrock	82818.94 ± 98832.77
Rastrigin	242.12 ± 38.18
Griewank	1.45 ± 0.42
Schwefel	-9484.07 ± 977.79
Salomon	0.33 ± 0.05

Για το πρόβλημα του Rosenbrock :Στην εικόνα 3 στο box-plot παρουσιάζονται τα αποτελέσματα που λήφθηκαν από τους τρεις αλγόριθμους στις 30 ανεξάρτητες εκτελέσεις. Αμφότεροι οι CACS και MACACO παρουσιάζουν μία πολύ όμοια συμπεριφορά ειδικότερα ως προς τους διαμέσους τους, αλλά ο MACACO παρουσίασε μερικές ακραίες τιμές που αύξησαν σημαντικά τις μέσες τιμές που δίνονται από το Πίνακα 3. Αυτά τα αποτελέσματα επιβεβαιώθηκαν επίσης από την υψηλή τυπική απόκλιση δοσμένη από το Πίνακα 3. Αυτό ενδέχεται να οφείλεται στο γεγονός ότι η συνάρτηση του Rosenbrock για ένα αριθμό διαστάσεων μεγαλύτερων του 2 γίνεται πολύτροπος (multimodal) όπως επισημαίνεται στο [13], και απο τη φύση του αλγόριθμου MACACO αυτού καθαυτού.

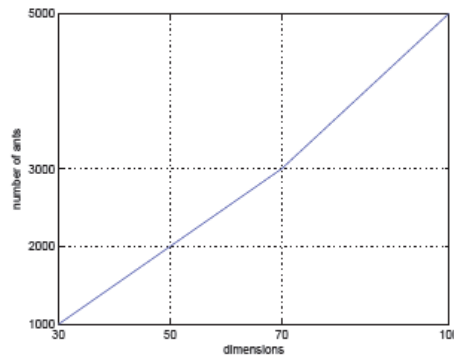
**Εικόνα 3 : Κουτί γράφημα για τη συνάρτηση Rosenbrock.****Πίνακας 5 : Μέσοι χρόνοι εκτέλεσης (σε δευτερόλεπτα) ληφθέντες από τους ACO<sub>R</sub> ,CACS και MACACO , για τα έξι προβλήματα αναφοράς που μελετήθηκαν**

	ACO <sub>R</sub>	CACS	MACACO
Sphere	429.19	19.40	41.61
Rosenbrock	2113.82	110.56	222.75
Rastrigin	445.19	23.75	45.28
Griewank	385.26	17.52	47.26
Schwefel	411.11	18.93	47.30
Salomon	443.06	15.93	41.65

Ο Πίνακας 5 παρουσιάζει το μέσο χρόνο εκτέλεσης για κάθε ένα από τους τρεις αλγόριθμους που μελετήθηκαν σε αυτήν την εργασία. Όπως μπορεί να γίνει ορατό από το πίνακα, ο αλγόριθμος MACACO παρουσιάζει μέσο χρόνο εκτέλεσης περίπου δύο φορές περισσότερο από τον αλγόριθμο CACS, αλλά είναι 10 φορές γρηγορότερος από τον ACO<sub>R</sub>. Αν και μία ανάλυση των χρόνων εκτέλεσης έχει πάντα θέμα με ένα αριθμό από τεχνικά ζητήματα - συμπεριλαμβανομένων τακτικών υλοποίησης- οι οποίες απαιτούν μεγάλη φροντίδα, τα αποτελέσματα του Πίνακα 5 μπορούν να επεξηγηθούν σαν μία καλή υπόδειξη της συγκριτικής απόδοσης των τριών αλγόριθμων που μελετήθηκαν σε αυτή την εργασία.

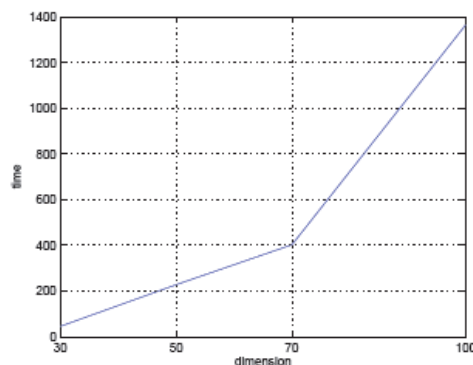
Ως ένα τελικό πείραμα, ο αριθμός των μυρμηγκιών που απαιτούνται από τον MACACO για να επιτευχθεί μία προκαθορισμένη λύση του προβλήματος Rastrigin με μεγαλύτερες διαστάσεις (30,50,70 και 100) ήταν εμπειρικά επαληθεύσιμο. Ο αριθμός των μυρμηγκιών που πήραμε έτσι

ώστε ο αλγόριθμος να μπορεί να φτάσει τη μέση λύση του τουλάχιστον 0.5 για αυτό το πρόβλημα, το οποίο είναι κοντά στο ολικό βέλτιστο, κρατώντας τον αριθμό των επαναλήψεων σταθερό στις 1000 (η ίδια τιμή υιοθετήθηκε για τα πειράματα που εκτελέστηκαν πριν). Η εικόνα 4 δείχνει την εξέλιξη του απαιτούμενου αριθμού των μυρμηγκιών με την αύξηση της διάστασης του προβλήματος, και προτείνει ότι αυτές οι δύο μεταβλητές είναι γραμμικά συσχετισμένες.



**Εικόνα 4 : Εξέλιξη του αριθμού των μυρμηγκιών που απαιτούνται για να λύσουμε το πρόβλημα Rastrigin με διακριτές διαστάσεις.**

Επιπλέον, ο χρόνος που απαιτείται για να εκτελεστεί μία διαδικασία βελτιστοποίησης για κάθε διάσταση του προβλήματος Rastrigin έχει σχεδιαστεί (και φαίνεται στην Εικόνα 5) για να υπολογιστεί ο αντίκτυπος στον αλγόριθμο όταν αριθμός των μυρμηγκιών και η διάσταση του προβλήματος αυξάνονται. Η Εικόνα 5 δείχνει ότι η αρχική αύξηση είναι γραμμική με τη διάσταση του προβλήματος αλλά, μετά από ένα αριθμό από 70 μεταβλητές ο απαιτούμενος χρόνος φαίνεται να μεγαλώνει πιο απότομα.



**Εικόνα 5 : Εξέλιξη του μέσου χρόνου (σε δευτερόλεπτα) απαιτούμενα για να λυθεί το πρόβλημα Rastrigin με διακριτές διαστάσεις.**

### 3.1.5. Σχόλια

Εδώ προτάθηκε ένας καινοφανής βασισμένος στα μυρμηγκία αλγόριθμος για συνεχή περιβάλλοντα. Η τεχνική, επωνομαζόμενη MACACO, η οποία βασίζεται στην παραγωγή τυχαιών διανυσμάτων από πολυμεταβλητές Γκαουσιανές συναρτήσεις πυκνότητας πιθανότητας pdf, συγκρίθηκε με τις προσεγγίσεις από τους  $ACO_R$  και CACS.

Τονίστηκαν δύο από αυτούς του συνεχούς χωρίου ACO αλγορίθμους: ο CACS και ο  $ACO_R$  όπου επεξηγήθηκαν και περιγράφηκαν οι θετικές και αρνητικές όψεις τους.

Ο προτεινόμενος αλγόριθμος περιγράφηκε και εφαρμόστηκε μαζί με τους αλγορίθμους CACS και  $ACOR_R$ , σε έξι αρκετά καλά προβλήματα αναφοράς βελτιστοποίησης. Τα αποτελέσματα έδειξαν ότι η νέα πρόταση ήταν ικανή να παράγει σημαντικά καλύτερα αποτελέσματα στα τέσσερα από τα έξι προβλήματα αναφοράς (για το πρόβλημα του Sphere, όλοι οι αλγόριθμοι ήταν ικανοί στην εύρεση του ολικού βέλτιστου). Μόνο για το πρόβλημα του Rosenbrock η πρόταση παρουσίασε μία μέση απόδοση χειρότερη από εκείνη που λήφθηκε από τον CACS

Ωστόσο, αναλύοντας το box-plot του προβλήματος, θα μπορούσαμε ένα δούμε ότι αμφότεροι οι CACS και MACACO παρουσίασαν μία πολύ όμοια συμπεριφορά, αν και ο MACACO παρουσίασε κάποιες αποκλίσεις, πιθανόν εξαιτίας της αδυναμίας του να πηδήξει από ένα τοπικό βέλτιστο σε ένα καλύτερο όταν αμφότερα είναι σε μία υπολογίσιμη απόσταση το ένα από το άλλο στο χώρο αναζήτησης.

Αν και ο MACACO απαιτεί τον υπολογισμό των ιδιοδιανυσμάτων και των ιδιοτιμών του πίνακα συσχετισμού σε κάθε επανάληψη, ο μέσος υπολογιστικός χρόνος που πήρε για την βελτιστοποίηση των προβλημάτων αναφοράς ήταν μικρότερος από εκείνον που παρουσιάστηκε από τον  $ACO_R$  και μόνο δύο φορές περίπου υψηλότερος από εκείνον που παρουσιάστηκε από τον CACS. Ωστόσο, η σημαντική βελτίωση στα αποτελέσματα παρουσιάστηκε από τον MACACO, όταν συγκρινόμενος με το CACS, δείχνει ότι το τμήμα αξίζει.

Αυτή η επιστημονική δημοσίευση δείχνει ότι οι ACO προσεγγίσεις είναι πολλά υποσχόμενες όσον αφορά την αντιμετώπιση της βελτιστοποίησης σε συνεχής χώρους, αφού αυτοί οι αλγόριθμοι μπορούν να δώσουν καλής ποιότητας λύσεις για προκλητικά προβλήματα με ένα σχετικά μικρό υπολογιστικό κόστος.

## **3.2 Βελτιστοποίηση Αποικίας Μυρμηγκιών για Συνεχή Χωρία με Μεταφορά Συνάθροισης Φερομονών**

### **3.2.1. Εισαγωγή**

Ως ένα βιο-εμπνευσμένο υπολογιστικό παράδειγμα, η βελτιστοποίηση Αποικίας Μυρμηγκιών (ACO) έχει εφαρμοστεί με επιτυχία σε ένα μεγάλο αριθμό από απαιτητικά υπολογιστικά προβλήματα. Η ACO προσομοιώνει τη συλλογική συμπεριφορά των μυρμηγκιών, τα οποία επικοινωνούν χρησιμοποιώντας ίχνη φερομόνης. Ωστόσο, η ACO είναι κυρίως εφαρμόσιμη σε διακριτά προβλήματα βελτιστοποίησης όπως το πρόβλημα του περιοδεύοντα πωλητή (TSP)

Αν και χρησιμοποιώντας μεταφορά ίχνους φερομόνης είναι πολύ αποτελεσματικό στη λύση διακριτών προβλημάτων βελτιστοποίησης, μία άμεση εφαρμογή της μεταφοράς ίχνους φερομόνης σε προβλήματα βελτιστοποίησης πραγματικών παραμέτρων είναι δύσκολη. Σε αυτή την ενότητα, εισάγουμε "φερομόνες συνάθροισης", επίσης παρατηρηθείσες στη φύση, και προτείνουμε έναν αλγόριθμο "συστήματος συνάθροισης φερομόνης (APS)" για τη λύση προβλημάτων βελτιστοποίησης πραγματικών παραμέτρων.

Όταν ένα άτομο ίδιου είδους έρχεται σε επαφή με τη φερομόνη, εξάγει μία αντίδραση, εξαρτώμενη από το τύπο της φερομόνης. Με αυτό το τρόπο μεταβιβάζεται συγκεκριμένη πληροφορία. Οι φερομόνες που προκαλούν σύμπλεγμα συμπεριφορών στα είδη, οι οποίες φέρνουν τα άτομα πιο κοντά, αναφέρονται ως φερομόνες συνάθροισης [13]. Πολλές συναρτήσεις αθροιστικής συμπεριφοράς έχουν παρατηρηθεί. Αυτές περιέχουν σημάδεμα και κάλυψη χώρου αναζήτησης τροφής [1], εύρεσης καταφύγιου, και άμυνας. Οι κατσαρίδες παράγουν μία συγκεκριμένη φερομόνη με το περίπωμα τους όταν βρίσκουν ένα ασφαλές καταφύγιο, το οποίο ελκύει άλλα μέλη των ειδών τους [18]. Ως αποτέλεσμα, οι φερομόνες συνάθροισης λειτουργούν έτσι ώστε τα άτομα να συναθροίζονται γύρω από μία "καλή θέση" με θετική ανάδραση. Οι φερομόνες εξατμίζονται με κάποιο ρυθμό. Αυτό αποτρέπει τα άλλα άτομα να συναθροιστούν σε μία τοπική θέση.

Το Σύστημα Συνάθροισης Φερομόνης (APS) που προτάθηκε εδώ χρησιμοποιεί αυτές τις φερομόνες συνάθροισης ως μία βασική μεταφορά του μοντέλου. Ενώ η ACO είναι εφαρμόσιμη κυρίως σε διακριτά προβλήματα, το APS μπορεί να λύσει προβλήματα βελτιστοποίησης πραγματικών παραμέτρων. Αν και χρειάζομαστε περισσότερη μελέτη, τα αποτελέσματα δείχνουν ότι δουλεύει αρκετά καλά σε ποικίλους ελέγχους συναρτήσεων που χρησιμοποιήθηκαν στην εξελικτική υπολογιστική κοινότητα.

### **3.2.2. Το Σύστημα Συνάθροισης Φερομόνης**

Αφού ο προτεινόμενος αλγόριθμος είναι μία παραλλαγή του ACO και χρησιμοποιεί κανόνες ενημέρωσης φερομόνης όμοιους με το ACO, το ACO περιγράφεται σύντομα στο τμήμα 2.1. Το APS παρουσιάζεται με λεπτομέρειες στα τμήματα 2.2 και 2.3.

### Μία Σύντομη Περίληψη του ACO

Στο ACO, οι δραστηριότητες αναζήτησης εκτελούνται με τα αποκαλούμενα μυρμήγκια. Ένα κινούμενο μυρμήγκι απλώνει ένα ίχνος φερομόνης στο έδαφος. Ένα μυρμήγκι συναντώντας απρόσμενα ένα προγενέστερα απλωμένο ίχνος μπορεί να το εντοπίσει και να αποφασίσει με μεγάλη πιθανότητα να το ακολουθήσει, συνεπώς ενισχύει το ίχνος με τη δική του φερομόνη.

Στο [5], ο αλγόριθμος ονομάστηκε σύστημα μυρμηγκιού (AS). Για ένα TSP, το AS δουλεύει ως εξής. Έστω  $m$  ο συνολικός αριθμός των μυρμηγκιών. Κάθε μυρμήγκι έχει τα ακόλουθα χαρακτηριστικά: (i) επιλέγει τη πόλη που θα πάει με μία πιθανότητα που είναι μία συνάρτηση της απόστασης της πόλης και της ποσότητας του ίχνους που υπάρχει στην συνδεδεμένη ακμή (ii) για να αναγκάσουμε το μυρμήγκι να κάνει μία επιτρεπόμενη περιήγηση, οι μεταβάσεις σε ήδη επισκεφθείσες πόλεις δεν επιτρέπονται μέχρι η περιήγηση να ολοκληρωθεί (iii) όταν ολοκληρώνει μία περιήγηση, απλώνει ένα ίχνος σε κάθε ακμή  $(i, j)$  που επισκέφθηκε.

Έστω  $\tau_{ij}(t)$  η ένταση του ίχνους της ακμής  $(i, j)$  στην επανάληψη  $t$ . Για  $t=0$ , οι αρχικές τιμές  $\tau_{ij}(0)$  για την ένταση του ίχνους έχουν τεθεί στις ακμές. Όταν όλα τα μυρμήγκια ολοκληρώνουν τις περιηγήσεις τους, η ένταση του ίχνους σε κάθε ακμή ενημερώνεται σύμφωνα με τον ακόλουθο τύπο:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij} \quad (1)$$

όπου  $\rho$  ( $0 \leq \rho < 1$ ) είναι ένας συντελεστής τέτοιος ώστε το  $(1-\rho)$  να αναπαριστά την εξάτμιση του ίχνους μεταξύ της επανάληψης  $t$  και  $t+1$ , και  $\Delta \tau_{ij}^k$  είναι η ποσότητα ανά μονάδα μήκους του ίχνους απλωμένο στην ακμή  $(i, j)$  από το  $k$ -στό μυρμήγκι μεταξύ της επανάληψης  $t$  και  $t+1$ . Είναι καθορισμένο να είναι αντιστρόφως ανάλογο με το μήκος της περιήγησης του  $k$ -οστού μυρμηγκιού έτσι ώστε οι μεγάλες τιμές του  $\Delta \tau_{ij}^k$  να δίνονται εάν το μήκος της περιήγησης του  $k$ -οστού μυρμηγκιού είναι συντομότερο. Αυτή η διαδικασία είναι επαναλαμβανόμενη μέχρι οι δοθείσες συνθήκες τερματισμού να ικανοποιηθούν.

### Το Βασικό Μοντέλο του Συστήματος Συνάθροισης Φερομόνης

Υπάρχει μία μεγάλη διαφορά μεταξύ των ACO και APS στο πώς η φερομόνη λειτουργεί στο χώρο αναζήτησης. Η πυκνότητα της φερομόνης στην ACO ορίζεται ως ένα ίχνος σε μία ακμή μεταξύ κόμβων ενός δοθέντος προβλήματος αλληλουχίας. Στο APS, η πυκνότητα της φερομόνης ορίζεται από μία συνάρτηση πυκνότητας στο χώρο αναζήτησης  $X$  στον  $R^n$ . Στο πραγματικό κόσμο, οι φερομόνες συνάθροισης χρησιμοποιήθηκαν από τα είδη για να επικοινωνούν με μέλη της κοινότητας τους για να μοιράζονται πληροφορίες σχετικά με τη τοποθεσία της τροφής, ασφαλές καταφύγιο, τους συντρόφους, ή τους εχθρούς.

Όπως στην ACO, το APS αποτελεί ένα κυκλικό μοντέλο. Σε κάθε κύκλο του APS, δανεισμένο από το φυσικό μοντέλο, τα  $m$  άτομα προσελκύονται σε θέσεις από το άθροισμα της φερομόνης στο χώρο αναζήτησης  $X$ . Αυτά προσελκύονται περισσότερο σε θέσεις όπου η πυκνότητα της φερομόνης είναι υψηλότερη, και λιγότερο στις θέσεις όπου η πυκνότητα είναι χαμηλότερη. Συνεπώς, στο σύστημα μας θεωρούμε ότι τα άτομα επιλέγουν τις θέσεις τους ανάλογα με τη πυκνότητα της φερομόνης, ή πιο συγκεκριμένα, με πιθανότητα ανάλογη με τη συνάρτηση του αθροίσματος της πυκνότητας στο χώρο αναζήτησης  $X$  ως εξής.

Έστω  $x$  μεταβλητή στο χώρο αναζήτησης  $X$ , δηλαδή το  $x \in X$ , και  $\tau(t, x)$  είναι η συνάρτηση πυκνότητας του αθροίσματος της φερομόνης στο APS στο κύκλο  $t$ . Αρχικά, στο APS κύκλο ( $t=0$ ), το άθροισμα της φερομόνης κατανέμεται ομοιόμορφα, δηλαδή  $\tau(0, x)=c$ , όπου  $c$  είναι μία σταθερά. Η συνάρτηση πυκνότητας πιθανότητας της συνάθροισης φερομόνης στο APS στο κύκλο  $t$ ,  $p_t(t, x)$ , ορίζεται ως εξής:

$$\dots p_r(t, x) = \frac{\tau(t, x)}{\int_x \tau(t, x) dx} \quad (2)$$

Ανάλογα με το πρόβλημα που λύνουμε, μία συνάρτηση καταλληλότητας  $f(x)$  αντιστοιχεί το  $x \in X$ . Κάθε ένα από τα  $m$  άτομα εκπέμπει άθροισμα φερομόνης γύρω από τη θέση του  $x$  στο  $X$  ανάλογα με τη συνάρτηση καταλληλότητας  $f(x)$ . Ενδέχεται να υπάρχουν ερωτήσεις ως προς το ποιά προσέγγιση είναι περισσότερο όμοια με αυτή που χρησιμοποιείται στο φυσικό κόσμο, για να χρησιμοποιηθεί μία απόλυτη τιμή του  $f(x)$ , μία σχετική τιμή, ή ένα σύστημα κατάταξης. Ωστόσο, στο σύστημα μας χρησιμοποιούμε σειρά  $r$  ( $r=1,2,\dots, m$ ) του κάθε ατόμου για να κάνει τη διαφορά καταλληλότητας μεταξύ των ατόμων πιο ευδιάκριτη. Το υψηλότερο σε κατάταξη άτομο έχει τιμή  $m$  και το χαμηλότερο σε κατάταξη άτομο έχει τιμή  $1$ . Ένα άτομο το οποίο έχει σειρά  $r$  εκπέμπει άθροισμα φερομόνης γύρω στο  $x_{t,r}$  με συνάρτηση πυκνότητας αναπαριστώμενη από  $\Delta\tau'(t,r,x_{t,r},x)$  όπου  $x_{t,r}$  είναι η θέση του ατόμου με σειρά  $r$ . Τότε το νέο άθροισμα πυκνότητας φερομόνης που εκπέμπεται σε ένα APS κύκλο  $t$  από  $m$  άτομα είναι

$$\dots \Delta\tau(t, x) = \sum_{r=1}^m \Delta\tau'(t, r, x_{t,r}, x) \quad (3)$$

Εδώ, υποθέτουμε ότι το συνολικό άθροισμα του όγκου φερομόνης που εκπέμπεται από  $m$  άτομα σε κάθε κύκλο  $t$  είναι ίσος με  $C$ , δηλαδή,

$$\int_x \Delta\tau(t, x) dx = C. \quad (4)$$

Αυτή η υπόθεση είναι σημαντική όταν μελετούμε την δειγματοληπτική μέθοδο στο τμήμα 2.3. Όταν όλα τα  $m$  άτομα ολοκληρώσουν την επιλογή των θέσεων τους, το συνολικό άθροισμα της πυκνότητας φερομόνης στο  $X$  ενημερώνεται σύμφωνα με τον ακόλουθο τύπο, όπως στην Εξίσωση 1 :

$$\tau(t+1, x) = \rho \cdot \tau(t, x) + \Delta\tau(t, x). \quad (5)$$

Αφού πραγματοποιηθεί η ενημέρωση φερομόνης, τα άτομα επανέρχονται στις αρχικές συνθήκες και ο επόμενος κύκλος APS ξεκινάει. Αυτή η διαδικασία επαναλαμβάνεται. Συνεπώς, η πυκνότητα της φερομόνης στα επιθυμητά σημεία στο χώρο αυξάνεται όσο ο APS κύκλος επαναλαμβάνεται και το APS αναμένεται να συγκλίνει σε μία επιθυμητή λύση. Εδώ, συζητάμε τη μορφή του  $\Delta\tau'(t,r,x_{t,r},x)$  της Εξίσωσης 3. Υποθέτουμε ότι (i)  $\Delta\tau'(t,r,x_{t,r},x)$  είναι επικεντρωμένο γύρω από τη θέση  $x_{t,r}$  (η θέση  $x_{t,r}$  έχει την υψηλότερη πυκνότητα), (ii) ένα χαρακτηριστικό με υψηλότερη σειρά εκπέμπει περισσότερη φερομόνη από ότι ένα άτομο σε χαμηλότερη σειρά. Επιπλέον, (iii) εισάγουμε ένα φαινόμενο συνεργασίας στο οποίο το  $\Delta\tau'(t,r,x_{t,r},x)$  επηρεάζεται από τη κατανομή των άλλων ατόμων και επιμηκνεται προς τη κατεύθυνση της κατανομής. Σε αυτή την έρευνα, χρησιμοποιούμε την ακόλουθη Γκαουσιανή συναρτησιακή μορφή για το  $\Delta\tau'(t,r,x_{t,r},x)$  ( $r=1,2,\dots,m$ ) για την Εξίσωση 3.

$$\Delta\tau'(t, r, x_{t,r}, x) = \frac{C}{\sum_{k=1}^m k^\alpha} r^\alpha N(x_{t,r}, \beta^2 \Sigma_t). \quad (6)$$

Εδώ  $\alpha(>0)$  είναι μία παράμετρος για να ρυθμίσουμε τη σχετική σπουδαιότητα της σειράς κατάταξης, ο  $\Sigma_t$  είναι ο πίνακας συνδιακύμανσης εκτιμώμενος από τη κατανομή  $m$  ατόμων στο χώρο αναζήτησης  $X$  για το κύκλο  $t$ , το  $\beta$  ( $\beta>0$ ) είναι μία παράμετρος για τον έλεγχο του εύρους της κατανομής των φερομόνων, η  $N(x_{t,i}, \beta^2 \Sigma_t)$  είναι μία πολυμεταβλητή κανονική κατανομή.

Η συνάρτηση που ορίστηκε από την Εξίσωση 6 ικανοποιεί και τις τρεις συνθήκες που αναφέρθηκαν παραπάνω. Έχει τη μεγαλύτερη τιμή στο  $x_{t,r}$ , τα άτομα σε υψηλότερη κατάταξη έχουν μεγαλύτερες συναρτησιακές τιμές, και ο πίνακας συνδιακύμανσης  $\beta^2 \Sigma_t$  αντανακλά μία κατανομή των ατόμων. Με μεγαλύτερες τιμές του  $\alpha$ , η αναλογία της συνολικής ποσότητας της φερομόνης που παράγεται από υψηλότερης κατάταξης άτομα αυξάνεται. Με μεγαλύτερες τιμές του  $\beta$ , το άθροισμα φερομόνης διαδίδεται ευρύτερα στο χώρο αναζήτησης  $X$  και με μικρότερες τιμές του  $\beta$ , το άθροισμα φερομόνης η φερομόνη συνάθροισης διαδίδεται λιγότερο. Τότε, το  $\Delta\tau(t, x)$ , η συνολική πυκνότητα φερομόνης που εκπέμπεται από  $m$  άτομα στο κύκλο  $t$ , μπορεί να ληφθεί από την Εξίσωση 3. Μπορούμε να δούμε επίσης ότι το  $\Delta\tau(t, x)$  ικανοποιεί την Εξίσωση 4, δηλαδή, η συνολική ποσότητα της φερομόνης σε κάθε κύκλο είναι ίση με  $C$ .

## Τεχνική Δειγματοληψίας

Σε αυτή την ενότητα, συζητάμε πώς δειγματίζουμε τα άτομα από την αθροιστική συνάρτηση πυκνότητας της φερομένης  $\tau(t+1, x)$  ληφθείσα από την Εξίσωση 5. Όπως περιγράφηκε στο 2.2, τα άτομα του δείγματος λαμβάνονται με πιθανότητα ανάλογη με τη αθροιστική πυκνότητα  $\tau(t+1, x)$ . Για να εκτελεστεί μία αναλογική δειγματοληψία, χρειάζεται να λάβουμε τη συνάρτηση πυκνότητας πιθανότητας  $p_{\tau}(t+1, x)$  από  $\tau(t+1, x)$ . Το  $\tau(t+1, x)$  στην Εξίσωση 5 μπορεί να επαναδιατυπωθεί ως

$$\tau(t+1, x) = \rho^{t+1} \tau(0, x) + \sum_{h=0}^t \rho^h \Delta \tau(t-h, x). \quad (7)$$

Από τις Εξισώσεις 2, 4, και 7, το  $p_{\tau}(t+1, x)$  λαμβάνεται ως

$$p_{\tau}(t+1, x) = \frac{\rho^{t+1}}{\sum_{k=0}^{t+1} \rho^k} \cdot \frac{\tau(0, x)}{C} + \sum_{h=0}^t \frac{\rho^h}{\sum_{k=0}^{t+1} \rho^k} \cdot \frac{\Delta \tau(t-h, x)}{C}. \quad (8)$$

Γενικότερα, εάν μία συνάρτηση πυκνότητας πιθανότητας (pdf)  $f(x)$  μπορεί να διασπασθεί σε υπό-συναρτήσεις πυκνότητας πιθανότητας  $f_k(x)$  αφού  $f(x) = p_1 f_1(x) + p_2 f_2(x) + \dots + p_s f_s(x)$ , τότε μπορούμε να δειγματίζουμε την  $f(x)$  ως εξής: πρώτα επιλέγουμε  $f_s(x)$  ( $s=1, 2, \dots, S$ ) σύμφωνα με τη πιθανότητα  $(p_1, p_2, \dots, p_s)$ , και κατοπιν δειγματίζουμε την  $f_s(x)$ , όπου,  $(p_1, p_2, \dots, p_s)$  είναι μία συνάρτηση κατανομής. Χρησιμοποιώντας αυτή τη μέθοδο, μπορούμε να δειγματίζουμε το  $p_{\tau}(t+1, x)$  της Εξίσωσης 8 με τον ακόλουθο τρόπο. Στην Εξίσωση 8, οι όροι  $\tau(0, x)/C$  και  $\Delta \tau(t-h, x)/C$  μπορούν να είναι συναρτήσεις πυκνότητας πιθανότητας από την Εξίσωση 4. Συνεπώς, πρώτα επιλέγουμε  $\tau(0, x)/C$  ή  $\Delta \tau(t-h, x)/C$  ( $h=0, 1, \dots, t$ ) σύμφωνα με τη πιθανότητα  $\rho^h / \sum_{k=0}^{t+1} \rho^k$  ( $h=0, 1, \dots, t, t+1$ ). Αυτό το ονομάζουμε κυκλική δειγματοληψία. Μετά, δειγματίζουμε  $\tau(0, x)/C$  ή  $\Delta \tau(t-h, x)/C$ . Δειγματίζοντας το  $\tau(0, x)/C$  είναι απλό διότι είναι ομοιόμορφη δειγματοληψία. Η δειγματοληψία του  $\Delta \tau(t-h, x)/C$  είναι όμοια με τη κυκλική δειγματοληψία. Γενικώς,  $\Delta \tau(t-h, x)/C$  μπορεί να ληφθεί από την Εξίσωση 6 ως

$$\frac{\Delta \tau(t-h, x)}{C} = \sum_{r=1}^m \frac{r^\alpha}{L} N(x_{t-h, r}, \beta^2 \Sigma_{t-h}), \quad (9)$$

όπου  $L = \sum_{k=1}^m k^\alpha$ . Όπως με τη κυκλική δειγματοληψία, πρώτα επιλέγουμε τάξη  $r$  σύμφωνα με τη

πιθανότητα  $r^\alpha/L$  ( $r=1, 2, \dots, m$ ). Ονομάζουμε αυτή τη δειγματοληψία, δειγματοληψία τάξης. Μετά από τη δειγματοληψία τάξης, δειγματίζουμε τη  $N(x_{t-h, r}, \beta^2 \Sigma_{t-h})$ , χρησιμοποιώντας την διάσπαση του Cholesky [20]. Για να εκτελέσουμε αυτή τη δειγματοληψία, βασισμένη στην Εξίσωση 8, χρειαζόμαστε μεγάλη μνήμη για να αποθηκεύσουμε τις τιμές του διανύσματος  $x_{t-h, r}$  και το πίνακα συνδιακύμανσης  $\beta^2 \Sigma_{t-h}$  όταν ο  $t$  κύκλος APS γίνεται μεγαλύτερος. Σε αυτή τη περίπτωση  $\rho^h \rightarrow 0$  για μεγάλα  $h$  αφού  $0 < \rho < 1$ . Συνεπώς, μπορούμε να περιορίσουμε το μέγιστο αριθμό των κύκλων για να διατηρήσουμε τα δεδομένα μέχρι μία σταθερά  $H$ . Τότε, η Εξίσωση 8 για  $t \geq H$  μπορεί να παρασταθεί ως

$$p_{\tau}(t+1, x) = \sum_{h=0}^{H-1} \frac{\rho^h}{\sum_{l=0}^{H-1} \rho^l} \cdot \frac{\Delta \tau(t-h, x)}{C}. \quad (10)$$

### 3.2.3. Πειραματική Μελέτη

#### Πειραματική Μεθοδολογία

Στο APS κυκλικό μοντέλο, κρατάμε τα καλύτερα  $e$  ( $e = m \times E_{rate}$ ) άτομα σε κάθε κύκλο. Αυτά μεταφέρονται στον επόμενο κύκλο. Αυτό δρα ως μία εκλεκτική στρατηγική στο GAs [8]. Η παράμετρος  $E_{rate}$  χρησιμοποιήθηκε για να ελέγχουμε τον αριθμό των εκλεκτών σε κάθε κύκλο. Χρησιμοποιούμε τις ακόλουθες τρεις συναρτήσεις ελέγχου: την Ελλειψοειδή συνάρτηση ( $F_{Ellipsoidal}$ ), τη συνάρτηση Ridge ( $F_{Ridge}$ ), και τη συνάρτηση Rosenbrock ( $F_{Rosenbrock}$ ). Η  $F_{Ridge}$  έχει αδύναμη διασύνδεση μεταξύ των μεταβλητών. Η  $F_{Rosenbrock}$  έχει ισχυρή διασύνδεση μεταξύ των

μεταβλητών. Η  $F_{\text{Ellipsoidal}}$  δεν έχει διασύνδεση μεταξύ των μεταβλητών. Ένα πρόβλημα μεγέθους  $n=20$  χρησιμοποιείται για να όλες τις συναρτήσεις ελέγχου.

$$F_{\text{Ellipsoidal}} = \sum_{i=1}^n ix_i^2, \quad (-3.12 \leq x_i < 7.12), \quad (11)$$

$$F_{\text{Ridge}} = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2, \quad (-44 \leq x_i < 84), \quad (12)$$

$$F_{\text{Rosenbrock}} = \sum_{i=2}^n (100(x_i - x_{i-1}^2) + (x_i - 1)^2), \quad (-2.048 \leq x_i < 2.048), \quad (13)$$

Η προκαθορισμένη τιμή για τις παραμέτρους είναι η εξής:  $m=100$ ,  $\rho=0.92$ ,  $\alpha=4$ ,  $\beta=0.6$ ,  $E_{\text{rate}}=0.1$ , και  $H=200$ . Αυτές οι προκαθορισμένες τιμές επιλέχθηκαν για να συντονίσουν το APS για να λειτουργεί καλά όταν ελέγχθηκε με τη συνάρτηση  $F_{\text{Rosenbrock}}$ . Υπολογίσαμε την APS μετρώντας τον #OPT τους (αριθμός των εκτελέσεων στις οποίες το APS πέτυχε την εύρεση ολικού βέλτιστου) και MNE (μέσος αριθμός υπολογισμών συναρτήσεων για να βρούμε το ολικό βέλτιστο σε αυτές τις εκτελέσεις όπου βρήκε το βέλτιστο). Θεωρήσαμε ότι ο εντοπισμός της επιτυχούς λύσης εάν όλοι οι παράμετροι  $(x_1, \dots, x_n)$  του καλύτερου χαρακτηριστικού ήταν μέσα στο εύρος  $[(o_j - 0.0001), (o_j + 0.0001)]$  για όλα τα  $j$ , όπου  $(o_1, \dots, o_n)$  είναι το βέλτιστο σημείο. 20 εκτελέσεις πραγματοποιήθηκαν. Κάθε εκτέλεση συνεχίστηκε μέχρι το ολικό βέλτιστο να βρεθεί ή να πραγματοποιηθεί ένα μέγιστο από 500,000 υπολογισμούς.

## Αποτελέσματα

### (1) Αποτελέσματα με προκαθορισμένες τιμές παραμέτρων

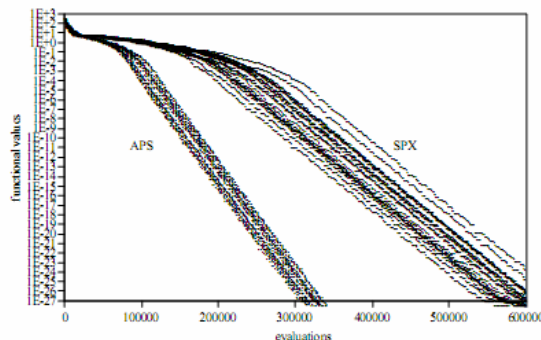
Ο Πίνακας 1 δείχνει τα αποτελέσματα του APS χρησιμοποιώντας τις προκαθορισμένες τιμές των παραμέτρων που περιγράφονται από το τμήμα 3.1.

**Table 1 Results of APS with default values**

Function		$F_{\text{Ellipsoidal}}$	$F_{\text{Ridge}}$	$F_{\text{Rosenbrock}}$
APS	#OPT	20	20	20
	MNE	80300.0	101660.0	121760.0
	STD*	1469.7	1765.4	8882.3
SPX	#OPT	20	20	20
	MNE	138900.0	179078.3	265766.3
	STD*	1362.2	2337.1	27152.5

\* STD: Standard Deviation

Τα αποτελέσματα συγκρίνονται με τα αποτελέσματα του SPX, μία τυπική κατάσταση της τέχνης του τελεστή διασταύρωσης σε ένα πραγματικά κωδικοποιημένο GAs [9]. Το APS έδειξε καλύτερα αποτελέσματα από το SPX στις  $F_{\text{Ellipsoidal}}$  και  $F_{\text{Ridge}}$ , αν και είναι δυνατόν να ρυθίσουμε περαιτέρω το APS όπως περιγράφεται παρακάτω (βλέπε εικόνα 3). Στην  $F_{\text{Rosenbrock}}$ , το APS έδειξε πολύ καλύτερα αποτελέσματα από ότι το SPX.

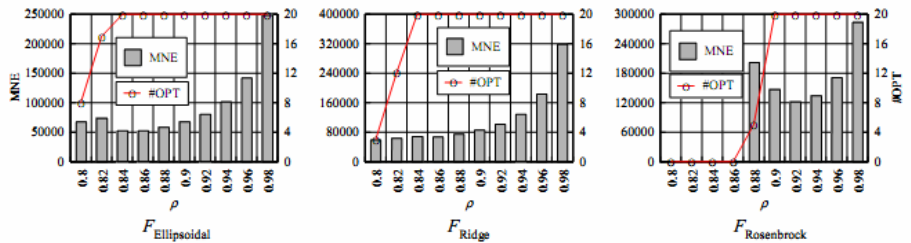


**Fig. 2** Convergence process on  $F_{\text{Rosenbrock}}$

Η Εικόνα 2 δείχνει τις διαδικασίες σύγκλισης αμφοτέρων των APS και SPX στην  $F_{\text{Rosenbrock}}$ . Μπορούμε να δούμε ότι το APS συγκλίνει περισσότερο σταθερά και γρήγορα από ότι το SPX.

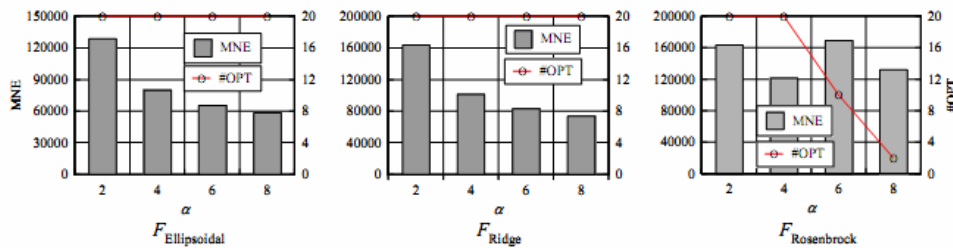
**(2) Ευαισθησία των παραμέτρων**

Για να δούμε τη δράση του συντελεστή εξάτμισης  $\rho$  στο APS, μεταβάλλαμε τη τιμή του  $\rho$  από τη προκαθορισμένη τιμή στο εύρος  $[0.8, 0.98]$  (Εικόνα 3).



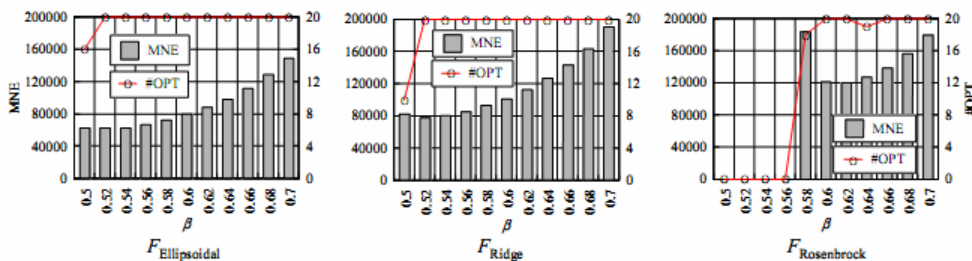
**Fig. 3** Effect of the coefficient of pheromone evaporation  $\rho$

Όταν η τιμή του  $\rho$  γίνει μεγαλύτερη, η φερομόνη που εξατμίστηκε σε προγενέστερους κύκλους παραμένει για ένα μεγάλο χρονικό διάστημα. Με τις  $F_{Ellipsoidal}$  και  $F_{Ridge}$ , το APS με  $\rho$  τιμές στο  $[0.84, 0.86]$  έδειξε ότι  $\#OPT=20$  και τις καλύτερες (μικρότερες) MNE τιμές. Με τη  $F_{Rosenbrock}$ , η οποία έχει ισχυρή διασύνδεση μεταξύ των μεταβλητών, η APS με τιμή  $\rho$  0.92 έδειξε μικρότερο MNE με  $\#OPT = 20$ . Για αυτό το πρόβλημα η τιμή του  $\rho$  είναι μεγαλύτερη από αυτές που προκύπτουν από τις  $F_{Ellipsoidal}$  και  $F_{Ridge}$ . Με μεγαλύτερες τιμές του  $\rho$ , το APS χρησιμοποιεί τη πληροφορία της πυκνότητας φερομόνης σε προγενέστερους κύκλους περισσότερο απ' ό,τι με μικρότερες τιμές του  $\rho$  και μπορεί να βρει λύση αποτελεσματικά σε συναρτήσεις που έχουν ισχυρή σύνδεση μεταξύ των μεταβλητών. Για να δούμε την επίδραση της παραμέτρου  $\alpha$ , η οποία ρυθμίζει τη σχετική σπουδαιότητα της σειράς όταν κάθε άτομο εκπέμπει φερομόνη, δοκιμάσαμε για το  $\alpha$  στο εύρος του  $[2, 8]$  με μέγεθος βήματος 2 (Εικόνα 4). Με μεγαλύτερες τιμές του  $\alpha$ , τα υψηλότερης σειράς άτομα εκπέμπουν φερομόνη με αυξανόμενους ρυθμούς. Με μεγαλύτερες τιμές του  $\alpha$ , η απόδοση του APS αυξάνεται, όπως φαίνεται από τα αποτελέσματα με τις  $F_{Ellipsoidal}$  και  $F_{Ridge}$ . Με τη  $F_{Rosenbrock}$ , η οποία έχει ισχυρή διασύνδεση μεταξύ των μεταβλητών, μεγαλύτερες τιμές του  $\alpha$  προκαλούν παράπλευρες απώλειες στη σύγκλιση



**Fig. 4** Effect of parameter  $\alpha$

Η παράμετρος  $\beta$  ελέγχει το εύρος της κατανομής της φερομόνης που εκπέμπεται από τα άτομα. Με αυξανόμενες τις μεγαλύτερες τιμές του  $\beta$ , τα άτομα θα εκπέμπουν περισσότερη φερομόνη. Ελέγξαμε το  $\beta$  στο εύρος  $[0.5, 0.7]$  με μήκος βήματος 0.02 (Εικ 5).



**Fig. 5** Effect of parameter  $\beta$

Με συνάρτηση  $F_{Ridge}$  και  $F_{Ellipsoidal}$ , μη έχοντας και έχοντας αδύναμη αντίστοιχα διασύνδεση μεταξύ των μεταβλητών, οι μικρότερες τιμές του  $\beta$  έχουν καλύτερη απόδοση. Με τη συνάρτηση  $F_{Rosenbrock}$ , η οποία έχει μία ισχυρή διασύνδεση μεταξύ των μεταβλητών, αμφότερες οι



μικρότερες και οι μεγαλύτερες τιμές του  $\beta$  έχουν φτωχή απόδοση, δείχνοντας υπέρ-εξερευνητική και υπέρ-εκμετάλλευτική συμπεριφορά.

#### **3.2.4. Συμπεράσματα**

Περιγράψαμε το *σύστημα συνάθροισης φερομόνης (APS)*, το οποίο χρησιμοποιεί φερομόνες συνάθροισης σαν μία βασική μεταφορά του μοντέλου, για τη λύση προβλημάτων βελτιστοποίησης πραγματικών μεταβλητών. Μελετήσαμε το APS με αρκετά προβλήματα δοκιμασίας. Τα αποτελέσματα έδειξαν ότι το APS μπορούσε να λύσει προβλήματα βελτιστοποίησης πραγματικών παραμέτρων αρκετά καλά. Επίσης εξερευνήσαμε την ευαισθησία των παραμέτρων ελέγχου του APS .

## Κεφάλαιο 4

### Μια Εισαγωγή Στη Σύγκλιση Των Αλγορίθμων ACO

#### 4.1 Μία Σύντομη Απόδειξη Σύγκλισης για μία Κλάση από Αλγορίθμους Βελτιστοποίησης Αποικίας μυρμηγκίων

Εδώ αποδεικνύονται κάποιες ιδιότητες σύγκλισης για μία κλάση από αλγορίθμους βελτιστοποίησης αποικίας μυρμηγκίων. Συγκεκριμένα, αποδεικνύεται ότι για κάθε μικρή σταθερά  $\varepsilon > 0$  και για ένα αρκετά μεγάλο αριθμό από  $t$  επαναλήψεις του αλγορίθμου, η πιθανότητα εύρεσης μίας βέλτιστης λύσης τουλάχιστον μία φορά είναι  $P^*(t) \geq 1 - \varepsilon$  και ότι αυτή η πιθανότητα τείνει στο 1 για  $t \rightarrow \infty$ . Αποδεικνύεται επίσης, ότι μετά την εύρεση μία βέλτιστης λύσης, χρειάζεται ένας πεπερασμένος αριθμός από επαναλήψεις για τα ίχνη φερομόνης που σχετίζονται με την εύρεση βέλτιστης λύσης για να αυξηθούν περισσότερο από κάθε άλλο ίχνος φερομόνης και ότι, για  $t \rightarrow \infty$  κάθε μυρμηγκί θα παράγει τη βέλτιστη λύση κατά τη διάρκεια της  $t$ -οστής επανάληψης με πιθανότητα  $P \geq 1 - \hat{\varepsilon}(\tau_{\min}, \tau_{\max})$ , όπου το  $\tau_{\min}$  και το  $\tau_{\max}$  είναι η ελάχιστη και η μέγιστη τιμή που μπορούν να ληφθούν από τα ίχνη φερομόνης.

##### 4.1.1 Εισαγωγή

Η βελτιστοποίηση αποικίας μυρμηγκίων (ACO) είναι μία μετα-ευριστική για τη προσεγγιστική λύση προβλημάτων συνδυαστικής βελτιστοποίησης που εμπνεύστηκε από την ερευνητική συμπεριφορά για τις μυρμηγκικές αποικίες. Στους ACO αλγορίθμους, οι υπολογιστικές πηγές είναι προσδιορισμένες σε ένα σύνολο από σχετικά απλούς πράκτορες (τεχνητά μυρμηγκία) που εκμεταλλεύονται την στιγμεργική (stigmergic) επικοινωνία, δηλαδή μία μορφή έμμεσης επικοινωνίας που οφείλεται στο περιβάλλον [4],[11] για τη κατασκευή λύσεων στο υπό μελέτη πρόβλημα. Η κατασκευή κάλων λύσεων είναι αποτέλεσμα της συνεργατικής αλληλεπίδρασης των πρακτόρων.

Τα τελευταία δέκα χρόνια, ένας αριθμός από εφαρμογές σε διάφορα NP-hard προβλήματα συνδυαστικής βελτιστοποίησης [5],[6] έδειξαν εμπειρικά την αποτελεσματικότητα της βελτιστοποίησης αποικίας μυρμηγκίων.

Εδώ παρουσιάζεται μία απλή απόδειξη σύγκλισης που εφαρμόζεται απευθείας σε τουλάχιστον δύο από τους πιο (πειραματικά επιτυχείς) ACO αλγορίθμους:

1) σύστημα αποικίας μυρμηγκίων (ACS) [7] και 2) MAX-MIN σύστημα μυρμηγκίων (MMAS) [19].

##### 4.1.2 Το Πρόβλημα Και Ο Αλγόριθμος

Θεώρησε ένα πρόβλημα ελαχιστοποίησης  $(S, f, \Omega)$ , όπου  $S$  είναι το σύνολο από τις (υποψήφιος) λύσεις,  $f$  είναι η αντικειμενική συνάρτηση, η οποία αντιστοιχεί σε κάθε υποψήφια λύση  $s \in S$  μία τιμή (κόστος)  $f(s)$  της αντικειμενικής συνάρτησης  $f(s)$ , και  $\Omega$  είναι ένα σύνολο από περιορισμούς, οι οποίοι ορίζουν το σύνολο των εφικτών υποψήφιων λύσεων. Ο στόχος της ελαχιστοποίησης είναι να βρούμε μία βέλτιστη λύση  $s^*$ , δηλαδή μία εφικτή υποψήφια λύση ελαχίστου κόστους.

Το πρόβλημα συνδυαστικής βελτιστοποίησης  $(S, f, \Omega)$  απεικονίζεται σε ένα πρόβλημα το οποίο μπορεί να χαρακτηριστεί από τα ακόλουθα.

1) Ένα πεπερασμένο σύνολο  $C = \{c_1, c_2, \dots, c_{N_c}\}$  από συνιστώσες.

2) Ένα πεπερασμένο σύνολο  $X$  από καταστάσεις του προβλήματος, με στοιχεία όλες τις δυνατές ακολουθίες  $x = \langle c_i, c_j, \dots, c_k, \dots \rangle$  από τα στοιχεία του  $C$ .

3) Το σύνολο από (υποψήφιος) λύσεις  $S$  είναι το υποσύνολο του  $X$  (δηλαδή  $S \subseteq X$ ).

4) Ένα σύνολο από εφικτές καταστάσεις  $\tilde{X}$ , με  $\tilde{X} \subseteq X$ , ορισμένες με δοκιμές στο πρόβλημα και που επαληθεύεται ότι δεν είναι αδύνατο να ολοκληρωθεί μία ακολουθία  $x \in \tilde{X}$  που ικανοποιεί τους περιορισμούς του  $\Omega$ .

5) Ένα μη κενό σύνολο  $S^*$  από βέλτιστες λύσεις, με  $S^* \subseteq \tilde{X}$  και  $S^* \subseteq S$ .

Δοθίσας της παραπάνω τυποποίησης, τα τεχνητά μυρμήγκια κατασκευάζουν υποψήφιες λύσεις εκτελώντας τυχαίες διαδρομές στον πλήρως συνεκτικό σταθμισμένο γράφο  $G=(C,L,T)$ , όπου οι κορυφές είναι οι συνιστώσες  $C$ , το σύνολο  $L$  συνδέει πλήρως τις συνιστώσες  $C$ , και  $T$  είναι ένα διάνυσμα των επονομαζόμενων ιχνών φερομόνης  $\tau$ . Ο γράφος  $G$  ονομάζεται κατασκευαστικός γράφος.

Κάθε τεχνητό μυρμήγκι τοποθετείται σε μία τυχαία επιλεγμένη κορυφή του γράφου και τότε εκτελεί ένα περίπατο υπό τυχαίες συνθήκες κινούμενο σε κάθε βήμα από κορυφή σε κορυφή στο γράφο με τέτοιο τρόπο που η επόμενη κορυφή να επιλέγεται στοχαστικά σύμφωνα με την ισχύ της φερομόνης στα τόξα. Ενώ κινείται από το μία κορυφή του γράφου  $G$  στην άλλη, οι περιορισμοί  $\Omega$  χρησιμοποιούνται για να αποτρέψουν τα μυρμήγκια από τη κατασκευή μη πρακτικών λύσεων. Μόλις τα μυρμήγκια ολοκληρώσουν το περίπατο τους, ενημερώνονται τα ίχνη φερομόνης. Τυπικά η κατασκευαστική συμπεριφορά της λύσης ενός μυρμηγκιού μπορεί να περιγραφεί ακολούθως.

#### ANT\_SOLUTION\_CONSTRUCTION

While ( $x_k \in \tilde{X}$  and  $x_k \notin S$ ) do:

at each step  $k$  after building the sequence

$x_k = \langle c_1, c_2, \dots, c_k \rangle$ , select the next vertex (component)  $c_{k+1}$  randomly following

$$P(c_{k+1} = c | T, x_k) = \begin{cases} \frac{\tau(c_k, c)^a}{\sum_{\substack{y \in C \\ (c_k, y) \in J_{c_k}}} \tau(c_k, y)^a}, & \text{if } (c_k, c) \in J_{c_k} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Όπου  $0 < a < +\infty$  είναι μία παράμετρος και μια συνδεση  $(c_k, y)$  ανηκει στο  $J_{c_k}$  αν η ακολουθία  $x_{k+1} = \langle c_1, c_2, \dots, c_k, y \rangle$  είναι τέτοια ώστε  $x_{k+1} \in \tilde{X}$ . Αν σε κάποια σημεία στη κατασκευή της λύσης το σύνολο  $J_{c_k}$  είναι κενό το μυρμήγκι απορρίπτεται και τερματίζεται η κατασκευή της λύσης του

Μόλις όλα τα μυρμήγκια τερματίσουν την διαδικασία ANT\_SOLUTION\_CONSTRUCTION, ξεκινάει μία φάση ενημέρωσης φερομόνης για τα ίχνη φερομόνης που έχουν τροποποιηθεί. Έστω  $\hat{s}$  η καλύτερη εφικτή λύση μέχρι τώρα και  $s_t$  η καλύτερη εφικτή λύση στην επανάληψη  $t$  του αλγορίθμου.  $f(\hat{s})$  και  $f(s_t)$  είναι οι τιμές των αντιστοίχων αντικειμενικών συναρτήσεων. Η διαδικασία ενημέρωσης φερομόνης μειώνει κατά ένα μικρό παράγοντα  $\rho$ , επονομαζόμενο ρυθμό εξάτμισης, τη τιμή των ιχνών φερομόνης σε όλες τις συνδέσεις στο  $L$  και τότε αυξάνει τη τιμή των ιχνών φερομόνης στις συνδέσεις που ανήκουν στο  $\hat{s}$  (προσθέτοντας φερομόνη μόνο σε εκείνες τις ακμές που ανήκουν στη καλύτερη μέχρι τώρα λύση, γνωστή στη βιβλιογραφία ως η καθολικά καλύτερη αδρανής ενημέρωση φερομόνης [5],[7]).

Η διαδικασία ενημέρωσης φερομόνης περιγράφεται όπως παρακάτω.

#### PHEROMONE\_UPDATE

$\forall (i, j): \tau(i, j) \leftarrow (1 - \rho) \cdot \tau(i, j)$

If  $f(s_t) < f(\hat{s})$ , then  $\hat{s} \leftarrow s_t$

$\forall (i, j) \in \hat{s}: \tau(i, j) \leftarrow \tau(i, j) + g(\hat{s})$

$\forall (i, j): \tau(i, j) \leftarrow \max\{\tau_{\min}, \tau(i, j)\}$

όπου  $\rho, 0 < \rho < 1$  είναι ο ρυθμός εξάτμισης,  $\tau_{\min} > 0$

είναι μία παράμετρος, και  $g(s), 0 < g(s) < +\infty$  είναι μία συνάρτηση

με  $g: S \mapsto \mathbb{R}^+$ ,  $f(s) < f(s') \Rightarrow g(s) \geq g(s')$ .

Ο αλγόριθμος αρχικοποιείται όπως παρακάτω.

**INITIALIZE**

Generate a feasible solution  $s'$  and set  $\hat{s} = s'$

$\forall (i, j)$ , set  $\tau(i, j) = \tau_0$

For each ant

select a start vertex  $c_1$  according to some problem dependent criterion

set  $k = 1$  and  $x_k = \langle c_1 \rangle$

where  $\tau_0, \tau_{\min} \leq \tau_0 < +\infty$ , is a parameter

Μετά τη τοποθέτηση αρχικών τιμών, ο αλγόριθμος επαναλαμβάνεται συνεχώς διαμέσου των διαδικασιών ANT\_SOLUTION\_CONSTRUCTION και PHEROMONE\_UPDATE, μέχρι να βρεθεί κάποια συνθήκη τερματισμού. Ακολουθως, θα ονομάσουμε τον ACO αλγόριθμο  $ACO_{gb, \tau_{\min}}$  όπου το  $gb$  δείχνει ότι χρησιμοποιήθηκε ο καθολικά καλύτερος κανόνας αναβάθμισης, ενώ το  $\tau_{\min}$  δείχνει ότι ενισχύεται το χαμηλότερο όριο στο εύρος των εφικτών ίχνων φερομόνης. Για τα ακόλουθα υποθέτουμε ότι  $\tau_{\min} < g(s^*)$ , που μπορεί να επιτευχθεί από τη ρύθμιση, δηλαδή,  $\tau_0 = g(s') / 2$ , όπου  $s'$  είναι η λύση που χρησιμοποιήθηκε για την τοποθέτηση αρχικών τιμών στον  $ACO_{gb, \tau_{\min}}$ .

**4.1.3 Απόδειξη Σύγκλισης**

Αποδεικνύουμε δύο θεωρήματα για τον αλγόριθμο που προτάθηκε στο προηγούμενο τμήμα. Πρώτα, δείχνουμε ότι ο  $ACO_{gb, \tau_{\min}}$  εγγυάται την εύρεση της βέλτιστης λύσης, με πιθανότητα που μπορεί να είναι κοντά στο ένα εάν δοθεί αρκετός χρόνος. Δεύτερον, αποδεικνύουμε ότι, μετά από ένα προκαθορισμένο αριθμό από επαναλήψεις  $t_0$ , αφού πρώτα έχει βρεθεί η βέλτιστη λύση, τα ίχνη φερομόνης στις συνδέσεις της βέλτιστης λύσης είναι μεγαλύτερα από εκείνα σε καθε άλλη σύνδεση. Αυτό το αποτέλεσμα επεκτείνεται για να δείξει ότι μία βέλτιστη λύση μπορεί να κατασκευαστεί με μία πιθανότητα μεγαλύτερη από  $1 - \hat{\epsilon}(\tau_{\min}, \tau_{\max})$ , όπου  $\tau_{\max}$  είναι η μέγιστη τιμή των φερομονών που μπορούμε να παρουμε.

Πριν να αποδείξουμε το πρώτο θεώρημα, είναι βολικό να δείξουμε ότι, εξαιτίας της εξάτμισης της φερομόνης, το μέγιστο δυνατό επίπεδο φερομόνης  $\tau_{\max}$  είναι φραγμένο ασυμπτωτικά.

**Πρόταση 1 :** Για κάθε  $\tau_{ij}$ , ισχύουν τα ακόλουθα :

$$\lim_{t \rightarrow \infty} \tau_{ij}(t) \leq \tau_{\max} = \frac{1}{\rho} \cdot g(s^*). \quad (2)$$

**Απόδειξη :** Η μέγιστη δυνατή ποσότητα φερομόνης που προστέθηκε σε κάθε ακμή  $(i, j)$  μετά από κάθε επανάληψη είναι  $g(s^*)$ . Καθαρά, στην επανάληψη 1, το μέγιστο δυνατό ίχνος φερομόνης είναι  $(1 - \rho) \cdot \tau_0 + g(s^*)$ , στην επανάληψη 2, είναι  $(1 - \rho)^2 \cdot \tau_0 + (1 - \rho) \cdot g(s^*) + g(s^*)$ , κ.τ.λ. Γι'αυτό το λόγο σύμφωνα με την εξάτμιση φερομόνης, το ίχνος φερομόνης στην επανάληψη  $t$  είναι φραγμένο από

$$\tau_{ij}^{\max}(t) = (1 - \rho)^t \cdot \tau_0 + \sum_{i=1}^t (1 - \rho)^{t-i} \cdot g(s^*).$$

Ασυμπτωτικά, αφού  $0 < \rho < 1$ , αυτό το άθροισμα συγκλίνει στο  $\tau_{\max} = \frac{1}{\rho} \cdot g(s^*)$

**Πρόταση 2 :** Μετά την εύρεση μίας βέλτιστης λύσης, ενθυμούμενοι ότι  $\tau_{ij}^*(t) \geq \tau_{\min}$  και ότι χρησιμοποιήθηκε ο καθολικά καλύτερος κανόνας αναβάθμισης φερομόνης, έχουμε ότι η  $\tau_{ij}^*(t)$  αυξάνεται. Η απόδειξη

$$\forall (i, j) \in s^* : \lim_{t \rightarrow \infty} \tau_{ij}^*(t) = \tau_{\max} = \frac{1}{\rho} \cdot g(s^*)$$

όπου  $\tau_{ij}^*$  είναι η τιμή του ίχνους φερομόνης στις συνδέσεις  $(i, j) \in s^*$ .

**Απόδειξη :** Η απόδειξη της πρότασης 2 είναι βασικά μία επανάληψη της απόδειξης της πρότασης 1, περιορισμένη στις συνδέσεις της βέλτιστης λύσης ( το  $\tau_0$  αντικαταστάθηκε από το  $\tau_{ij}^*(t^*)$  στην απόδειξη της πρότασης 1, όπου  $t^*$  είναι η επανάληψη όταν έχει βρεθεί η πρώτη βέλτιστη λύση )

Η πρόταση 1 λέει ότι για την παρακάτω απόδειξη του θεωρήματος 1, το μόνο απαραίτητο σημείο είναι ότι  $\tau_{\min} > 0$  διότι το  $\tau_{\max}$  θα είναι φραγμένο από την εξάτμιση φερομόνης. Η Πρόταση 2 δηλώνει επιπροσθέτως ότι τα ίχνη φερομόνης σε όλες τις συνδέσεις του  $s^*$ , μόλις έχει βρεθεί μία βέλτιστη λύση, συγκλίνει στο  $\tau_{\max} = g(s^*) / \rho$ .

Τώρα μπορούμε να αποδείξουμε το παρακάτω θεώρημα

**Θεώρημα 1 :** Έστω  $P^*(t)$  η πιθανότητα ότι ο αλγόριθμος βρίσκει μία βέλτιστη λύση τουλάχιστον μία φορά μέσα στις πρώτες  $t$  επαναλήψεις. Τότε για μία τυχαία επιλογή ενός μικρού  $\varepsilon > 0$  και για ένα αρκετά μεγάλο  $t$ , ισχύει ότι

$$P^*(t) \geq 1 - \varepsilon$$

και ασυμπτωτικά

$$\lim_{t \rightarrow \infty} P^*(t) = 1$$

**Απόδειξη :**

Εξαιτίας των ορίων των ιχνών φερομόνης  $\tau_{\min}$  και  $\tau_{\max}$ , μπορούμε να εγγυηθούμε ότι κάθε εφικτή λύση στην (1) γίνεται με πιθανότητα  $p_{\min} > 0$ . Ένα τετριμμένο κάτω φράγμα για το  $p_{\min}$  μπορεί να δοθεί ως εξής :

$$p_{\min} \geq \hat{p}_{\min} = \frac{\tau_{\min}^a}{(N_C - 1) \cdot \tau_{\max}^a + \tau_{\min}^a} \quad (3)$$

Για την παραγωγή αυτού του φράγματος θεωρούμε την ακόλουθη "χειρότερη περίπτωση" κατάσταση. Το ίχνος φερομόνης συνδεδεμένο με την επιθυμητή απόφαση είναι  $\tau_{\min}$ , ενώ όλες οι άλλες εφικτές λύσεις (υπάρχουν το πολύ  $N_C - 1$ ) έχουν ένα συνδεδεμένο ίχνος φερομόνης του  $\tau_{\max}$ , όπου  $N_C$  είναι η πληθικότητα του συνόλου  $C$  των συνιστωσών. Τότε, κάθε γενική λύση  $s'$ , συμπεριλαμβανομένης κάθε βέλτιστης λύσης  $s^* \in S^*$ , μπορεί να παραχθεί με πιθανότητα  $\hat{p} \geq \hat{p}_{\min}^n > 0$ , όπου  $n < +\infty$  είναι το μέγιστο μήκος της ακολουθίας. Επειδή είναι αρκετό ότι ένα μурμηγκι βρίσκει μία βέλτιστη λύση, ένα κάτω φράγμα για το  $P^*(t)$  δίνεται από :

$$\hat{P}^*(t) = 1 - (1 - \hat{p})^t.$$

Επιλέγοντας το  $t$  αρκετά μεγάλο, αυτή η πιθανότητα μπορεί να γίνει μεγαλύτερη από κάθε τιμή  $1 - \varepsilon$  επειδή έχουμε ότι  $\lim_{t \rightarrow \infty} \hat{P}^*(t) = 1$ .

**Θεώρημα 2 :** Έστω  $t^*$  η επανάληψη όταν έχει βρεθεί η πρώτη βέλτιστη λύση. Τότε μία τιμή  $t_0$  υπάρχει έτσι ώστε να ισχύουν τα ακόλουθα:

$$\tau_{ij}(t) > \tau_{kl}(t)$$

$$\forall (i, j) \in s^*, \forall (k, l) \in L \wedge (k, l) \notin s^* \text{ και } \forall t > t^* + t_0 = t^* + [(1 - \rho) / \rho]$$

**Απόδειξη :** Μετά από μία μεταβατική περίοδο  $t_0$  που έχει διανυθεί από την επανάληψη στην οποία βρέθηκε η πρώτη βέλτιστη λύση (δηλαδή, για  $t > t^* + t_0$ ), το ίχνος φερομόνης που χρησιμοποιήθηκε στις συνδέσεις στη βέλτιστη λύση είναι μεγαλύτερο από ότι σε κάθε άλλη εφικτή σύνδεση. Στη πραγματικότητα, εξαιτίας της χρήσης του καθολικά καλύτερου κανόνα ενημέρωσης, μόνο οι συνδέσεις που ανήκουν στο  $s^*$  αυξάνουν τα ίχνη φερομόνης τους, ενώ τα ίχνη φερομόνης όλων των υπολοίπων συνδέσεων μειώνονται κατά ένα παράγοντα  $\rho$  μετά από κάθε επανάληψη μέχρι να φτάσουν το χαμηλότερο φράγμα  $\tau_{\min}$ . Εμείς τώρα δίνουμε ένα φράγμα στο μήκος της μεταβατικής περιόδου  $t_0$ . Για να το κάνουμε αυτό, υποθέτουμε την ακόλουθη περίπτωση χειρότερης κατάστασης. Έστω  $(i, j)$  μία σύνδεση που ανήκει στο  $s^*$  με ένα συνδεδεμένο ίχνος φερομόνης στην επανάληψη  $t^*$  του  $\tau_{ij}^*(t^*) = \tau_{\min}$ . Επίσης, έστω  $(k, l)$  μία

σύνδεση που δεν ανήκει στο  $s^*$  με ένα συνδεδεμένο ίχνος φερομόνης στην επανάληψη  $t^*$  του  $\tau_{ij}(t^*) = \tau_{\max}$ . Στην επανάληψη  $t^* + t'$ ,  $\tau_{ij}^*(t)$  γίνεται

$$\tau_{ij}^*(t^* + t') = (1-\rho)^{t'} \cdot \tau_{\min} + \sum_{i=0}^{t'-1} (1-\rho)^i \cdot g(s^*) > t' \cdot (1-\rho)^{(t'-1)} \cdot g(s^*)$$

ενώ η τιμή του  $\tau_{kl}(t)$  στην επανάληψη  $t^* + t'$  είναι

$$\tau_{kl}(t^* + t') = \max\{\tau_{\min}, (1-\rho)^{t'} \cdot \tau_{\max}\}.$$

Για τους σκοπούς μας, η ενδιαφέρουσα περίπτωση είναι όταν αυτό το μέγιστο αντιστοιχεί στο  $(1-\rho)^{t'} \cdot \tau_{\max}$ . Τότε έχουμε ότι  $\tau_{ij}^*(t^* + t') > \tau_{kl}(t^* + t')$  όταν  $t' \cdot (1-\rho)^{(t'-1)} \cdot g(s^*) > (1-\rho) \cdot \tau_{\max}$  η οποία είναι η περίπτωση όταν

$$t' > \left[ \frac{\tau_{\max} \cdot (1-\rho)}{g(s^*)} \right] = \left[ \frac{(1-\rho)}{\rho} \right] \equiv t_0$$

Από το Θεώρημα 2 αμεσα βλέπουμε ότι, για κάθε επανάληψη  $t > t^* + t_0$ , κάθε μυρμήγκι θα είναι ικανό να κατασκευάσει το  $s^*$  δημιουργώντας, σε κάθε κατασκευαστικό βήμα, καθοριστικά την επιλογή που συνδέεται με το μεγαλύτερο ίχνος φερομόνης. Στην οριακή περίπτωση, μόλις η βέλτιστη λύση έχει βρεθεί, μπορούμε να δώσουμε μία χαμηλότερη εκτίμηση φράγματος για μία πιθανότητα το μυρμήγκι να κατασκευάσει μία βέλτιστη λύση όταν ακολουθεί τη στοχαστική πολιτική του αλγορίθμου. Πριν να αποδείξουμε αυτό, δείχνουμε στη Πρόταση 3 ότι το ίχνος φερομόνης των συνδέσεων που δεν είναι στη βέλτιστη λύση συγκλίνει στο όριο της τιμής  $\tau_{\min}$ .

**Πρόταση 3:** Μόλις μία βέλτιστη λύση έχει βρεθεί και για κάθε  $\tau_{ij}(t)$  τέτοιο ώστε  $(i, j) \notin s^*$ , ισχύει ότι  $\lim_{t \rightarrow \infty} \tau_{ij}(t) = \tau_{\min}$ .

**Απόδειξη:** Επειδή μετά από την εύρεση βέλτιστης λύσης οι συνδέσεις που δεν ανήκουν στη βέλτιστη λύση δεν λαμβάνουν καθόλου φερομόνη, η τιμή τους μπορεί μόνο να μειωθεί. Συγκεκριμένα, μετά από μία επανάληψη,  $\tau_{ij}(t^* + 1) = \max\{\tau_{\min}, (1-\rho) \cdot \tau_{\max}\}$  και μετά από  $t'$  επαναλήψεις,  $\tau_{ij}(t^* + t') = \max\{\tau_{\min}, (1-\rho)^{t'} \cdot \tau_{\max}\}$ . Τότε είναι φανερό ότι για  $t \rightarrow \infty$ ,  $\tau_{ij}(t) \rightarrow \tau_{\min}$ . Στη πραγματικότητα, η Πρόταση 3 μπορεί να γίνει ισχυρότερη δείχνοντας ότι η τιμή  $\tau_{\min}$  επιτυγχάνεται σε ένα πεπερασμένο αριθμό επαναλήψεων.

**Πρόταση 4:** Ξεκινώντας από την επανάληψη  $t' \geq t^* + t_0$ , ισχύει ότι

$$\forall (i, j) \notin s^*, \tau_{ij}(t) = \tau_{\min}$$

όπου  $t^*$  είναι η επανάληψη όταν η πρώτη βέλτιστη λύση  $s^*$  έχει βρεθεί και

$$t_0 = \left[ \frac{(\ln \tau_{\min} - \ln \tau_{\max})}{\ln(1-\rho)} \right]$$

**Απόδειξη:** Μπορούμε να δώσουμε ένα φράγμα στο  $t_0$  για τουλάχιστον ένα  $(i, j) \in s^*$ , ισχύει ότι  $\tau_{ij}(t^*) = \tau_{\max}$ . Ακολουθώντας την ίδια αιτιολογία όπως στη Πρόταση 3, σημειώνουμε ότι μετά από  $t'$  επαναλήψεις  $\tau_{ij}(t^* + t') = \max\{\tau_{\min}, (1-\rho)^{t'} \cdot \tau_{\max}\}$ . Τότε το  $t_0$  είναι η πρώτη επανάληψη έτσι ώστε  $(1-\rho)^{t_0} \cdot \tau_{\max} \leq \tau_{\min}$ . Είναι εύκολο να δείξουμε ότι αυτή είναι η περίπτωση για

$$t_0 = \left[ \frac{(\ln \tau_{\min} - \ln \tau_{\max})}{\ln(1-\rho)} \right].$$

**Πόρισμα 1:** Έστω  $t^*$  να είναι η επανάληψη όταν έχει βρεθεί η πρώτη βέλτιστη λύση και  $P(s^*, t, k)$  είναι η πιθανότητα ένα τυχαίο μυρμήγκι  $k$  να κατασκευάσει το  $s^*$  στην  $t$ -οστή επανάληψη, με  $t > t^*$ .

Τότε ισχύει ότι

$$\lim_{t \rightarrow \infty} P(s^*, t, k) \geq 1 - \hat{\epsilon}(\tau_{\min}, \tau_{\max}).$$

**Απόδειξη:** Έστω ένα μυρμήγκι  $k$  τοποθετημένο στο στοιχείο  $i$  και  $(i, j)$  να είναι μία σύνδεση του  $s^*$ . Ενθυμούμενοι ότι τα μυρμήγκια επιλέγουν την επόμενη συνιστώσα  $j$  στο σύνολο  $J_i$ , ένα

κατώτερο φράγμα  $\hat{p}_{ij}^*(t)$  η πιθανότητα  $p_{ij}^*(t)$  ότι το μυρμήγκι  $k$  κάνει τη "σωστή επιλογή"  $(i,j)$  δίνεται από τον όρο

$$\hat{p}_{ij}^*(t) = \frac{(\tau_{ij}^*(t))^\alpha}{(\tau_{ij}^*(t))^\alpha + \sum_{(i,k) \neq s} (\tau_{ik}(t))^\alpha}.$$

Εξαιτίας των Προτάσεων 2 και 3, έχουμε

$$\hat{p}_{ij}^* = \lim_{t \rightarrow \infty} \hat{p}_{ij}^*(t) = \frac{\lim_{t \rightarrow \infty} (\tau_{ij}^*(t))^\alpha}{\lim_{t \rightarrow \infty} \left[ (\tau_{ij}^*(t))^\alpha + \sum_{(i,k) \neq s} (\tau_{ik}(t))^\alpha \right]} = \frac{\tau_{\max}^\alpha}{\tau_{\max}^\alpha + (N_C - 1) \cdot \tau_{\min}^\alpha} \quad (5)$$

Γι'αυτό το λόγο στο όριο, ένα κάτω φράγμα για το  $P(s^*, t, k)$  είναι το  $\hat{p}_k = (\hat{p}_{ij}^*)^n$ .

Θέτοντας  $\hat{\varepsilon} = 1 - \hat{p}_k$  αποδεικνύεται το πόρισμα.

#### 4.1.4 Σχόλια

Στο τμήμα 4.1.3, αποδείξαμε δύο θεωρήματα σχετικά με τη σύγκλιση του  $\text{ACO}_{\text{gb}, \tau_{\min}}$ . Σε αυτό το τμήμα, συζητάμε σύντομα τη σημασία των δύο αυτών θεωρημάτων και δείχνουμε πώς αυτά σχετίζονται με την απόδειξη του Gutjahr's[14].

*A. Τι λέει στη πραγματικότητα η απόδειξη;*

Είναι χρησιμο να κατανοήσουμε τι μας λένε πραγματικά οι αποδείξεις που παρουσιάζονται στο τμήμα 4.1.3. Το Θεώρημα 1 λέει ότι ο αλγόριθμος μας δεν αποκλείει την πιθανότητα εύρεσης της βέλτιστης λύσης, ενώ το Θεώρημα 2 λέει ότι, μόλις έχει βρεθεί η βέλτιστη λύση τα ίχνη φερομόνης στις συνδέσεις που ανήκουν στη βέλτιστη λύση θα γίνουν μεγαλύτερα από αυτά κάθε άλλης σύνδεσης. Βασισμένο σε αυτό το πρόσφατο αποτέλεσμα, το Πόρισμα 1 δίνει ένα φράγμα στη πιθανότητα κατασκευής μίας βέλτιστης λύσης. Από την άλλη μεριά, οι αποδείξεις δεν λένε τίποτα σχετικά με το χρόνο που απαιτείται για να βρεθεί μία βέλτιστη λύση, που μπορεί να είναι αστρονομικά μεγάλος.

Είναι σημαντικό να τονίσουμε το ρόλο της αυστηρής ανισότητας στην ενημέρωση φερομόνης του  $\text{ACO}_{\text{gb}, \tau_{\min}}$  (βλέπε το αντικείμενο 2 στο κανόνα ενημέρωσης φερομόνης του τμήματος 4.1.2).

Υποθέτουμε ότι αντικαθιστούμε το σύμβολο  $<$  με το σύμβολο  $\leq$  στο κανόνα αναβάθμισης φερομόνης. Για ένα πρόβλημα με διάφορα διακριτά βέλτιστα, αυτό θα μπορούσε να οδηγήσει σε μία κατάσταση στην οποία πηγαίνουμε πίσω και εμπρός μεταξύ διακριτών καθολικών βέλτιστων. Αν και αυτό δεν επηρεάζει το Θεώρημα 1, το Θεώρημα 2 δεν θα μπορούσε να ισχύει πλέον, δηλαδή, δεν θα μπορούσαμε να αποδείξουμε ότι όλες οι φερομόνες εγκαθίστανται σε μία καθολική βέλτιστη λύση. Είναι ενδιαφέρον, ότι όλες αυτές οι εφαρμογές των ACO αλγορίθμων που χρησιμοποιούν το καλύτερο κανόνα καθολικής ενημέρωσης επίσης χρησιμοποιούν μία αυστηρή ανισότητα, δηλαδή, ενημερώνουν τη καθολικά καλύτερη λύση μόνο όταν μία βελτιωμένη λύση έχει βρεθεί.

Ένα σημαντικό ρόλο στην απόδειξη του Θεωρήματος 1 έπαιξαν τα  $\tau_{\min}$  και  $\tau_{\max}$ . Όσο μικρότερη είναι η αναλογία  $\tau_{\max} / \tau_{\min}$ , τόσο μεγαλύτερο είναι το κάτω φράγμα  $\hat{p}_{\min}$  που δίνεται σε αυτήν την απόδειξη. Αυτό είναι σημαντικό διότι όσο μεγαλύτερο είναι το  $\hat{p}_{\min}$ , τόσο μικρότερη είναι η χειρότερη περίπτωση εκτίμησης του αριθμού των επαναλήψεων  $t$  που απαιτούνται για να εξασφαλίσουν ότι μία βέλτιστη λύση βρέθηκε με πιθανότητα μεγαλύτερη από  $1 - \varepsilon$ . Στην πραγματικότητα, το πιο στενό φράγμα ελήφθη αν όλα τα ίχνη φερομόνης είναι τα ίδια, δηλαδή, για τη περίπτωση ομοιόμορφης κατασκευής τυχαίας λύσης. Σε αυτή τη περίπτωση θα έπρεπε να έχουμε  $\hat{p}_{\min} = 1 / N_C$ . Αυτό, το κατά κάποιο τρόπο αντιφατικό αποτέλεσμα είναι εξαιτίας του γεγονότος ότι η απόδειξη μας είναι βασισμένη στη χειρότερη περίπτωση ανάλυσης: χρειάζεται να θεωρήσουμε τη χειρότερη κατάσταση όπου η μεροληψία στη κατασκευή λύσης εισαχθείσα από τα ίχνη φερομόνης είναι αντιπαραγωγική και οδηγεί σε υποβέλτιστες λύσεις, δηλαδή, υποθέτουμε ότι το επίπεδο ίχνους φερομόνης συνδεδεμένο με τη σύνδεση που χρειάζεται ένα

μυρμήγκι για να περάσει και να κατασκευάσει μία βέλτιστη λύση είναι  $\tau_{\min}$ , ενώ στις άλλες συνδέσεις είναι πολύ μεγαλύτερο-στη χειρότερη περίπτωση, αντιστοιχεί στο  $\tau_{\max}$ . Όπως είπαμε, παράλα αυτά, η κύρια συμβολή του θεωρήματος, σε πρακτικούς όρους, είναι να εξασφαλίσουμε ότι εάν ο αλγόριθμος τρέχει αρκετά, θα βρει μία βέλτιστη λύση με μια μεγάλη πιθανότητα. Αυτό το αποτέλεσμα είναι σημαντικό διότι, όπως θα δούμε στο τμήμα V-A, η απόδειξη σύγκλισης του  $ACO_{gb, \tau_{\min}}$  μπορεί να επεκταθεί για να καλύψει δύο από τους πειραματικά καλύτερους εκτελεστικά ACO αλγορίθμους.

#### B. Σχέση με την απόδειξη σύγκλισης του Gutjahr

Πρόσφατα, ο Gutjahr [14] πρότεινε μία απόδειξη σύγκλισης για το GBAS, ένα αλγόριθμο που ανήκει στη κλάση των ACO. Ο GBAS είναι πολύ όμοιος με τον  $ACO_{gb, \tau_{\min}}$  εκτός από το ότι  $\tau_{\min} = 0$  και ο κανόνας ενημέρωσης φερομόνης αλλάζει τις φερομόνες μόνο όταν, στη τρέχουσα επανάληψη, έχει παραχθεί μία λύση τουλάχιστον τόσο καλή όσο η καλύτερη λύση που έχει βρεθεί μέχρι τώρα. Απέδειξε το παρακάτω θεώρημα:

1) Για κάθε  $\varepsilon > 0$ , για ένα προκαθορισμένο  $\rho$ , και για ένα αρκετά μεγάλο αριθμό από μυρμήγκια, η πιθανότητα  $P$  ότι ένα προκαθορισμένο μυρμήγκι κατασκευάζει τη βέλτιστη λύση στην επανάληψη  $t$  είναι  $P \geq 1 - \varepsilon$  για όλα τα  $t \geq t_0$ , με  $t_0 = t_0(\varepsilon)$ .

2) για κάθε  $\varepsilon > 0$ , για ένα προκαθορισμένο αριθμό από μυρμήγκια, για ένα ρυθμό εξάτμισης  $\rho$  αρκετά κοντά στο μηδέν, η πιθανότητα  $P$  ότι ένα προκαθορισμένο μυρμήγκι κατασκευάζει τη βέλτιστη λύση στην επανάληψη  $t$  είναι  $P \geq 1 - \varepsilon$  για όλα τα  $t \geq t_0$ , με  $t_0 = t_0(\varepsilon)$ .

Υπάρχει ένας αριθμός από διαφορές μεταξύ της απόδειξης του Gutjahr και αυτήν εδώ, η πιο σημαντική αφορά το τύπο της σύγκλισης που αποδείχθηκε. Στη πραγματικότητα, στο Θεώρημα 1 αποδεικνύουμε σύγκλιση σε τιμή (δηλαδή, αποδεικνύουμε ότι ο αλγόριθμος τελικά θα βρει τη βέλτιστη λύση), ενώ ο Gutjahr αποδεικνύει σύγκλιση στη λύση (δηλαδή, αποδεικνύει ότι ο αλγόριθμος θα συγκλίνει σε μία κατάσταση στην οποία παράγει τη βέλτιστη λύση ξανά και ξανά). Η απόδειξη του Gutjahr είναι ισχυρότερη από τη δική μας (αυτή συνεπάγεται το αποτέλεσμα μας, αλλά το αντίστροφο δεν είναι αληθές), αλλά η απόδειξη μας ισχύει για κάθε ACO αλγόριθμο μέχρι να υπάρχει ένα χαμηλότερο φράγμα  $\tau_{\min} > 0$  και ένα άνω φράγμα  $\tau_{\max} < +\infty$  για τα ίχνη φερομόνης. Συνεπώς, ενώ η απόδειξη του Gutjahr ισχύει μόνο για το GBAS, ένας αλγόριθμος ACO που ποτέ δεν εκτελέστηκε και για τον οποίο δεν είναι διαθέσιμα ερευνητικά αποτελέσματα, η απόδειξη εδώ ισχύει, όπως φαίνεται στο τμήμα V, για κάποιους από τους καλύτερα εκτελεσμένους αλγορίθμους ACO που υπάρχουν στη βιβλιογραφία.

Από μία πιο τεχνική σκοπιά, οι δύο αποδείξεις διαφέρουν σε ένα αριθμό σημείων που συνοψίζονται στα ακόλουθα.

1) Το Θεώρημα 1 ισχύει ανεξάρτητα από το τρόπο όπου οι φερομόνες ενημερώνονται (για να είμαστε πιο συγκεκριμένοι, ισχύει, δεδομένου ότι  $\tau_{\min} > 0$ , για κάθε κανόνα ενημέρωσης φερομόνης με  $0 < \rho < 1$  και αυτό προσθέτει μία πεπερασμένη ποσότητα από ίχνος φερομόνης), ενώ η απόδειξη του Gutjahr ισχύει μόνο για GBAS συγκεκριμένους κανόνες ενημέρωσης φερομόνης (σε μία επέκταση του θεωρήματος του [13], αποδεικνύει σύγκλιση κάτω από τη συνθήκη ότι ο GBAS κανόνας ενημέρωσης φερομόνης εφαρμόζεται τουλάχιστον στις τελικές φάσεις του αλγορίθμου).

2) Τα Θεωρήματα 1 και 2, ως επίσης και το Πρόσχημα 1, ισχύουν για  $\tau_{\min} > 0$ , ενώ για την απόδειξη του Gutjahr, τα ίχνη φερομόνης μπορούν να τείνουν στο μηδέν.

3) Ο Gutjahr αποδεικνύει ότι η πιθανότητα παραγωγής βέλτιστης λύσης σε κάθε επανάληψη τείνει στο ένα όσο ο αριθμός των επαναλήψεων τείνει στο άπειρο, ενώ μπορούμε μόνο να αποδείξουμε, εξαιτίας του  $\tau_{\min}$ , ότι τείνει στο  $1 - \varepsilon$  (δες το Πρόσχημα 1).

4) Το αποτέλεσμα εδώ είναι ανεξάρτητο από τον αριθμό των βέλτιστων λύσεων στο  $S^*$ , ενώ μία από τις συνθήκες για το θεώρημα του Gutjahr είναι ότι υπάρχει μία μόνο βέλτιστη λύση (αυτός ο περιορισμός αφαιρέθηκε από τον Gutjahr στο [13]).

5) Στο Θεώρημα 1, η σύγκλιση είναι μία συνάρτηση του  $\tau_{\min}$  και του  $\rho$ , ενώ στο θεώρημα του Gutjahr, είναι μία συνάρτηση του αριθμού των μυρμηγκιών και του  $\rho$ .



#### 4.1.5 ACO Αλγόριθμοι Και Σύγκλιση

Όπως ήδη αναφέραμε, από τη σκοπιά του ερευνητή που ενδιαφέρεται για τις εφαρμογές του αλγορίθμου, το ενδιαφέρον σημείο της απόδειξης σύγκλισης είναι εκείνο που αντιστοιχεί στο Θεώρημα 1, που δηλώνει ότι ο  $ACO_{gb, \tau_{min}}$  βρίσκει μία βέλτιστη λύση με μεγάλη πιθανότητα εάν τρέχει αρκετό χρόνο.

Είναι, συνεπώς, ενδιαφέρον ότι αυτό το θεώρημα εφαρμόζεται επίσης σε ACO αλγόριθμους που διαφέρουν από το  $ACO_{gb, \tau_{min}}$  με το τρόπο που εκτελείται η διαδικασία ενημέρωσης φερομόνης. Γενικά, το Θεώρημα 1 εφαρμόζεται σε κάθε ACO αλγόριθμο για τον οποίο η πιθανότητα  $P(s)$  κατασκευής μίας λύσης  $s \in S$  πάντα παραμένει μεγαλύτερη από μία μικρή σταθερά  $\epsilon > 0$ . Στον  $ACO_{gb, \tau_{min}}$ , αυτό είναι μία απευθείας συνέπεια του γεγονότος ότι  $0 < \tau_{min} < \tau_{max} < +\infty$ , η οποία οφείλεται: 1) στην άμμεση ανάθεση μίας ελάχιστης τιμής  $\tau_{min}$  για τα ίχνη φερομόνης 2) περιορίζοντας τη ποσότητα της φερομόνης που μπορούν να εναποθέσουν τα μυρμηγκία μετά από κάθε επανάληψη, δηλαδή,  $\forall s, g(s) < +\infty$  και 3) αφήνοντας τη φερομόνη να εξατμίζεται στη διάρκεια του χρόνου, δηλαδή, θέτοντας  $\rho > 0$ . Θα ονομάσουμε τη κλάση των ACO αλγορίθμων που ικανοποιούν αυτές τις τρεις συνθήκες  $ACO_{\tau_{min}}$ . Ο  $ACO_{gb, \tau_{min}}$  διαφέρει από τον  $ACO_{\tau_{min}}$  στο ότι επιπρόσθετα απαιτεί τη χρήση του καθολικά καλύτερου κανόνα ενημέρωσης φερομόνης. Συνεπώς, ο  $ACO_{gb, \tau_{min}}$  μπορεί να θεωρηθεί ως μία ειδική περίπτωση του  $ACO_{\tau_{min}}$ . Εξ ορισμού, το Θεώρημα 1 ισχύει για κάθε αλγόριθμο στο  $ACO_{\tau_{min}}$ . Αντιθέτως, αυτό δεν ισχύει γενικά για το Θεώρημα 2 και το Πόρισμα 1.

##### A. Αλγόριθμοι στο $ACO_{\tau_{min}}$

Ακολουθώντας, δείχνουμε ότι τα MMAS και ACS, δύο από τους πειραματικά επιτυχεστέους αλγορίθμους ACO, περιλαμβάνονται στη κλάση  $ACO_{\tau_{min}}$ .

1) *MAX-MIN Σύστημα Μυρμηγκιών*: Ο MMAS (*MAX-MIN Ant System*) είναι ένας από τους καλύτερους εκτελεστικά ACO αλγόριθμους και εφαρμόστηκε επιτυχώς σε προβλήματα όπως το γνωστό πρόβλημα του περιοδεύοντα πωλητή (TSP) και το πρόβλημα τετραγωνικής ανάθεσης (QAP) [18],[19]. Είναι εύκολο να δείξουμε ότι ο MMAS ανήκει στη κλάση  $ACO_{\tau_{min}}$ . Στην πραγματικότητα, υπάρχουν δύο μόνο ασήμαντες διαφορές μεταξύ του MMAS και του  $ACO_{gb, \tau_{min}}$ . Πρώτα, ο MMAS χρησιμοποιεί μία άμμεση τιμή για το  $\tau_{max}$  αντί μίας εμμεσης όπως έγινε στον  $ACO_{gb, \tau_{min}}$ . Ο κύριος λόγος για αυτή την επιλογή είναι ότι ο MMAS περιστασιακά επαναρχικοποιεί τα ίχνη φερομόνης στο  $\tau_{max}$ . Αναφερόμαστε στο [19] για περισσότερες λεπτομέρειες. Δεύτερον, ο MMAS χρησιμοποιεί έναν κάπως πιο γενικό κανόνα ενημέρωσης φερομόνης από τον  $ACO_{gb, \tau_{min}}$ . Όπως στον  $ACO_{gb, \tau_{min}}$ , στον MMAS, μόνο μία λύση χρησιμοποιήθηκε για να επιλεγθούν οι συνδέσεις στις οποίες θα προστεθεί φερομόνη, αλλά επιτρέπει να επιλέξουμε μεταξύ της καλύτερης λύσης  $s_t$  στην  $t$  επανάληψη και της καθολικά καλύτερης λύσης  $\hat{s}$ . Συνεπώς, είναι φανερό ότι το Θεώρημα 1 ισχύει για τον MMAS.

Στο MMAS, εδείχθη πειραματικά ότι μία καλή στρατηγική είναι να επιλέξουμε όλο και συχνότερα τη καθολικά καλύτερη λύση για την ενημέρωση φερομόνης, μέχρι να φτάσουμε μία κατάσταση στην οποία η φερομόνη προστίθεται μόνο στις συνδέσεις που ανήκουν στο  $\hat{s}$ . Σε αυτή τη περίπτωση, είναι εύκολο να προσαρμόσουμε το Θεώρημα 2 έτσι ώστε να ισχύει (υποθέτουμε εδώ ότι δεν εφαρμόστηκε επαναρχικοποίηση φερομόνης). Αρκεί να υπολογίσουμε τη περίοδο μετάβασης ξεκινώντας από  $t' = \max\{t^*, \hat{t}\}$ , όπου  $\hat{t}$  είναι η επανάληψη μετά από την οποία μόνο η καθολικά καλύτερη λύση προσθέτει φερομόνη. Μία παρόμοια αιτιολογία εφαρμόζεται στις αποδείξεις των Προτάσεων 2, 3 και 4, και, συνεπώς, στο Πόρισμα 1.

2) *Σύστημα Αποικίας μυρμηγκιών*: Ο ACS (*Ant Colony System*) [7], άλλος ένας πολύ επιτυχημένος ACO αλγόριθμος, επίσης ανήκει στο  $ACO_{\tau_{min}}$ . Ακόμα, αυτό δεν είναι τόσο άμεσο να το δούμε όπως για τον MMAS. Συνεπώς, πρώτα δίνουμε κάποιες περισσότερες λεπτομέρειες στον ACS και τότε δείχνουμε γιατί το Θεώρημα 1 εφαρμόζεται σ αυτόν.

Ο ACS διαφέρει σε τρία σημεία από τον  $ACO_{gb, \tau_{min}}$ . Πρώτα, ο ACS χρησιμοποιεί τον ψευδοτυχαίο αναλογικό κανόνα δράσης επιλογής. Σε κάθε κατασκευαστικό βήμα, ένα μυρμήγκι έχει δύο πιθανές επιλογές: να επιλέξει τη σύνδεση με τη μεγαλύτερη τιμή ίχνους φερομόνης ή να εκτελέσει μία μεροληπτική εξερεύνηση σύμφωνα με το (1). Η πρώτη επιλογή έγινε με πιθανότητα  $q_0$ , η επόμενη με πιθανότητα  $(1-q_0)$ , όπου  $0 \leq q_0 < 1$  είναι μία παράμετρος. Δεύτερον, ο ACS δεν εφαρμόζει εξάτμιση φερομόνης σε όλες τις συνδέσεις. Ο κανόνας ενημέρωσης χρησιμοποιήθηκε στον ACS ως εξής.

ACS\_OFFLINE\_PHEROMONE\_UPDATE

If  $f(s_t) < f(\hat{s})$ , then  $\hat{s} \leftarrow s_t$   
 $\forall (i, j) \in \hat{s}: \tau(i, j) \leftarrow (1-\rho) \cdot \tau(i, j) + \rho \cdot g(\hat{s})$   
 όπου  $\rho$  είναι η εξάτμιση φερομόνης.

Τρίτον, κάθε μυρμήγκι στον ACS χρησιμοποιεί ένα κανόνα αναβάθμισης τοπικού ίχνους φερομόνης που τα μυρμήγκια εφαρμόζουν αμέσως αφού έχουν διανύσει μία σύνδεση κατά τη διάρκεια της κατασκευής λύσης.

ACS\_ONLINE\_STEP\_BY\_STEP\_PHEROMONE\_UPDATE

$(c_k, c) \in x_{k+1}: \tau(c_k, c) \leftarrow (1-\xi) \cdot \tau(c_k, c) + \xi \cdot \tau_0$

όπου  $\xi$ ,  $0 < \xi < 1$ , και  $\tau_0$  είναι δύο παράμετροι.

Η αποτέλεσμα του τοπικού κανόνα αναβάθμισης είναι να κάνει μία επιλεγμένη σύνδεση λιγότερο επιθυμητή για τα επόμενα μυρμήγκια. Είναι ευκολό να παρατηρήσουμε ότι οι δύο κανόνες αναβάθμισης φερομόνης που χρησιμοποιήθηκαν στον ACS είναι της μορφής  $a_{k+1} = (1-\psi) \cdot a_k + c \cdot b$ , για  $k \geq 1$ , όπου  $a_{k+1}$  και  $a_k$  είναι  $\tau_{ij}(t+1)$  και  $\tau_{ij}(t)$ , και αντίστοιχα,  $b = g(\hat{s})$ ,  $\tau_0$  και  $\psi = \rho, \xi$ . Τότε, έχουμε

$$a_k = (1-\psi)^k \cdot a_0 + b \cdot [1 - (1-\psi)^k]$$

του οποίου το όριο είναι  $b$  όταν  $k \rightarrow \infty$ . Η ακολουθία μειώνεται για  $a_0 > b$  (με μέγιστο  $a_0$ ) και αυξάνεται για  $a_0 < b$  (με μέγιστο το  $b$ ).

Τώρα, η ερώτηση είναι: Πώς το αποτέλεσμα της σύγκλισης μεταφέρεται από τον  $ACO_{gb, \tau_{min}}$  στον ACS; Πρώτα παρατηρούμε, ότι στον ACS η μέγιστη ποσότητα φερομόνης είναι περιορισμένη από το  $\tau_{max} = \tau_{max}^{ACS} = g(\hat{s})$  (αυτό το φράγμα λήφθηκε χωρίς να θεωρήσουμε την τοπική ενημέρωση φερομόνης). Επιπλέον, η παράμετρος  $\tau_0$  στο ACS αντιστοιχεί στο  $\tau_{min}$  στον  $ACO_{gb, \tau_{min}}$ , δηλαδή, καμία τιμή ίχνους φερομόνης δεν μπορεί να πέσει κάτω από  $\tau_0$ . Αυτή είναι η περίπτωση διότι το  $\tau_0$  επιλέχθηκε με τέτοιο τρόπο που είναι μικρότερο από το  $g(\hat{s})$  και σε αυτή τη περίπτωση το όριο της ακολουθίας  $a_k$  από παραπάνω που αντιστοιχεί στο  $\tau_0$ , δίνοντας ένα κάτω φράγμα στο ίχνος φερομόνης κάθε συνιστώσας της λύσης  $(i, j)$ .

Το δεύτερο βήμα είναι να δείξουμε ότι κάθε εφικτή λύση μπορεί να κατασκευαστεί με μη μηδενική πιθανότητα. Ο ευκολότερος τρόπος για να το δούμε αυτό είναι να κάνουμε κάποιες προκαθορισμένες επιλογές  $(i, j)$  στον ACS. Ας υποθέσουμε ότι στη σύνδεση  $(i, j)$  δεν έχουμε αντιστοιχίσει το μεγαλύτερο ίχνος φερομόνης. Τότε, η πιθανότητα επιλογής της σύνδεσης  $(i, j)$  μπορεί να υπολογιστεί ως το γινόμενο της πιθανότητας πραγματοποίησης τυχαίας επιλογής, η οποία είναι  $1-q_0$ , και της πιθανότητας επιλογής της σύνδεσης  $(i, j)$  σε αυτή τη τυχαιοποιημένη επιλογή. Ένα φράγμα για τη τελευταία δίνεται από το  $\hat{p}_{min}$  στο (3). Συνεπώς, ένα κάτω φράγμα για τη πιθανότητα πραγματοποίησης μία συγκεκριμένης επιλογής του κατασκευαστικού βήματος είναι  $(1-q_0) \cdot \hat{p}_{min}$  και το **Θεώρημα 1** εφαρμόζεται στον ACS.

Το **θεώρημα 2**, εφαρμόζεται επίσης στον ACS, εκτός του ότι δεν είναι πλέον δυνατό να παραχθεί εύκολα σε ένα καθοριστικό φράγμα στο μήκος της περιόδου μετάβασης  $t_0$ . Παρόλα αυτά, θα πρέπει να σημειωθεί ότι, μόλις βρεθεί μία βέλτιστη λύση  $s^*$ , εξαιτίας του καλύτερου καθολικού

κανόνα αναβάθμισης φερομόνης, οι συνδέσεις που ανήκουν στο  $s^*$  είναι οι μόνες που συνεχίζουν να δέχονται φερομόνη και αυτό ενδέχεται να αυξάνει τα ίχνη φερομόνης, ενώ τα ίχνη φερομόνης σε όλες τις άλλες συνδέσεις μπορούν μόνο να μειωθούν, μέχρι το κάτω φράγμα  $\tau_0$  (αυτό συμβαίνει κάθε φορά που χρησιμοποιήθηκαν από κάποιο μυρμήγκι εξαιτίας του κανόνα αναβάθμισης τοπικής φερομόνης).

### B. Επιπρόσθετα Χαρακτηριστικά των ACO Αλγορίθμων.

Πολλοί ACO αλγόριθμοι [5],[6] περιλαμβάνουν κάποια χαρακτηριστικά που δεν υπάρχουν στον  $ACO_{gb,\tau_{min}}$ . Το πιο σημαντικό είναι η χρήση της τοπικής αναζήτησης για να βελτιώσουμε τις λύσεις που κατασκευάστηκαν από τα μυρμήγκια και η χρήση ευριστικής πληροφορίας στην επιλογή της επόμενης σύνδεσης ή συνιστώσας. Στην πραγματικότητα, αυτά τα δύο χαρακτηριστικά χρησιμοποιήθηκαν συχνά στον MMAS και στον ACS. Συνεπώς, μία φυσική ερώτηση είναι πώς αυτές οι επεκτάσεις επιδρούν στις αποδείξεις σύγκλισης για τον  $ACO_{gb,\tau_{min}}$  και, γι'αυτό το λόγο, επίσης για τον  $ACO_{\tau_{min}}$ .

Ας θεωρήσουμε πρώτα την πρόσθετη χρήση της τοπικής αναζήτησης. Η τοπική αναζήτηση προσπαθεί να βελτιώσει μία λύση μυρμηγκιού  $s$  εφαρμόζοντας επαναληπτικά, μικρές τοπικές αλλαγές σε αυτό. Τυπικά, η καλύτερη λύση  $s'$  που βρίσκουμε από τη τοπική αναζήτηση επέστρεφει και χρησιμοποιείται για να ενημερώσει τα ίχνη φερομόνης. Είναι ευκολο να δούμε ότι η χρήση τοπικής αναζήτησης δεν επιδρά στις ιδιότητες σύγκλισης του  $ACO_{gb,\tau_{min}}$  - αυτές αναφέρονται μόνο στο τρόπο που οι λύσεις κατασκευάστηκαν και ισχύουν άσχετα από τη χρήση της τοπικής αναζήτησης. Παρά το γεγονός ότι η τοπική αναζήτηση δεν επιδρά στη θεωρητική συμπεριφορά σύγκλισης του  $ACO_{gb,\tau_{min}}$ , είναι γνωστο ότι πρακτικά οι αλγόριθμοι ACO συχνά γίνονται πιο αποτελεσματικοί όταν εφαρμόζονται σε τοπική αναζήτηση [5].

Εκ των προτέρων διαθέσιμη πληροφορία στο πρόβλημα μπορεί να χρησιμοποιηθεί για να παράγουμε ευριστική πληροφορία που μεροληπτεί τις πιθανοθεωρητικές αποφάσεις που παίρνονται από τα μυρμήγκια. Όταν ενσωματώνονται τέτοιες ευριστικές πληροφορίες στον  $ACO_{gb,\tau_{min}}$ , ο (1) γίνεται :

$$P(c_{k+1} | T, x_k) = \begin{cases} \frac{\left[ \tau(c_k, c)^\alpha \cdot \eta(c_k, c)^\beta \right]}{\sum_{\substack{y \in J_k \\ (c_k, y) \in J_k}} \left[ \tau(c_k, y)^\alpha \cdot \eta(c_k, y)^\beta \right]}, & \text{if } (c_k, c) \in J_k \dots \\ 0, & \text{otherwise} \end{cases}$$

όπου  $\eta(c_k, c)$  μετράει την ευριστική επιθυμητότητα της πρόσθετης συνιστώσας λύσης  $c$ , και  $\beta$  είναι μία παράμετρος. Το Θεώρημα 1 δεν επηρεάστηκε από την ευριστική πληροφορία εάν έχουμε  $0 < \eta(i, j) < +\infty$  για κάθε  $(i, j) \in L$  και  $\beta < \infty$ . Σε μία επέκταση του Θεωρήματος 2, μπορούμε να εγγυηθούμε ότι οι συνιστώσες του  $s^*$  είναι οι πιο πιθανές να επιλεγούν, εάν σε κάθε κατασκευαστικό βήμα το γινόμενο  $\tau(c_k, c^*)^\alpha \cdot \eta(c_k, c^*)^\beta$ , όπου  $c^*$  είναι η συνιστώσα που πρέπει να επιλέξουμε για να κατασκευάσουμε την  $s^*$  είναι μέγιστο. Στη πραγματικότητα, αυτό μπορεί να εγγυηθεί εάν  $r_\tau^\alpha > r_\eta^\beta$ , όπου  $r_\tau = \tau_{max} / \tau_{min}$  και  $r_\eta = \frac{\eta_{max}}{\eta_{min}}$  με  $\eta_{min}$  και  $\eta_{max}$  να είναι η μικρότερη και μεγαλύτερη ευριστική πληροφορία για κάθε σύνδεση  $(i, j) \in L$ . Κάτω από αυτή τη συνθήκη, δημιουργήθηκε μία παραπλανητική ευριστική πληροφορία από ένα μεγαλύτερο εύρος του δυνατού ίχνους φερομόνης. Εάν αυτή η συνθήκη αναγνωριστεί, μπορούμε να υπολογίζουμε επίσης ένα φράγμα για το μήκος  $t_0$  της μεταβατικής περιόδου όπως στο Θεώρημα 2, το οποίο τότε γίνεται

$$t_0^n > \left[ (1-\rho) \cdot \tau_{max} \cdot \eta_{max} / (g(s^*) \cdot \eta_{min}) \right] = \left[ r_\eta \cdot (1-\rho) / \rho \right].$$

Σαν μία τελευταία παρατήρηση, σημειώνουμε ότι ο AS, "ένας ιδιαίτερος ξεχωριστός ACO αλγόριθμος επειδή είναι ο πρόγονος όλων των ACO αλγορίθμων [3], [8], [9], ως επίσης και

κάποιες από τις μεταβλητές του (για παράδειγμα εκλεκτικοί AS [3],[9]) και η rank-based έκδοση του AS [2]) δεν ανήκει στον  $ACO_{\tau_{min}}$ . Στην πραγματικότητα, σε αυτούς τους τρεις αλγόριθμους, δεν υπάρχει κάτω φράγμα στη τιμή των ιχνών φερομόνης που μπορεί συνεπώς να γίνει μηδέν. Είναι ενδιαφέρον να σημειώσουμε ότι ο ACS και MMAS δείχνουν να εκπληρώνονται καλύτερα από τον AS και οι μεταβλητές τους σε πολλά πρότυπα προβλήματα αναφοράς όπως το TSP και QAP. Συνεπώς, είμαστε στην ευνοϊκή περίπτωση στην οποία οι αλγόριθμοι ACO για τους οποίους η σύγκλιση μπορεί να αποδειχθεί θεωρητικά επίσης δείχνουν καλύτερη απόδοση πρακτικά.

#### 4.1.6 Συμπέρασμα

Εδώ, αποδείξαμε δύο θεωρήματα που εφαρμόζονται στον αλγόριθμο ACO επονομαζόμενο  $ACO_{gb, \tau_{min}}$ . Το πρώτο θεώρημα ισχυρίζεται ότι η πιθανότητα εύρεσης τουλάχιστον μία φορά μίας βέλτιστης λύσης  $P^*$  μπορεί να γίνει μεγαλύτερη από  $1-\epsilon$  για κάθε μικρή σταθερά  $\epsilon > 0$  εάν ο αλγόριθμος τρεξει για ένα αρκετά μεγάλο αριθμό επαναλήψεων. Δείξαμε, ότι το Θεώρημα 1 εφαρμόζεται σε μία μεγαλύτερη κλάση αλγορίθμων ACO τους επονομαζόμενους  $ACO_{\tau_{min}}$ , οι οποίοι διαφέρουν από τον  $ACO_{gb, \tau_{min}}$  στο ότι μπορούν να χρησιμοποιήσουν κάθε λογικό κανόνα ενημέρωσης ίχνους φερομόνης.

Το δεύτερο θεώρημα, το οποίο εφαρμόζεται στον  $ACO_{gb, \tau_{min}}$  ισχυρίζεται ότι ξεκινώντας από ένα προκαθορισμένο αριθμό από επαναλήψεις μετά την εύρεση της βέλτιστης λύσης, τα ίχνη φερομόνης θα είναι μεγαλύτερα στις συνδέσεις που ανήκουν στη βέλτιστη λύση παρά σε κάθε άλλη λύση. Συνεπώς, ένα μυρμηγκί που σε κάθε κατασκευαστικό βήμα διαλεγει τη σύνδεση με το μεγαλύτερο ίχνος φερομόνης θα κατασκευάσει σιγουρα τη βέλτιστη λύση. Επιπλέον στο Πόρισμα 1, αποδείξαμε ότι για  $t \rightarrow \infty$  κάθε προκαθορισμένο μυρμηγκί θα παράγει τη βέλτιστη λύση κατά τη διάρκεια της  $t$ -οστής επανάληψης με πιθανότητα  $P \geq 1 - \hat{\epsilon}(\tau_{min}, \tau_{max})$  όπου  $\tau_{min}$  και  $\tau_{max}$  είναι οι ελάχιστες και οι μέγιστες τιμές που μπορούν να πάρουν τα ίχνη φερομόνης.

Τελικά, δείξαμε ότι κάποια από αυτά τα αποτελέσματα μπορούν να επεκταθούν σε δύο από τους πιο χρησιμοποιημένους και επιτυχείς αλγορίθμους, στον MMAS και στον ACS

## 4.2 Σύγκλιση Βελτιστοποίησης Αποικίας μυρμηγκιών σε Πρώτης-Τάξης Παραπλανητικά Συστήματα (first-order deceptive system)

Τα παραπλανητικά προβλήματα θεωρήθηκαν δύσκολα για τη βελτιστοποίηση αποικίας μυρμηγκιών (ACO) και πιστεύαμε ότι ο ACO θα αποτύχει να συγκλίνει στο ολικό μέγιστο των παραπλανητικών προβλημάτων (deceptive problems).

Εδώ παρουσιάζεται μία ανάλυση σύγκλισης του ACO σε παραπλανητικά προβλήματα. Αποδεικνύεται ότι ο ACO μπορεί να πετύχει σύγκλιση προσβασιμότητας αλλά όχι ασυμπτωτική σύγκλιση για μία κλάση πρώτης τάξης παραπλανητικά προβλήματα (FODS) χωρίς να υποθέσουμε μία ελάχιστη φερομόνη σε κάθε επανάληψη.

### 4.2.1 Εισαγωγή

Η βελτιστοποίηση αποικίας μυρμηγκιών (ACO) είναι μία δημοφιλής μέθοδος για δύσκολα διακριτά προβλήματα βελτιστοποίησης [6, 5, 2, 9].

Αν και υπάρχει μία τεράστια ποσότητα από πειραματισμούς και παραλλαγές του ACO, η θεωρητική του θεμελίωση είναι ακόμα στην πρώιμη του ανάπτυξη [5]. Πρόσφατα, υπήρξε μία αυξημένη προσπάθεια να εμβαθύνουν στη κατανόηση της συμπεριφοράς σύγκλισης του ACO. Ο Guijahr [7, 8] αποδεικνύει τη σύγκλιση μίας ιδιαίτερης υλοποίησης του ACO επονομαζόμενη graph-based ant system (GBAS). Παρόλα αυτά, ο GBAS είναι λίγο διαφορετικός από τις συνήθεις υλοποιήσεις του ACO και η πρακτική του απόδοση είναι άγνωστη. Ο Dorigo και άλλοι δείχνουν τη σύγκλιση μίας άλλης κλάσης του ACO [10, 6], στην οποία υπάρχει ένα κάτω

φράγμα  $\tau_{\min}$  για όλες τις τιμές της φερομόνης. Τέτοια μέθοδος δηλώθηκε ως  $ACO_{\tau_{\min}}$ . Τυπικά, υπάρχουν δύο τύποι σύγκλισης ενός αλγορίθμου στοχαστικής βελτιστοποίησης:

- **Ασυμπτωτική σύγκλιση. (Asymptotic convergence)** Ένας αλγόριθμος έχει ασυμπτωτική σύγκλιση εάν  $\lim_{t \rightarrow \infty} P_s(t) = 1$ , όπου το  $P_s(t)$  είναι η πιθανότητα ο αλγόριθμος να παράγει μία βέλτιστη λύση στην  $t^{\text{οστή}}$  επανάληψη.
- **Σύγκλιση Προσβασιμότητας. (Reachability convergence)** Ένας αλγόριθμος έχει σύγκλιση προσβασιμότητας εάν  $\lim_{t \rightarrow \infty} P_r(t) = 1$ , όπου το  $P_r(t)$  είναι η πιθανότητα ο αλγόριθμος να παράγει μία βέλτιστη λύση τουλάχιστον μία φορά από τη  $1^{\text{η}}$  μέχρι την  $t^{\text{οστή}}$  επανάληψη.

Ο Dorigo και άλλοι [10, 6] δείχνουν ότι ο  $ACO_{\tau_{\min}}$  επιτυγχάνει σύγκλιση προσβασιμότητας κάτω από κάποιες υποθέσεις. Ο Dorigo και άλλοι [10, 6] δείχνουν επίσης την ασυμπτωτική σύγκλιση του  $ACO_{\tau_{\min}(t)}$ , στην οποία το κάτω φράγμα της φερομόνης  $\tau_{\min}(t)$  αλλάζει κατά τη διάρκεια του χρόνου, κάτω από την υπόθεση  $\tau_{\min}(t) = d / \ln(t + 1)$ , όπου  $d$  είναι μία σταθερά.

Ο κύριος περιορισμός της μελέτης του  $ACO_{\tau_{\min}}$  και  $ACO_{\tau_{\min}(t)}$  είναι ότι δεν επιτρέπουν για μια εκθετικά γρήγορη μείωση των ιχνών φερομόνης αποτέλεσμα της χρήσης ενός σταθερού παράγοντα εξάτμισης, που χρησιμοποιήθηκε από τις περισσότερες ACO υλοποιήσεις. Εδώ θεωρούμε έναν ACO αλγόριθμο που χρησιμοποιεί την εκθετική μείωση της φερομόνης και αποδεικνύει τη σύγκλιση προσβασιμότητας.

Τα αποτελέσματα επαληθεύτηκαν για μία συγκεκριμένη κλάση από προβλήματα βελτιστοποίησης, επονομαζόμενα  $n$ -bit πρόβλημα παγίδας. Το  $n$ -bit πρόβλημα παγίδας θεωρήθηκε ως ένα δύσκολο πρόβλημα επειδή είναι ένα **πρώτης-τάξης παραπλανητικό σύστημα (FODS)**. Το FODS χαρακτηρίστηκε από τοπικά βέλτιστα προκαθορισμένα σημεία με μεγάλες περιοχές έλξεων. Είναι γνωστό ότι και οι δύο γενετικοί αλγόριθμοι ο (GA) και ACO ενδέχεται να πέσουν μέσα σε παγίδες με τοπικά μέγιστα και να αποτύχουν να βρουν τη καθολική βέλτιστη λύση [1, 3, 4]. Παρουσιάζουμε έναν ACO αλγόριθμο που ακολουθεί από κοντά τις πιο κοινές ACO υλοποιήσεις και αποδεικνύει ότι ο αλγόριθμος μας επιτυγχάνει σύγκλιση προσβασιμότητας για ένα  $n$ -bit πρόβλημα παγίδας.

Επίσης δείχνουμε ότι ο αλγόριθμος δεν επιτυγχάνει ασυμπτωτική σύγκλιση. Αποδεικνύουμε ότι, ο κλασικός ACO αλγόριθμος με εκθετική μείωση της φερομόνης, δεν μπορεί να συγκλίνει σε βέλτιστες λύσεις των  $n$ -bit προβλημάτων παγίδας. Αυτό το αποτέλεσμα είναι σημαντικό καθώς παρέχει μία πρώτη εξήγηση της έλλειψης ασυμπτωτικής σύγκλισης του ACO σε παραπλανητικά προβλήματα.

#### Algorithm 1: Ant colony optimization(ACO)

```

initialize pheromone values  $\tau_i^1$  for each  $i = 1, \dots, n$ 
and  $j = 1, \dots, |D_i|$ ;
foreach iteration  $t = 1, 2, \dots$  do
  foreach ant  $k = 1, 2, \dots, m$  do
     $s \leftarrow \text{ConstructSolution}(\tau)$ ;
    if  $s$  is feasible solution then
      if  $f(s) < f(s_b)$  or  $s_b = \text{null}$  then
         $s_b \leftarrow s$ ;
         $\Psi_t \leftarrow \Psi_t \cup \{s\}$ ;
  Update Pheromone( $\tau, \Psi_t, s_b$ );

```

### 4.2.2 Βελτιστοποίηση Αποικίας μυρμηγκιών και Παραπλανητικά Προβλήματα

Σε αυτό το τμήμα, εισάγουμε το πλαίσιο εργασίας του ACO αλγορίθμου που θα χρησιμοποιήσουμε και θα αναθεωρήσουμε την αρχή των παραπλανητικών προβλημάτων. Ο ACO σχεδιάστηκε για τη λύση περιορισμένων προβλημάτων βελτιστοποίησης (CO) ορισμένα ως εξής [5].

**Ορισμός 1** Ένα πρόβλημα περιορισμένης βελτιστοποίησης ορίστηκε από ένα μοντέλο  $P=(S,\Omega,f)$ , όπου: 1)  $S$  είναι ένας χώρος αναζήτησης ορισμένος πάνω σ ένα πεπερασμένο σύνολο από διακριτων μεταβλητών · 2)  $\Omega$  είναι ένα σύνολο από περιορισμούς στις μεταβλητές · και 3)  $f:S \rightarrow \mathcal{R}$  είναι η αντικειμενική συνάρτηση που πρέπει να μεγιστοποιηθεί.

Στο χώρο αναζήτησης  $S$ , υπάρχουν  $n$  μεταβλητές αποφάσεων  $X_i$ , για  $i=1,\dots,n$ , όπου το  $X_i$  μπορεί να πάρει τιμές από ένα σύνολο από  $D_i = \{c_i^1, \dots, c_i^{|D_i|}\}$ . Μία εκχώρηση μεταβλητής γράφεται ως  $X_i = c_i^j$ . Μία πλήρης εκχώρηση σε όλα τα  $X_i$  δίνει μία άμεση λύση. Το  $S$  είναι το σύνολο όλων των πλήρων εκχωρήσεων. Δηλώνουμε επίσης το σύνολο όλων των συνιστώσων της λύσης ως  $R = \{c_i^j \mid i = 1, \dots, n, j = 1, \dots, |D_i|\}$ .

Μία λύση  $s \in S$  ονομάζεται εφικτή (**feasible solution**) εάν ικανοποιεί όλους τους περιορισμούς στο  $\Omega$ . Μία εφικτή λύση  $s^*$  είναι ένα καθολικό βέλτιστο (**global optimum**) εάν  $f(s^*) \geq f(s)$  για κάθε  $s \in S$ . Το σύνολο όλων των καθολικών βέλτιστων δηλώνεται ως  $S^* \subseteq S$ . Ο γενικός ACO αλγόριθμος φαίνεται στον **Αλγόριθμο 1**. Στον ACO αλγόριθμο, αναθέτουμε μία τιμή  $\tau_i^j$ , επονομαζόμενη φερομόνη, σε κάθε μία από τις συνιστώσες της λύσης  $c_i^j$ . Το διάνυσμα όλων των φερομονών δηλώνεται με  $\tau$ .

Η κεντρική συνιστώσα του ACO είναι η διαδικασία  $\text{ConstructSolution}(\tau)$  που κάθε μυρμήγκι χρησιμοποιεί για να κατασκευάσει μία λύση. Αυτή η διαδικασία κατασκευής δημιουργεί λύσεις ως ακολουθίες στοιχείων από ένα πεπερασμένο σύνολο από συνιστώσες της λύσης  $S$ . Έστω  $s^p$  μια μερική λύση κατασκευασμένη από ένα μυρμήγκι. Αρχικά, το  $s^p = \{\}$  είναι κενό. Το μυρμήγκι σε κάθε βήμα της κατασκευής προσθέτει μία συνιστώσα της εφικτής λύσης από το σύνολο  $R(s^p)$ , όπου  $R(s^p) \subseteq R \setminus \{s^p\}$  είναι το σύνολο των εφικτών λύσεων που ικανοποιούν τους περιορισμούς στο  $\Omega$  δεδομένης της μερικής λύσης  $s^p$ . Όταν επιλέγουμε τη συνιστώσα για τη μεταβλητή  $X_i$ , η πιθανότητα το μυρμήγκι να επιλέξει το  $c_i^j$  είναι,  $\forall j = 1, \dots, |D_i|$ ,

$$P(c_i^j, s^p) = \frac{[\tau_i^j]^\alpha [\eta(c_i^j)]^\beta}{\sum_{c_i^k \in R(s^p)} [\tau_i^k]^\alpha [\eta(c_i^k)]^\beta} \quad (1)$$

Εδώ, το  $\eta(c_i^j)$  είναι μία ευριστική συνάρτηση για τη συνιστώσα  $c_i^j$ . Τα  $\alpha > 0$  και  $\beta > 0$  είναι δύο παράμετροι που ελέγχουν τη σχετική σημαντικότητα της φερομόνης και της ευριστικής πληροφορίας.

Στην επανάληψη  $t$ , και αφού όλα τα μυρμήγκια κατασκεύασαν λύσεις, ο ACO καλεί την  $\text{UpdatePheromone}(\tau, \Psi_t, s_b)$  για να ενημερώσουμε το διάνυσμα φερομόνης  $\tau$ . Εδώ, το  $\Psi_t$  είναι το σύνολο των λύσεων που κατασκευάζονται από τα μυρμήγκια στην επανάληψη  $t$  και  $s_b$  είναι η υπερκείμενη (incumbent) καλύτερη λύση. Ο κανόνας αναβάθμισης φερομόνης είναι,  $\forall i = 1, \dots, n, j = 1, \dots, |D_i|$ ,

$$\tau_i^j = \rho \tau_i^j + \frac{1}{|S_i^j|} \sum_{s \in S_i^j} F(s), \quad (2)$$

όπου  $\rho \in (0,1)$  είναι ένας παράγοντας εξάτμισης και  $S_i^j$  είναι το σύνολο των λύσεων στο  $\Psi_t$  που έχουν συνιστώσα το  $c_i^j$ , και  $F : S \rightarrow \mathcal{R}^+$  είναι μία συνάρτηση ποιότητας τέτοια ώστε, για κάθε  $s, s' \in S$ , εάν  $f(s) > f(s')$ , τότε  $F(s) \geq F(s')$ .

Η αναμενόμενη ποιότητα επανάληψης δηλώνεται ως  $W_F(\tau|t)$  όπου  $t$  είναι ο αριθμός της επανάληψης. Το  $W_F(\tau|t)$  δηλώνεται ως :

$$W_F(\tau|t) = \sum_{s \in S} F(s)P(s|\tau), \quad (3)$$

όπου  $P(s|\tau)$  είναι η πιθανότητα η λύση  $s$  να παραχθεί από ένα μυρμήγκι δοθέντος τους διανύσματος φερομόνης  $\tau$ .

**Ορισμός 2** Δοθέντος ενός περιορισμένου προβλήματος βελτιστοποίησης  $P$ , ένας ACO αλγόριθμος είναι ένας τοπικός βελτιστοποιητής για το  $P$  εάν για κάθε αρχική τιμή φερομόνης, η αναμενόμενη ποιότητα επανάληψης ικανοποιεί :

$$W_F(\tau|t+1) \geq W_F(\tau|t), \quad \forall t \geq 0 \quad (4)$$

Μπορούμε τώρα να εισάγουμε την έννοια της παραπλάνησης για τοπικούς βελτιστοποιητές [3].

**Ορισμός 3** Δοθέντος ενός περιορισμένου προβλήματος βελτιστοποίησης  $P$ , αυτο ονομάζεται πρώτης-τάξης παραπλανητικό πρόβλημα (FODS) για ένα ACO αλγόριθμο, εάν ο ACO αλγόριθμος είναι ένας τοπικός βελτιστοποιητής, και υπάρχει μία αρχική ρύθμιση των τιμών φερομόνης έτσι ώστε ο αλγόριθμος να μην αναμενεται να συγκλίνει σε ένα καθολικό βέλτιστο.

### 4.2.3 Ανάλυση Σύγκλισης του ACO στο FODS

Επί του παρόντος, το εάν ένα πρόβλημα είναι FODS αποδεικνύεται εμπειρικά και στερείται θεωρητικής ανάλυσης. Για παράδειγμα, πειράματα εκτελέστηκαν για να αποδειχτεί ότι τα προβλήματα  $n$ -bit παγίδας και χρονοδρομολόγησης είναι παραπλανητικά συστήματα δείχνοντας ότι ο ACO δεν συγκλίνει σε καθολικά βέλτιστες λύσεις [3]. Παρόλα αυτά, δεν υπάρχει απόδειξη γιατί ο ACO δεν μπορεί να επιτύχει ασυμπτωτική σύγκλιση για τέτοια προβλήματα. Επίσης, θα πρέπει ακόμα να καθοριστεί εάν ο ACO μπορεί να επιτύχει σύγκλιση προσβασιμότητας σε παραπλανητικά συστήματα. Επιπλέον, η κατανόηση της πολυπλοκότητας του χρόνου για τη λύση παραπλανητικών προβλημάτων είναι σημαντική, αφού τα παραπλανητικά προβλήματα είναι τα δυσκολότερα προβλήματα για τον ACO. Εδώ μελετάμε το πρόβλημα  $n$ -bit παγίδας, ένα γνωστό παράδειγμα του FODS που επίσης μελετήθηκε για εξελικτικούς αλγορίθμους.

$s$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
bit 4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
bit 3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
bit 2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
bit 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$f(s)$	5	1	1	2	1	2	2	3	1	2	2	3	2	3	3	4

Πίνακας 1. Τιμή της συνάρτησης προσαρμογής του 4-bit προβλήματος παγίδας

### Το πρόβλημα $n$ -bit παγίδας

Το πρόβλημα  $n$ -bit παγίδας είναι να βρούμε μεταξύ των  $2^n$  δυαδικών αριθμών, από το 0 έως το  $2^n - 1$ , τον ένα με τη μεγαλύτερη προσαρμογή. Η προσαρμογή του δυαδικού αριθμού  $s$  ορίζεται ως :

$$f(s) = \begin{cases} h(s) & s \neq 0 \\ n+1 & s = 0 \end{cases} \quad (5)$$

όπου το  $h(s)$  είναι η απόσταση του Hamming μεταξύ του  $s$  και του 0. Προφανώς, το καθολικό μέγιστο είναι  $s^* = 0$ .

Για να λύσουμε το πρόβλημα  $n$ -bit παγίδας θα σχεδιάσουμε τον ακόλουθο ACO αλγόριθμο που ταιριάζει μέσα στο γενικότερο πλαίσιο ACO στον Αλγόριθμο 1. Χρησιμοποιούμε  $m$  μυρμήγκια. Σε κάθε επανάληψη ένα μυρμήγκι διαδοχικά διορθώνει το  $j$ -οστό bit του δυαδικού αριθμού για  $j=1,2,\dots,n$ . Για το  $j$ -οστό bit, το μυρμήγκι έχει δύο επιλογές  $c_0^j$  και  $c_1^j$ , αντίστοιχα με τη ρύθμιση του  $j$ -οστού bit σε 0 και 1. Η φερομόνη του  $c_0^j$  και του  $c_1^j$  είναι  $\tau_0^j$  και  $\tau_1^j$  αντίστοιχα.

Ορίζουμε το σύνολο  $G_j^k = \{(b_1 b_2 \dots b_n) \in S \mid b_j = k\}$ , όπου  $k \in \{0,1\}$  και το  $b_j$  είναι η τιμή του  $j$ -οστού bit. Γι'αυτό το λόγο, το  $G_j^k$  είναι ένα σύνολο από δυαδικούς αριθμούς των οποίων το  $j$ -οστό bit είναι το  $k$ . Ορίζουμε τη συνάρτηση  $F(G_j^k)$  ως τη μέση προσαρμογή όλων των αριθμών στο  $G_j^k$  :

$$F(G_j^k) = \frac{1}{|G_j^k|} \sum_{s \in G_j^k} f(s) \quad (6)$$

Για παράδειγμα, όταν  $n=4$ , έχουμε ότι  $F(G_j^0) = 2.125$  και το  $F(G_j^1) = 2.5$  για όλα τα  $j=1,2,3,4$ .

Αρχικοποιούμε τη φερομόνη  $\tau_j^k$  ως  $\tau_j^k(0) = F(G_j^k)$ , για όλα τα  $j=1, \dots, n$  και  $k \in \{0,1\}$ . Για κάθε  $j$ -οστό bit, η πιθανότητα ένα μυρμήγκι να επιλέξει τη τιμή  $k$ , για  $k=0$  ή  $1$ , είναι :

$$P(c_j^k, t) = \frac{\tau_j^k(t)}{\tau_j^0(t) + \tau_j^1(t)}, \quad (7)$$

η οποία παράγεται από την (1) θέτοντας  $\alpha=1$  και  $\beta=0$ . Σε κάθε επανάληψη, η φερομόνη ενημερώνεται ως :

$$\tau_j^k(t+1) = \rho \tau_j^k(t) + \sum_{s \in S_j^k} f(s), \quad (8)$$

όπου το  $S_j^k$  είναι το σύνολο των λύσεων που παράγονται από την  $t$ -οστή επανάληψη και έχουν το  $c_j^k$  όπως στο  $j$ -οστό bit.

### Ασυμπτωτική Σύγκλιση του ACO σε ένα πρόβλημα $n$ -bit παγίδας

**Λήμμα 1** Για κάθε πρόβλημα  $n$ -bit και  $F(G_j^k)$  συνάρτηση ορισμένη στο (6) έχουμε ότι :

$$F(G_j^1) = \frac{n+1}{2} \quad \text{και} \quad F(G_j^0) = \frac{n-1}{2} + \frac{n+1}{2^{n-1}}. \quad (9)$$

**Απόδειξη.** Έχουμε από τον ορισμό στο (6) ότι :

$$F(G_j^1) = \frac{1}{2^{n-1}} \sum_{k=1}^n C_{n-1}^{k-1} k = \frac{n+1}{2} \quad (10)$$

$$F(G_j^0) = \frac{1}{2^{n-1}} \left( (n+1) + \sum_{k=1}^{n-1} C_{n-1}^k k \right) = \frac{n-1}{2} + \frac{n+1}{2^{n-1}} \quad (11) \quad \blacksquare$$

**Λήμμα 2** Έστω  $\beta > 0$  να είναι μία σταθερά, έχουμε ότι

$$\lim_{T \rightarrow \infty} \prod_{t=[\beta]+1}^T \left( 1 - \frac{\beta}{t} \right) = 0 \quad (12)$$

**Απόδειξη.** Θεωρούμε τέσσερις περιπτώσεις.

1) Στη περίπτωση που  $\beta=1$ , έχουμε

$$\lim_{T \rightarrow \infty} \prod_{t=[\beta]+1}^T \left( 1 - \frac{\beta}{t} \right) = \lim_{T \rightarrow \infty} \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{3}{4} \cdots \frac{T-1}{T} = \lim_{T \rightarrow \infty} \frac{1}{T} = 0. \quad (13)$$

2) Στη περίπτωση  $\beta > 1$ , έχουμε  $\prod_{t=[\beta]+1}^T \left( 1 - \frac{\beta}{t} \right) < \prod_{t=[\beta]+1}^T \left( 1 - \frac{1}{t} \right)$ .

Έστω  $\prod_{t=2}^{[\beta]} \left( 1 - \frac{1}{t} \right) = r$ , έχουμε ότι:

$$r \prod_{t=[\beta]+1}^T \left( 1 - \frac{\beta}{t} \right) = \prod_{t=2}^T \left( 1 - \frac{1}{t} \right). \quad (14)$$

Ξανά, επειδή  $\lim_{T \rightarrow \infty} \prod_{t=2}^T \left( 1 - \frac{1}{t} \right) = 0$ , έχουμε  $\lim_{T \rightarrow \infty} r \prod_{t=[\beta]+1}^T \left( 1 - \frac{\beta}{t} \right) = 0$  που σημαίνει ότι  $\lim_{T \rightarrow \infty} \prod_{t=[\beta]+1}^T \left( 1 - \frac{\beta}{t} \right) = 0$

3) Στη περίπτωση  $\beta < 1$  και  $\beta = \frac{1}{q}$  όπου το  $q$  είναι ένας ακέραιος μεγαλύτερος από 1.

Έστω  $T = p \cdot q$  και  $c = \prod_{t=2}^{q-1} \left( 1 - \frac{1}{t} \right)$ , τότε έχουμε ότι:

$$\prod_{t=2}^T \left( 1 - \frac{1}{t} \right) = c \prod_{t=q}^T \left( 1 - \frac{1}{t} \right) = c \prod_{j=1}^{p-1} \prod_{i=0}^{q-1} \left( 1 - \frac{1}{jq+i} \right) > c \prod_{j=1}^{p-1} \left( 1 - \frac{1}{jq} \right)^q$$



$$\text{Συνεπώς, } \lim_{T \rightarrow \infty} \prod_{t=2}^T \left(1 - \frac{1}{t}\right) > c \lim_{p \rightarrow \infty} \prod_{j=1}^p \left(1 - \frac{1}{jq}\right)^q = c \left[ \lim_{p \rightarrow \infty} \prod_{j=1}^p \left(1 - \frac{1}{jq}\right)^q \right] = 0 \quad (15)$$

Αφού γνωρίζουμε το  $\lim_{T \rightarrow \infty} \prod_{t=2}^T \left(1 - \frac{1}{t}\right) = 0$  και  $c \left[ \lim_{p \rightarrow \infty} \prod_{j=1}^p \left(1 - \frac{1}{jq}\right)^q \right] \geq 0$ , απο τη (15) παίρνουμε  $c \left[ \lim_{p \rightarrow \infty} \prod_{j=1}^p \left(1 - \frac{1}{jq}\right)^q \right] = 0$ , η οποία συνεπάγεται  $\lim_{p \rightarrow \infty} \prod_{j=1}^p \left(1 - \frac{1}{jq}\right) = 0$ , το οποίο είναι ισοδύναμο με :

$$\lim_{T \rightarrow \infty} \prod_{t=1}^T \left(1 - \frac{1}{qt}\right) = \lim_{T \rightarrow \infty} \prod_{t=1}^T \left(1 - \frac{\beta}{t}\right) = 0.$$

4) Στη περίπτωση που  $\beta < 1$ , και  $\beta = \frac{p}{r}$  όπου  $p$  και  $r$  είναι και οι δύο ακέραιοι μεγαλύτεροι από 1.

Μπορούμε πάντα να βρούμε  $\beta' = \frac{1}{q}$  όπου  $q$  είναι ένας ακέραιος μεγαλύτερος από ένα έτσι ώστε

$$\beta' < \beta.$$

Αφού  $\prod_{t=1}^T \left(1 - \frac{\beta'}{t}\right) > \prod_{t=1}^T \left(1 - \frac{\beta}{t}\right)$  και  $\lim_{T \rightarrow \infty} \prod_{t=1}^T \left(1 - \frac{\beta'}{t}\right) = 0$  από τη περίπτωση 3), έχουμε ότι

$$\lim_{T \rightarrow \infty} \prod_{t=1}^T \left(1 - \frac{\beta}{t}\right) = 0. \quad \blacksquare$$

Θεωρούμε τον αλγόριθμο ACO που περιγράφηκε στον **Αλγόριθμο1** χρησιμοποιώντας την (7) και την (8) ως συναρτήσεις επιλογής και ενημέρωσης, αντίστοιχα. Ονομάζουμε αυτόν τον αλγόριθμο  $ACO_{n\text{-bit}}$ .

Για τον  $ACO_{n\text{-bit}}$  αλγόριθμο, ορίζουμε :

$$\alpha_j(t) = \frac{\tau_j^0(t)}{\tau_j^1(t)}, \quad \forall j = 1, 2, \dots, n, \quad t = 0, 1, 2, \dots \quad (16)$$

Δηλαδή, το  $\alpha_j(t)$  είναι ο λόγος της φερομόνης στη τιμή 0 προς τη φερομόνη στη τιμή 1 για το  $j$  bit στην επανάληψη  $t$ .

**Θεώρημα 1** Ο αλγόριθμος  $ACO_{n\text{-bit}}$  δεν μπορεί να επιτύχει ασυμπτωτική σύγκλιση σε ένα πρόβλημα  $n$ -bit παγίδας όταν  $n > 3$ .

**Απόδειξη.**

Για το πρόβλημα  $n$ -bit παγίδας, αφού η καθολικά βέλτιστη λύση είναι  $s^* = \{0, 0, \dots, 0\}$ , δείχνουμε ότι  $\forall j = 1, 2, \dots, n$ ,

$$\lim_{t \rightarrow \infty} E[a_j(t)] = \lim_{t \rightarrow \infty} E\left[\frac{\tau_j^0(t)}{\tau_j^1(t)}\right] = 0. \quad (17)$$

Υποθετούμε ότι έχουμε  $n$  μυρμηγκία. Πρώτα αποδεικνύουμε με επαγωγή ότι

$$E[a_j(t)] < 1, \quad \forall j = 1, 2, \dots, n, \quad t = 0, 1, 2, \dots \quad (18)$$

Όταν το  $t=0$ , το  $\tau_j^0(0)$  αρχικοποιείται στο  $F(G_j^0)$  και το  $\tau_j^1(0)$  αρχικοποιείται στο  $F(G_j^1)$ . Από το

Λήμμα 1, ενώ  $n > 3$ ,  $F(G_j^1) > F(G_j^0)$ . Για ένα προκαθορισμένο  $n$ , θέτουμε  $F(G_j^0) = \alpha_0 F(G_j^1)$ , όπου  $\alpha_0$  είναι μία σταθερά και  $0 < \alpha_0 < 1$ . Συνεπώς, έχουμε τη βασική περίπτωση :

$$E[\alpha_j(0)] = \frac{\tau_j^0(0)}{\tau_j^1(0)} = \frac{F(G_j^0)}{F(G_j^1)} = \alpha_0 < 1.$$

Για να αποδείξουμε τη (18) με επαγωγή, υποθέτουμε ότι  $E[\alpha_j(t)] < 1$ .

Σύμφωνα με το (8), έχουμε ότι στην  $(t+1)^{\text{οστή}}$  επανάληψη, η αναμενόμενη τιμή της φερομόνης ικανοποιεί

$$E[\tau_j^0(t+1)] \leq \rho E[\tau_j^0(t)] + E\left[\frac{m\alpha_j(t)}{1+\alpha_j(t)} F(G_j^0)\right],$$

$$E[\tau_j^1(t+1)] \geq \rho E[\tau_j^1(t)] + E\left[\frac{m}{1+\alpha_j(t)} F(G_j^1)\right].$$

Απο τη (16), έχουμε ότι

$$\begin{aligned} E[\alpha_j(t+1)] &= \frac{E[\tau_j^0(t+1)]}{E[\tau_j^1(t+1)]} \leq \frac{\rho E[\tau_j^0(t)] + \frac{mE[\alpha_j(t)]}{1+E[\alpha_j(t)]} F(G_j^0)}{\rho E[\tau_j^1(t)] + \frac{m}{1+E[\alpha_j(t)]} F(G_j^1)} \\ &= \frac{\rho(1+E[\alpha_j(t)])E[\tau_j^0(t)] + E[\alpha_j(t)]mF(G_j^0)}{\rho(1+E[\alpha_j(t)])E[\tau_j^1(t)] + mF(G_j^1)} \end{aligned}$$

Συνεπώς, έχουμε ότι :

$$E[\alpha_j(t+1)] = E[\alpha_j(t)] \left( 1 - \frac{(1-\alpha_0)mF(G_j^1)}{\rho(1+E[\alpha_j(t)])E[\tau_j^1(t)] + mF(G_j^1)} \right) < E[\alpha_j(t)]$$

Επομένως, παίρνουμε, για κάθε  $j=1, \dots, n$ ,

$$1 > \alpha_0 = E[\alpha_j(0)] > E[\alpha_j(1)] > \dots > E[\alpha_j(t)] > E[\alpha_j(t+1)] > \dots > 0. \quad (19)$$

Συνεπώς, αποδείξαμε τη (18) με επαγωγή.

Επιπλέον, αφού

$$E[\tau_j^1(t)] = \rho E[\tau_j^1(t-1)] + \frac{1}{1+E[\alpha_j(t)]} mF(G_j^1) < E[\tau_j^1(t-1)] + mF(G_j^1), \quad (20)$$

εφαρμόζοντας την (20) οδηγούμαστε στο

$$E[\tau_j^1(t)] < E[\tau_j^1(0)] + tmF(G_j^1) < (t+1)mF(G_j^1). \quad (21)$$

Αντικαθιστώντας την (21) στην (19) δίνει

$$E[\alpha_j(t+1)] < E[\alpha_j(t)] \left( 1 - \frac{(1-\alpha_0)}{\rho(1+E[\alpha_j(t)])(t+1)+1} \right)$$

Από τη (19), γνωρίζουμε ότι  $1+E[\alpha_j(t)] < 2$ .

$$\text{Συνεπώς, } E[\alpha_j(t+1)] < E[\alpha_j(t)] \left( 1 - \frac{(1-\alpha_0)}{2\rho(t+1)+1} \right) < E[\alpha_j(t)] \left( 1 - \frac{(1-\alpha_0)}{(2\rho+3)^t} \right)$$

Έστω  $\beta = \frac{1-\alpha_0}{2\rho+3}$ , έχουμε

$$E[\alpha_j(t+1)] < E[\alpha_j(t)] \left( 1 - \frac{\beta}{t} \right) < \prod_{i=1}^t \left( 1 - \frac{\beta}{i} \right) \alpha_0$$

Από το Λήμμα 2, παίρνουμε ότι :  $\lim_{t \rightarrow \infty} E[\alpha_j(t)] = 0$ . Συνεπώς, ο αλγόριθμος  $ACO_{n\text{-bit}}$  δεν συγκλίνει στη βέλτιστη λύση  $s^* = \{0, 0, \dots, 0\}$ . ■

Το Θεώρημα 1 δίνει μία επεξήγηση γιατί οι αλγόριθμοι ACO έχουν δυσκολίες στην επίλυση παραπλανητικών προβλημάτων τέτοια όπως το πρόβλημα n-bit παγίδας.

### Σύγκλιση προσβασιμότητας στον ACO σε ένα πρόβλημα n-bit παγίδας

Αν και ο  $ACO_{n\text{-bit}}$  δεν μπορεί να επιτύχει ασυμπτωτική σύγκλιση, θα δείξουμε ότι μπορεί να πετύχει σύγκλιση προσβασιμότητας. Χρειαζόμαστε πρώτα κάποια προκαταρκτικά αποτελέσματα.

**Λήμμα 3** Η φερομένη στην t-οστή επανάληψη του  $ACO_{n\text{-bit}}$  ικανοποιεί τη  $\tau_j^0(t) \geq \frac{1-\rho^{\tau+1}}{1-\rho} \phi F(G_j^0)$ ,

όπου  $\phi$  είναι μία σταθερά,  $0 < \phi < 1$ , για  $j=1, 2, \dots, n$  και  $t=1, 2, \dots$

**Απόδειξη.** Από το κανόνα ενημέρωσης φερομένης στο (8), έχουμε

$$\tau_j^k(t) = \rho \tau_j^k(t-1) + \sum_{s \in S_j^k} f(s), \quad (22)$$

Έστω  $\theta_j^0 = \min \{f(s) \mid s = (b_1 b_2 \dots b_n), s \in S, b_j = 0\}$  και  $\theta_j^0 = \varphi F(G_j^0)$ .

Προφανώς,  $0 < \rho < 1$ . Αφού  $\tau_j^0(t) \geq \rho \tau_j^0(t-1) + \theta_j^0$ , έχουμε ότι  $\tau_j^0(t) \geq \rho \tau_j^0(t-1) + \varphi F(G_j^0)$ .

Επομένως, παίρνουμε

$$\begin{aligned} \tau_j^0(t) &\geq \rho^t \tau_j^0(0) + (\rho^{t-1} + \rho^{t-2} + \dots + 1) \varphi F(G_j^0) = (\rho^t + \rho^{t-1} + \rho^{t-2} + \dots + 1) \varphi F(G_j^0) \\ &= \frac{1 - \rho^{t+1}}{1 - \rho} \varphi F(G_j^0). \end{aligned}$$

**Λήμμα 4** Για κάθε θετικό ακέραιο  $n$  και ένα πραγματικό αριθμό  $\rho \in (0, 1)$  υπάρχει ένας θετικός

ακέραιος  $t_0$  έτσι ώστε  $\frac{1 - \rho^{t+1}}{1 - \rho} > t^{-\frac{1}{n}}$  για κάθε  $t > t_0$ .

**Απόδειξη.** Βλέπουμε το γεγονός ότι

$$\lim_{t \rightarrow \infty} \frac{t^{-\frac{1}{n}}}{\frac{1 - \rho^{t+1}}{1 - \rho}} = \lim_{t \rightarrow \infty} \frac{1 - \rho}{(1 - \rho^{t+1})^{\frac{1}{n}} t} = 0.$$

Συνεπώς, υπάρχει ένας ακέραιος  $t_0 > 0$  έτσι ώστε  $\frac{1 - \rho^{t+1}}{1 - \rho} > t^{-\frac{1}{n}}$  για κάθε  $t > t_0$ . ■

Από το Λήμμα 3 και το Λήμμα 4, βλέπουμε ότι όταν το  $t$  είναι αρκετά μεγάλο, το  $\tau_j^0 \geq t^{-\frac{1}{n}} \varphi F(G_j^0)$ .

Θέτουμε  $\tau_{\min}^0(j, t) = t^{-\frac{1}{n}} \varphi F(G_j^0)$  και  $\tau_{\min}^0(t) = \min_{1 \leq j \leq n} \tau_{\min}^0(j, t)$ . Εάν θέσουμε  $\theta_{\min} = \min_{1 \leq j \leq n} \varphi F(G_j^0)$ , τότε από το Λήμμα 3 και το Λήμμα 4,

$$\tau_{\min}^0(t) \geq t^{-\frac{1}{n}} \theta_{\min}, \quad \forall t > t_0. \quad (23)$$

**Λήμμα 5** Η φερομένη στην  $t$ -οστή επανάληψη του  $ACO_{n\text{-bit}}$  ικανοποιεί τη  $\tau_j^k(t) \leq \frac{1}{1 - \rho} \varphi F(G_j^k)$  για  $k=0, 1, j=1, 2, \dots$

**Απόδειξη.** Από το κανόνα ενημέρωσης φερομένης στο (8), έχουμε

$$\tau_j^k(t) = \rho \tau_j^k(t-1) + \sum_{s \in S_j^k} f(s), \quad (24)$$

Έστω  $\lambda_j^k = \max \{f(s) \mid s = (b_1 b_2 \dots b_n), s \in S, b_j = k\}$  και  $\lambda_j^k = \varphi F(G_j^k)$  όπου  $\varphi > 1$  είναι μία σταθερά για ένα δοσμένο  $n$ . Αφού το  $\tau_j^k(t) \leq \rho \tau_j^k(t-1) + m \lambda_j^k$ , έχουμε ότι  $\tau_j^k(t) \leq \rho \tau_j^k(t-1) + \varphi m F(G_j^k)$ .

Επομένως, παίρνουμε  $\tau_j^k(t) \leq \rho^t \tau_j^k(0) + (\rho^{t-1} + \rho^{t-2} + \dots + 1) \varphi F(G_j^k) \leq \frac{1}{1 - \rho} \varphi m F(G_j^k)$ . ■

Δηλώνουμε  $\tau_{\max}^k(j) = \frac{1}{1 - \rho} \varphi m F(G_j^k)$  και  $\tau_{\max}^k = \max_{1 \leq j \leq n, k=0,1} \tau_{\max}^k(j)$ .

**Θεώρημα 2** Ο αλγόριθμος  $ACO_{n\text{-bit}}$  επιτυγχάνει σύγκλιση προσβασιμότητας στο πρόβλημα  $n$ -bit παγίδας.

**Απόδειξη.** Αν και υπάρχουν πολλαπλά μυρμήγκια, χρειάζεται μόνο να αποδείξουμε ότι ένα μυρμήγκι μπορεί να επιτύχει σύγκλιση προσβασιμότητας για να επαληθεύσει το αποτέλεσμα. Από το Λήμμα 4 και το Λήμμα 5, γνωρίζουμε ότι για κάθε  $t > t_0$ , όπου  $t_0$  ορίστηκε στο Λήμμα 4,

$$P(c_j^0, t) = \frac{\tau_j^0(t)}{\tau_j^0(t) + \tau_j^1(t)} \geq \frac{\tau_{\min}^0(t)}{2\tau_{\max}^k}. \quad (25)$$

Συνεπώς, έστω  $P(t)$  να είναι η πιθανότητα το μυρμήγκι να μπορεί να παράγει τη βέλτιστη λύση  $s^* = (0, 0, \dots, 0)$  στην επανάληψη  $t$ , έχουμε ότι

$$P(t) \geq \left[ \frac{\tau_{\min}^0(t)}{2\tau_{\max}} \right]^n. \quad (26)$$

Έστω  $P_{\text{succ}}(T)$  και  $P_{\text{fail}}(T)$ , αντίστοιχα, να είναι η πιθανότητα το μυρμήγκι να βρει και να μην βρει το  $s^*$  στις πρώτες  $T$  επαναλήψεις. Από το (23) και το Λήμμα 5, γνωρίζουμε ότι

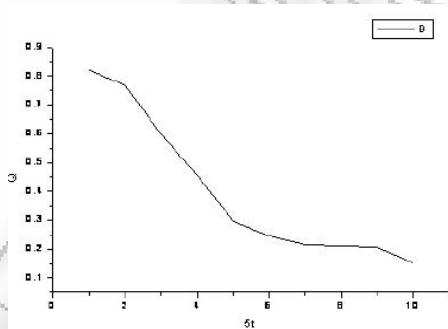
$$P_{\text{fail}}(T) = \prod_{t=1}^T (1 - P(t)) \leq \prod_{t=t_0}^T \left[ 1 - \left( \frac{\tau_{\min}^0(t)}{2\tau_{\max}} \right)^n \right] \leq \prod_{t=t_0}^T \left[ 1 - \left( \frac{\theta_{\min}}{2\tau_{\max}} \right)^n \left( \frac{1}{t} \right)^n \right].$$

Δηλώνω  $\left( \frac{\theta_{\min}}{2\tau_{\max}} \right)^n = \gamma$ , τότε  $\lim_{T \rightarrow \infty} P_{\text{fail}}(T) \leq \lim_{T \rightarrow \infty} \prod_{t=t_0}^T \left[ 1 - \frac{\gamma}{t} \right]$ . Από το Λήμμα 2, έχουμε ότι

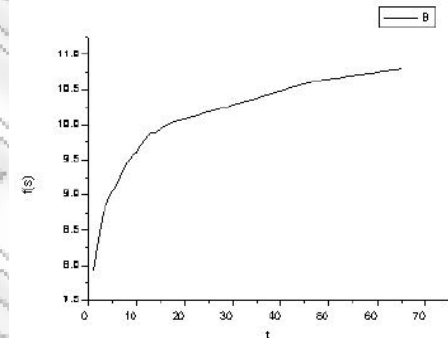
$$\lim_{T \rightarrow \infty} P_{\text{succ}}(T) = 1 - \lim_{T \rightarrow \infty} P_{\text{fail}}(T) = 1. \quad \blacksquare$$

#### 4.2.4 Πειραματικά Αποτελέσματα

Υλοποιήσαμε ένα ACO στο πρόβλημα n-bit παγίδας. Θέσαμε  $n=11$ , βαθμός εξάμισης  $\rho=0.95$ , και χρήση 20 μυρμήγκια. Πρώτα, για να δοκιμάσουμε την ασυμπτωτική σύγκλιση, κάνουμε 100 εκτελέσεις και καταγράφουμε  $Q$ , η λύση φερομένης σε κάθε επανάληψη  $t$ . Η Εικόνα 1 δείχνει τη μέση φερομένη σε κάθε επανάληψη 100 εκτελέσεων. Βλέπουμε ότι ο  $Q$  μειώνεται καθώς το  $t$  αυξάνεται, υποδεικνύοντας ότι ο ACO δεν έχει ασυμπτωτική σύγκλιση στο πρόβλημα n-bit παγίδας.



Εικόνα 1. Μέση λύση φερομένης



Εικόνα 2. Μέση υπερκείμενη λύση

Κατόπιν, για να δείξουμε σύγκλιση προσβασιμότητας, σχεδιάζουμε στην Εικόνα 2 τη μέση προσαρμογή της υπερκείμενης λύσης  $f$  ως προς τον αριθμό επαναλήψεων  $t$ . Βλέπουμε ότι το μέσο  $f$  προσεγγίζει τη βέλτιστη τιμή 11 όσο το  $t$  αυξάνεται, δείχνοντας τη σύγκλιση προσβασιμότητας reachability convergence του ACO.

#### 4.2.5 Συμπεράσματα

Εδώ αναλύσαμε θεωρητικά τους ACO αλγόριθμους σε παραπλανητικά προβλήματα, μία κλάση προβλημάτων που θεωρούνται δύσκολα για τον ACO. Παρουσιάσαμε μία πρώτη προσπάθεια σε σχέση με μία ανάλυση σύγκλισης για τον ACO σε παραπλανητικά συστήματα.

Χρησιμοποιώντας ένα πρόβλημα n-παγίδας ως παράδειγμα πρώτης τάξεως παραπλανητικών προβλημάτων, αποδείξαμε ότι ο ACO μπορεί να πετύχει σύγκλιση προσβασιμότητας αλλά ότι ασυμπτωτική σύγκλιση για κάποια παραπλανητικά προβλήματα πρώτης τάξης χωρίς να υποθέσουμε μία ελάχιστη φερομένη σε κάθε επανάληψη. Παρουσιάσαμε επίσης πειραματικά αποτελέσματα που επιβεβαιώνουν την ανάλυση.

## Αναφορές

### 3.1

- [1] G. Bilchev and I. C. Parmee. The ant colony metaphor for searching continuous design spaces. In T. C. Fogarty, editor, *Evolutionary Computing, AISB Workshop*, volume 993 of *Lecture Notes in Computer Science*, pages 25–39. Springer, 1995.
- [2] G. E. P. Box and M. A. Muller. A note on the generation of random normal deviates. *Annals. Math. Stat.*, 29:610–611, 1958.
- [3] F. O. de Franca, F. J. Von Zuben, and L. N. deCastro. Max min ant system and capacitated p-medians: Extensions and improved solutions. *Informatica (Slovenia)*, 29(2):163–172, 2005.
- [4] M. Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992.
- [5] M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 11–32. McGraw-Hill, London, 1999.
- [6] M. Dorigo and T. Stützle. The ant colony optimization metaheuristic: Algorithms, applications, and advances. In F. W. Glover and G. A. Kochenberger, editors, *Handbook of Metaheuristics*, pages 251–286. Kluwer Academic Press, 2003.
- [7] J. Dréo and P. Siarry. A new ant colony algorithm using the heterarchical concept aimed at optimization of multim minima continuous functions. In M. Dorigo, G. D. Caro, and M. Sampels, editors, *Ant Algorithms*, volume 2463 of *Lecture Notes in Computer Science*, pages 216–221. Springer, 2002.
- [8] M. Guntsch and M. Middendorf. A population based approach for ACO. In S. Cagnoni, J. Gottlieb, E. Hart, M. Middendorf, and G. Raidl, editors, *Applications of Evolutionary Computing, Proceedings of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim*, volume 2279 of *LNCS*, pages 72–81, Kinsale, Ireland, 3-4 2002. Springer-Verlag.
- [9] I. T. Hernádvölgyi. Generating random vectors from the multivariate normal distribution. Technical Report TR-98-07, University of Ottawa, Aug. 20 1998.
- [10] G. Marsaglia and W. W. Tsang. The ziggurat method for generating random variables. *Journal of Statistical Software*, 5(8):1–7, 2000.
- [11] N. Monmarché, G. Venturini, and M. Slimane. On how *Pachycondyla apicalis* ants suggest a new search algorithm. *Future Generation Computer Systems*, 16(8):937–946, 2000.
- [12] S. H. Pourtakdoust and H. Nobahari. An extension of ant colony system to continuous optimization problems. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stützle, editors, *ANTS Workshop*, volume 3172 of *Lecture Notes in Computer Science*, pages 294–301. Springer, 2004.
- [13] Y.-W. Shang and Y.-H. Qiu. A note on the extended Rosenbrock function. *Evolutionary Computation*, 14(1):119–126, March 2006.
- [14] K. Socha and M. Dorigo. Ant colony optimization for continuous domains. *European Journal of Operational Research*, In Press, Corrected Proof, 2006.
- [15] T. Stützle and M. Dorigo. ACO algorithms for the quadratic assignment problem. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 33–50. McGraw-Hill, London, 1999.

### 3.2

- [1] Bell, W. J.: Chemo-orientation in Walking Insects, *Chemical Ecology of Insects*, Bell, W. J. and Carde, R. T. Eds, 93-109 (1984).
- [2] Bullnheimer, B., Hartl, R. F., and Strauss, C.: A New Rank Based Version of the Ant System: A Computational Study, *Central European Journal for Operations Research and Economics*, 7(1):25-38 (1999).
- [3] Bullnheimer, B., Hartl, R. F., and Strauss, C.: Applying the Ant System to the Vehicle Routing

- Problem, *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Voss, S., et al (Eds.), Kluwer (1999).
- [4] Costa, D. and Hertz, A.: Ants Can Colour Graphs. *Journal of the Operational Research Society*, 48:295-305 (1997).
- [5] Dorigo M., Maniezzo, V., and Colorni, A.: The Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Trans. on SMC-Part B*, 26(1):29-41(1996).
- [6] Dorigo M. & L.M. Gambardella (1997). Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE TEC*, 1(1):53-66 (1997).
- [7] Forsyth P. and Wren, A.: An Ant System for Bus Driver Scheduling. *Proc. of the 7th Int. Workshop on Computer-Aided Scheduling of Public Transport* (1997).
- [8] Goldberg, D. E.: *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley publishing company (1989).
- [9] Higuchi, T., Tsutsui, S., and Yamamura, M.: Theoretical analysis of simplex crossover for real-coded Genetic Algorithms, *Proc. of the PPSN VI*, 365-374 (2000).
- [10] Kuntz P., Layzell, P., and Snyers, D.: A Colony of Ant-like Agents for Partitioning in VLSI Technology. *Proc. of the 4th European Conf. on Artificial Life*, 417-424, MIT Press (1997).
- [11] Larranaga, P. and Lozano, J. A. (eds): *Estimation of distribution algorithms*, Kluwer Academic Publishers (2002).
- [12] Law, J. H. and Regnier, F. E.: *Annual Review of Biochemistry*, 40:533-548 (1971).
- [13] Lorenzo Figueiras, A.N. and Lazzari, C. R.: Aggregation behavior and interspecific responses in three species of Triatomine, *Memórias do Instituto Oswaldo Cruz* . 93(1):133-137 (1998).
- [14] Maniezzo, V.: Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem, *CSR 98-1*, Scienze dell'Informazione, Università di Bologna (1998).
- [15] Ono, I. and Kobayashi, S.: A Real-Coded Genetic Algorithm for Function Optimization Using the Unimodal Normal Distribution Crossover, *Proc. of the 7th ICGA*, 246-253, (1997).
- [16] Pelikan, M., Goldberg, D. E., and Lobo, F.: A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1), 5-20 (2002). Also Technical Report IlliGAL Report 99018, University of Illinois at Urbana-Champaign (1999).
- [17] Pena, J. M., Lozano, J. A., and Larranaga, P.: Benefit of Data Clustering in Multimodal Function Optimization via EDAs, *Estimation of Distribution Algorithms*, Larranaga, P. and Lozano, J. A. (Eds), Kluwer Academic Publishers, Chap. 10:211-229 (2002).
- [18] Sakuma, M. and Fukami, H.: Aggregation arrestant pheromone of the German cockroach, *Blattella germanica* (L.) (Dictyoptera: Blattellidae): isolation and structure elucidation of blasttellanostanamide-A and -B, *Journal of Chem. Ecol.* 19:2521-2541 (1993).
- [19] Stützle, T. and Hoos, H.: The MAX-MIN Ant System and local Search for Combinatorial Optimization Problems: Towards Adaptive Tools for Global Optimization, *Proc. of the 2nd Metaheuristics Int. Conf. (MIC-97)* (1997).
- [20] Schatzman, M., Taylor, J., and Schatzman, M.: *Numerical Analysis: A Mathematical Introduction*, Oxford Univ Press (2002).
- [21] Schoonderwoerd, R., Holland, O., Bruten, J., and Rothkrantz, L.: Ant-based Load Balancing in Telecommunications Networks. *Adaptive Behavior*, 5(2):169-207 (1997).

#### 4.1

- [1] M. Birattari, G. Di Caro, and M. Dorigo, "For a formal foundation of the ant programming approach to combinatorial optimization. Part 1: The problem, the representation, and the general solution strategy," ATR-Human Information Processing Laboratories, Kyoto, Japan, Tech Rep. TR-H-301, Dec. 2000
- [2] B. Bullnheimer, R.F. Hartl, and C. Strauss, "A new rank-based version of the ant system: A computational study," *Central Eur. J. Oper. Res. Econom.* , vol. 7, no. 1, pp. 25-38, 1999.
- [3] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation (in Italian), Dipartimento di Elettronica, Politecnico di Milano, Milano, Italy, 1992.
- [4] M. Dorigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," *Future Gener. Comput. Syst.* , vol. 16, no. 8, pp. 851-871, 2000.

- [5] M. Dorigo and G. Di Caro, "The ant colony optimization meta-heuristic," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. London, U.K. : McGraw-Hill, 1999, pp. 11-32.
- [6] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artif. Life*, vol. 5, no. 2, pp. 137-172, 1999.
- [7] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 53-66, Apr. 1997.
- [8] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive feedback as a search strategy," *Dipartimento di Elettronica, Politecnico di Milano, Milano, Italy, Tech. Rep. 91-016*, 1991.
- [9] ---, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst. Man Cybern. B*, vol. 26, pp. 29-41, Feb. 1996.
- [10] M. Dorigo and T. Stützle, "The ant colony optimization metaheuristic: Algorithms, applications and advances," in *Handbook of Mathematics*, F. Glover and G. Kochenberger, Eds. Norwell, MA :Kluwer, to be published.
- [11] P. P. Grassé, "La reconstruction du nid et les coordinations interindividuelles chez *bellicositermes natalensis et cubitermes sp.* La théorie de la stigmergie essai d'interprétation du comportement des termites constructeurs," *Insectes Sociaux*, vol. 6, pp. 41-81, 1959.
- [12] W. J. Gutjahr, "ACO algorithms with guaranteed convergence to the optimal solution," *Info. Processing Lett.*, vol. 82, no. 3, pp. 145-153, 2002.
- [13] ---, "A generalized convergence result for the graph-based ant system metaheuristic," *Dept. Statistics and Decision Support Systems, Univ. Vienna, Vienna, Austria, Tech. Rep. 99-09*, 1999.
- [14] ---, "A graph-based ant system and its convergence," *Future Gener. Comput. Syst.*, vol. 16, no. 8, pp. 873-888, 2000.
- [15] B. Hajek, "Cooling schedules for optimal annealing," *Math. Oper. Res.*, vol. 13, no. 2, pp. 311-329, May 1988.
- [16] N. Meuleau and M. Dorigo, "Ant colony optimization and stochastic gradient descent," *Artif. Life*, vol. 8, no. 2, pp. 103-121, 2002.
- [17] F. Romeo and A. Sangiovanni-Vincentelli, "A theoretical framework for simulated annealing," *Algorithmica*, vol. 6, no. 3, pp. 302-345, May 1991.
- [18] T. Stützle and H. H. Hoos, "The MAX-MIN ant system and local search for the traveling salesman problem," in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, T. Bäck, Z. Michalewicz, and X. Yao, Eds. Piscataway, NJ: IEEE Press, 1997, pp. 309-314.
- [19] ---, "MAX-MIN ant system," *Future Gener. Comput. Syst.*, vol. 16, no. 8, pp. 889-914, 2000.

## 4.2

- [1] C. Blum. Theoretical and practical aspects of ant colony optimization. *Dissertations in Artificial Intelligence*, 2463, 2004.
- [2] C. Blum. Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews*, 2(4):353-373, December 2005.
- [3] C. Blum and M. Dorigo. Deception in ant colony optimization. In *Proceedings of ANTS2004, Lecture Notes in Computer Science*, pages 119-130, 2004.
- [4] C. Blum and M. Dorigo. Search bias in ant colony optimization: on the role of competition-balanced systems. *IEEE Transactions on Evolutionary Computation*, 9: 159-174, 2005.
- [5] M. Dorigo and C. Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344:243-278, 2005.
- [6] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, 2004.
- [7] W. Gutjahr. A Graph-based Ant System and its convergence. *Applied Soft Computing*, 16:873-888, 2000.
- [8] W. Gutjahr. ACO algorithms with guaranteed convergence to the optimal solution. *Info. Processing Lett.*, 82:145-153, 2002.
- [9] S. Shtovba. Ant algorithms: Theory and applications. *Programming and Computer Software*, 31: 167-178, 2005.



[10] T. Stützle and M. Dorigo. A Short Convergence Proof for a Class of Ant Colony Optimization Algorithms. IEEE Transactions on Evolutionary Computation, 6:358-365, 2002.