

---

Università degli Studi di Padova – Dipartimento di Ingegneria dell'Informazione  
Corso di Laurea in Ingegneria Informatica

**Relazione per la prova finale:**  
**«Swarm Intelligence: Analisi Comparativa di  
Algoritmi Metaeuristici ispirati dalla Natura»**

Relatore: Prof. Mirco Rampazzo

Laureanda: Chiara Vigna  
Matricola n.: 2109951

Padova, 22/09/2026

## I Sistemi Complessi in Natura

- **Comportamento emergente:** Da reti di interazioni locali non lineari tra agenti semplici emerge un'intelligenza collettiva globale.
- **Swarm Intelligence:** Massima espressione biologica in colonie di formiche, banchi di pesci, stormi di uccelli o sciami di api.
- **Decentralizzazione:** Risoluzione di problemi complessi senza un controllo centralizzato o un leader globale.
- **Stigmergia:** Coordinamento guidato da stimoli tramite l'alterazione dell'ambiente condiviso.



## La Sfida Ingegneristica

- **Problemi di classe NP-hard:** I metodi esatti falliscono a causa dell'esplosione esponenziale dei tempi di calcolo.
- **Approcci Metaeuristici:** Rinunciano all'ottimalità globale per fornire **soluzioni sub-ottime** di elevata qualità in tempi computazionali ragionevoli.
- **Versatilità:** Richiedono poche assunzioni strutturali sul dominio.



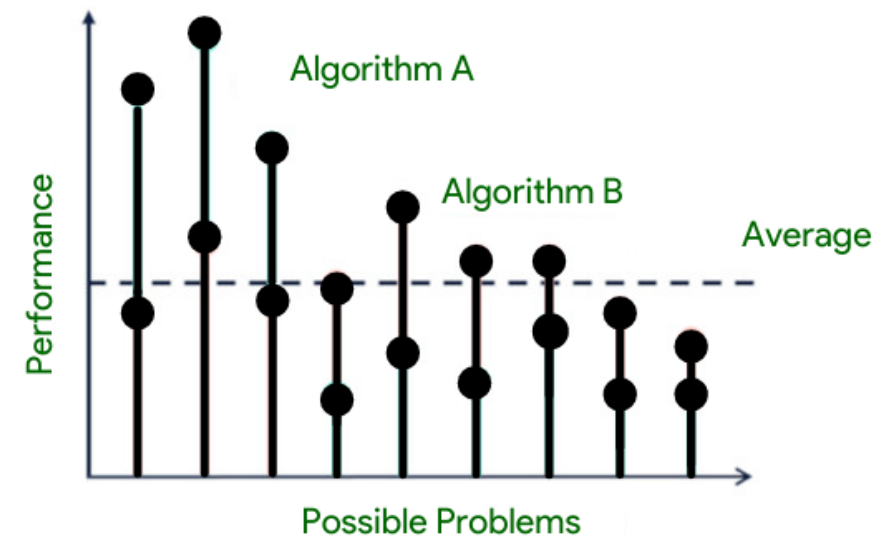
# NFL Theorem

## No Free Lunch Theorem (NFL)

*Formulato da David Wolpert e William Macready nel 1997.*

**Se le prestazioni di tutti gli algoritmi di ottimizzazione sono mediate sull'insieme di tutti i problemi configurabili, esse risultano identiche.**

**Conseguenza fondamentale:** non ha senso cercare una metaeuristica universalmente «perfetta». Le prestazioni superiori su un problema sono compensate da carenze su problemi diversi.



# Il trade-off tra Exploration ed Exploitation

## Exploration (Esplorazione)

- **Obiettivo:** Mappare ampie porzioni del dominio di ricerca saltando in regioni distanti.
- **Necessità:** Indispensabile nelle fasi iniziali dell'ottimizzazione per individuare le zone promettenti e sfuggire agli ottimi locali.
- **Rischio di sbilanciamento:** Un algoritmo puramente esplorativo si comporta come una ricerca casuale, non convergendo mai stabilmente.

## Exploitation (Sfruttamento)

- **Obiettivo:** Concentrare la ricerca in un intorno ristretto raffinando le soluzioni già scoperte.
- **Necessità:** Indispensabile nelle fasi finali dell'ottimizzazione per trovare con precisione la migliore soluzione.
- **Rischio di sbilanciamento:** Un algoritmo focalizzato solo sullo sfruttamento soffre di convergenza prematura.

**Strategia Ideale: Transizione fluida dall'esplorazione globale iniziale allo sfruttamento locale finale.**

# Accelerated PSO (APSO)



## La semplificazione di Xin-She Yang

- **Riconduzione della diversità:** Nella PSO classica, la memoria individuale ( $\mathbf{x}_i^*$ ) evita il collasso, ma Yang dimostra che tale diversità si può simulare **introducendo casualità stocastica controllata**.

$\epsilon \in [0, 1]$ : variabile casuale  
uniforme con shift

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \alpha \left( \epsilon - \frac{1}{2} \right) + \beta (\mathbf{g}^* - \mathbf{x}_i^t)$$

- **Rimozione della velocità:** Per accelerare il calcolo, viene eliminata l'equazione della velocità, aggiornando direttamente la posizione della particella  $i$  all'iterazione  $t+1$ .

## Equazione del modello:

$$\mathbf{x}_i^{t+1} = (1 - \beta) \mathbf{x}_i^t + \beta \mathbf{g}^* + \alpha \epsilon_n$$

- $\mathbf{g}^*$ : Migliore soluzione globale dello sciame
- $\beta$ : Parametro di attrazione (0.1 ~ 0.7)
- $\epsilon_n$ : Vettore casuale stocastico (N(0,1) o U(-0.5, 0.5))

## Decadimento esponenziale di $\alpha$ :

$$\alpha = \alpha_0 e^{-\gamma t}$$

(o  $\alpha_0 \gamma^t$ , con  $\gamma = 0.7$ )

Garantisce una transizione naturale e fluida dall'**esplorazione** (alto  $\alpha$  iniziale) al **raffinamento locale** (basso  $\alpha$  finale).

# Artificial Bee Colony (ABC)

| Karaboga (2005) modella la Swarm Intelligence con la specializzazione dei tre ruoli:



## Employed Bees

- **Associazione:** Una per ciascuna delle **S fonti di cibo** (soluzioni).
- **Perturbazione monodimensionale:** Modificano una singola coordinata **j** della fonte  $x_i$  prendendo spunto da un'altra fonte casuale **k**:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

- **Selezione Greedy:** Mantengono la nuova posizione solo se migliora la fitness.

## Onlookers Bees

- **Scelta probabilistica:** Selezionano la fonte tramite **roulette wheel** proporzionale alla **fitness**:

$$P_i = \frac{F(x_i)}{\sum_{j=1}^S F(x_j)}$$

- **Compressione del range:** Probabilità compressa in  $[0.1, 1.0]$  per non escludere mai del tutto le fonti peggiori.

## Scout Bees

- **Abbandono delle fonti:** Se una fonte non migliora dopo un 'limit' di iterazioni (**limit** =  $SN * D$ ).
- **Re-inizializzazione:** L'ape employed associata diventa scout e rigenera la soluzione in modo casuale nel dominio.
- **Evitare stalli:** Introduce una forte diversificazione stocastica per sfuggire ad ottimi locali.

# Metodologia e Fair Comparison

## Principi di Fair Comparison

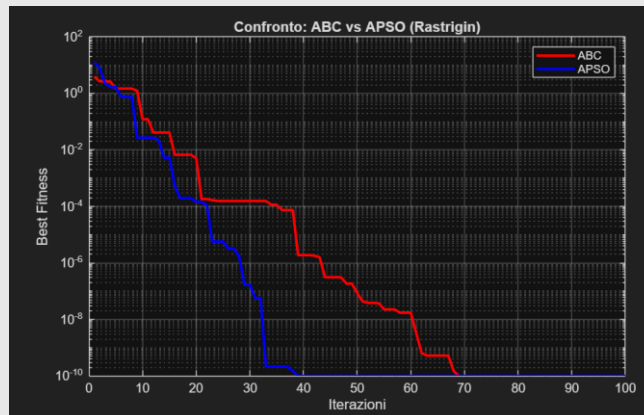
- **Equità Computazionale:** occorre concedere ad entrambe le metaeuristiche lo stesso sforzo computazionale (stesso numero di valutazioni della funzione obiettivo, FEs).
- **Regolazione dei Parametri:** si adatta la popolazione e le iterazioni in modo che il prodotto totale delle FEs coincida.
- **Campagna di test:** 100 iterazioni totali, popolazione  $N=50$  su funzioni 2D.

| Caratteristica        | APSO (Yang)                                           | ABC (Karaboga)                                                                   |
|-----------------------|-------------------------------------------------------|----------------------------------------------------------------------------------|
| Spazio di Ricerca     | Continuo                                              | Continuo                                                                         |
| Valutazioni per iter. | 1 per particella ( $N_{\text{APSO}}$ )                | 2 per fonte di cibo ( $N_{\text{employed}} + N_{\text{onlooker}}$ )              |
| Agenti di ricerca     | 50 particelle                                         | 50 api (25 fonti di cibo)                                                        |
| Formula Equivalenza   | $N_{\text{APSO}} \times \text{MaxIter}_{\text{APSO}}$ | $(N_{\text{employed}} + N_{\text{onlooker}}) \times \text{MaxIter}_{\text{ABC}}$ |

# Convergenza su Rastrigin e Ackley

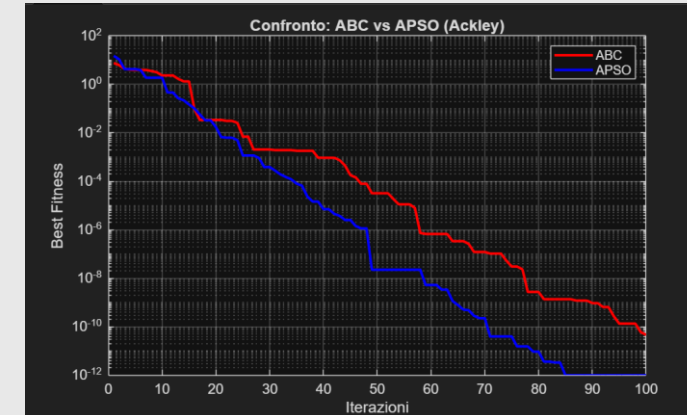
## Funzione Rastrigin (Multimodale, Separabile)

- **Morfologia:** Superficie complessa ricca di minimi locali periodici disposti a griglia.
- **Risultati:** Entrambi gli algoritmi azzerano l'errore raggiungendo il minimo globale noto ( $f(x^*)=0$ ).
- **Dinamica:** APSO converge fluidamente all'iterazione 40 grazie alla spinta diretta verso il global best. ABC converge all'iterazione 66 mostrando un andamento a gradini, in cui i plateau di stallo vengono superati dall'azione stocastica delle api scout.



## Funzione Ackley (Multimodale, Non Separabile)

- **Morfologia:** Ampia periferia quasi piatta con deboli oscillazioni ed un solco centrale profondo.
- **Risultati di precisione alla centesima iterazione:**
  - APSO:  $3.99 \times 10^{-15}$  (all'iterazione 85)
  - ABC:  $2.41 \times 10^{-11}$
- **Dinamica:** L'aggressività vettoriale di APSO permette di superare rapidamente la regione piatta esterna e scendere nel solco centrale, ottenendo un'accuratezza numerica superiore.



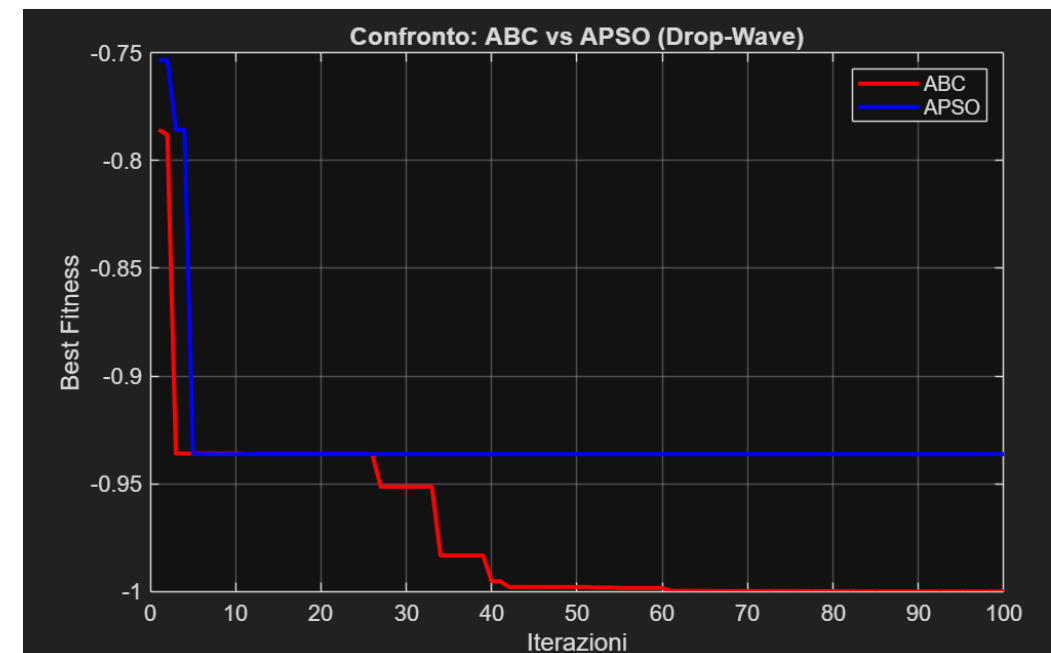
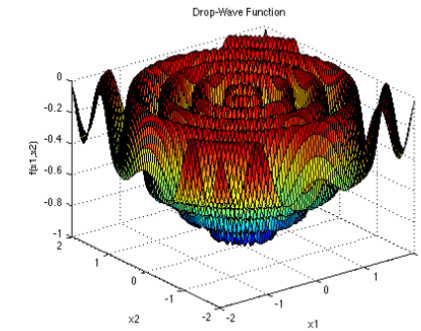
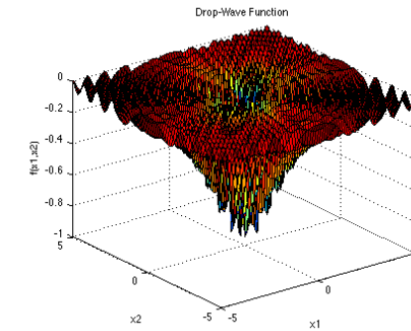
# Convergenza su Drop-Wave

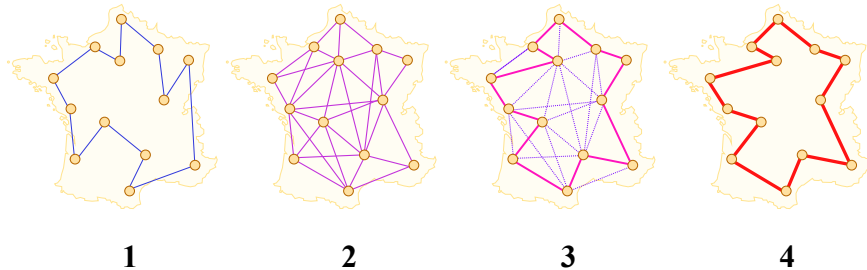
## Il Fallimento di APSO

- La Funzione **Drop-Wave**: Fortemente multimodale e non separabile, caratterizzata da onde concentriche.
- **APSO cade in trappola**: La forte accelerazione fa precipitare le particelle in uno dei minimi ad anello periferici, che agiscono da potenti attrattori locali.
- **Plateau prematuro**: APSO si blocca a una Best Fitness di -0.9362453 (lontano dal minimo globale noto di -1.0) e vi rimane fino all'iterazione 100.

## L'Efficacia di Esplorazione di ABC

- La colonia raggiunge un valore finale di -0.9999998 (all'iterazione 42).
- **Ruolo delle api scout**: Fornisce la diversità stocastica necessaria per evadere dall'avvallamento secondario in cui APSO rimane intrappolato.





## ACO applicato al Traveling Salesman Problem

- **Il Problema (TSP):** Individuare il percorso di lunghezza minima che visiti  $n$  città esattamente una sola volta per poi ritornare alla città d'origine. **NP-hard combinatorio.**
- **Eliminazione dei sotto-tour:** Anziché imporre complessi vincoli matematici della ricerca operativa classica (**DFJ** o **MTZ**), la prevenzione è garantita in ACO da una memoria locale delle singole formiche (la lista dei nodi già visitati, '**visited**').
- **Regola di transizione probabilistica:**

$$p_{ij} = \frac{\text{concentrazione di feromone} \cdot \text{desiderabilità}}{\sum_{l \in N_i} \text{concentrazione di feromone} \cdot \text{desiderabilità}}$$

$\phi_{ij}^\alpha d_{ij}^\beta$

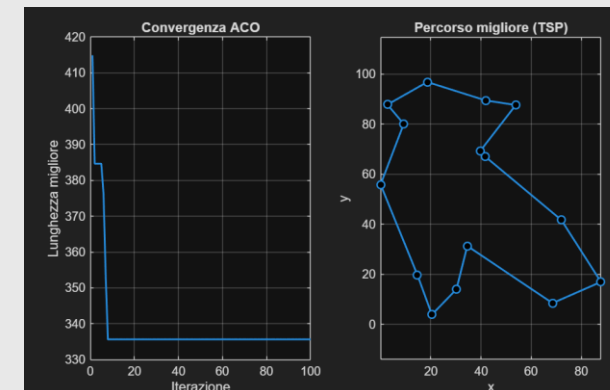
## Risultati Baseline Sperimentale

### Parametri di Baseline:

- $n = 15$  città
- $\alpha = 2$  (peso del feromone)
- $\beta = 2$  (peso dell'euristica)
- $\gamma = 0.5$  (tasso di evaporazione)

### Migliore Lunghezza Trovata: 335.660

- **Convergenza rapidissima:** Raggiunta prima dell'iterazione 10.
- **Proprietà geometrica:** Generazione di un ciclo chiuso privo di intersezioni o incroci degli archi.



# Risultati: TSP-ACO al variare dei principali iper-parametri

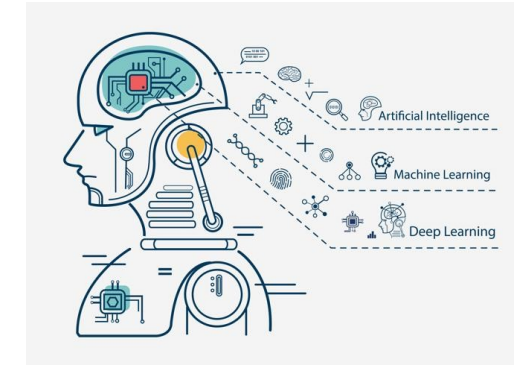


| Configurazione Parametrica                     | Effetto su Grafo Piccolo (n = 15)                                                                                           | Effetto su Grafo Grande (n = 100/150)                                                                                                |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Dominanza Feromone ( $\alpha=5$ , $\beta=1$ )  | Raggiunge una lunghezza sub-ottimale di circa 363 (con incroci geometrici). Stagnazione all'iterazione 7.                   | Forte tendenza alla stagnazione prematura su soluzioni pessime, a causa della dominanza delle prime tracce.                          |
| Dominanza Euristica ( $\alpha=1$ , $\beta=5$ ) | Comportamento greedy di prossimità geometrica immediata. Raggiunge comunque l'ottimo di 335.660 alla terza iterazione.      | Privando lo sciame della memoria collettiva, l'algoritmo si blocca in minimi locali: L=834.032 contro L=823.104 della baseline.      |
| Bassa Evaporazione ( $\gamma=0.1$ )            | Il feromone svanisce molto lentamente. Rallenta la convergenza sull'ottimo 335.660 (raggiunta solo all'iterazione 27 vs 8). | La lentezza nel dismettere percorsi inefficienti inizialmente esplorati si amplifica ostacolando l'ottimizzazione.                   |
| Alta Evaporazione ( $\gamma=0.9$ )             | La traccia si azzerà al 90% ad ogni passo. Consente comunque di trovare l'ottimo di 335.660 pulendo le vecchie tracce.      | Azzerà la memoria e impedisce l'apprendimento cooperativo. Su n=150 nodi, degrada ad una ricerca stocastica: L=1132.481 vs 1063.444. |

# Conclusione

## Sintesi dei risultati sperimentali e limiti del lavoro

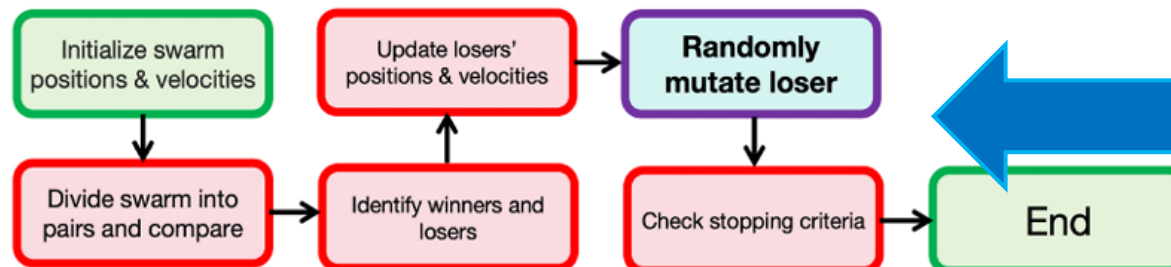
- **Compromesso Confermato:** APSO garantisce elevata velocità ed exploitation, mentre ABC e ACO mostrano robustezza nell'evitare ottimi locali tramite meccanismi di diversificazione stocastica.
- Limiti Metodologici delle analisi:
  - Spazio continuo testato solo in 2D ( $D=2$ ).
  - Mancanza di un'analisi statistica approfondita (media e deviazione standard) su run multipli stocastici.
  - Contesti statici e deterministici.



## Algoritmi metaeuristici come pilastro dell'IA

L'aspetto più interessante è come questi modelli possano essere applicati allo sviluppo dell'AI:

- Superamento dei metodi classici
- Varianti avanzate (CSO-MA)
- Sinergia con il Machine Learning





<https://github.com/ChiaraVigna/swarm-intelligence-thesis>

- [1] M. C. Gianluca Bocchi, *La sfida della complessità*, Milano: Bruno Mondadori, 1985.
- [2] M. D. G. T. E. Bonabeau, *Swarm intelligence. From natural to artificial systems*, Oxford University Press, 1999.
- [3] D. H. Wolpert e W. G. Macready, «"No Free Lunch Theorems for Optimization",» in *IEEE Transactions on Evolutionary Computation*, 1997.
- [4] K. K. Mishra, "Nature-inspired algorithms for Engineers and scientists", 2023.
- [5] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2010.
- [6] D. & B. B. Karaboga, «A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,» *Journal of Global Optimization*, 2007.
- [7] D. Karaboga, «An idea based on honey bee swarm for numerical optimization,» *Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department*, 2005.
- [8] M. Dorigo, *Optimization, Learning and Natural Algorithms*, Politecnico di Milano: PhD thesis, 1992.
- [9] «Virtual Library of Simulations Experiments: Test Functions and datasets,» [Online]. Available: <https://www.sfu.ca/~ssurjano/optimization.html>.
- [10] M. K. A.-B. A. T. G. T. A. N. V. K. M. Z. Naser, «A Review of Benchmark and Test Functions for Global Optimization Algorithms and Metaheuristics,» *Wiley*, 2025.
- [11] «"Der Handlungsreisende – wie er sein soll und was er zu tun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiß zu sein – von einem alten Commis-Voyageur"».
- [12] L. a. W. Biggs, *Graph Theory*, Clarendon Press, 1986.
- [13] G. F. R. & J. S. Dantzig, «Solution of a large-scale traveling-salesman problem,» *Journal of the Operations Research Society of America*, 1954.
- [14] M. Pisani, «How Nature is Inspiring AI Algorithms,» towards data science, 2022. [Online].
- [15] Z. Z. C. J. C. & W. K. W. Elvis Han Cui, «Applications of nature-inspired metaheuristic algorithms for tackling optimization problems across disciplines,» *scientific reports*, 2024.