CSE/ECE 848 Introduction to Evolutionary Computation

Module 3, Lecture 13, Part 1

Metaheuristics and Automatically Tuned EC Methods

Kalyanmoy Deb, ECE

Koenig Endowed Chair Professor

Content of Three Parts

- Part 1:
  - Popular Metaheuristics
- Part 2:
  - Self-adaptive EC Methods
  - Automatic Parameter Tuning in EC Methods
- Part 3:
  - **irace**: Automatic Algorithm Configuration Method

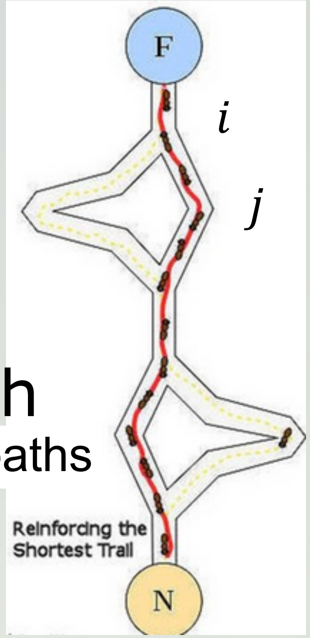
Popular Metaheuristics

- EC: Mimics natural evolution in arriving at increasingly efficient population of solutions in generations
- Metaheuristics: High-level problem-independent strategies to develop heuristics-based algorithms
  - Heuristics: Problem-dependent algorithms
- Many (>100) metaheuristics based on nature-inspired or metaphor-based phenomena
  - Usually, population-based approaches
  - Nature-inspired, metaphor-based
- Memetic algorithms: Hybrid local search and population-based metaheuristics

Reference: Moscato, P. (2012). Metaheuristic optimization frameworks: A survey and benchmarking. *Soft Computing 16*(3): 527–561.

Ant Colony Optimization (ACO)

- Ants find the shortest path to arrive at food
- By "pheromone" release and tracking (τ)
- Initialize random pheromone level
- Multiple ants (pop) move in iterations
- In a TSP, choose an edge (ij) for k-th ant with



$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{l \in C_k} \tau_{ij}^\alpha \eta_{ij}^\beta}, & \text{if } j \in C_k, \\ 0, & \text{otherwise,} \end{cases}$$

η_{ij}: $1/d_{ij}$, $\alpha \geq 0, \beta \geq 1, C_k$: allowed paths

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

- Phermone evaporates with coeff. ρ

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{if ant } k \text{ used the edge } ij \text{ in its tour,} \\ 0, & \text{otherwise.} \end{cases}$$

Q: const., L_k: cost fn.

- Population of ants settle on the shortest path

Reference: M. Dorigo and L. M. Gambardella. *Ant colony system: a cooperative learning approach to the traveling salesman problem*. IEEE Trans. Evol. Comput., 1(1):53, 1997.

ACO Versions and Applications

- Ant System (AS): First ACO algorithm (Dorigo, 1996)
- Ant Colony System (ACS): Local-global pheromone update, select shorter edges with larger phermone
- Elitist Ant System: Best ant always deposit phermone
- Max-min Ant System: Controlled pheromone limit
- Rank-based Ant System: Only best ranked ants deposit
- Cont. Orthogonal Any colony (COAC): Orthogonal array based search by collaborative ants
- Recursive Ant Colony: Subdomain based AS
- Applications: Scheduling, Vehicle routing, Assignment, Set covering, Device sizing, image processing, classification and machine learning problems

Artificial Bee Colony Optimization (ABCO)

- Mimics Bee's strategy of foraging for nectar
- Initialize N food sources in n-dim space randomly
- Three types of Bees: $v_{e,i} = x_{m,i} + \phi_i \left(x_{m,i} - x_i^{(k)} \right)$ $\phi_i \in [-1, 1]$
  - Employed Bees (N): A new food source v_e is created by using its own food location x_m. Better of v_e and x_m in nectar content (f) is kept in memory. (*Mutation+Selection*)
  - Onlooker Bees ($0.5N$): Chooses a new food source v_e based on its nectar content (f). v_e is then modified to obtain v_0. Better of v_e and v_0 is kept in memory. (*Global Selection*)
  - Scout Bees (1): Chooses a random food source location. If an Employed Bee is not performing well, it is converted to a Scout Bee. (*Restart*)
- Numerical problems, combinatorial problems, multi-objective optimization

Reference: D. Karaboga and B. Basturk, (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *Journal of global optimization, 39*(3). 459–471, 2007..

Other Metaheuristics

- Harmony Search
- Bacterial Foraging
- Gene Expression Programming
- Firefly Algorithms
- Gravitational Search Algorithm
- Cuckoo Search
- Shuffled Frog Leaping Algorithm
- Bees Algorithms
- Bee Colony Optimization
- Bat Algorithms
- Invasive Weed Optimization
- Grammatical Evolution
- Charged System Search
- Groundwater flow algorithm
- Big Bang-Big Crunch Optimization
- Honey-Bee Mating Optimization
- Krill Herd Algorithm
- Bee-Hive Algorithm
- Glowworm Swarm Optimization
- Intelligent Water Droplet Algorithm
- Stochastic Diffusion Search
- Cat Swarm Optimization
- Central Force Optimization
- Artificial Fish Swarm Optimization
- Black Hole Optimization
- Monkey Search

Reference: Sunith Bandaru and Kalyanmoy Deb. Metaheuristic techniques. *Decision Sciences*. CRC Press, 2016. 693-750.

Simulated Annealing Method

- Mimics controlled cooling of molten metal to achieve equilibrium state – Annealing process
- Start with a random point $x^{(0)}, T(0), t = 0$
- Perturb in the neighborhood to create $x^{(t+1)}$
- Choose $x^{(t+1)}$ using prob = min[1, $e^{(f(x^{(t)}) - f(x^{(t+1)}))/T(t)}$]
- Temp T is reduced with iteration t
- Asymptotic convergence proof for infinitesimally small change in T
- Combinatorial problems, numerical optimization problems
- SA as a mutation within a GA (SAGA)

References: 1. Kirkpatrick, S.; Gelatt Jr, C. D.; Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science, 220* (4598): 671–680.
2. H. Esbensen and P. Mazumder, "SAGA : a unification of the genetic algorithm with simulated annealing and its application to macro-cell placement," Proceedings of 7th International Conference on VLSI Design, Calcutta, India, 1994, pp. 211-214.

Tabu Search Method

- Start with a single random solution s^t, a Tabu list T, med-term memory M and long-term memory L
- Create N neighbors of s^t, making sure they do not exist in T (avoid repeats and cycles)
- Find best s'^t of N. If $f(s'^t) < f(s^t)$, select s'^t and add it to T. Set $s^t = s'^t$
- Delete oldest solution of T
- For intensification, modify s^t to have common properties of elements in M. Update M.
- For diversification, modify s^t to have least common properties of elements in L. Update L.
- Go back to Second Line
- Combinatorial problems, vehicle routing, etc.

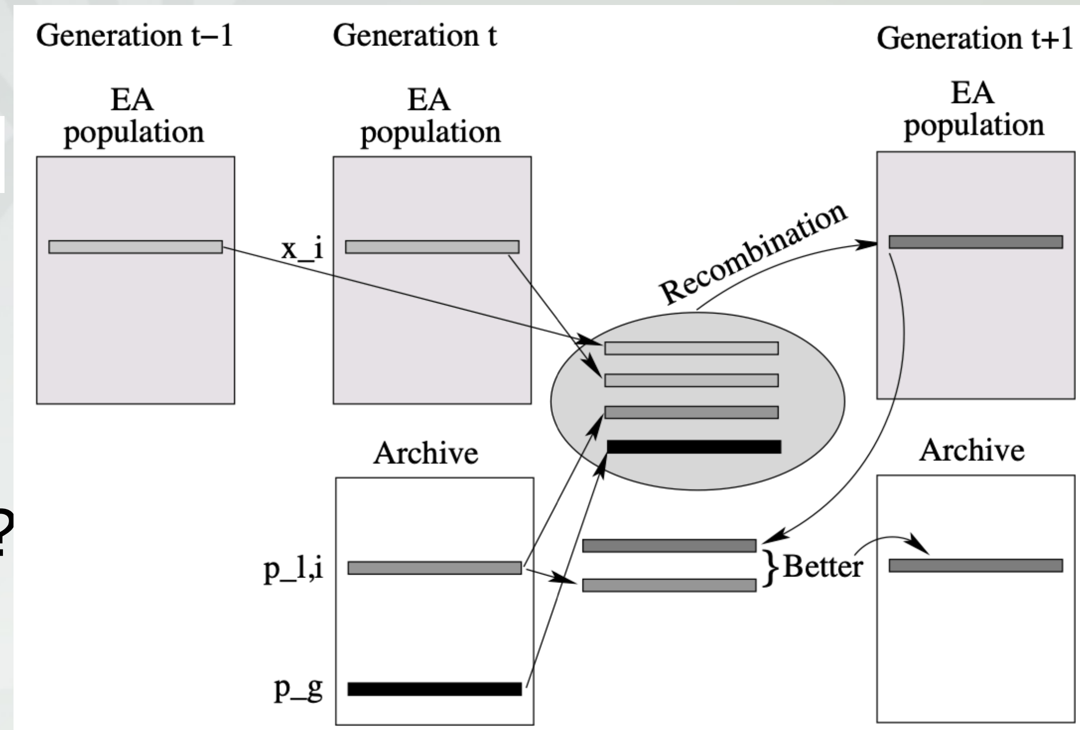
Reference: F. Glover and E. Taillard. (1993). A user's guide to tabu search, *Annals of operations research, 41* (1), 1–28.

.

Algorithmic Equivalence

- Certain metaheuristics methods have internal similarity or one can be expressed as another
- Consider a modified GA:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + w\left(\mathbf{x}_i^t - \mathbf{x}_i^{t-1}\right) + c_1 r_1 \left(\mathbf{p}_{l,i}^t - \mathbf{x}_i^t\right) + c_2 r_2 \left(\mathbf{p}_g^t - \mathbf{x}_i^t\right)$$



- Update GA Pop and an Archive in each iteration
- Is it not same as PSO?
- At the core, they are interchangeable

Reference: Deb, K. and Padhye, N. (2014). Enhancing Performance of Particle Swarm Optimization Through an Algorithmic Link with Genetic Algorithms. *Computational Optimization and Applications, 57*(3), Springer, 761–794.

.

End of Module 3, Lecture 13, Part 1

- Metaheuristics methods are discussed
- About 100+ methods exist
- Which one to use? Any automated way to choose?
- At the core, they may have similarities, which are worth exploring for improving one with the other

- Part 2:
  - Self-adaptive EC methods
  - Automatic parameter tuning in EC
- Part 3:
  - **irace** method