Bee Colony Optimization (BCO)
The first fifteen years

Tatjana Davidović

Mathematical Institute
Serbian Academy od Sciences and Arts

The sixth Symposium
"Mathematics and Applications" 2015
Oct. 16, 2015

Presentation outline

BCO

- Optimization framework, meta-heuristic method;
- Nature-Inspired Algorithm;
- Population based method;
- Imitates swarm behavior;
- Explores collective (swarm) intelligence;
- Based on foraging behavior of honeybees;
- Proposed by Lučić and Teodorović, 2001.

Other bees foraging algorithms

- Artificial Bee Colony (ABC)
  [1] Karaboga, D., "An idea based on honey bee swarm for numerical optimization", Technical report, Erciyes University, Engineering Faculty Computer Engineering Department Kayseri/Turkiye, (2005).
  [2] Karaboga, D., and Basturk, B., "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", Journal of global optimization, 39(3), (2007), 459-471.
- Bees Algorithm (BA)
  [1] Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., and Zaidi, M., "The bees algorithm - a novel tool for complex optimisation problems", Proc. 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006), Elsevier, Cardiff, Wales, UK, (2006) 454-459.
  [2] Pham, D., T., Soroka, A. J., Ghanbarzadeh, A., and Koc, E., "Optimising neural networks for identification od wood defects using the bees algorithm", Proc. IEEE International Conference on Industrial Informatics, Singapore, (2006) 1346-1351.

Bees in the nature

[1] S. Camazine, and J. Sneyd, "A model of collective nectar source by honey bees: Self-organization through simple rules", J. Theor. Biol. vol. 149, 1991, pp. 547-571.

- Scout bees look for a food in the neighborhood of the hive;

Bees in the nature

[1] S. Camazine, and J. Sneyd, "A model of collective nectar source by honey bees: Self-organization through simple rules", J. Theor. Biol. vol. 149, 1991, pp. 547-571.

- Scout bees look for a food in the neighborhood of the hive;
- They return to the hive and opt to one of the possibilities:
  1. become *recruiters*, i.e. to dance and inform their hive-mates about locations (directions and distances), quantities, and qualities of the available food sources;
  2. return to the discovered nectar source and continue collecting nectar;
  3. abandon the food location and become *uncommitted followers*.

Bees in the nature

[1] S. Camazine, and J. Sneyd, "A model of collective nectar source by honey bees: Self-organization through simple rules", J. Theor. Biol. vol. 149, 1991, pp. 547-571.

- Scout bees look for a food in the neighborhood of the hive;
- They return to the hive and opt to one of the possibilities:
  1. become *recruiters*, i.e. to dance and inform their hive-mates about locations (directions and distances), quantities, and qualities of the available food sources;
  2. return to the discovered nectar source and continue collecting nectar;
  3. abandon the food location and become *uncommitted followers*.
- Followers select recruiters and follow them to the nectar source;

Bees in the nature

[1] S. Camazine, and J. Sneyd, "A model of collective nectar source by honey bees: Self-organization through simple rules", J. Theor. Biol. vol. 149, 1991, pp. 547-571.

- Scout bees look for a food in the neighborhood of the hive;
- They return to the hive and opt to one of the possibilities:
  1. become *recruiters*, i.e. to dance and inform their hive-mates about locations (directions and distances), quantities, and qualities of the available food sources;
  2. return to the discovered nectar source and continue collecting nectar;
  3. abandon the food location and become *uncommitted followers*.
- Followers select recruiters and follow them to the nectar source;
- The loyalty and recruitment among bees are always a function of the quantity and quality of the food source.

Waggle dance

Foraging of honey bees

(PceliceSaVirtuelnomKamerom.swf)

Differences between bees in nature and artificial bees

- All artificial bees are included in the search;

Differences between bees in nature and artificial bees

- All artificial bees are included in the search;
- Hive is virtual, it has no specific location;

Differences between bees in nature and artificial bees

- All artificial bees are included in the search;
- Hive is virtual, it has no specific location;
- Communication is synchronous;

Differences between bees in nature and artificial bees

- All artificial bees are included in the search;
- Hive is virtual, it has no specific location;
- Communication is synchronous;
- Artificial bees are divided into two groups:
  1. recruiters;
  2. followers.

Differences between bees in nature and artificial bees

- All artificial bees are included in the search;
- Hive is virtual, it has no specific location;
- Communication is synchronous;
- Artificial bees are divided into two groups:
    1. recruiters;
    2. followers.
- Probabilities and roulette wheel are used to handle loyalty and recruitment.

Method overview

- Builds/improves solutions through iterations (fwd+bck passes);

Method overview

- Builds/improves solutions through iterations (fwd+bck passes);
- Searches solution space through iterations consisting of:
  1. Building/improving solutions (forward pass);
  2. Knowledge exchange (backward pass);

Method overview

- Builds/improves solutions through iterations (fwd+bck passes);
- Searches solution space through iterations consisting of:
  1. Building/improving solutions (forward pass);
  2. Knowledge exchange (backward pass);
- Communication assumes exchange of (partial) solution qualities:

Method overview

- Builds/improves solutions through iterations (fwd+bck passes);
- Searches solution space through iterations consisting of:
  1. Building/improving solutions (forward pass);
  2. Knowledge exchange (backward pass);
- Communication assumes exchange of (partial) solution qualities:
- Consequently, each bee takes one of the following options:
  1. Abandons current solution and decides to follow another bee (uncommitted);
  2. Continues to build current solution and recruits other bees (recruiter).

Method overview

- Builds/improves solutions through iterations (fwd+bck passes);
- Searches solution space through iterations consisting of:
  1. Building/improving solutions (forward pass);
  2. Knowledge exchange (backward pass);
- Communication assumes exchange of (partial) solution qualities:
- Consequently, each bee takes one of the following options:
  1. Abandons current solution and decides to follow another bee (uncommitted);
  2. Continues to build current solution and recruits other bees (recruiter).
- Best obtained solution is reported as the final one;

Method overview

- Builds/improves solutions through iterations (fwd+bck passes);
- Searches solution space through iterations consisting of:
  1. Building/improving solutions (forward pass);
  2. Knowledge exchange (backward pass);
- Communication assumes exchange of (partial) solution qualities:
- Consequently, each bee takes one of the following options:
  1. Abandons current solution and decides to follow another bee (uncommitted);
  2. Continues to build current solution and recruits other bees (recruiter).
- Best obtained solution is reported as the final one;
- Parameters:
  1. B - number of bees;
  2. NC - number of moves during one forward pass.

Bee Colony Optimization - pseudocode

Initialization: Read problem data, parameter values (B and NC),
and stopping criterion.

Do

(1) Assign a(n) (empty) solution to each bee.

(2) *For* $(i = 0; i < NC; i++)$

//**forward pass**

(a) *For* $(b = 0; b < B; b++)$

For $(s = 0; s < f(NC); s++)$//count moves

(i) Evaluate possible moves;

(ii) Choose one move using the roulette wheel;

//**backward pass**

(b) *For* $(b = 0; b < B; b++)$

Evaluate the (partial/complete) solution of bee b;

(c) *For* $(b = 0; b < B; b++)$

Loyalty decision for bee b;

(d) *For* $(b = 0; b < B; b++)$

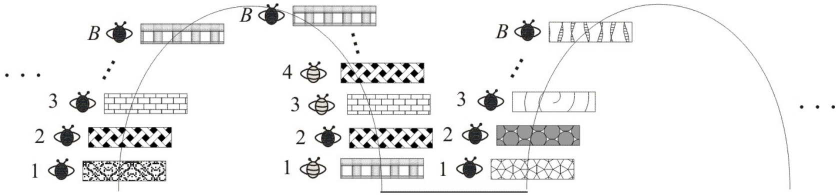
If (b is uncommitted), choose a recruiter by the roulette wheel.

(3) Evaluate all solutions and find the best one. Update x_{best} and $f(x_{best})$

while stopping criterion is not satisfied.

return $(x_{best}, f(x_{best}))$

Bee Colony Optimization - illustration



BCO - Forward pass

- Problem dependent;

BCO - Forward pass

- Problem dependent;
- Builds/improves solutions associated to bees;

BCO - Forward pass

- Problem dependent;
- Builds/improves solutions associated to bees;
- Uses greedy randomized (stochastic) procedures;

BCO - Forward pass

- Problem dependent;
- Builds/improves solutions associated to bees;
- Uses greedy randomized (stochastic) procedures;
- Components/transformations with better characteristics have higher chances to be chosen.

BCO - Backward pass

- Evaluation (Normalization - min.)

$$O_b = \frac{y_{max} - y_b}{y_{max} - y_{min}}, \qquad O_b \in [0,1], \qquad b = 1, 2, \ldots, B$$

BCO - Backward pass

- Evaluation (Normalization - min.)

$$O_b = \frac{y_{max} - y_b}{y_{max} - y_{min}}, \qquad O_b \in [0,1], \qquad b = 1, 2, \ldots, B$$

- Loyalty decision:

$$p_b^{u+1} = e^{-\frac{O_{max} - Ob}{u}}, \qquad b = 1, 2, \ldots, B$$

BCO - Backward pass

- Evaluation (Normalization - min.)

$$O_b = \frac{y_{max} - y_b}{y_{max} - y_{min}}, \qquad O_b \in [0,1], \qquad b = 1, 2, \ldots, B$$

- Loyalty decision:

$$p_b^{u+1} = e^{-\frac{O_{max} - O_b}{u}}, \qquad b = 1, 2, \ldots, B$$

- Recruitment:

$$p_b = \frac{O_b}{\sum_{k=1}^{R} O_k}, \qquad b = 1, 2, \ldots, R$$

BCO modifications

- Initially: Constructive algorithm with independent iterations;

BCO modifications

- Initially: Constructive algorithm with independent iterations;
- Introduction of global knowledge;

BCO modifications

- Initially: Constructive algorithm with independent iterations;
- Introduction of global knowledge;
- Improvement variant: Transformation of complete solutions;

BCO modifications

- Initially: Constructive algorithm with independent iterations;
- Introduction of global knowledge;
- Improvement variant: Transformation of complete solutions;
- Combination of construction and improvement;

BCO modifications

- Initially: Constructive algorithm with independent iterations;
- Introduction of global knowledge;
- Improvement variant: Transformation of complete solutions;
- Combination of construction and improvement;
- Various loyalty functions;

BCO modifications

- Initially: Constructive algorithm with independent iterations;
- Introduction of global knowledge;
- Improvement variant: Transformation of complete solutions;
- Combination of construction and improvement;
- Various loyalty functions;
- Heterogeneous bees;

BCO modifications

- Initially: Constructive algorithm with independent iterations;
- Introduction of global knowledge;
- Improvement variant: Transformation of complete solutions;
- Combination of construction and improvement;
- Various loyalty functions;
- Heterogeneous bees;
- Parallelization;

BCO modifications

- Initially: Constructive algorithm with independent iterations;
- Introduction of global knowledge;
- Improvement variant: Transformation of complete solutions;
- Combination of construction and improvement;
- Various loyalty functions;
- Heterogeneous bees;
- Parallelization;
- Hybridization.

Theoretical verification

- Convergence analysis;

Theoretical verification

- Convergence analysis;
- Best-so-far and model convergence;

Theoretical verification

- Convergence analysis;
- Best-so-far and model convergence;
- Constructive variant is considered in details;

Theoretical verification

- Convergence analysis;
- Best-so-far and model convergence;
- Constructive variant is considered in details;
- Necessary and sufficient conditions are identified;

Theoretical verification

- Convergence analysis;
- Best-so-far and model convergence;
- Constructive variant is considered in details;
- Necessary and sufficient conditions are identified;
- Learning rate is established.

Scheduling independent tasks to identical machines

$T = \{1, 2, \ldots, n\}$ - set of independent tasks,
$M = \{1, 2, \ldots, m\}$ - set of identical machines,
l_i - processing time of task i ($i = 1, 2, \ldots, n$).

Objective: Minimization of completion time of all tasks (*makespan*).

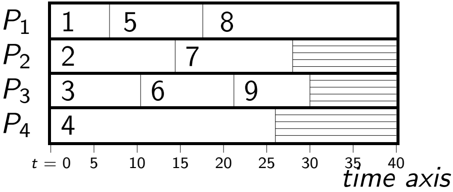


Figure: Gantt diagram–schedule of tasks to processors

Scheduling - mathematical formulation

In order to present a mathematical programming formulation of the problem, let us introduce the binary variables x_{ij} defined in the following way:

$$x_{ij} = \begin{cases} 1, & \text{if task } i \text{ is assigned to processor } j, \\ 0, & \text{otherwise.} \end{cases}$$

The considered scheduling problem is formulated in the following way:

$$\min y \tag{1}$$

$$s.t. \quad \sum_{j=1}^{m} x_{ij} = 1, \;\; 1 \le i \le n, \tag{2}$$

$$y - \sum_{i=1}^{n} l_i x_{ij} \ge 0, \;\; 1 \le j \le m, \tag{3}$$

$$x_{ij} \in \{0, 1\}, \;\; 1 \le i \le n, \;\; 1 \le j \le m, \tag{4}$$

BCO - steps

Construction of solutions: NC tasks are added to the current solution in each forward pass.
Probability of choosing task i equals:

$$p_i = \frac{l_i}{\sum_{k=1}^{K} l_k}, \quad i = 1, 2, \ldots, n \tag{5}$$

with l_i representing the processing time of the i-th task and K being the number of "available" tasks (not previously chosen).
Corresponding processor is selected by a **best fit** rule in such a way that the new solution is not worse than the **current global best** - greedy concept.

The p-Center Problem

- Given is a set of n nodes (locations, customers);
- $D = [d_{ij}]_{n \times n}$ matrix of Euclidean distances between nodes i and j;
- The goal is to locate p facilities (centers) in such a way to minimize the maximum of the distances from each customer to its nearest facility;
- Facilities could be located at any of the given n nodes;
- Customer is assigned to the nearest located facility.

Integer linear program

Binary variables:

$$x_{ij} = \begin{cases} 1, & \text{if user from node } i \text{ is assigned to facility located at node } j, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_j = \begin{cases} 1, & \text{if facility is located at node } j, \\ 0, & \text{otherwise.} \end{cases}$$

The objective to minimize maximum distance between customer and the corresponding facility can be given as

$$\min \max \sum_{j=1}^{n} d_{ij} x_{ij}$$

Integer linear program (Constraints)

$$\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n, \tag{6}$$

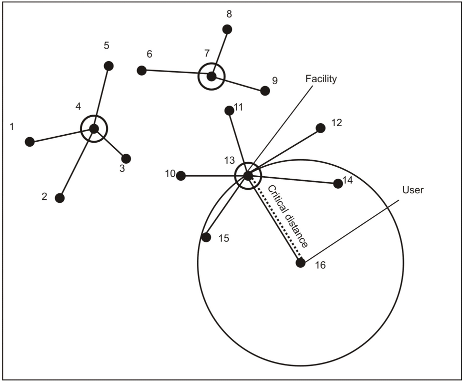
$$x_{ij} \le y_j, \quad 1 \le i \le n, \quad 1 \le j \le n, \tag{7}$$

$$\sum_{j=1}^{n} y_j = p, \tag{8}$$

$$z - \sum_{j=1}^{n} d_{ij} x_{ij} \ge 0, \quad 1 \le i \le n, \tag{9}$$

$$x_{ij}, y_j \in \{0, 1\}, \quad 1 \le i \le n, \quad 1 \le j \le n. \tag{10}$$

Solution example



BCOi - Solution modification

- For the first time improvement variant of BCO is proposed;

BCOi - Solution modification

- For the first time improvement variant of BCO is proposed;
- Different treatment of same solutions has to be assured;

BCOi - Solution modification

- For the first time improvement variant of BCO is proposed;
- Different treatment of same solutions has to be assured;
- A number of facilities Q is substituted by non facility locations;

BCOi - Solution modification

- For the first time improvement variant of BCO is proposed;
- Different treatment of same solutions has to be assured;
- A number of facilities Q is substituted by non facility locations;
- Q - is chosen randomly for each bee;

BCOi - Solution modification

- For the first time improvement variant of BCO is proposed;
- Different treatment of same solutions has to be assured;
- A number of facilities Q is substituted by non facility locations;
- Q - is chosen randomly for each bee;
- Q non centers are added (the solution feasibility is ruined) in such a way to reduce "critical distance";

BCOi - Solution modification

- For the first time improvement variant of BCO is proposed;
- Different treatment of same solutions has to be assured;
- A number of facilities Q is substituted by non facility locations;
- Q - is chosen randomly for each bee;
- Q non centers are added (the solution feasibility is ruined) in such a way to reduce "critical distance";
- Q locations are removed from the center list in a greedy manner;

BCOi - Solution modification

- For the first time improvement variant of BCO is proposed;
- Different treatment of same solutions has to be assured;
- A number of facilities Q is substituted by non facility locations;
- Q - is chosen randomly for each bee;
- Q non centers are added (the solution feasibility is ruined) in such a way to reduce "critical distance";
- Q locations are removed from the center list in a greedy manner;
- $Q \in [0, p]$ if $5 \cdot p < n$, otherwise $Q \in [0, \frac{n-2.5\cdot p}{2.5}]$.

Summary and classification

- Routing: the traveling salesman problem, vehicle routing problem, vehicle routing problem with time windows, Vehicle rerouting in the case of unexpectedly high demand in distribution systems, routing and wavelength assignment (RWA) in all-optical networks;

Summary and classification

- Routing: the traveling salesman problem, vehicle routing problem, vehicle routing problem with time windows, Vehicle rerouting in the case of unexpectedly high demand in distribution systems, routing and wavelength assignment (RWA) in all-optical networks;
- Location: the p-median problem, traffic sensors locations problem on highways, inspection stations locations in transport networks, anti-covering location problem, p-center problem, location of distributed generation resources, and capacitated plant location problem;

Summary and classification

- Routing: the traveling salesman problem, vehicle routing problem, vehicle routing problem with time windows, Vehicle rerouting in the case of unexpectedly high demand in distribution systems, routing and wavelength assignment (RWA) in all-optical networks;
- Location: the p-median problem, traffic sensors locations problem on highways, inspection stations locations in transport networks, anti-covering location problem, p-center problem, location of distributed generation resources, and capacitated plant location problem;
- Scheduling: static scheduling of independent tasks on homogeneous multiprocessor systems, scheduling dependent tasks to homogeneous systems, open-shop scheduling, the ride-matching problem, job shop scheduling, task scheduling in computational grids, backup allocation problem, and berth allocation problem;

Summary and classification (cont.)

- Medicine with chemistry: cancer therapy, chemical process optimization.

Summary and classification (cont.)

- Medicine with chemistry: cancer therapy, chemical process optimization.
- Networks: network design, transit network design problem, urban transit network design;

Summary and classification (cont.)

- Medicine with chemistry: cancer therapy, chemical process optimization.
- Networks: network design, transit network design problem, urban transit network design;
- Continuous and mixed optimization problems: numerical function minimization, the satisfiability problem in probabilistic logic, management of the access charges level for the use of railway infrastructure;

Summary and classification (cont.)

- Medicine with chemistry: cancer therapy, chemical process optimization.
- Networks: network design, transit network design problem, urban transit network design;
- Continuous and mixed optimization problems: numerical function minimization, the satisfiability problem in probabilistic logic, management of the access charges level for the use of railway infrastructure;
- Selection: feature selection problem.

PhD thesis

[1] M. Šelmić, *Location problems on transport networks by computational intelligence methods*, PhD thesis, Faculty of Traffic and Transportation, University of Beograde, 2011.

[2] M. Nikolić, *Resolving the consequences of traffic disturbances by bee colony optimization*, PhD thesis, Faculty of Traffic and Transportation, University of Beograde, 2015.

[3] T. Stojanović, *The development and analisys of metaheuristics for satisfiability in probabilistic logics*, Faculty of Science, University of Kragujevac, 2015.

[4] T. Jakšić Krüger, *The development, parallelization and theoretical verification of bee colony optimization*, Faculty of Technical Sciences, University of Novi Sad, 2015.

Future trends

- Asynchronous communication;

Future trends

- Asynchronous communication;
- New collaboration (e.g., solution decomposition);

Future trends

- Asynchronous communication;
- New collaboration (e.g., solution decomposition);
- Advanced hybridization;

Future trends

- Asynchronous communication;
- New collaboration (e.g., solution decomposition);
- Advanced hybridization;
- Advanced parallelization;

Future trends

- Asynchronous communication;
- New collaboration (e.g., solution decomposition);
- Advanced hybridization;
- Advanced parallelization;
- New applications.

Thank you for the attention!

Questions?

Tatjana Davidović
tanjad@mi.sanu.ac.rs