

# Hamiltonian Mechanics unter besonderer Berücksichtigung der höheren Lehranstalten

Ivar Ekeland<sup>1</sup>, Roger Temam<sup>2</sup> Jeffrey Dean, David Grove, Craig Chambers, Kim B. Bruce, and Elsa Bertino

<sup>1</sup> Princeton University, Princeton NJ 08544, USA,  
I.Ekeland@princeton.edu,

WWW home page: <http://users/~iekeland/web/welcome.html>

<sup>2</sup> Université de Paris-Sud, Laboratoire d'Analyse Numérique, Bâtiment 425,  
F-91405 Orsay Cedex, France

**Abstract.** The abstract should summarize the contents of the paper using at least 70 and at most 150 words. It will be set in 9-point font size and be inset 1.0 cm from the right and left margins. There will be two blank lines before and after the Abstract. . .

**Keywords:** computational geometry, graph theory, Hamilton cycles

## 1 Introduction

The ABC simulates foraging behavior of natural honey bees for getting to the bottom of multi-dimensional and multi-model problems [6]. D. Karaboga in 2005 [4] identified that bees follow a systematic process while collecting nectar in order to produce honey. Since its inception ABC gone through numerous modifications due to its wide applicability in optimization of complex problems. The mathematical formulation of ABC algorithm in analogous to the natural behaviour of real honey bees.

The remaining article is arranged as follow: section 2 explain basic ABC algorithm. Section 3 deliberates some modification in position update equation of ABC. The Arrhenius ABC algorithm is discussed in section 4. In section 5, the RPP problem explained and simulation results are discussed. Section 6 concludes this paper.

## 2 Artificial Bee Colony Algorithm

The ABC algorithm is inspired cooperative and intelligent behavior of honey bees during the search of better quality food sources. The real honey bees lives in hive and has separation of work. These bees are cooperative with each other not competitive. Dervis Karaboga [4] proposed a mathematical model for working of these honey bees and divided them into two classes: employed and unemployed bees. This classification is based on task assigned to them. New food sources are identified by employed bees. The ABC algorithm divided into three phases:

employed bee phase, onlooker bee phase and scout bee phase. These phases are repeated for large number of iteration to get good solutions. These steps are discussed here.

**Initialization:** First of all, ABC initializes all the parameters and random population of bee using Eq. 1 that is analogous to solutions.

$$p_{ij} = LB_j + rand \times (UB_j - LB_j) \quad (1)$$

here,  $i$  varies from 1 to population size and  $j = 1, 2, \dots, D$ . Dimension of selected problem denoted by  $D$ . Position of  $i^{th}$  solution in  $j^{th}$  direction denoted by  $p_{ij}$ . Lower and upper bounds for search space represented by  $LB_j$  and  $UB_j$  accordingly.  $rand$  denotes a randomly value that is selected from the range (0,1).

**Employed Bee Phase:** After initialization, next step is to select better solutions nearby the existing solutions. In this phase solution update themselves using Eq. 2.

$$V_{ij} = p_{ij} + \phi_{ij} \times (p_{ij} - p_{kj}) \quad (2)$$

here,  $\phi_{ij} \in [-1, 1]$  is an arbitrary number,  $k$  varies from 1 to population size.  $k$  should be different from  $i$ . Remaining all symbols have their usual meaning.

**Onlooker Bee Phase:** In this phase a greedy selection approach used to identify better solution for next iteration. This selection takes place with the help of probability. The probability should be a function of fitness for each solution and computed by using Eq. 3.

$$Prob_i = \frac{fitness_i}{\sum_{i=1}^{colony\ size/2} Fitness_i} \quad (3)$$

**Scout Bee Phase:** If a solution is not able to update itself for long time (limit should be pre defined), then it is re-initialised using 1.

### 3 Recent Development in ABC Algorithm

Initially ABC algorithm was developed and implemented for numerical optimization problems only. But now it is very popular for complex optimization problem with different characteristics. It is very simple and easy to implement. The performance of ABC algorithm mainly depends on position update equation (2).

$$ox_{ij} = UB_j + LB_j - x_{ij} \quad (4)$$

Where  $i^{th}$  solution in  $j^{th}$  direction is denoted by  $x_{ij}$  and lower and upper bounds are denoted by LB and UB in that order. Sharma et al. [9] initiated the idea of Levy flight search in ABC. The new position update equation depicted in Eq. 5.

$$x'_{bestj}(t+1) = x_{bestj}(t) + step_{size}(t) \times U(0, 1) \quad (5)$$

Where  $step_{size}$  computed using Eq. 6.

$$step_{size}(t) = 0.001 \times s(t) \times (x_{bestj}(t) = x_{kj}(t)) \quad (6)$$

In Eq. 6 the component  $s$  decided by using Eq. 7 as follow:

$$s = \frac{u}{|v|^{\frac{1}{\beta}}} \quad (7)$$

Where,  $u$  and  $v$  are computed using normal distribution and discussed in detail in [9]. Zhu et al. [14] proposed a unique variant of ABC that is to some extent inspired from particle swarm optimization and named it as Gbest-guided ABC (GABC) algorithm. Here [14] a new approach suggested by Zhu et al. for position update and shown in Eq. 8.

$$v_{ij} = x_{ij} + \phi_{ij} \times (x_{ij} - x_{kj} + \psi_{ij} \times (y_j - x_{ij})) \quad (8)$$

Where  $\psi_{ij} \times (y_j - x_{ij})$  is newly added term that guiding solution search process towards best in current swarm.  $\phi$  denotes an evenly distributed arbitrary number in interval  $[0, C]$ , where  $C$  denotes some positive constant and  $y_j$  represent global best solution in current swarm. The Eq. 8 improve global best solution in present swarm by improving exploitation process. The GABC is one of the effective modification in ABC and attracted number of researchers to work on it. Like Sharma et al. [10] in recent times proposed modification in GABC by taking into consideration local and global best solutions. The employed bee and onlooker bee phase customized as shown in Eq. 9 and Eq. 10 correspondingly.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(x_{Lbestj} - x_{ij}) \quad (9)$$

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(x_{Gbestj} - x_{ij}) \quad (10)$$

Where, local and global best values in current population denoted by  $x_{Lbestj}$  and  $x_{Gbestj}$  respectively. Another variant of gbest ABC proposed by Bhambu et al. in [3] with two amendment in basic GABC [14]. Here both the employed and onlooker bee phase are modified using Eq. 11 and Eq. 12 likewise.

$$x_{newij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \frac{(1-t)}{T} \times \psi_{ij}(y_j - x_{ij}) \quad (11)$$

$$x_{newij} = x_{ij} + \frac{(1-t)}{T} \times \phi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(y_j - x_{ij}) \quad (12)$$

Where,  $t$  and  $T$  represent present and maximum iteration count accordingly. In Eq. 11 and Eq. 12 two important components are defined, first one is:  $\phi_{ij}(x_{ij} - x_{kj})$  that preserve the randomness of the proposed algorithm and another is:  $\psi_{ij}(y_j - x_{ij})$ . The second component move the solutions in the direction of global optimum and accelerate the convergence. The weight-age of these components keep changing and depends on iteration counter to maintain balancing between exploration and exploitation.

A memetic approach incorporated in ABC to enhance the performance of ABC by Bansal et al. [1]. In [1] a complex process based on GSS strategy used to decide the value of  $\phi$  in Eq. 2. This approach update only best solutions in current swarm as it assume that better solutions lies inn closeness of highly fitted solution. A fitness based position update method established in basic ABC by Kumar et al. [7] with supposition that there are more probability to find the majority feasible solution in closeness of solutions with higher fitness. In [7] position of employed bees updated using Eq. 13.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + (2.0 - prob_i)(x_{bestj} - x_{ij}) \quad (13)$$

Tiwari et al. [13] proposed a different position update strategy based on weightage of fitness of each individual. Apart from these strategies discussed here, a number of strategies developed and applied for complex problems very efficiently in ABC. A detailed study of ABC algorithm with step by step derivation available in [6].

#### 4 Arrhenius ABC Algorithm

Some recent studies [2, 6] pointed out that major problem with ABC is unbalanced exploration and exploitation while searching for optimal solution in the given search area. This balancing may be accomplished with appropriate step size at the same time as updating position of individuals.

#### 5 Robot Path Planning Problem (RPPP)

The considered problem has of a robot and a  $2D$  search space with some obstacles. Both the starting and target positions are pre-decided for robot. There are a number of obstacles in the given search space whose co-ordinates and radius are defined by  $r$ ,  $x$ -axis, and  $y$ -axis. The robot is moving in the direction of target position along with a path and in each successive iteration it tries to improve it. In order to change its direction/rotation either in left or right some handle points are there in path. The points construct an entire path from source  $S$  to target  $T$  characterizes as  $(S, n1, n2, n3, \dots, n, T)$ . It is assumed that the robot are capable to turn in both directions (left or right) in case of clashing while moving towards target. The new position of robot at time  $t + 1$  decided by Eq. 14 and Eq. 15, where  $(x, y)$  denotes its current location at time  $t$ ,  $\delta t$  denotes change in time instance,  $\theta$  represents the angle of rotation and  $v$  is the velocity of robot.

$$x' = x + v \times \cos(\theta\delta t) \quad (14)$$

$$y' = y + v \times \sin(\theta\delta t) \quad (15)$$

The Eq. 16 shows the distance ( $d$ ) covered by a robot with velocity  $v$ .

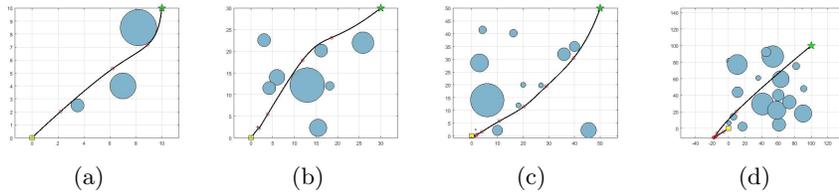


Fig. 1: Simulation results of ABC for RPP problem

$$d = \sqrt{(x' - x)^2 + (y' - y)^2} \quad (16)$$

The overall goal of the RPPP is to diminish the travelling cost i.e. total distance covered from source to destination. In order to apply the aABC algorithm to solve this problem, a 2-D workspace of the robot's motion designed according to the initial and the target positions, number of obstacles and handles. All the required parameters (maximum iteration, population, number of runs) initialized [11]. Using aABC fitness values are computed and possible collisions are detected till the termination criteria meet. Based on the fitness a greedy selection approach used to select next position for robot and it stops when reached to the target. The simulation results of the aABC [8] and ABC [4] are computed and compared with Differential Evolution (DE) [12], and Particle Swarm Optimization (PSO) [5] on Intel core-i5 machine in MATLAB 12.1 using Windows 7 operating system. Maximum iterations for all algorithms are taken 5000. The experiments have been carried out for four different cases, shown in Table 1:

Table 1: Cases considered for experiment

| Case No. | of Obstacles | No. of Handle Points | Start Point | Target Point |
|----------|--------------|----------------------|-------------|--------------|
| 1        | 3            | 3                    | (0,0)       | (10,10)      |
| 2        | 9            | 5                    | (0,0)       | (30,30)      |

Figure 1 shows the simulation result of RPP problem by using ABC, DE, PSO and aABC algorithms respectively. Numerical results are listed in Table 2. The Table 2 demonstrates the outcomes of simulation for the anticipated aABC and other selected algorithms DE, ABC and PSO. The aABC is capable in finding optimal path as the optimal distance measured by the aABC is very less in contrast to the ABC, DE and PSO algorithms.

Table 2: Comparison of Results for the Optimal Path

| Number of obstacles and handles | Algorithm | Optimal Distance |
|---------------------------------|-----------|------------------|
| 3 & 3                           | ABC       | 14.564           |
|                                 | DE        | 14.5825          |
|                                 | PSO       | 14.6891          |
|                                 | aABC      | 14.5448          |
| 9 & 5                           | ABC       | 43.5514          |
|                                 | DE        | 43.4917          |
|                                 | PSO       | 45.4477          |
|                                 | aABC      | 43.4502          |
| 12 & 8                          | ABC       | 72.8696          |
|                                 | DE        | 70.7745          |
|                                 | PSO       | 73.0731          |
|                                 | aABC      | 70.4475          |
| 20 & 12                         | ABC       | 184.0479         |
|                                 | DE        | 154.9557         |
|                                 | PSO       | 146.5722         |
|                                 | aABC      | 144.4057         |

## 6 Conclusion

This paper focuses on a solving robot path planning problem with a unique variant of ABC algorithm, namely Arrhenius ABC algorithm. The just anticipated aABC strategy is inspired by an exclusive concept given by Arrhenius. The exploration capability of the ABC algorithm is improved by the new method. After analysing results it can be concluded that aABC is better choice to get rid of complex optimization problems like RPPP. The results prove that aABC is best choice to solve RPP problem in comparison to other considered algorithms.

Further, the proposed method may be applied on different categories of problems. Like optimization of extracted feature set for image datasets and an attempt of improving its accuracy can be made by incorporating some preprocessing. Also, some new strategy for path set may be implemented for RPPP.

## Bibliography

- [1] JC Bansal, H Sharma, KV Arya, and A Nagar. Memetic search in artificial bee colony algorithm. *Soft Computing*, 17(10):1911–1928, 2013.
- [2] JC Bansal, H Sharma, and SS Jadon. Artificial bee colony algorithm: a survey. *International Journal of Advanced Intelligence Paradigms*, 5(1-2):123–159, 2013.
- [3] P Bhambu, S Sharma, and S Kumar. Modified gbest artificial bee colony algorithm. In *Soft Computing: Theories and Applications*, pages 665–677. Springer, 2018.
- [4] D. Karaboga. An idea based on honey bee swarm for numerical optimization. *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes*, 2005.
- [5] J Kennedy and R Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks IV*, pages, volume 1000, 1995.
- [6] S Kumar and R Kumari. Artificial bee colony, firefly swarm optimization, and bat algorithms. In *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*, pages 145–182. Chapman and Hall/CRC, 2018.
- [7] S Kumar, VK Sharma, A Kumar, and H Sharma. Fitness based position update in artificial bee colony algorithm. *International Journal of Engineering Research & Technology*, 3(5):636–641, 2014.
- [8] Sandeep Kumar, Anand Nayyar, and Rajani Kumari. Arrhenius artificial bee colony algorithm. In *International Conference on Innovative Computing and Communications*, pages 187–195. Springer, 2019.
- [9] H Sharma, JC Bansal, KV Arya, and XS Yang. Lévy flight artificial bee colony algorithm. *International Journal of Systems Science*, 47(11):2652–2670, 2016.
- [10] H Sharma, S Sharma, and S Kumar. Lbest gbest artificial bee colony algorithm. In *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*, pages 893–898. IEEE, 2016.
- [11] P Sharma, H Sharma, S Kumar, and K Sharma. Black-hole gbest differential evolution algorithm for solving robot path planning problem. In *Harmony Search and Nature Inspired Optimization Algorithms*, pages 1009–1022. Springer, 2019.
- [12] R Storn and K Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
- [13] P Tiwari and S Kumar. Weight driven position update artificial bee colony algorithm. In *Advances in Computing, Communication, & Automation (ICACCA)(Fall), International Conference on*, pages 1–6. IEEE, 2016.
- [14] G Zhu and S Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied mathematics and computation*, 217(7):3166–3173, 2010.