ADV MATH SCI JOURNAL

Advances in Mathematics: Scientific Journal **9** (2020), no.6, 3907–3916
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.6.70 Spec Issiue on ICAML-2020

AN IMPROVED ANT COLONY ALGORITHM BASED ON LEVY FLIGHT DISTRIBUTION

SANDHYA BANSAL, RAJEEV GOEL, AND RAMAN MAINI[1]

ABSTRACT. In the present paper, an improved ant colony algorithm hybridized with Levy distribution has been proposed. Ant colony algorithm is a swarm intelligent algorithm inspired from food search strategy of natural ants. It is a very efficient in solving complex optimization problems however, its performance degrades with increase in problem size. It generally suffers from local optima trapping problem for large size combinatorial optimization problems. In this paper, this problem is handled using the Levy flight distribution, in such a way that some of ants will take long jumps according to Levy distribution to jump out from local optima situations. Better performance of proposed approach has been demonstrated by testing it on well-known CEC-2014 non-constrained data-sets.

1. INTRODUCTION

Ant colony optimization (ACO) algorithm is one of the nature inspired algorithm proposed in 1996 [1]. It is a heuristic approach inspired by food searching strategy of real ants. Natural ants work cooperatively to find the food sources, in which some of the ants search for new food sources while other follow previously traversed paths by getting feedback from other ants. ACO preserves good balancing between exploration (using heuristics) and exploitation (positive feedback). Ants which searching for food excrete some scented material (called pheromone) on the traversed path. Next group of ants gets inspiration

[1]*corresponding author*

2010 *Mathematics Subject Classification.* 68W99.

Key words and phrases. Ant Colony Algorithm, Levy Flight, Combinatorial Optimization, Global Optima.

from this pheromone to track already found food sources. This pheromone gets evaporated with the passage of time. So, the paths which are shorter in length have higher concentration of pheromone and have higher probability of being chosen.

It has been proved that ACO performs excellently among its other nature inspired approaches viz. genetic algorithms (GAs), Simulated annealing (SA) approach, Tabu Search (TS) and Particle Swarm Optimization (PSOs) etc. Due to its advantages over other algorithms, ACO has been widely used for solving complex combinatorial optimization problems like, Travelling Salesman Problem (TSP), Bin-Packing problem, Job-shop scheduling and Vehicle Routing Problems (VRP), etc. [2]. ACO generally suffers from some of most common problems of heuristic algorithms like slow convergence and being trapped into local optima [3, 4].

Ant colony system (ACS), Multi-ant colony system, Max-Min ACS, ACS with multi-pheromone matrices, hybrid ACOs has been developed to improve efficiency of ACO. These advanced/improved versions of ACO found better/improved optimized results of many combinatorial optimization problems. However, with such improvements the complexity of ACO increases significantly. So, still researchers are working towards the development of more efficient versions of ant based algorithms with comparatively less complexity and much higher performance rate in solving large scale optimization problems. So, here we are proposing a new hybrid of ACS with sine-cosine algorithm having high performance rate.

Rest of the paper has been organised as follows. Section 2 presents literature related to ACS, while the proposed approach is discussed in section 3. In section 4, experimental work and results obtained has been discussed and finally the work has been concluded in section 5.

2. Related work

An improved ant colony algorithm for mobile robot path planning was proposed in [5]. Local pheromone diffusion process has been introduced for local search space. The paths are further localised using geometric distribution. In [6], ACO has been hybridized with firefly optimization algorithm to avoid local optimization problem and better performance of proposed algorithms over other state of art has been presented by testing on vehicle routing problems (VRP) and stochastic VRP in [6] and [7] respectively. Another improved version of ACO with novel pheromone updation and pheromone diffusion mechanisms is presented in [2]. ACO based financial crisis prediction (FCP) problem has been modelled in [8]. ACO based feature selection and classification algorithms are proposed to differentiate reliable and unreliable clients to avoid bankruptcy. In [9], memetic ant colony algorithm has been developed to generate test instances for reliable software testing. Evolution strategies (ES) is incorporated into ACS to enhance search space for branch coverage s/w testing approach.

Multiple ant colony optimization based upon Pearson correlation coefficient (PCCACO) has been presented in [10]. Pearson correlation coefficient has been used to erect communication between multiple colonies using with help of adapting frequency. Efficiency of proposed algorithm compared to other approaches has been demonstrated by testing all the approaches on standard travelling salesman problem (TSP) datasets. Levy flight based ACO known as Levy-ACO has been proposed in [11]. The original uniformly distributed next location selection probability has been replaced by Levy distributed step length.

3. Proposed algorithm

Ant Colony Optimization: ACO is a swarm based algorithm, which gets inspiration from natural ants. Each ant search for the food source using heuristic information (its own intuition) and getting inspiration from previous ants (cooperative behaviour) [1,4,6]. The information from previous ants get deposited as the pheromone on the paths chosen. Hence the transition from current location to next location is a function of heuristic information as well as the pheromone concentration.

3.1. Solution construction.

Solution construction. In general, the movement of an ant from current location (i) to next location (j) is given by eq. (3.1):

$$j = \begin{cases} \operatorname{argmax}_{j\in\psi}\left((\eta_{ij})^{a}(\tau_{ij})^{b}\right) & if\ q \geq q_0 \\ J & otherwise \end{cases} . \tag{3.1}$$

Here, η_{ij} is the heuristic value (exploration) in going from location (i) to next location (j) and τ_{ij} is the pheromone concentration (exploitation) on the path travelled between these two locations. J is a variable selected randomly according to following probability distribution:

$$p_{ij} = \begin{cases} \frac{(\eta_{ij})^a (\tau_{ij})^b}{\sum_{u \in \psi} (\eta_{iu})^a (\tau_{iu})^b} & \forall j \in \psi \\ 0 & otherwise \end{cases}.$$

where p_{ij} is the probability of choosing the next location j from the set ψ of next available locations.

3.2. **Pheromone update.** Initially, the pheromone is initialized for each path as inverse function of that path length. This pheromone gets updated in two ways: (i) Local Pheromone Update and (ii) Global pheromone update. In each iteration let there are k number of ants which independently work towards solution construction process. For each iteration, each of the ant updates the pheromone concentration locally with formula given by eq. (3.2):

$$\tau_{ij}(t+1) = (1-\rho) . \tau_{ij}(t) + \rho . \sum_{k=1}^{K} \Delta\tau_{ij}^{k}, \tag{3.2}$$

where ρ is local pheromone evaporation constant and $\Delta\tau_{ij}^{k}$ is the amount of pheromone deposited by k^{th} ant which depends upon the total path length (L) of the tour followed by that ant and can be calculated by following formula:

$$\Delta\tau_{ij}^{k} = \frac{1}{L_k}.$$

Similarly, pheromone can be updated globally after each iteration as follows:

$$\tau_{ij}(t+1) = (1-\rho') . \tau_{ij}(t) + \rho' . \Delta\tau_{ij}^{G},$$

where ρ' is global pheromone evaporation constant and $\Delta\tau_{ij}^{G}$ is the amount of pheromone deposited at the end of each iteration given by the formula

$$\Delta\tau_{ij}^{G} = \frac{1}{L_{best}}.$$

Here L_{best} is the total path length of best tour found till now.

Hence, the candidate solutions with smaller distances (i.e. nearly equal objective function values) and higher concentration of pheromone i.e. more attractive solutions (chosen by more number of ants) are more likely to be chosen. However, the solutions which are highly deviated from the current best solutions are less likely to be chosen as candidate sites. Moreover, with the passage of time, the pheromone concentration on the more likely chosen paths get high while concentration of pheromone on other paths gradually get decreased.

So, ACS always focuses on more attractive solutions i.e. can search local regions more effectively but are not equally capable of finding more diversified solutions. Such a situation leads to getting trapped into local optima or sub-optimal solutions. So, more high jumps towards more promising regions can avoid from being trapped in local optima and improve the search efficiency of ACS. Levy flights have the capability of very high jumps and exploring the new regions of search space so, in the present work Levy flights are incorporated into ACS to avoid the problem of sub-optimal solutions. Moreover, to keep balance between exploration and exploitation Levy flights are implemented in a greedy manner to avoid the ants being diverge highly from promising solution space.

3.3. **Levy Flight Optimization.** Levy flights are Markovian stochastic processes whose jumps follows are distributed according to Levy probability distribution functions. These are named after French mathematician named Paul Pierre Levy [12,13]. This distribution decays at large as

$$L\left(s\right) \sim \left|S\right|^{-1-\beta} 0 < \beta \leq 2,$$

$$S = \frac{\mu}{\left|v\right|^{\frac{1}{\beta}}}, \tag{3.3}$$

$$\mu \sim N\left(0, {\sigma_\mu}^2\right), \quad v \sim N(0, {\sigma_v}^2),$$

i.e., follows simple power law or are non-Gaussian distributions. Here S is the step length and β is Levy index. Unlike other common distributions like Gaussian distribution, Cauchy's distribution Levy distributions are fat tailed distributions. Due to very high divergence, $\langle s^2(t)\rangle \to \infty$, extremely long jumps may occurs.

Furthermore,

$$\sigma_\mu = \left(\frac{\Gamma\left(1+\beta\right)\sin\left(\frac{\pi\beta}{2}\right)}{\beta\Gamma\left(1+\beta\right)2^{(\beta-1)/2}}\right)^{1/\beta},$$

$$\sigma_v = 1\ \Gamma\left(1+\beta\right) = \int_0^\infty x^\beta e^{-x} dx.$$

Levy distribution in its simplified form is given by

(3.4)

$$L\left(s,\alpha,\gamma\right)=\begin{cases}\sqrt{\frac{\alpha}{2\pi}}\exp\left[-\frac{\alpha}{2(s-\gamma)}\right]\left[-\frac{1}{(s-\gamma)^{3/2}}\right] & if \quad 0<\gamma<s<\infty \\ 0 & if \quad s\leq 0\end{cases}.$$

Here α controls the distribution scale so called scale parameter and γ is location or shift parameter and s is sample set of the distribution. In the proposed approach, the position of i^{th}ant in j^{th} dimension is updated according to following rule:

$$x_{ij}^{new}=x_{ij}*S,$$

where x_{ij}^{new} is newly updated position of ant and S is step length which distribution according to Levy distribution which is calculated according to eq. (3.3). Interested reader may refer to [13] for more information about Levy flights.

As step size depends significantly on Levy index β so different values of β, results in different step sizes. Larger values of β cause large jumps i.e. exploration of more diversified regions and smaller values of it cause smaller jumps i.e. exploitation of local regions. Presently, we use dynamic values of β according to following equation.

$$\beta=\beta_{max}\left(1-\frac{Curr\_Itr}{Max\_Itr}\right).$$

Initially higher values (exploration) of β are generated while with increase in number of iterations the value of β decreases accordingly (exploitation).

In order to incorporate the presented Levy Flight approach into ACS, we divide the number of ants into two different colonies: (i) Col_A and (ii) Col_B. 75% of ants are dedicated to Col_A and remaining 25% ants works as Col_B. The ants of Col_A search new solutions according to transition probability (eq. (3.1)) while Col_B with explore new regions according to eq. (3.4).

Basic steps of proposed approach are given as follows:

Step 1: Initialize ACS parameters like, number of ants, maximum number of iterations Max_itr, heuristic vs pheromone importance (a,b).

Step 2: Construct initial solution randomly for each of the ant and divide those into 2 different pools i.e. Col_A containing first 75% and Col_B containing rest 25% solutions.

Step 3: Set $Curr\_Itr=1$ and repeat Step 4 to Step 8 till $Curr\_Itr\leq Max\_Itr$.

Step 4: Each colony i.e. $Col\_A\ and\ Col\_B$ generate new solutions using state transition probability rule, eq. (3.1) and Levy distribution, eq. (3.4) respectively.

Step 5: If any (1 or more) of newly generated solution of Col_A is better than any of solution of that pool then replace (1 or more) worst old solutions of that pool by new ones. Do the same for newly generated solutions of Col_B.

Step 6: Apply local pheromone update for each pool separately.

Step 7: Globally update the pheromone for best solution found till now in any of ant pool.

Step 8: $Curr\_Itr = Curr\_Itr + 1$.

Step 9: Present the best solution and stop.

4. RESULTS AND DISCUSSION

The performance of proposed algorithm has been validated by testing it on CEC-2014 unconstrained problems [14]. Each instance has been tested for 100 times and in each run maximum number of iterations are $10^4 \times D$, where D is the dimension of the problem. Number of ants for each run is $5 \times D$. Numerical results obtained by applying basic ACS and proposed approach on 10 dimension problems are presented in Table 1. The best, worst, average and median values obtained among all the 100 runs of each problem instance are reported for comparison. From the Table 1, it is clear that while comparing best values obtained in all the 100 runs GLF_ACS performs much better than ACS in all the 30 instances. Similarly, while comparing average values GLF_ACS provides better average values than ACS in test problems except for f3, f8, f21, f23 and f27 instances. Similarly, on the basis of median values, GLF_ACS performs better as compared to median values obtained from ACS algorithm in instances except for f1, f10, f26 and f27.

However, ACS performs better as compared to GLF_ACS in terms of comparison on obtained worst results. Here, GLF_ACS is better than ACS only in f1, f10, f21, f27 and f30 and in all the remaining instances GLF_ACS provides worse results than ACS. This uncommon behaviour is probably due to long jumps towards unexplored results sometimes may degrade the objective function values. Hence, overall analysis of Table 1 demonstrates that GLF_ACS more efficiently explores the unexplored solution space and thus avoids local optima stagnation problem.

5. Conclusion

Present work focuses on improving the performance of ACO by hybridizing with Levy flight optimization. Local optima trapping and premature convergence problem of ACO has been solved by using large step size jumping of Levy flights. Entire ant colony is divided into two colonies such that one colony constructs solution using ant based transition rule on the other hand another colony uses Levy distribution based transition for finding next moves. Non-constrained CEC-2014 datasets has been used to validate the performance of proposed GLF_ACS. Results shows better ability of proposed algorithm to find optimal solutions as compared to classical ACS. For the future work, to evaluate the performance of proposed algorithm for complex real life problems.

Table 1: Comparison of ACS and GLF_ACS based on CEC-2014 Unconstrained datasets

Dataset	**Algorithm**	**Best**	**Average**	**Worst**	**Median**
f1	ACS	6359	8266.7	10776.6	**8902.6**
	GLF_ACS	**5987**	**7783.1**	**10174.4**	8980.5
f2	ACS	45.4	59.02	**84.21**	86.26
	GLF_ACS	**40.1**	**52.13**	90.8	**72.18**
f3	ACS	1.062	**1.3806**	**1.1664**	1.9116
	GLF_ACS	**0.729**	2.9477	2.4426	**1.0935**
f9	ACS	3.09	4.326	**7.29744**	5.253
	GLF_ACS	**3.0406**	**3.34466**	7.416	**4.5609**
f10	ACS	8.007	10.4091	13.9344	**10.4091**
	GLF_ACS	**6.9672**	**8.36064**	**12.8112**	13.2378
f11	ACS	7.296	10.2144	**7.45584**	12.4032
	GLF_ACS	**3.5504**	**3.90544**	11.6736	**4.61552**
f12	ACS	0.0293	0.04395	**0.033198**	0.05274
	GLF_ACS	**0.01509**	**0.022635**	0.07032	**0.02235**
f13	ACS	0.8047	1.20705	**0.1329**	1.4446
	GLF_ACS	**0.0604**	**0.07254**	1.5293	**0.08463**
f14	ACS	0.425	0.5525	**0.0589**	0.6375

	GLF_ACS	**0.031**	**0.0434**	0.6375	**0.0558**
f17	ACS	0.106	0.1166	**0.048672**	0.1484
	GLF_ACS	**0.02028**	**0.022308**	0.2014	**0.038532**
f18	ACS	49.057	58.8684	**57.35018**	78.4912
	GLF_ACS	**33.7354**	**50.6031**	103.0197	**43.85602**
f19	ACS	2.086	2.2946	**1.468817**	2.7118
	GLF_ACS	**0.86401**	**1.296015**	5.0064	**1.209614**
f20	ACS	3.955	5.9325	**6.362**	7.119
	GLF_ACS	**3.181**	**3.8172**	8.701	**5.4077**
f22	ACS	5.9218	7.10616	**7.59322**	9.47488
	GLF_ACS	**3.3014**	**4.29182**	13.62014	**5.94252**
f23	ACS	3079.82	**3387.802**	**6303.57**	5543.676
	GLF_ACS	**3001.7**	3602.04	6775.604	**5403.06**
f24	ACS	132.04	198.06	**239.338**	250.876
	GLF_ACS	**108.79**	**141.427**	290.488	**163.185**
f25	ACS	150.014	210.0196	**192.057**	225.021
	GLF_ACS	**128.038**	**179.2532**	360.0336	**192.057**
f26	ACS	109.72	164.58	**150.003**	**175.552**
	GLF_ACS	**100.002**	**110.0022**	219.44	180.0036
f27	ACS	2.505	**3.7575**	3.990378	**3.2565**
	GLF_ACS	**1.90018**	3.96652	**3.7575**	3.610342
f30	ACS	376.12	**451.344**	761.19472	601.792
	GLF_ACS	**345.9976**	484.39664	**639.404**	**484.39664**

References

[1] M. Dorigo, V. Maniezzo, A. Colorni: *The Ant System: Optimization by a colony of cooperating agents*, EEE Trans. Syst. Man, Cybern. B, **26**(1) (1996), 1–13.

[2] W. Deng, J. Xu, H. Zhao: *An improved ant colony optimization algorithm based on immunization strategy*, IEEE Access, **7**(2019), 20281–20292.

[3] R. K. Goel, S. R. Bansal: *Hybrid algorithms for rich vehicle routing problems: a survey*, Elsevier Inc, 2020.

[4] R. Jovanovic, M. Tuba, S. Vos: *An efficient ant colony optimization algorithm for the blocks relocation problem*, Eur. J. Oper. Res., **274**(1) (2019), 78–90.

[5] J. Liu, J. Yang, H. Liu, X. Tian, M. Gao: *An improved ant colony algorithm for robot path planning*, Soft Comput., **21**(19) (2017), 5829–5839.

[6] R. GOEL, R. MAINI: *A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems*, J. Comput. Sci., **25**(2018), 28–37.

[7] R. GOEL, R. MAINI, S. BANSAL: *Vehicle routing problem with time windows having stochastic customers demands and stochastic service times: Modelling and solution*, J. Comput. Sci., **34**(2019), 1–10.

[8] U. J. N. METAWA, K. SHANKAR, S. K. LAKSHMANAPRABU: *Financial crisis prediction model using ant colony optimization*, Int. J. Inf. Manage., **50**(2018), 538–556.

[9] H. SHARIFIPOUR, M. SHAKERI, H. HAGHIGHI: *Structural test data generation using a memetic ant colony optimization based on evolution strategies*, Swarm Evol. Comput., **40**(2018), 76–91.

[10] H. ZHU, X. YOU, S. LIU: *Multiple Ant Colony Optimization Based on Pearson Correlation Coefficient*, IEEE Access, **7**(2019), 61628–61638.

[11] Y. LIU, B. CAO: *Improving Ant Colony Optimization algorithm with Levy Flight [C]*, MISTA 2017 Conference, 2017.

[12] A. CHECHKIN, R. METZLER, J. KLAFTER, V. Y. GONCHAR: *Introduction to the Theory of Lévy Flights*, In book: Anomalous Transport: Foundations and Applications, 2008.

[13] A. F. KAMARUZAMAN, A. M. ZAIN: *Levy Flight Algorithm for Optimization Problems- A Literature Review*, Applied Mechanics and Materials, **421**(2013), 496–501.

[14] J. J. LIANG, B. Y. QU, P. N. SUGANTHAN: *Problem Definitions and Evaluation Criteria for the CEC* , 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, 2014.

DEPT. OF CSE, M.M.D.U, MULLANA, INDIA
Email address: sandhya12bansal@gmail.com

CS DEPT. GOVT. COLLEGE, NARAINGARH, AMBALA, INDIA
Email address: rcse123@gmail.com

DEPT. OF CSE, PUNJABI UNIVERSITY, PATAILA, INDIA
Email address: research_raman@yahoo.com