Accelerated Shuffled Frog-Leaping Algorithm

Shweta Sharma, Tarun Kumar Sharma, Millie Pant, Jitendra Rajpurohit and Bhagyashri Naruka

Abstract Shuffled frog-leaping algorithm (SFLA) is a recent addition to the family of stochastic search methods that mimic the social and natural behavior of species. SFLA combines the advantages of local search process of particle swarm optimization (PSO) and mixing of information of the shuffled complex evolution. The basic idea behind modeling of such algorithms is to achieve near to global solutions to the large-scale optimization problems and complex problems which cannot be solved using deterministic or traditional numerical techniques. In this study, the searching process is accelerated using golden section-based scaling factor and the constraints are handled by the penalty functions. Penalty functions are used to find the optimal solution for restrained optimization problems in the feasible region of the total search space. The resulting algorithm is named as Accelerated-SFLA. The proposal is implemented to solve the problem of optimal selection of processes. The results illustrate the efficacy of the proposal.

Keywords Shuffled frog-leaping algorithm · Constrained optimization · Memetic · Swarm intelligence

S. Sharma (✉) · T.K. Sharma ·
J. Rajpurohit · B. Naruka
Amity University, Jaipur, Rajasthan, India
e-mail: shweta_sharma0287@yahoo.com

T.K. Sharma
e-mail: taruniitr1@gmail.com

J. Rajpurohit
e-mail: jiten_rajpurohit@yahoo.com

B. Naruka
e-mail: bhagyashree.naruka@gmail.com

M. Pant
IIT Roorkee, Roorkee, India
e-mail: millidma@gmail.com

K.N. Das et al. (eds.), *Proceedings of Fourth International Conference on Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing 336,
DOI 10.1007/978-81-322-2220-0_15

1 Introduction

Since the last few decades, stochastic search techniques gather the attention of many researchers, scientist, and academicians to solve larger-scale and complex optimization problems arising in the domain of engineering, science, and management. The advantage of such techniques over traditional techniques is their simplicity and easy to implement without requiring the derivation of the objective function and constraints. These techniques require only auxiliary knowledge about the problem.

The stochastic techniques are formulated by inspiring from the natural and social behavior of species. Some of the stochastic techniques are evolutionary programming (EP) [1], genetic algorithms (GA) [2], evolution strategies (ES) [3], particle swarm optimization (PSO) [4], differential evolution (DE) [5], bacterial foraging optimization algorithm (BFOA) [6], artificial bee colony algorithm (ABC) [7, 8], and ant colony optimization (ACO) [9].

Following the same trend, Eusuff and Lansey [10] proposed SFLA, based on evolution of memeplexes. A detailed note is given in Sect. 2. Having the advantage of both PSO and mixing of the information (taken from GA), SFLA has also proved its efficacy and ability in discovering global optimal solutions to several combinatorial optimization problems [10]. In this paper, we have a proposed a penalty-guided SFLA to solve constrained optimization problems.

The rest of the paper is structured as follows: Sect. 2 describes the process of handling constraints. Section 3 briefs the basic SFLA followed by Sect. 4, which briefs the proposal. Optimal selection of processes is described in Sect. 5, and experimental setup and simulated results are defined in Sect. 6. Finally, the conclusions drawn from the study are presented in Sect. 7.

2 Constraints Handling Process

Handling of the constraint in solving constraints optimization problems is an important and key issue. To find the feasible solution for a problem with the presence of equalities and inequalities in constraints, optimization problem is not an easy task. Many techniques have been proposed to handle such constraints. Penalty functions are one of the well-known approaches to handle constraints. Penalty functions, in spite of their popularity, have certain limitations like there are too many parameters to be adjusted. It is too tough to identify or fix the parameter values in order to balance the penalty and objective functions. Further, the search process is comparatively slow, and there is no assurance of attaining the global optimal solution. Deb [11] modified these algorithms to overcome this limitation by giving the concept of parameter-free penalty functions, i.e., one attempt to solve an unrestrained minimization problem in a search space S using a modified objective function F such as

$$F(x) = \begin{cases} f(x) & \text{if } x \in S \\ f_{\mathrm{w}} + \sum_{z=1}^{p+q} g_z(x) & \text{if } x \notin S \end{cases} \tag{1}$$

where x are solutions obtained by approaches, f_{w} is the worst feasible solution in the population, p and q are the number of equality and inequality constraints, S is the set of feasible solutions, and g_z is the set of constraints.

3 Overview of SFLA

SLFA is a stochastic search algorithm based on evolution of memeplexes. In essence, SLFA contains the element of both the local search method of PSO and the concept of mixing information of the shuffled complex evolution. SFLA has also proved its efficacy in finding global solutions to several combinatorial optimization problems [10]. In SFLA, a set of frogs represents the population of possible solutions, which is partitioned into subsets called memeplexes. Different subsets are having frogs from different cultures, each frog carries out a local search, and the position of the worst frog is modified or updated so that the frogs can move toward optimization. When each subset evolves through the fixed number of generations or memetic evolution steps, the ideas hold by the frogs within the subset are passed among subsets through shuffling process. This process of local search and shuffling of information continues until the termination criterion is satisfied.

There are four steps in SFLA:

A. *Initialization Process*
The initialization of a set of frogs (solutions) is similar to initialization process of other stochastic techniques, i.e., using Eq. (2). The population of frogs (P) be represented by $X_i = (x_{i1}, x_{i2}, \ldots, x_{iS})$, and then, position of each frog is generated by

$$x_{ij} = \mathrm{lb}_j + \mathrm{rand}(0, 1) \times (\mathrm{ub}_j - \mathrm{lb}_j) \tag{2}$$

for $i = 1, 2, \ldots, P$ (*set of frogs*); $j = 1, 2, \ldots, S$ (*S-dimensional vector*) and lb_j and ub_j are the lower and upper bounds, respectively, for the dimension j.

B. *Sorting and Division Process*
The frogs, based on their fitness evaluations, are sorted in descending order. Then, the sorted population of P frogs is distributed into m subsets (memeplexes), and each subset holds n frogs such that $P = m \times n$. The distribution is done such that the frog with maximum fitness value will go into subset first, accordingly the next frog into second subset, and so on. Then, X_{b} (best) and X_{w} (worst) individuals in each subset are determined.

C. *Local Search Process*
Worst individual position is improved using Eqs. (3) and (4):

$$D_i = \text{rand}(0,1) \times (X_b - X_w) \tag{3}$$

$$X_w = X_w + D_i; \quad -D_{max} \leq D_i \leq D_{max} \tag{4}$$

where i = 1, 2, …, N_{gen}; D is the movement of a frog, whereas D_{max} represents the maximum permissible movement of a frog in feasible domain; N_{gen} is maximum generation of evolution in each subset. The old frog is replaced if the evolution produces the better solution, or else X_b is replaced by X_g (optimal solution). If no improvement is observed, then a random frog is generated and replaces the old frog. This process of evolution continues till the termination criterion met.

D. *Shuffling Process*
The frogs are again shuffled and sorted to complete the round of evolution. Again follow the same four steps until the termination condition is met.

4 Accelerated-SFLA

Each frog in its memeplex explores the solution space locally, and then, all the memeplexes are shuffled and again divided into new subset of memeplexes. This information exchanging between memeplexes results in optimal search. As it can be analyzed from Eq. (3), when the difference between the position of the best frog (X_b) and the worst frog (X_w) decreases, the perturbation decreases on the position of the worst frog.

Thus, the search process might stagnate and lead to premature convergence. To avoid such incident, we have modified the local searching process in SFLA.

Searching mechanism for the worst frog is accelerated by embedding scalar factor component in improving the position of the worst frog. The modified equation is given below:

$$D_i = \text{rand}(0,1) \times \text{SF}(X_b - X_w) \tag{5}$$

SF is a scaling factor that controls the amplification or length of exploration of ($X_b - X_w$) vector. SF is computed using golden section search [12] and is given in Fig. 1. Two intermediate points SF_i^1 and SF_i^2 are generated using Eqs. (6) and (7). $\frac{1+\sqrt{5}}{2}$ is the golden ratio. The fitness values of $f(\text{SF}_i^1)$ and $f(\text{SF}_i^2)$ are evaluated and compared. If the fitness of $f(\text{SF}_i^1) < f(\text{SF}_i^2)$, then $\alpha = \text{SF}_i^1$.

If the resulting value falls outside the acceptable range for parameter j, it is set to the corresponding extreme value in that range. The pseudo-code of Accelerated-SFLA is given below:

Fig. 1 SF computation

$$SF_i^1 = \alpha - \frac{\beta - \alpha}{\frac{1+\sqrt{5}}{2}} \tag{6}$$

$$SF_i^2 = \beta + \frac{\beta - \alpha}{\frac{1+\sqrt{5}}{2}} \tag{7}$$

where $\alpha = 0.5$ and $\beta = 1.5$

Following steps are followed to compute SF:

Step 1: Evaluate $f(SF_i^1)$ and $f(SF_i^2)$

Step 2: If $f(SF_i^1) < f(SF_i^2)$ Then

Step 3: $\alpha = SF_i^1$ Else

Step 4 : $\beta = SF_i^2$

Begin:

Initialize the random population of frogs P [using Eq. (2)];
Evaluate the fitness of each frog [using Eq. (1)];
Sort the population of frogs (P) based on their fitness function value;
Distribute the population of frogs (P) into m memeplexes;

For each m;

X_b (best frog) and X_w (worst frog) are identified;
Update the position of the worst frog using Eqs. (5) and (4);
Repeat until the fix number of iterations;

End;

Evolved memeplexes are combined;
Evaluate fitness using Eq. (1) and arrange the population of frogs (based on their fitness value) in descending order.
Repeat till the termination criterion is true;

End;

Impact of the proposal: This process widens the searching area and balances local and global searching capabilities. Initially, it explores and then converges toward the optimal solutions with the process of combination and shuffling.

5 Optimal Selection of Processes

This problem has been taken from Floudas [13]. There are three processes (P_1, P_2, and P_3) in a company (Fig. 2) that are used to produce a chemical C. Process 1 uses B as a raw material to produce C. However, B can be either produced via two

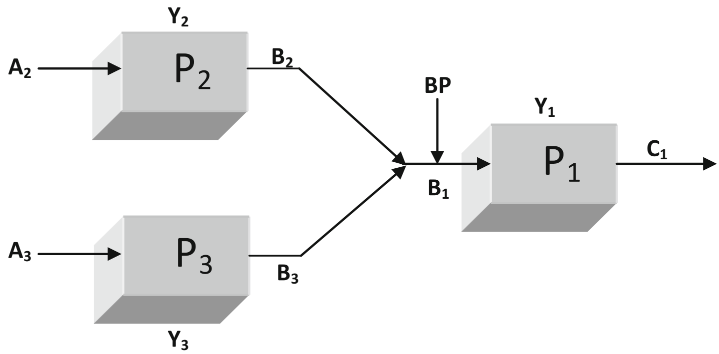

Fig. 2 Processes to produce a chemical C

processes (P_2 or P_3), or purchased from other producer. Chemical A is used as a raw material in processes P_2 and P_3. The related data with specifications (nonlinear I/O relationship) are presented in Table 1.

The objective of the problem is to select the appropriate processes (based on their level of production) to maximize the profit.

Processes P_1 and P_2 consume A_2 and A_3 amounts of chemical A. As a result, P_1 and P_2 produce B_2 and B_3 amounts of B. BP is the quantity of B purchased from some external entity. P_1 produces C_1 amount of C. The existence of the three processes is defined by 0–1 variables (Y_1, Y_2, and Y_3).

Table 1 Problem data

Conversions	P_1	$C = 0.9B$	
	P_2	$B = \ln(1 + A)$	
	P_3	$B = 1.2 \ln(1 + A)$	
		(A, B, and C are in ton/h)	
Maximum capacity	P_1	2 ton/h of C	
	P_2	4 ton/h of B	
	P_3	5 ton/h of B	
Demand of C	1 ton/h maximum		
Prices	A	\$1,800/ton	
	B	\$7,000/ton	
	C	\$13,000/ton	
Costs		*Fixed* (10^3 \$/ton)	*Variable* (10^3 \$/ton of product)
	P_1	3.5	2
	P_2	1	1
	P_3	1.5	1.2

$$\text{Max Profit } F = 11C_1 - 3.5Y_1 - Y_2 - B_2 - 1.5Y_3 - 1.2B_3 - 7BP - 1.8A_2 - 1.8A_3$$

w.r.t. the constraints:

- Conversion
  $C_1 = 0.9B_1$
  $B_2 = \ln(1 + A_2)$
  $B_3 = 1.2\ln(1 + A_3)$
- Mass balance for B
  $B_1 = B_2 + B_3 + BP$

The applied limits and specifications are as follows:

- Condition for non-negativity continuous variables
  $A_2, A_3, B_1, B_2, B_3, C_1 \geq 0$
- Integer constraints:
  $Y_1, Y_2, Y_3 = 0$ or 1
- Maximum demand for C
  $C_1 \leq 1$
- Plant capacity limit
  $B_2 \leq 4Y_2$
  $B_3 \leq 5Y_3$
  $C_1 \leq 2Y_1$

Finally, for the objective function, the terms for the profit PR expressed in $\$10^3$/h are given as follows:

1. Revenue generated from sales of product $C = 13C$
2. Expense incurred in purchasing chemical $B = 7BP$
3. Expense incurred in purchasing chemical A:
   $1.8A_2 + 1.8A_3A + 1.8A_2 + 1.8A_3$
4. Preset (fixed) cost for the P_1, P_2, and P_3:
   $3.5Y_1 + 2C_1 + Y_2 + B_2 + 1.5Y_3 + 1.2B_3$

6 Experimental Setup and Results

The above-stated problem is simulated on Deb C++ with the following parameters.

- All experiments were repeated 25 times independently with 24,000 objective function evaluations for each problem.
- Population size of frogs is fixed to 100.
- Number of function evaluations (*NFEs*) fixed to 5,000.

- m (number of memeplexes) = 10.
- n (number of iterations evolves in each memeplexes) = 10.
- N_{gen} = 10.
- D_{max} = 100 % of variable range.
- Binary variables and integers are handled by rounding of the decision variables to nearest integer [14].

The optimal solution for the problem of selecting process is achieved by both SFLA and Accelerated-SFLA with 100 % success rate (both are able to reach the optimum solution in all 25 trial runs). The difference lies in the time and NFE taken to achieve the optimal values. Accelerated-SFLA took only 1.5 s and 1,095 NFEs, whereas SFLA took 2 s and 1,978 NFEs to reach the optimal solution. The simulated results show that the proposal is 44 % faster than SFLA. The results for the optimal selection of processes are as follows:

$Y_1 = Y_3 = 1$ and $Y_2 = 0$; hence, processes P_1 and P_3 are selected to maximum profit with the following details:

$A_2 = 0.0000$; $A_3 = 1.5201$; $B_1 = 1.1110$; $B_2 = 0.0000$; $B_3 = 1.1114$; $BP = 0.0000$; $C_1 = 1.0000$. The subtotal of fix cost will be 5 $(Y_1 + Y_3)$, operating cost will be 3.333 $(C_1 + B_2 + B_3)$, and the raw material cost is 2.744 $(A_2 + A_3 + BP)$; hence, the total cost will be 11.077, and the net profit will be $F = \mathbf{1.923}$ (total revenue − total cost). The success rate of both the algorithms is 100 %.

7 Conclusions

The paper proposes Accelerated-SFLA variant of SFLA, a memetic algorithm based on the improvement done in shuffled leap frog algorithm. A simple modification is proposed in SFLA. In the proposal, scaling factor based on golden section in searching process of SFLA is introduced. The aim of the study is to accelerate the convergence speed of SFLA and preventing it from trapping into local optima. This kind of hybridization seems to be very efficient in solving computational optimization problems. We have tested the efficacy of the proposal on an optimal selection of processes in chemical manufacturing company.

In the future, we will try to further investigate the proposal and enhance the employment of the proposal on large-scale optimization problems.

References

1. Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence Through Simulated Evolution. Wiley, New York (1966)
2. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, Reading (1989)

3. Rechenberg, I.: Evolutions strategie: optimierung technischer systeme nach prinzipien der biologischen evolution. Ph.D. Thesis, Technical University of Berlin, Department of Process Engineering (1971)
4. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceeding of IEEE International Conference on Neural Networks, pp. 1942–1948, Perth, Australia. IEEE Service Center, Piscataway, NJ (1995)
5. Price, K., Storn, R.: Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report, International Computer Science Institute, Berkley (1995)
6. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst. Mag. **22**(3), 52–67 (2002)
7. Karaboga, D.: An idea based on bee swarm for numerical optimization. Technical Report, TR-06, Erciyes University Engineering Faculty, Computer Engineering Department (2005)
8. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim. **39**, 459–471 (2007)
9. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern B **26**(1), 29–41 (1996)
10. Eusuff, M., Lansey, K.E.: Optimization of water distribution network design using the shuffled frog leaping algorithm. Water Resour. Plan. Manage. **129**(3), 210–225 (2003)
11. Deb, K.: An efficient constraint handling method for genetic algorithms. Comput. Methods Appl. Mech. Eng. **186**, 311–338 (2000)
12. Kiefer, J.: Sequential minimax search for a maximum. Proc. Am. Math. Soc. **4**, 502–506 (1953)
13. Floudas, C.: Nonlinear and Mixed-Integer Optimization. Oxford University Press, New York (1995)
14. Srinivas, M., Rangaiah, G.P.: Differential evolution with tabu list for solving nonlinear and mixed-integer nonlinear programming problems. Ind. Eng. Chem. Res. **46**(22), 7126–7135 (2007)