Solving the Tension/Compression Spring Design Problem by an Improved Firefly Algorithm

Yuksel Celik [1] [0000-0002-7117-9736] and Hakan Kutucu [1] [0000-0001-7144-7246]

[1] Karabuk University, Department of Computer Engineering, Karabuk, Turkey
yukselcelik@karabuk.edu.tr

Abstract. Since the 1970s, nature inspired meta-heuristic algorithms have become increasingly popular. These algorithms include a set of algorithmic concepts that can be used to identify heuristic methods that are used for a wide range of different tasks. The use of meta-heuristics greatly increases the possibility of finding a qualitative solution for complex, combinatorial optimization problems in a reasonable time. The most popular nature inspired meta-heuristics are those methods representing successful animal and micro-organism swarm behaviors. Firefly Algorithm (FA) is a recent one of such meta-heuristic algorithms It is based on a swarm intelligence and inspired by the social behaviors of fireflies. In this paper, we adapt the neighborhood method to FA and propose an improved firefly algorithm (IFA) to solve a well-known engineering problem, the so-called Tension/Compression Spring Design. We test the proposed IFA on this problem and compare the results with those obtained by some other meta-heuristics. The experimental modeling shows that the proposed IFA is competitive and improves the quality of solutions for the aforementioned engineering design problem.

Keywords: Firefly Algorithm (FA), Tension/Compression Spring Design, Swarm Intelligence, Metaheuristic

1 Introduction

The goal of heuristic or metaheuristic algorithms for solving combinatorial optimization problems is to find an optimal value under specified constraints. A few general approaches to optimization are available; analytical methods, numerical methods, heuristic methods. Numerical optimization methods rely on computation of gradients in determine the solution with maximum fitness. The standard assumptions in optimization is a multimodal search space by specific techniques that have additional constraints imposed on the search space such as linearity of constraints and objective function in linear programming and assumption of discrete variables in combinatorial optimization [1]. If the search space is non-convex, then an optimal solution cannot be guaranteed. The problem with numerical optimization technique is the locality of optima i.e. the solution depends on the starting solution wherein the gradients force the optimum to a local optimum even though a better solution would exist elsewhere in the search space. Stochastic optimization techniques rely on random perturbations to the solution space and are more adept at preventing a solution from being trapped in a local optimum for non-convex problems [2]. The obvious problem with stochastic optimization techniques is that on an average it requires a lot more computations of alternate solutions compared to gradient-based techniques. Many practical problems of importance are not convex and difficult to solve in reasonable amount of time; consequently, heuristics are deployed to make the solution feasible in a reasonable amount of time. Heuristics is a way of approximation of solution by trading optimality for speed. Meta-heuristic algorithms are generic algorithmic frameworks that are often rooted in natural processes, such as simulated annealing, genetic algorithm and behavior of insects. Meta-heuristic algorithms cannot guarantee a global optimum however they are able to provide good solutions in reasonable timeframe and are typically able to avoid local optima [3].

The work is organized as follows. We first give in Section 2 the formulation of the tension/compression spring design problem. In Section 3, we review firefly algorithm and propose an improved firefly algorithm. We present in Section 4 some computational results for the problem and finally, give some concluding remarks in Section 5.

1 The Tension/Compression Spring Design

The Tension/Compression Spring Design problem (TCSD) illustrated in Fig 1 is a continuous constrained problem. The problem is to minimize the volume V of a coil spring under a constant tension/compression load. The problem consists of three design variables. These are:

- the number of spring's active coils $P = x_1 \in [2, 15]$;
- the diameter of the winding $D = x_2 \in [0.25, 1.3]$;
- the diameter of the wire $d = x_3 \in [0.05, 2]$.

The mathematical formulation of the TCSD problem is as follows: [4]

$$\min \quad f(x) = (x_3 + 2)x_2 x_1^2 \tag{1}$$

subject to

$$\begin{aligned} q_1(x) &= 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0 \\ \mathbf{q_2(x)} &= \frac{\mathbf{4x_2^2 - x_1x_2}}{\mathbf{12566(x_2x_1^3 - x_1^4)}} + \frac{\mathbf{1}}{\mathbf{5108x_1^2}} - \mathbf{1} \le \mathbf{0} \\ q_3(x) &= 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0 \\ q_4(x) &= \frac{x_2 + x_1}{1.5} - 1 \le 0 \end{aligned} \tag{2}$$

The design problem upper and lower bounds variables are

$$2 \leq x_1 \leq 15,\ 0.25 \leq x_2 \leq 1.3,\ 0.05 \leq x_3 \leq 2 \qquad (3)$$

This is a convex optimization problem and the closed form optimum solution of the problem is f(X) =0.0126652327883 for X = [x_1, x_2, x_3] = [0.051689156131, 0.356720026419, 11.288831695483].

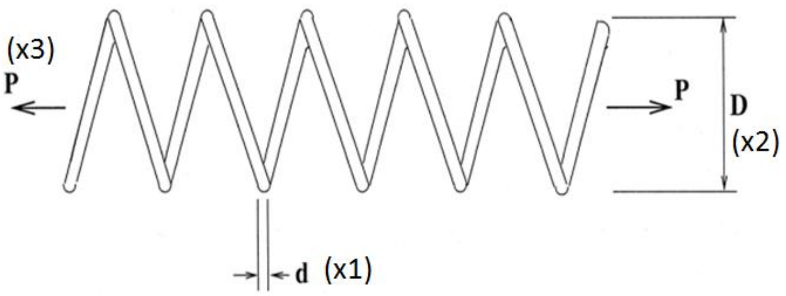

Fig 1. Schematic of tension/compression spring design problem [5].

4 Firefly Optimization Algorithm

Firefly algorithm (FA) is a recent nature inspired approach based on swarm intelligence and inspired by the social behaviors of fireflies in tropical zones for solving optimization problems developed by Yang in 2008 [2]. This algorithm is based on the phenomenon of bioluminescence. The light produced by the special photogenic bodies acts as a communication channel. The main task of flashing light is to attract a partner.

The mathematical form of the algorithm is based on the following assumptions. First of all, all fireflies are unisex and therefore can communicate with anyone else [4]. The attractiveness, in this case, is determined by the level of brightness of the individual. Brighter light attracts the others. [5]. This is performed for any binary combination of fireflies in the population, on every algorithm’s iteration. The brightness of a firefly is determined by the objective function of the problem [6].

Optimization algorithms typically employ either global or local search methods. Global search methods aim to find the best solution in the entire search space often by seeding multiple initial solutions in the search space and randomly perturbing the solution. Local search methods typically start from a single initial solution and improve the results of a global search by computing gradients in the local search space to reach a local minima/maxima [7]. Classical firefly algorithm aims to find an optimal solution. In this paper, we introduce local search methods with the best global solution to improve the results of the classical firefly algorithm. By deploying local search methods, we borrow a random element from the best global solution and process it with some random variables to see if the global solution can be enhanced any further as described in equation (4) below:

$$
\begin{aligned}
&\alpha = rand(\beta) \quad 0 \leq \beta \leq 1 \\
&j = rand(\Omega) \quad 0 \leq \Omega \leq m \\
&X_{ij} = x_{ij} + \left(x_{ij} - b_j\right)\alpha
\end{aligned} \tag{4}
$$

where m is the dimension of the solution for each firefly, a value in the current solution set (population) is defined by x_i ; j denotes the randomly selected item from the solution set. The best solution (global minimum) is represented by b.. The main feature of x_{ij}, is that, the b_i is multiplied by a random value α in [0,1]. The improvement of the algorithm is that the obtained result is added to the overall solution. The pseudo code of the proposed method is given as Algorithm 1.

Algorithm 1: The Improved Firefly Algorithm

Objective function f(x), $x=(x_1,\ldots,x_d)^T$
Generate initial population of fireflies x_i $(i=1,2,\ldots,n)$
Light intensity I_i *at* x_i *is determined by* $f(x_i)$
Define light absorption coefficient γ
while *(t<MaxGeneration)*
  for *i=1:n all n fireflies*
    for *j=1: n all n fireflies (inner loop)*
      if $(I_i<I_j)$*, Move firefly i towards j;* **end if**
      Vary attractiveness with distance r via exp [-γr]
      Evaluate new solutions and update light intensity
    end for *j*
  end for *i*
  for *i=1:n all n fireflies*
    Improved select random a value by Equation (4)
  end
  for *i=1:n all n fireflies*
    rank the fireflies and find the current global best (b)
end while
Post process results and visually observe

---

4 Computational Results

The experimental test is carried out with the following parameters for both classical FA and the proposed IFA.

- Number of iterations = 20.000
- Number of fireflies = 20
- Randomness factor, α = 0.5

- Attractiveness of a firefly, β = 0.2
- Absorption coefficient, Ω = 1

The worst ,best and also the average results by the parameters obtained in the computational tests to solve TCSD problem are shown in Table 1. Moreover, the number of fireflies, the worst, best, average results and standard deviations obtained by Firefly Algorithm in literature as well as firefly algorithm and improved firefly algorithm algorithms used in this paper are presented in Table 2.

Table 1. Comparison of computational tests between classical FA and improved FA

Algorithm		$d(x_1)$	$D(x_2)$	$P(x_3)$	Cost F(x)	Error Rate%
FA	Best	0.051622816	0.355101557	11.38560208	0.012667003	0.014
	Mean	-	-	-	0.012811505	1.155
	Worst	0.062460428	0.675464982	3.545398454	0.014613206	15.380
IFA	Best	0.051604177	0.354674682	11.41061417	0.012666265	0.008
	Mean	-	-	-	0.012755131	0.710
	Worst	0.058164347	0.533417917	5.414915162	0.013380967	5.651

Table 2. Comparison of computational tests between FAs

Researcher	Algorithm	Best	Mean	Worst	SD	No. fireflies
Parsi [4]	MFA	0.01269	0.01513	0.02320	0.00147	15
Present Work	FA	0.01269251	0.01281114	0.014613206	0.0002348	40
Present Work	IFA	0.01266626	0.01275513	0.013380967	0.0001205	40

Tables 1 and 2 demonstrate that IFA yields better results in terms of all indicators. In addition, an experimental comparison of the work of an improved IFA algorithm with fifteen existing ones was carried out. The results of this comparison are given in Table 3.

Table 3. Comparison of computational tests between several FAs

Researcher(s)	Algorithm	$d(x_1)$	$D(x_2)$	$P(x_3)$	Cost F(x)
Coello 2000 [5]	GA	0.05148	0.351661	11.632201	0.01270478
Ray and Liew 2003 [6]	-	0.05216022	0.3681587	10.648442	0.012669249
Raj et al. 2005 [7]	ADE	0.053862	0.41128365	8.6843798	0.0127484
Hedar et al 2006 [8]	FSA	0.0517425	0.35800478	11.213907	0.012665285
He and Wang 2007 [9]	CPSO	0.051728	0.357644	11.244543	0.0126747
Zhang et al. 2008 [10]	DSS-MDE	0.05168906	0.35671775	11.288965	0.012665233
Montes et al. 2008 [11]	ES	0.051643	0.35536	11.397926	0.012698
Shen et al. 2009 [12]	iGSO	0.05169	0.35677	11.28617	0.01267
Omran et al. 2009 [13]	CODEQ	0.05168375	0.35658984	11.296471	0.012665238
Coelho 2010 [14]	QPSO	0.051515	0.352529	11.538862	0.012665
Gandomi et al. 2013 [15]	BAT	0.05169	0.35673	11.2885	0.01267

Akay and Karaboga 2012 [16]	ABC	0.051749	0.358179	11.203763	0.012665
Garg 2014 [17]	ABC	0.05168916	0.35672003	11.288831	0.012665233
Modified FA [4]	MFA	0.05173	0.3577	11.2595	0.01269
Present work, IFA	IFA	0.05160418	0.35467468	11.410614	0.012666265

As it can be seen from Table 3, proposed IFA finds better results than eight out of 8 existing ones algorithms in comparison and gives worse results than the rest.

5 Discussion & Conclusion

At the present time, optimization algorithms are ubiquitously used to solve problems in many domains, many engineering finance and operations research. Heuristic and meta-heuristic optimization algorithms are commonly used where the search space is complex. Meta-heuristics can be a general algorithmic wrapper that can be applied to solving various optimization tasks with a relatively small number of modifications to make it adapted to a particular problem. Its application greatly enhances the possibility of finding a qualitative solution for complex, topical combinatorial optimization problems in a reasonable time. Firefly algorithm is a recent swarm intelligence meta-heuristic algorithm. In this paper, a neighborhood method is integrated to the current FA and the new algorithm IFA is proposed. IFA integrates the stochastic randomness of the classical FA with the local search to maximize the outcome. To compare the performance of FA and IFA, we tested them on two well-known engineering design problem with a closed form solution.

A direct comparison of FA and IFA shows that IFA performs better than FA. We also compared IFA with the results of other optimization algorithms. Compression/Tension Spring Design problem is compared with 15 existing optimization algorithm. IFA is worse than seven and better than eight algorithms. Often multiple algorithms need to be tried to get the most optimum value for a solution and we believe that IFA would be a good algorithm in the mix for maximizing the returns.

Acknowledgement

This investigation was supported by Research Fund of the Karabuk University. Project Number: KBUBAP-18-DS-174.

References

1. Lemonge, A. C., Barbosa, H. J., Borges, C. C., Silva, F. B.: Constrained Optimization Problems in Mechanical Engineering Design Using a Real-Coded Steady-State Genetic Algorithm. Mecánica Computacional, 29, 9287-9303 (2010).
2. Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press, (2008).

3. Mladenović, N., Hansen, P.: Variable neighborhood search. Computers & Operations Research 24.11 (1997): 1097-1100.
4. Kazemzadeh-Parsi, M. J.: A Modified Firefly Algorithm for Engineering Design Optimization Problems. IJST, Transactions of Mechanical Engineering 38(M2), 403-421 (2014).
5. Carlos, A., Coello, C.: Use of a self-adaptive penalty approach for engineering optimization problems. Computers in Industry 41(2), 113–127 (2000)
6. Ray, T., Liew. K. M.: Society and Civilization: An Optimization Algorithm Based on the Simulation of Social Behavior. IEEE Transactions on Evolutionary Computation 7, 386-396 (2003).
7. Raj, K. H., Sharma, R. S., Mishra, G. S., Dua, A., Patvardhan, C.: An Evolutionary Computational Technique for Constrained Optimisation in Engineering Design. Journal of the Institution of Engineers India Part Me Mechanical Engineering Division 86,121-128 (2005).
8. Hedar, A.-R., Fukushima, M: Derivative-Free Filter Simulated Annealing Method for Constrained Continuous Global Optimization. Journal of Global Optimization 35(4), 521-549 (2006).
9. He, Q., Wang, L.: An Effective Co-Evolutionary Particle Swarm Optimization for Constrained Engineering Design Problems. Engineering Applications of Artificial Intelligence 20, 89-99 (2007).
10. Zhang, M., Luo, W., Wang, X.: Differential evolution with dynamic stochastic selection for constrained optimization. Information Sciences 178(15), 3043–3074 (2008).
11. Montes, E. M., Coello C. A. C.: An Empirical Study About The Usefulness of Evolution Strategies to Solve Constrained Optimization Problems. International Journal of General Systems 37, 443-473 (2008).
12. Shen, H., Zhu, Y., Niu, B., Wu, G. H.: An Improved Group Search Optimizer for Mechanical Design Optimization Problems. Progress in Natural Science 19, 91-97 (2009).
13. Omran, M. G. H., Salman, A.: Constrained optimization using CODEQ, Chaos. Solitons & Fractals 42, 662-668 (2009).
14. Coelho, L. S.: Gaussian Quantum-Behaved Particle Swarm Optimization Approaches for Constrained Engineering Design Problems. Expert Systems with Applications 37, 1676-1683 (2010).
15. Gandomi, A. H., Yang, X. S., Alavi, A. H. Talatahari, S.: Bat Algorithm for Constrained Optimization Tasks."Neural Computing and Applications 22(6), 1239-1255 (2013).
16. Akay, B., Karaboga. D.: Artificial bee colony algorithm for large-scale problems and engineering design optimization. Journal of Intelligent Manufacturing 23(4), 1001-1014 (2012).
17. Garg, H.: Solving Structural Engineering Design Optimization Problems Using an Artificial Bee Colony Algorithm. Journal of Industrial And Management Optimization 10, 777-794 (2014).