Pareto based Bat Algorithm for Multi Objectives Multiple Constraints Optimization in GMPLS Networks

Mohsin Masood[1], Mohamed Mostafa Fouad[2], and Ivan Glesk[3]

[1] Electronics and Electrical Engineering Department, University of Strathclyde, Glasgow UK,
[2] Arab Academy for Science, Technology and Maritime transport, Cairo, Egypt,
[3] Electronics and Electrical Engineering Department, University of Strathclyde, Glasgow UK,
mohsin.masood@strath.ac.uk, mohamed_mostafa@aast.edu,
ivan.glesk@strath.ac.uk

Abstract. Modern communication networks offer advance and diverse applications, which require huge usage of network resources while providing quality of services to the users. Advance communication is based on multiple switched networks that cannot be handle by traditional IP (internet protocol) networks. GMPLS (Generalized multiprotocol label switched) networks, an advance version of MPLS (multiprotocol label switched networks), are introduced for multiple switched networks. Traffic engineering in GMPLS networks ensures traffic movement on multiple paths. Optimal path (s) computation can be dependent on multiple objectives with multiple constraints. From optimization prospective, it is an NP (non-deterministic polynomial-time) hard optimization problem, to compute optimal paths based on multiple objectives having multiple constraints. The paper proposed a metaheuristic Pareto based Bat algorithm, which uses two objective functions; routing costs and load balancing costs to compute the optimal path (s) as an optimal solution for traffic engineering in MPLS/GMPLS networks. The proposed algorithm has implemented on different number of nodes in MPLS/GMPLS networks, to analysis the algorithm performance.

Keywords: Bat Algorithm • GMPLS Networks • Optimization • Particle Swarm Optimization • Routing Protocols • Traffic Engineering

1 Introduction

Advance telecommunication applications require a massive movement of data flow in the network, which causes various network problems such as congestion, packet delays, high utilization of network resources and bandwidth use [1]. To address these challenges, traffic engineering concept was introduced in the networks. Traffic engineering (TE), is used to optimize the network performance by ensuring massive data flow in the network with minimum utilization of network resources and with performance efficiency. TE can be applied to any range (from local area to wide area) of multiple switched networks. Recently, multiple path traffic engineering has been introduced as appealing approach to handle diverse applications with increased network performance [2]. Multiple path routing is the technique of traffic management, which balances large amount of traffic into multiple routes. It shows significant results compare to traditional routing techniques, which relies on forwarding traffic over shortest path routes. Multi-path traffic engineering optimizes network utilization and address various network problems effectively such as packet loss, congestion and link loads. Multipath routing traffic engineering requires algorithms which can compute optimal routes, having multiple objectives and constraints [3, 4]. In networking, it is known as multi-objective multiple constrained (MCOP) based optimization problem, which is an NP hard. This paper provides a metaheuristic pareto based bat algorithm, which will provide optimal solutions as paths for MCOP in communication networks.

Traditional IP networks has various limitation while using traffic engineering, which affects traffic engineering performance. Therefore, to improve network capabilities, multiprotocol label switched (MPLS) networks are introduced, which are based on label switched network. Furthermore, Generalized multiprotocol label switched (GMPLS) network is introduced, which is the extended version of MPLS networks. GMPLS networks provide the set of protocols which enable forwarding of traffic over multiple switched networks such as packet, time, wavelength and fiber switching networks [5, 6].

The proposed algorithm considers two objective functions; routing costs and load balancing costs with constraints and the task is to find the optimal paths (as solutions) in MPLS/ GMPLS networks.

2 MPLS/ GMPLS Networks

MPLS/ GMPLS uses labels over the packets and forward them in the network from source to destination routers. Routing protocols play an important role for label switching and forwarding of packets in MPLS/ GMPLS networks [6]. In MPLS/ GMPLS domain, a virtual connection is established known as label switched path (LSP) for forwarding user data. The establishment of the label switched paths (LSP) is done with the help of interior gateway routing protocols such as open shortest path first (OSPF) and intermediate system-to-intermediate system (IS-IS) protocols [7]. When the packet arrives from the source, the router connected to source site label the packet and forward to its next router towards the destination. Each intermediate router in the network lookup the label and forward the packet to the next routers in the network, unless the packet reaches to the router at destination site. The routers at source and destination site, are known as label edge routers (LER) while the routers, used for forwarding labelled packets, are known as label switched routers (LSR). Router connected to source site, which receives traffic request and take the initiative for label switched path (LSP) is known as ingress router. While the label edge router (LER) which is at destination side is known as egress router. Label switched path (LSP) develops between ingress and egress routers in MPLS/ GMPLS domain. Once the path or label switched path (LSP) has established, then the user data will forward from source to destination through label switched routers (LSRs) in the network. This label switching approach of MPLS/ GMPLS networks enhances network performance with minimum utilization of network resources compare to IP networks, where each router must look up the list of

IP addresses. Most of the service providers prefer GMPLS based routers for modern applications [7, 8].

3 Problem Evaluation

To provide the effective traffic engineering in MPLS/ GMPLS network and for handling massive amount of traffic flow, the techniques must be used which can enhance network performance and provide optimal solutions. In MPLS/ GMPLS networks, ingress receives a number of traffic requests, and the task is to find the number of optimal routes while considering multiple objectives and constraints. An algorithm can offer optimal paths as solutions for the given scenario. In the paper, we proposed pareto based bat algorithm, while considering two objective functions; routing costs and load balancing costs. The proposed algorithm will provide optimal solutions as paths having minimum routing costs and load balancing costs. The algorithm will be implemented on different number of nodes in MPLS/ GMPLS networks for analyzing network performance.

In the paper, we used notation for MPLS/ GMPLS networks as graph(G). The network / graph(G) is consist of number of routers and links, which are represented as; for routers set, vertices(V) is used and for links set, edges(L) is used. The graph with number of vertices and edges can be represent as $G = (V, L)$. The set of vertices (V) in the network is V = $\{v_1, v_2, v_3, \ldots, v_n)$ and links set is $L = \{l_1, l_2, l_3, \ldots, l_n\}$. The objective functions are explained as follow.

3.1 Total Routing Costs Objective Function

Service providers use specific link cost for per unit of data flow in MPLS/ GMPLS networks, which is described as follow [9, 10]:

$$R_{cost} = \sum T_{links} I_{traffic} \tag{1}$$

Where, R_{cost} represents the routing cost for a path. While T_{links} represents the connected links and $I_{traffic}$ is the i^{th} traffic over the path. The total routing costs objective function is mathematically described as follow [9, 10]:

$$1^{st} \text{ Objective Function} = \sum traffic \in T_{traffic} \sum R_{cost} \tag{2}$$

Where, *traffic* is member of all traffics set(T_{raffic}).

3.2 Total Load Balancing Costs Objective Function

The second objective function is to distribute the traffic evenly over multiple links, which is dependent on load balancing costs. Load balancing costs function consist of two parameters, known as link utilization(L_u) and link capacity(L_c). The load balancing function can be described as follow [9, 10]:

$$Load_{balancing} = \text{link utilization}(L_u) \,/\, \text{link capacity}(L_c) \tag{3}$$

In our experiments, the task of the proposed algorithm is to minimize the load balancing function. The mathematical expression for the total load balancing costs is given as follow [9, 10]:

$$2^{nd} \text{ Objective Function} = \min\left(\sum Load_{balancing}\right) \quad (4)$$

4 Proposed Algorithm

We proposed a metaheuristic algorithm to address the optimization problem in traffic engineering for MPLS/ GMPLS networks.

4.1 Pareto Based Bat Algorithm (PBA)

Bat algorithm is a mathematic bio-inspired technique introduced by X. Yang in 2010 [11], which is used for solving optimization problems in different applications. Bat algorithm is inspired by the bat technique for searching its prey in searching area. While searching for its prey, each bat periodically evaluates its searching as updated solutions with the given fitness function. The searching nature of bats dependent on echolocation parameters known as loudness(L_d) and pulse-rate(P_r). When the bat approaches towards its prey, the loudness(L_d) decreases while pulse-rate(P_r) increases [11,12]. In our paper, we modelled bat algorithm as Pareto based model, in which each bat will search for optimal solutions as minimum routing costs and minimum load balancing costs paths in n-dimension searching space. In bat algorithm, each (i^{th})bat is used as a candidate of searching optimal solution, where it updates its position(x_i^{ite}) and velocity(v_i^{ite}) in n-dimension searching space during each iteration, which is given as follow [11,12,13]:

$$freq_i = freq_{min} + \beta\ (freq_{max} + freq_{min}) \quad (5)$$

$$v_i^{ite} = v_i^{ite-1} + freq_i\ (x_i - x^{globalbest}) \quad (6)$$

$$x_i^{ite} = x_i + v_i^{ite} \quad (7)$$

Where *ite* represents the iterations used in the algorithm. $freq_i$ represents the initial frequency while $freq_{max}$ and $freq_{min}$ are the maximum and minimum frequencies, respectively. β is the random number within the range of 0 and 1. $x^{globalbest}$ is global best position of the i^{th} bat. The global best position($x^{globalbest}$) is accomplished by comparing all given solutions of n bats. Each bat, after updating its velocity(v_i^{ite}) and position(x_i^{ite}) takes a random walk for searching to achieve its local best solution based on the condition; *if rand > pulse-rate* (P_r), based on following [11,12,13]:

$$x_{i,\ best\text{-}local}^{ite} = x_i + \varepsilon < L_{d,Aveg} \quad (8)$$

where, $x_{i,\ best\text{-}local}^{ite}$ is used for local best position. ε is a random number, $\varepsilon \in [-1, 1]$. $L_{d,Aveg}$ represents the average loudness of the bats. During each iteration, bat updates its loudness(L_d) and pulse rate(P_r) value. If the bat is approaching to its optimal solution then the loudness(L_d) level will decrease while pulse-rate(P_r) level will increase, as given by following equations [11,12,13]:

$$L_{d,\ i}^{ite+1} = \alpha\ L_{d,\ i}^{ite} \quad (9)$$

$$P_{r,\ i}^{ite} = P_{r,\ i}[1 - e^{-\gamma t}] \quad (10)$$

Where, α and γ are constant values, set from the interval of [0, 1]. The pseudo code of the proposed pareto based bat algorithm is given in algorithm 1.

Algorithm 1. Pseudo code of Pareto based Bat Algorithm (PBA)

```
Routing costs objective function f_{x, routing} = [x_{r,1}, x_{r,2}, x_{r,3}, .... x_{r,n}]
Load balancing costs objective function f_{x, load} = [x_{l,1}, x_{l,2}, x_{l,3}, .... x_{l,n}]
Remove the links and routers from the matrix, after applying the constraints associated to routing costs
and load balancing costs functions
Initialize number of bats population
    At initial pulse-rate(P_r) and initial loudness(L_d), initialize pulse frequency(freq_i)
While (iterations < total number of iterations for routing costs function)
        Update frequency(freq_i) by adjusting maximum(freq_max) and minimum frequency(freq_min)
        Update bats position(x_i^ite) and velocities(v_i^ite) in the network (matrix)
        Apply the routing costs function constraints.
        Generate local best position of each i^th bat
 if (rand < Pulse-rate(P_r)
            Generate local optimal solution as a path having minimum routing costs
end if
        Generate random solutions (paths) in the matrix randomly
 if (rand < L_d & Present routing costs < Previous routing costs)
          Accept the new updated solution as optimal path
          Increase Pulse-rate(P_r) and decrease Loudness(L_d)
          Find the global best position(x^globalbest) of the i^th bat having optimal solution
end if
end While
Store the optimal solutions as paths having minimum routing costs
While (iterations < total number of iterations for Load balancing costs function)
          Update (freq_i) by adjusting freq_max and freq_min
          Update bats position(x_i^ite) and velocities(v_i^ite) in the network (matrix)
          Apply the load balancing costs function constraints.
          Generate local best position of each i^th bat
if (rand < Pulse-rate(P_r)
          Generate local optimal solution as a path having minimum load balancing costs
end if
         Generate random solutions (paths) in the matrix randomly
 if (rand < L_d & Present load balancing costs < Previous load balancing costs)
          Accept the new updated solution as optimal path
          Increase Pulse-rate(P_r) and decrease Loudness(L_d)
          Find the global best position(x^globalbest) of the i^th bat having optimal solution
         Store the optimal solutions as paths having minimum load balancing costs
end if
end While
Store the optimal solutions as paths having minimum routing costs
Generate Pareto archive of paths with minimum routing and load balancing costs
```

5 Experimental Setup

Throughout the experiments the algorithm had been implemented as pareto based bat algorithm using MATLAB tool. For analyzing performance analysis of the proposed algorithm, it was implemented over various scales of nodes in MPLS/GMPLS networks such as 80, 90 and 100 nodes, as presented in Fig. 1, 2 and 3 respectively. Furthermore, the proposed algorithm has been modified through changing its parameters and then divide them into five cases, entitled as PBA-1 (Pareto based bat algorithm), PBA-2, PBA-3, PBA-4 and PBA-5. In each PBA case, we changed the maximum loudness value ($L_{d,\,max}$) and minimum loudness value ($L_{d,\,min}$), which updates the loudness(L_d) value during iteration. In PBA-1; $L_{d,\,max} = 5$, in PBA-2; $L_{d,\,max} = 12$, in PBA-3; $L_{d,\,max} = 18$, in PBA-4; $L_{d,\,max} = 24$ and in PBA-5; $L_{d,\,max} = 30$, while $L_{d,\,min} = 0$ for all PBA cases. Pareto based optimal solutions of two objective functions simulated results are shown in Fig.1, Fig. 2 and Fig. 3, with Pareto frontiers. The paper highlighted the non-dominated solution of both objective functions with different signs and connect then with lines to draw a Pareto front for each case.

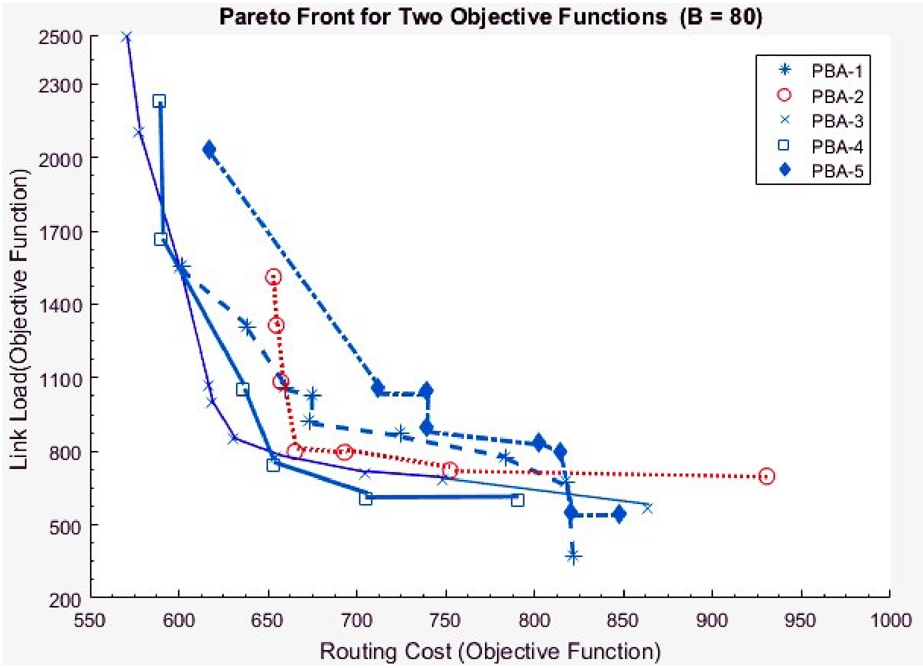

Fig. 1. Pareto front of routing costs and load balancing costs function for nodes (B) = 80

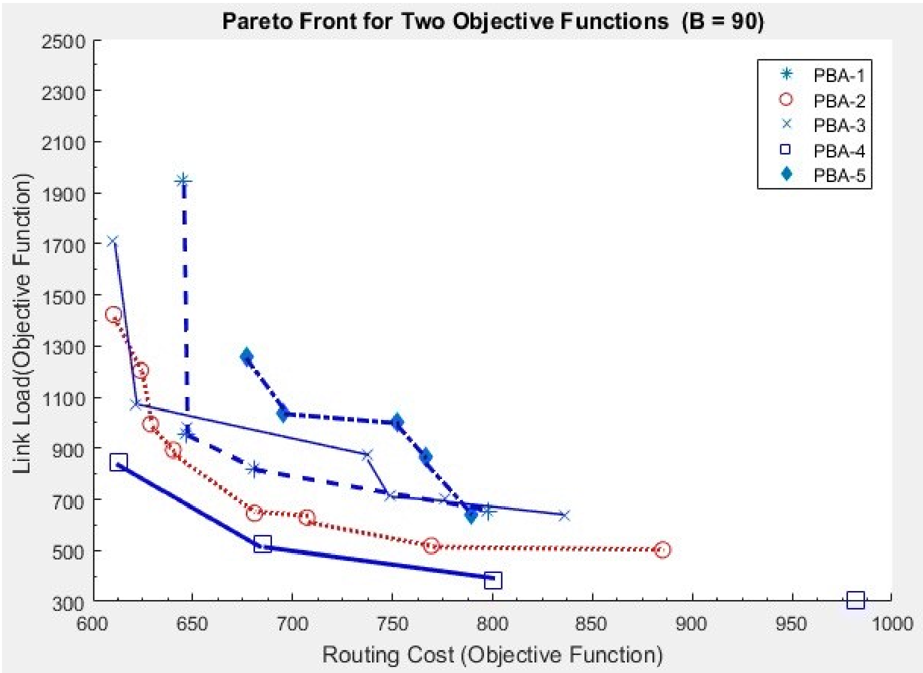

Fig. 2. Pareto front of routing costs and load balancing costs function for nodes (B) = 90

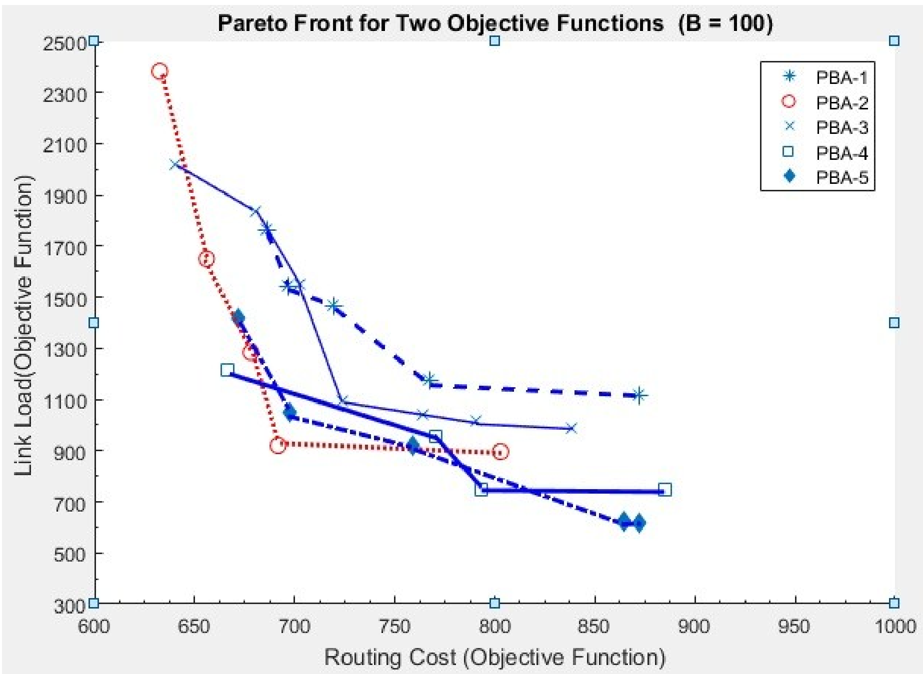

Fig. 3. Pareto front of routing costs and load balancing costs function for nodes (B) = 100

6 Result Analysis

The figures represent the optimal solutions (paths) for two objective functions, where each solution represents the minimum routing costs and load balancing costs. For example, in Fig. 1, for PBA-1 case in 80 nodes network, the Pareto curve shows the optimal solutions with highlighted points which are connected lines. It is also noticed that when routing costs increase, the load balancing costs decreases and vice versa. Routing costs and load balancing costs are minimum/ optimal values (as shown in Pareto front) in 80 nodes network compare to 90 and 100 nodes networks for all PBA scenarios. Similarly, 90 nodes network has better results compare to 100 nodes networks. These findings are same for PBA-2, PBA-3, PBA-4 and PBA-5 for 80, 90 and 100 MPLS/GMPLS nodes networks, as shown in all figures.

For comparative analysis, the proposed Pareto based BAT algorithm(PBA) is compared with particle swarm optimization algorithm (PSO). Each algorithm is implemented on 100 nodes GMPLS network. The parameters used for comparison are: minimum routing costs, minimum load balancing costs, mean values and standard deviation. Both algorithms run for 100 times to collect data and then analyze with mentioned parameters, which is presented in table 1. The results in table 1 show that proposed bat algorithm (PBA) has minimum or optimum values for both routing costs and load balancing costs function, in addition to reduction other measuring's parameters. For example, PBA algorithm has minimum routing costs value of 462 compare to PSO routing costs value of 1169, which means that PBA algorithm achieved optimum value compared to PSO algorithm. Similarly, for mean values and standard deviation values; PBA algorithm achieved minimum (optimum) values compare to PSO algorithm obtained values, which shows that PBA algorithm obtains optimum values as a mean with a small standard deviation from the mean. This may have related to the adjustment of the frequency of the bat based on how far is the object.

Table 1. Comparative study table between proposed Pareto BAT(PBA) and PSO

	100 Nodes MPLS/ GMPLS Network					
	Minimum Routing Costs	**Mean (Routing Costs)**	**Standard Deviation (Routing Costs**	**Minimum Load Balancing Costs**	**Mean (Load Balancing Costs)**	**Standard Deviation (Load Balancing Costs)**
Proposed PBA	463	865	150.29	87	150	100
PSO	1169	177	269	101	260	125

7 Conclusion

The paper has presented the metaheuristic based algorithm as a solution for multiple constrained based multi-objective optimization (MCOP) problem for traffic engineering in MPLS/ GMPLS networks. The proposed algorithm (with its presented pseudo code) is implemented on different number of nodes in MPLS/ GMPLS network with various algorithm cases such as PBA-1, PBA-2, PBA-3, PBA- 4 and PBA-5. The algorithm provides optimal solutions with Pareto front for minimum routing costs and load balancing costs. We also found that the routing costs increases when load balancing costs decreases and vice versa. Furthermore, the optimal solutions in the form of Pareto front have minimum routing costs and load balancing costs in small networks compare to large MPLS/ GMPLS networks.

References

1. Walrand, J. and Varaiya, P., *High-Performance Communication Networks*. San Francisco: Elsevier Science, textbook, (1999).
2. Mazandu, G., Traffic engineering using multipath routing approaches, textbook, (2007).
3. Liu, H., Zhang, X., Wang, D. and Xu, G., An algorithm for end-to-end performance analysis of network based on traffic engineering. *Journal of Electronics (China)*, 20(4), pp.293-298 (2003).
4. Girão-Silva, R., Craveirinha, J., Clímaco, J. and Captivo, M., Multiobjective routing in multiservice MPLS networks with traffic splitting — A network flow approach. *Journal of Systems Science and Systems Engineering*, 24(4), pp.389-432 (2015).
5. Ramadža, I., Ožegović, J. and Pekić, V., Network performance monitoring within MPLS traffic engineering enabled networks. In: *Software, Telecommunications and Computer Networks (SoftCOM), 2015 23rd International Conference on*. IEEE (2015).
6. Lv, M. and Ji, W., Research on GMPLS traffic engineering mechanism. In: *Communication Technology (ICCT), IEEE 13th International Conference on*. IEEE (2011).
7. Masood, M., Abuhelala, M. and Glesk, I., A comprehensive study of routing protocols performance with topological changes in standard networks. *International Journal of Electronics, Electrical and Computational System*, 5(8), pp.31-40 (2016).
8. Farrel, A. and Bryskin, I., *GMPLS*. San Francisco: Elsevier/Morgan Kaufman (2006).
9. El-Alfy, E., Mujahid, S. and Selim, S., A Pareto-based hybrid multiobjective evolutionary approach for constrained multipath traffic engineering optimization in MPLS/GMPLS networks. *Journal of Network and Computer Applications*, 36(4), pp.1196-1207 (2013).
10. SC, E. and C, E., A multiobjective off-line routing model for MPLS networks. In: *Proceedings of the 18th international teletraffic congress; (2003).*
11. Yang XS., A New Metaheuristic Bat-Inspired Algorithm. In: González J.R., Pelta D.A., Cruz C., Terrazas G., Krasnogor N. (eds) Nature Inspired Cooperative Strategies for Optimization. Studies in Computational Intelligence, vol 284. Springer, Berlin, Heidelberg (NICSO 2010)
12. Malakooti, B., Kim, H. and Sheikh, S., Bat intelligence search with application to multi-objective multiprocessor scheduling optimization. *The International Journal of Advanced Manufacturing Technology*, 60(9-12), pp.1071-1086 (2011).

13. Castelo Damasceno, N. and Gabriel Filho, O. (2017). PI controller optimization for a heat exchanger through metaheuristic Bat Algorithm, Particle Swarm Optimization, Flower Pollination Algorithm and Cuckoo Search Algorithm. *IEEE Latin America Transactions*, 15(9), pp.1801-1807.