IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

An Improved Particle Swarm Algorithm for Multi-Objectives based Optimization in MPLS/GMPLS Networks

MOHSIN MASOOD[1], MOHAMED MOSTAFA FOUAD[2], RASHID KAMAL[3], AND IVAN GLESK[1]

[1]Electronics and Electrical Engineering Department, University of Strathclyde, G11XW, Glasgow, UK
[2]Arab Academy for Science, Technology, and Maritime Transport, Cairo, Egypt
[3]Department of Computing and Technology, Abasyn University, Peshawar, Pakistan

Corresponding author: Rashid Kamal (e-mail: rashid.kamal@abasyn.edu.pk)

This work was supported by the European Union 's Horizon 2020 Research and Innovation Program through the Marie Sklodowska-Curie under Grant 734331

ABSTRACT Particle swarm optimization (PSO) is a swarm-based optimization technique capable of solving different categories of optimization problems. Nevertheless, PSO has a serious exploration issue that makes it a difficult choice for multi-objectives constrained optimization problems (MCOP). At the same time, Multi-Protocol Label Switched (MPLS) and its extended version Generalized MPLS, has become an emerging network technology for modern and diverse applications. Therefore, as per MPLS and Generalized MPLS MCOP needs, it is important to find the Pareto based optimal solutions that guarantee the optimal resource utilization without compromising the quality of services (QoS) within the networks. The paper proposes a novel version of PSO, which includes a modified version of the Elitist learning Strategy (ELS) in PSO that not only solves the existing exploration problem in PSO, but also produces optimal solutions with efficient convergence rates for different MPLS/ GMPLS network scales. The proposed approach has also been applied with two objective functions; the resource provisioning and the traffic load balancing costs. Our simulations and comparative study showed improved results of the proposed algorithm over the well-known optimization algorithms such as *standard* PSO, Adaptive PSO, Bat and Dolphin algorithm.

INDEX TERMS communication networks optimization, exploration problem, multi-objective constrained optimization, particle swarm optimization, swarm intelligence

I. INTRODUCTION

IN a network terminology, network efficiency term is used for the effective use of the network resources. Therefore, the target is to find the optimal route within the network that ensures the efficient utilization of the network resources. The computation of the optimal paths relays on optimization algorithms. Selecting the optimal paths in multi-constraints networks is still a major challenge. Finding the best path from the list of feasible paths under real networks scenarios is called a Multi-Constrained Optimal Path (MCOP) problem [1], [2]. In actual networks, the optimal paths can be derived by computing the multiple objectives with multi-constraints. These constraints might conflict with each other, which limits the feasible number of optimal paths. Therefore, network traffic engineers are trying to balance between multi-objective constrained functions. This concept is known as Pareto based optimal solutions. Pareto based solutions consist of Pareto set (multiple feasible solutions) in the multi-objective spaces. These optimization challenges can be solved either by exact or approximate/ meta-heuristic approaches. Meta-heuristic based methodologies are usually applied when exact-based approaches failed, or a computational complexity arose [3], [4].

Network vendors and network service providers usually offer MPLS (Multi-Protocol Label Switched) and GMPLS (Generalized MPLS) networks to satisfy the QoS requirements of miscellaneous applications desired by the end users. The QoS routing in such MPLS/ GMPLS networks is an MCOP optimization problem and is recognized as non-deterministic polynomial-time (NP) hard problem [2], [5]. Recently, numerous meta-heuristic optimization approaches (that include swarm, bio-inspired and genetic-based algorithms) have gained momentous popularity in various applications optimization, including the networks optimization.

Particle swarm optimization (PSO) technique is a swarm-based algorithm for solving both continues and combinatorial optimization problems and had been applied to several applications due to its fast convergence to find optimal solution. *Standard* PSO can be easily trap to its local and/or global optima that not only become the source of exploration problem but also effects its convergence. Multiple versions had been proposed, including hybrid and adaptive models of PSO to rescue algorithm's local and/or global optima problem [6]–[8]. However, PSO exploration problem is still a serious drawback for some modern applications.

This paper offers an adapted version of PSO algorithm for solving both local and global optima problem with a fast convergence rate for the MCOP optimization problem in MPLS/ GMPLS networks. The proposed adapted algorithm is based on the linear combination of objectives functions in order to produce optimal solutions and Pareto set that is offered as non-dominant solutions of the contrast based function. The rest of the paper is organized as follows; Section 2 represents an importance of meta-heuristic algorithms including PSO algorithm for solving various optimization problems and discusses the related work to PSO algorithm, subject to its various improved versions. Section 3 will formulates the problem of traffic engineering in MPLS/GMPLS networks as MCOP problem that includes multi-objectives subject to multiple constraints. Section 4 presents the solution in the form of proposed algorithm for MCOP problem in MPLS/GMPLS networks. Section 5 provides the results obtained and comparative analysis between the proposed algorithm and other meta-heuristic techniques. Finally, section 6 includes the conclusions. In the paper, for MPLS and GMPLS networks MPLS/ GMPLS keywords are used.

II. BACKGROUND AND RELATED WORK

A. META-HEURISTIC OPTIMIZATION ALGORITHMS

Numerous developed meta-heuristic algorithms are inspired by the behavioral study of birds, fishes, bees, ants, bats, and other animals. These algorithms have been used for solving complex optimization problems [9]. The meta-heuristic algorithms are classified on the foundation of inputs, searching tactics and the complication of exploration and exploitation capabilities [10]. Developing and selecting an efficient algorithm, that could manage multiple objective-based optimization problems, has become a core challenge. Moreover, local and global optima are known challenges in the convergence of meta-heuristic optimization techniques. Therefore, the best algorithm should offer global overview and have fewer chances to be trapped into its local optima [11].

B. PARTICLE SWARM OPTIMIZATION ALGORITHM AND EXPLORATION PROBLEM

Particle swarm optimization (PSO) algorithm was introduced by Kennedy and Eberhart to solve optimization problems [12]. The algorithm is dependent on a number of particles as searching agents, which disperse in the searching area hunting for solutions in a swarm formation. In such swarm, each particle depends on two parameters; a self-learning (cognitive) capability and social interactions with other swarm members through several iterations. During each algorithm's iteration, each particle locates the best position compared to the previous position, technically recognized as the local best position (local optima). Whereas, for global best position (global optima), each particle's best position is compared with all other particle's best positions in the swarm and choose the (global optima). The cognitive parameter is considered for tracing local optima, and for global optima, a social-interaction parameter is used. As per next iteration, the global optima will be considered as the reference that should be followed by the other swarm members [12], [13]. One of the basic problems of the PSO algorithm is the trapping of particles into local optima; a state where the particle flies to the same position in each iteration. This is identified as the local optima problem [6]. Whereas, global optima problem refers to a state where the number of particles in the swarm offer repetitive global position during iterations. Local and global optima problems not only occur in the PSO algorithm but are subjects of investigation for other optimization techniques such as Genetic and Ant Colony optimization algorithms [6], [14].

The enhancement of PSO technique performance by designing onto various types of topologies is an active research direction [15]. According to Dong et al [16], PSO algorithm for high dimension complex problem produces premature solutions due to its local optima problem. He further stated that present PSO has lack of effective global search capability and therefore, fail to find optimal solutions for complex and complicated optimization problems. Consequently, optimization algorithm designers are focusing on the fundamental improvement of local optima in detail with fast convergence or find all probabilities of exploring other searching areas in the searching domain. To increase the exploring capability of the PSO algorithm, the common strategy is to improve the search space diversity.

Further approaches are adapted through the enhancements in the algorithm's parameters or through proposing hybrid models with different algorithms [17]. According to Laura et al. [18], most of the techniques escape local optima trap by accepting non-improving or inferior neighbors. Liang et al. [15] presented a comprehensive learning particle swarm optimization (CLPSO), where all particles in the swarm maintain the particle's best positions and use it to update particle's velocity. A combination of local and global version is presented by Parsopoulos and Vrahatis [19] as a unified particle swarm optimizer (UPSO). Mendes and Kennedy proposed fully informed PSO, where all neighbor particles are used to update velocity instead of using particle's best and global best position parameters. Hence, the particle's influence on its neighbor is weighted the result is dependent on its fitness value and neighborhood size [20]. A fitness-distance-ratio-based PSO (FDR-PSO) has been developed by Veeramachaneni et al. [21], in which the strategy of near neighborhood interaction is employed. Other researchers pro-

posed a hybrid model of PSO with algorithms to enhance the algorithm performance and solve local and/ or global optima problem [22].

In actual networks, multiple objectives having multi-constrained based optimization problem (MCOP) is considered as complex optimization problem, which is also an NP hard problem. Therefore, meta-heuristic techniques have become appealing approaches, since they consume a small amount of the computing time to resolve described network optimization issues. Regarding multi-constrained optimal path (MCOP), PSO algorithm either in its original form or its modified versions has been applied for various optimization problems [1], [23]. Siddiqui et al. [24], proposed PSO based algorithm for the multi-constrained route optimization for electric vehicles (EVs). Jiuxin et al [25], propounded efficient multi-objective service selection (EMOSS) based on a multi-objective methodology to answer the time-complexity and non-global optimal solutions problems in quality of service levels requirements. Dekun [26], came up with an improved version of PSO algorithm for MCOP problem, which uses an external library to save its current best solution and maintains this archive during iterations.

III. PROBLEM FORMULATION AND NOTATIONS

According to several PSO based research papers, the application of the local best (P_{best}) PSO is superior to the global best (G_{best}) PSO. The local best PSO has better exploration abilities, not easily falling into a local optimum, and it is not suffering from an early convergence. While the global best version of particle swarm optimization (G_{best}) model is suitable for uni-modal optimization problems (single objective), but can be trapped into local optima for multi-modals problems. Whereas the local best version of PSO (P_{best}) can be used for solving multi-modal problems, it badly affects the convergence of the algorithm for uni-modal optimization problems [27].

To address the above, this paper proposes a modified version of PSO which is based on a combination of both, the local-best and the global-best versions of PSO algorithms. The method is entitled as "*Pareto based Modified Local Global Particle Swarm Optimization (PMLG-PSO)*" algorithm. The paper then evaluates the local and global optima problems and proposes solutions for MCOP optimization in MPLS/GMPLS networks. In the paper, an exploration problem and local/global optima problem are interchangeable. Before starting to investigate the exploration problem of the PSO algorithm, the paper first formulates the multi-objectives optimization problem in MPLS/GMPLS using a Pareto approach.

A. MPLS/ GMPLS NETWORKS: A BRIEF

MPLS/ GMPLS is a network protocols suite used for modern communication networks and supports all network switching technologies including time, wavelength, space, and packet switching. Using MPLS/ GMPLS protocols suite, a wireless connection establishes as a path between the edge routers of

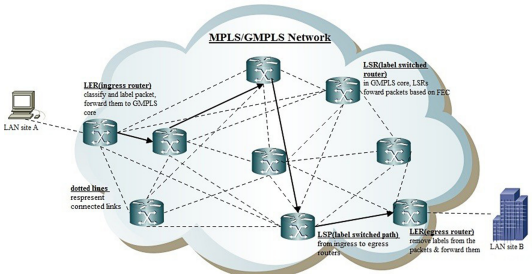


FIGURE 1. MPLS/GMPLS Network

the network, known as Label Switched Path (LSP). An edge router that is connected to sending host is called an ingress router. Whereas the router connected with the receiving host is the egress router. Ingress and egress routers are named as Label Edge Routers (LER). The present network domain includes all the connected nodes/ routers between LERs. Thus, based on the network topology, there are two basic types of routers; called interior routers and boundary routers. LERs are boundary routers. All connected routers between LERs are interior routers. MPLS/ GMPLS networking is built over the label switching terminology, where packets are identified and forwarded into the network, based on labels identification rather than IP addresses. All routers between LERs are Label Switched Routers (LSR). The job of LERs is to assign labels and LSRs will forward the packets in the network, dependent on the label identification rather than using IP addresses. Today, major networking vendors offer MPLS/ GMPLS enabled routers and many ISPs (Internet Service Providers) have deployed these routers in backbone networks [28].

In our study, the MPLS/ GMPLS network is represented as graphs, where nodes/routers are vertices and links signify edges, according to the graph theory. A route is computed path between the LERs. Each path indicates the sequentially connected links between LSRs. Therefore, the MPLS/ GMPLS network can be presented as *G*, where denotes for graph/ network, set of routers (vertices) are denoted by R_{Set}, where each router *r* / LER/ LSR is the member of R_{Set} and L_{Set} is the set of connected links *l*. Each link has a fixed capacity for traffic denoted as l_c. In the experiments, number of traffic requests will be arriving at ingress router, which computes the optimal path (s) based on proposed algorithm between ingress and the egress routers. Fig. 1 portrays the graphical representation of described MPLS/GMPLS networks. Further details on the topology notations are given in Table 1.

B. OBJECTIVE FUNCTIONS AND CONSTRAINTS (MCOP) FOR MPLS/ GMPLS NETWORKS

Our experimental scenarios of MPLS/ GMPLS networks consider two objective functions though out the optimization approach, such as; the optimal resource provisioning and the load balancing within the constructed network along with constraints, since it is MCOP problem. The following subsection is a detailed explanation for mentioned objective

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2934946, IEEE Access

TABLE 1. Topology Notations

G	graph / network topology
l	each link in the network
r	every router in the network (LERs and LSRs)
L_{Set}	$\{l_i, i= 1,\ 2\ 3,\ \ldots..,l_n\}$is the set of ;$\in L_{set}$
R_{Set}	$\{r_i, i= 1,\ 2\ 3,\ \ldots..,r_n\}$ is the set of routers r, $\quad r \in R_{set}$
l_r	is the link between r and r^*(connected LSR_s)
e	unit of traffic (traffic is no. of packets/ bits per sec)
E_{set}	$\{e_i,\ \mathrm{i} = 1,\ 2\ 3,\ \ldots..,e_n\ \}$ set of all traffics e ; $e \in E_{set}$
p	path defined for each traffic request e between LERs
P_{total}	total number of paths p
p_r^b	path bandwidth between connected routers
$v_{traffic}$	total volume of traffic requests for path p
l_u	link utilization of each link l
lc	link capacity of each link l
φ_l	load balancing objective function
$c_{l,r}$	routing cost for unit traffic e over link l
C_e^p	routing cost for unit traffic over path p
x_e^p	flow of traffic request on path p as continues variable ;$x_e^p > 0$
$i_{e,l}^p$	an indicator of traffic flow over path p
$i_{e,l}^p$	is 1 if link l is being used by path p for traffic request e

functions.

1) OPTIMAL RESOURCE PROVISIONING OBJECTIVE FUNCTION

In the experimental setup, the MPLS/ GMPLS domain-based network is designed, which receives traffic demands at ingress router of the network. The objective function for the routing algorithm is to compute the optimal path(s), where resources can be provisioned optimally without compromising on quality-of-services (QoS), to be granted to end users. In the network, each link l has assigned a specific cost per unit of traffic flow called as $C_{l,r}$, which is assigned by the service provider. Whereas C_e^p is the total routing cost of a path p for traffic request over a selected link l. The bandwidth price for the connection between each router of the network is p_r^b and thus, the bandwidth B cost is $C_e = B \times \mathrm{p}_r^b$ over the link l such as, $l = r, r^*$. Similarly, there is a delay d_e on each link l for forwarding the data traffic in the network. Let x_{ij} be the used as an indicator to show that whether the link l is selected for forwarding traffic from i to j.

$$X_{ij} = \begin{cases} 1, & \text{if the edge is selected} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$Cost_{resource} = Cost_{bandwidth} + Cost_{delay} + Cost_{Routing} \tag{2}$$

$$Cost_{bandwidth} = \alpha \left(\sum_{l\ \epsilon\ L_{set}} C_e i_{e,l}^p \right) \tag{3}$$

$$Cost_{delay} = \beta \left(\sum_{l\ \epsilon\ L_{set}} d_e \right) \tag{4}$$

$$Cost_{Routing} = \gamma \left(\sum_{l\ \epsilon\ L_{set}} C_{l,r} i_{e,l}^p \right) \tag{5}$$

$$\textbf{\textit{Objective Function :}} \qquad min\ (Cost_{resource}) \tag{6}$$

which subject to following constraints;

$\sum_{l\ \epsilon\ L_{set}} x_{ij} = 1 \quad for\ \ x_e^p > 0$	Constraint (6a)
$\sum_{l\ \epsilon\ L_{set}} x_{ij} = 1\ \ for \quad i_{e,l}^p = 1$	Constraint (6b)
$\sum_{l}^{L_{set}} x_e^p = v_{traffic}$	Constraint (6c)
$\sum_{l}^{L_{set}} p_r^b \ \ where\ p_r^b > 0,$ $p_r^b = \{10,\ 100\}$	Constraint (6d)
$d_e > 0\ \ for\ \forall\ l\ \in L_{set}$	Constraint (6e)

While $c_{l,r}$ is the total routing cost of a path p for traffic request over a selected link l. $alpha$, $beta$ and $gamma$ are positive constants that are used to denote the weight of each component. The objective function for offline resource provisioning problem is given by Eq. (6); Constraint (6a) ensures that all the links belonging to a set of links can only be selected if $x_{ij} = 1$, and when the variable x_e^p for the selection of the path value is greater than 0. From constraint (6b), the constraint identifies that the link l from L_{Set} can only be chosen if link$l$ is used for the traffic request. Constraint (6c) verifies that all selected links for the computed path must accommodate all number of traffic requests in the given topology. Constraint (6d) as a verification of bandwidth price is added on each selected link within limited bounds of [10, 100]. Constraint (6e) recognizes that traffic delay must be present to all links.

2) TRAFFIC LOAD BALANCING OBJECTIVE FUNCTION

There are a number of choices how to accomplish traffic load balancing metric and implement a piecewise approximation of a cost function as described by Fortzet et al. [29]. Each connected link between routers in the network is associated with two parameters identified as a link load or utilization l_u and a link capacity l_c. The Link utilization l_u is a number of traffic flows over the link. While the link capacity l_c can be defined as a capacity of a link to handle unit traffic/number of traffic flows [2], [30]. By having these two functions, traffic load balancing (φ_l) cost can be illustrated as the link utilization according to its capacity as follow;

$$\varphi_l = \frac{link\ utilization\ (l_u)}{link\ capacity\ (l_c)} \tag{7}$$

To achieve a resourceful use of links for traffic flows in the network, the links load balancing can be used as a fitness function that is to be minimize. This approach leads to an efficient utilization of network resources as well as to minimum packets loss, bound delays, and jitter etc. Consequently, this objective function can be described to minimize the sum of traffic load balancing costs for all the links(l) in the network as demonstrated by Eq. (8).

$$\textbf{\textit{Objective Function :}} \qquad mini(\sum_{l\ \epsilon\ L_{set}} \varphi_l) \tag{8}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2934946, IEEE Access

Subject to constraints:

$$l_u \leq l_c \quad \text{Constraint } (8a)$$

$$\varphi_l \geq l_u \; \forall \, l \in L_{total} \quad \text{Constraint } (8b)$$

$$l_u = \sum_{e \in E} \sum_{l=1}^{L_{set}} i_e^p x_e^p \quad \text{Constraint } (8c)$$

$$\varphi_l \geq 2.2\, l_u - 0.5\, 2\; l_c \quad \text{Constraint } (8d)$$

$$\varphi_l \geq 5.1\, l_u - 2.33\;\; l_c \quad \text{Constraint } (8e)$$

$$\varphi_l \geq 15.2\, l_u - 9.32\;\; l_c \quad \text{Constraint } (8f)$$

$$\varphi_l \geq 60.2\, l_u - 45.35\;\; l_c \quad \text{Constraint } (8g)$$

$$\varphi_l \geq 300.3\, l_u - 261.34\;\; l_c \quad \text{Constraint } (8h)$$

$$x_e^p > 0\,, \; \forall \, e \in E_{set} \quad and \quad \forall \, l \in L_{set} \quad \text{Constraint } (8i)$$

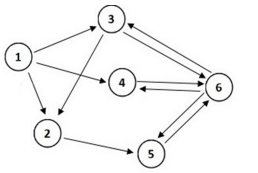
$$r \in R_{set} \; for \; v_{traffic} \quad \text{Constraint } (8j)$$

$$l > 0 \; for \; l = \{r, r^*\} \quad \text{Constraint } (8k)$$

Constraint (8a) ensures that the link utilization l_u in the network must have either the same or less than the link's capacity l_u. In modest wordings, the traffic flows over the link must not exceed the link's capacity. Constraint (8b) states that traffic load balancing φ_l must be greater than or equal to the link utilization l_u for all links. Further, this constraint guarantees that difference between the link utilization l_u and link capacity l_c must not be very high. Constraint (8c) precisely explains features of the link utilization l_u in the given topology. Link l can only be considered as utilized for a traffic flow, when the traffic flow is the member of the set of traffic demands, and the link l is also the member of the set in admissible links of the network. While $i_e^p x_e^p$ are the indicators which identifies the link l can be used for traffic flow over a path, such that a link l can be used for path computation. Constraints (8d) to (8h) defines some specific linear load balancing cost function within feasible region. Constraint (8i) represents the identifier of traffic flow over a path, when it is more than 0 value. In other words, a path can be used for a traffic request. Moreover, constraints (8j) and (8k) identify the connected links and routers in the topology, which are used for the requested path.

a: PARTICLES ENCODING

A Particle encoding is the first step towards further evaluation of the proposed algorithm. In this paper, the particles travel over the two-dimensional matrix, where each row represents a from-node and each column defines the to-node, in a searching (sparse) matrix. Each particle represents the traffic flow over each selected link for a given traffic request. To elaborate this section, let us consider a 6-nodes network as shown in Fig. 2 and the adjacency and demand matrices, as shown in Fig. 3, 4. In the given example, three traffic



$Network = (source, destination, traffic)$

FIGURE 2. Network Model

From Node \ To Node	1	2	3	4	5	6
1		1	1	1		
2					1	
3		1				1
4						1
5						1
6			1	1	1	

FIGURE 3. Adjacency Matrix for Traffic Model

demands (packets per unit secs) are considered as $N_1 = (1,\ 6,\ 30)$, $N_2 = (3,\ 5,\ 10)$ and $N_3 = (1,\ 4,\ 20)$. For the first traffic demand t_1, the traffic will be split over three paths; i.e., first path as (1, 3, 6), 2nd path (1, 4, 6) and (1, 2, 5, 6) as 3rd path. For N_2, traffic splits over two paths such as path-1: (3, 2, 5), and path-2: (3, 6, 5). Similarly, for N_3, the traffic will be split over two equal paths as path-1: (1, 4) and path-2: (1, 3, 6, 4).

IV. PROPOSED PMLG-PSO APPROACH

The paper proposes a novel meta-heuristic approach, named as *PMLG-PSO* algorithm for the MCOP optimization problem in MPLS/ GMPLS networks. The flow chart of the proposed *PMLG-PSO* algorithm is shown in Fig. 5. The flow chart is segmented into three main phases; initialization phase, processing phase and post-processing phase. Each of the phase will be described in pseudo code along with explanation in the following subsections.

A. INITIAL PHASE

This phase discusses the initial stage of the algorithm, where the searching agents (particles), searching space (matrix/graph) and algorithm's initial operations are engaged. In pre-processing stage, the constraints have been applied to limit the searching space and as a result obtain a skeleton matrix. A Skeleton matrix is used as a searching constrained dependent graph, where particles can hunt for the feasible solutions. Certain network administrative-based constraints are applied. For example, each traffic request for specific nodes and links are not allowed. In experiments, skeleton matrix is applied as a MPLS/ GMPLS network that is the collection of permissible links between connected nodes and can be used for traffic requests. Furthermore, the particles are initialized with random initial positions in the searching matrix for both fitness functions (resource provisioning and traffic load balancing).

Let us explore the case for a resource provisioning fitness function along with its constraints, where the random initial positions of particles are loaded as $init_1$. Whereas, for traf-

TABLE 2. Variables used in Pseudo code of PMLG-PSO Algorithm

i	i^{th} particle
N	Population size
x_i	i^{th} Particle position in network x_{i_r}
i^{th}	Particle position in resource provisioning matrix
x_{i_l}	i^{th} Particle Position in traffic load balancing matrix
v_i	velocity of the i^{th} particle (s)
v_{i_r}	i^{th} particle velocity in resource provisioning matrix
v_{i_l}	i^{th} particle velocity in traffic load balancing matrix
w	weight inertia
$c_{1(r)}$	cognitive coefficient (resource provisioning matrix)
$c_{2(r)}$	social behavior coefficient (resource provisioning)
$c_{1(l)}$	cognitive coefficient (load balancing matrix)
$c_{2(l)}$	social behavior coefficient (load balancing matrix)
$P_{(best_r)}$	i^{th} particle best position for resource provisioning
$P_{(best_l)}$	i^{th} particle best position for traffic load balancing
$G_{(best_r)}$	Global best position for resource provisioning cost
$G_{(best_l)}$	Global best position for traffic load balancing cost
$f_{(xi_r)}$	Resource provisioning cost objective function
$f_{(xi_l)}$	Traffic load balancing cost objective function
$f_{(xi previous)}$	Previous fitness function

fic load balancing fitness functions with its constraints, the initial particle positions are loaded in $init_2$. The *PMLG-PSO* algorithm and its proposed modifications are:

A Particle best position for a resource provision function is P_{best_r} . The particle best position for a load balancing function is P_{best_l} . Global-best-positions G_{best_r} and G_{best_l} are computed by P_{best} (P_{best_r} and P_{best_l}) . Solutions for (G_{best_r} and G_{best_l}) are collected as non-dominant solutions for generating the graph of Pareto front. However, the algorithm's optimal solutions are obtained as global-best-positions G_{best} from the foundation of the linear combination fitness function ($init_{LC}$). According to [31], [32] for an optimal solutions collection, there are various tactics that can be used for MCOP optimization problems. They are criteria-based, aggregation-based, Pareto-based dominant ranking and linear combination-based methods. In this Paper, we applied the parametric linear combination of fitness functions for the computation of optimal solutions (G_{best}, which can mathematically derive as;

$$Fitness\ Function_{(LC)}:$$
$$\alpha\ (Cost_{resource}) + (1-\alpha)\left(\sum_{l\ \epsilon\ L_{set}} \varphi_l\right) + \Phi \qquad (9)$$

In Eq. (9), α is a weight variable and Φ is called plenty. G_{best} of the linear combination fitness function ($Fitness\ Function_{(LC)}$ *or* $init_{LC}$) will be considered as the optimal solution. The pseudo code for the initialization phase is given below as *Algorithm* 1, whereas variables used in the algorithms are defined in Table 2.

Algorithm 1:Algorithm for Initialization Stage

Each particle $\boldsymbol{P}$ in population initialize w.r.t f_{xi_r} & f_{xi_l}
LOOP Process
for $i = 1 : N$ **do**
  $x_{i(r)}$: for $init_1$ subject to constraints
  $x_{i(l)}$: for $init_2$ subject to constraints
  $v_{i(r)}$: for $init_1$
  $v_{i(l)}$: for $init_2$
end for
w: initialize weight inertia parameter
($c_{1\ (r)}$, $c_{2(r)}$): initialize with the initial values of 2
for $i = 1 : N$ **do**
  evaluate $fintess_{Cost_{resource}}$ (f_{xi_r}): for $init_1$
  evaluate $fintess_{\sum_{l\ \epsilon\ L_{set}} \varphi_l}$ (f_{xi_l}): for $init_2$
  update P_{best_r}: based on f_{xi_r}
  update P_{best_l}: based on f_{xi_l}
end for
  select G_{best_r}: among updated P_{best_r}
  select G_{best_l}: among updated P_{best_l}
for $i = 1 : N$ **do**
  evaluate $fitness_{(LC)}$: $\alpha\ (f_{xi_r}) + (1-\alpha)\ (f_{xi_l}) + \Phi$
  update $P_{best_{LC}}$: based on $fitness_{(LC)}$
end for
Generate $G_{best_{LC}}$ among updated $P_{best_{LC}}$

B. PROCESSING PHASE

Processing fragment contains all those key components of the proposed algorithm, which help *PMLG-PSO* to resolve the traditional PSO algorithm's exploration problem and thus, find the optimal solutions for MCOP optimization in the network. During processing stage, algorithm's traditional parameters such as w, $c_1$and c_2 will contribute to the algorithm's convergence activity for $init_1$, $init_2$ and $init_{LC}$. The updated positions and velocities for $init_1$, $init_2$ and $init_{LC}$ are evaluated with respect to their fitness functions separately. The feasible solutions are referred here as feasible paths in the network. A condition is applied that if the updated fitness function used for path computation is optimal than the previous solution, then update particle's position (solution) as particle's best solution(P_{best}). This exercise occurs in $init_1$, $init_2$ as $P_{best_r}$and P_{best_l} respectively. This step ensures that the updated position is better than the previous position. If the latest position of the particle is not optimal then particle keeps the previous position as its best position and from P_{best} positions in the updated swarm, chooses the global best solution(G_{best}).

The algorithm uses evolutionary state estimation (ESE) technique, which is dependent on the particle's distribution information in the swarm [33]. The evolutionary factor is developed on the fuzzy classification method and is directed by evolutionary factor f . The steps that are been followed for ESE terminology can be described as follow:

1) Find each particle's mean distance d_i from all other particles in the swarm, using following expression;

$$d_i = \frac{1}{N-1} \sum_{j=1,\ j\neq x_i}^{N} \sqrt{\sum_{k=1}^{D} \left(x_i^k - x_j^k\right)^2} \qquad (10)$$

Where, j= another particle from x_i particle,
k=each dimension in the matrix

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2934946, IEEE Access

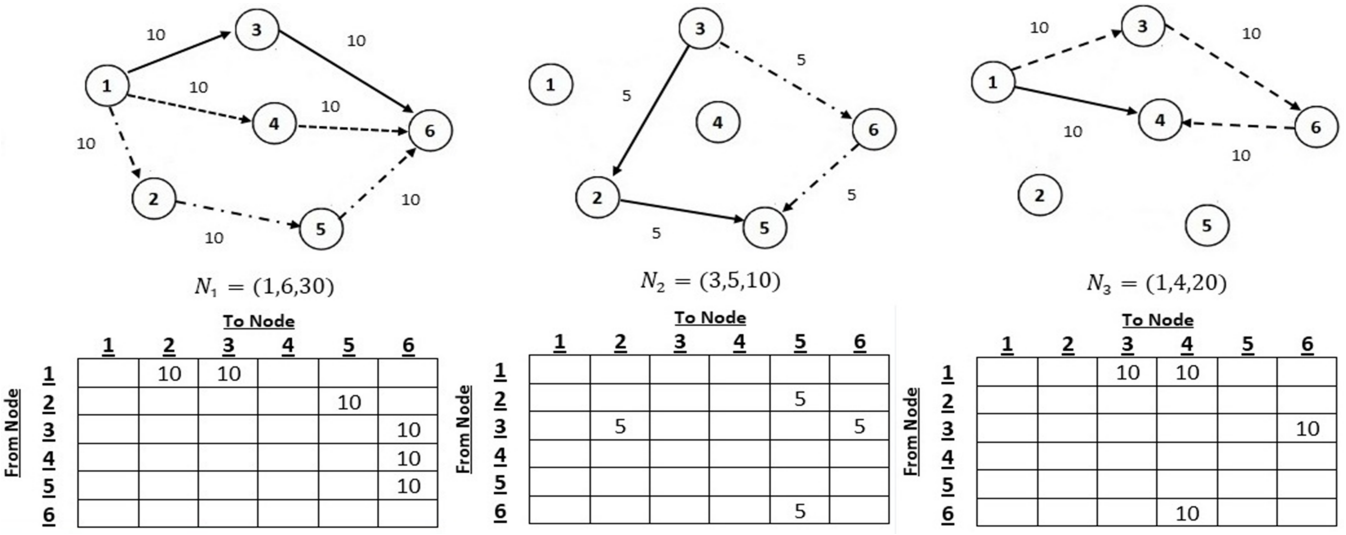


FIGURE 4. Demand Matrices

D= total dimensions in the matrix

1) Compare all computed distances (for each particle) for both $init_1$ and $init_2$ matrices and determine maximum distance(d_{max}), minimum distance(d_{min}) and distance of the computed global best particle(d_g).
2) Now, evaluate the evolutionary factor(f) by the following expression;

$$f = \frac{d_g - d_{min}}{d_{max} - d_{min}} \in [\mathbf{0}, \mathbf{1}] \tag{11}$$

3) After computing evolutionary factor (f), calculate the w as follow;

$$w = \frac{\mathbf{1}}{(\mathbf{1} + \mathbf{1.5}e^{-\mathbf{2.6}f})} \in [\mathbf{0.4}, \mathbf{0.9}], \forall f \in [\mathbf{0}, \mathbf{1}] \tag{12}$$

4) Evaluate $c_1$and $c_2$with the following expression;

$$c_i^{t+1} = c_i^t \pm \delta \tag{13}$$

Where i =1, 2 for c_1 and c_2, but c_1, c_2 values must be within the limits[1.5, 2.5] and is a uniformly generated random value from the range [0, 1]. Adaptive PSO used Elitist Learning Strategy (ELS) method to target the global optima problem in PSO technique [33].

In APSO based ELS model, the random dimension p^d from the particle's global historical best positions is selected, denoted as p^d. After that, find the maximum x^d_{max} and minimum x^d_{min} dimension ranges. Compute the value of P^D (dimension value) based on the following equation:

$$P^D = p^d + \left(x^d_{max} - x^d_{max}\right) Gaussian\left(\mu, \sigma^2\right) \tag{14}$$

where $\sigma = \sigma_{max}\left(\sigma_{max} - \sigma_{min}\right)$, $\sigma_{max} = 1$, $\sigma_{min} = 0.1$. Gaussian represents here as Gaussian distribution with mean value $\mu = 0$ and σ is time varying standard deviation value. σ_{max} and σ_{min} values are based on the empirical study of [33].

Despite the use of an ELS model in APSO, the algorithm can still suffer intense exploration problem, as the local-optima problem still exist during number of iterations. Such as, if only global optima problem is supervised with the assistance of ELS model (as proposed in [33]), then the local optima problem can still occur. This situation disturbs the overall performance of the PSO algorithm with respect to global optima problem along with poor convergence and thus, generate sub-optimal solutions. Moreover, during iterations, the increasing number of particles undergoing local optima may results the ELS technique as ineffective model.

In simple words, the ELS model that targets only global optima problem, does not guarantee the provision of optimal solutions if has the exploration problem in the form of the local optima issue. The research work presented in this paper proposes an algorithm that targets both local and global optima problems simultaneously, and to certify the algorithm's rationality on real-world application, and is applied for MCOP optimization in MPLS/ GMPLS networks.

1) PMLG-PSO Algorithm for Exploration Problem

In the proposed *PMLG-PSO* algorithm, a modified model of ELS is offered which is embedded with so called "Repairing Box". Once the algorithm identifies the exploration problem at any stage, it goes to its repairing box (embedded six registers). The repairing box is used to drag out the particle or group of particles is either stacked in its local optima or its global optima. This repairing process of the local/ global optima problem within the repairing box is briefly explained in the pseudo code along with description. However, before solving the problem, it is important to vigilantly detect the local/optima problem. Having such goal, six registers are implanted into the algorithm. Each register is specifically used for the detection of optima/ global problem during

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2934946, IEEE Access

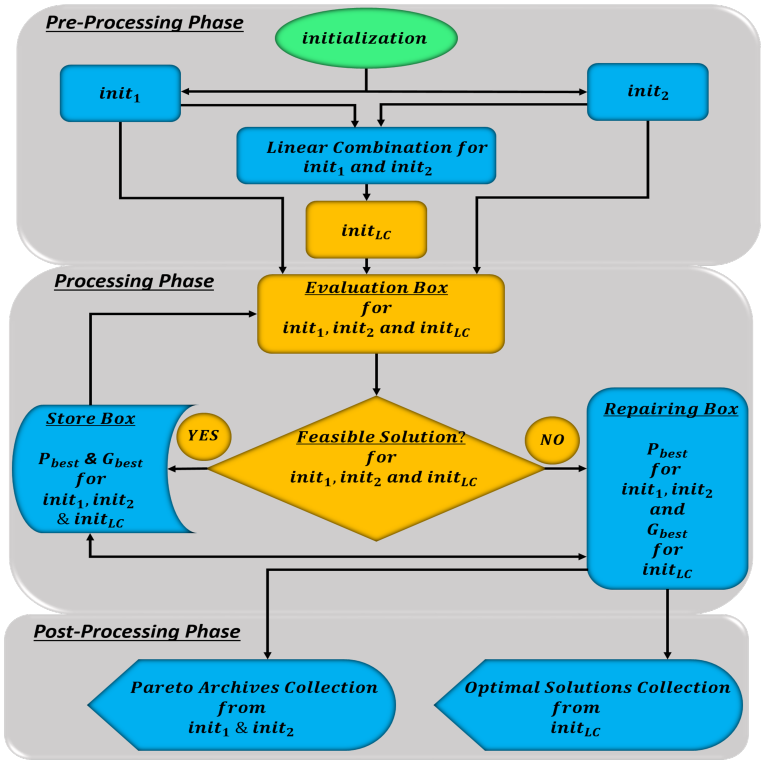


FIGURE 5. Proposed PMLG-PSO Algorithm Flow Chart

TABLE 3. Registers used for Detection of Exploration Problem

	Register Names	Registers Working
i.	$\boldsymbol{P_{best}}$ ***Register***	Register for each particle's local best position P_{best}
ii.	$\boldsymbol{P_{best}}$ **:** $\boldsymbol{p^d}$ ***Register***	Register for P_{best} random p^d values
iii.	$\boldsymbol{P_{best}}$ **:** $\boldsymbol{P^D}$ ***Register***	Register for P_{best} P^D $values$
iv.	$\boldsymbol{G_{best}}$ ***Register***	Storing register for global best position G_{best}
v.	$\boldsymbol{G_{best}}$ ***Register*** $\boldsymbol{p^d}$	Register of global best position p^d values
vi.	$\boldsymbol{G_{best}}$ **:** $\boldsymbol{P^D}$ ***Register***	Storing register for G_{best} P^D values

different stages of the algorithm. The purposes and names of the registers are mentioned in Table 3.

Registers in Table 3 work as a memory registers that store previous values thus help the algorithm to spot the exploration problem. For example, during algorithm's execution, each i particle in the swarm keeps storing its P_{best} values in the P_{best} register. If P_{best} fell into its local optima, then P_{best} register identifies the repetitive value/ position for P_{best}. Furthermore, it is also possible that P_{best} :p^d $Register$ and P_{best} :P^D $Register$ may detects the local optima problem. Similar situations can be reflected in G_{best} register, G_{best} : P^D $Register$ and G_{best} : P^D $Register$ during algorithm's processing stages. Once the local and/ or global optima problem are identified at any stage with the help of mentioned registers, then algorithm follows towards the repairing box. Repairing box contains the modified version of ELS model and is entitled as $Adaptive\ Elitist\ learning\ Strategy\ (AELS)$ model. In the repairing box, the offered $AELS$ model will be used to fix the local/ global optima problem. Fig. 6 and Fig. 7 are the pictorial demonstration of the proposed repairing box and *AELS* model.

For example, if P_{best} $Register$ found a local optima problem, this is resolved with the help of $AELS$. If in some way if the P_{best} : p^d $Register$ records still show the local optima problem, then the proposed $AELS$ technique will further choose another random p^d value and then computes the final P^D value. If the problem still occurs, the algorithm will repeat the random replacement of other p^d values until the condition is not satisfied. Furthermore, in worst-case scenario, when p^d is also trapped in a fixed value, then the new p^d will be fetched from P_{best} $register$.

Global optima problems can be exposed in G_{best} , G_{best} : p^d and $Registers$ and henceforth, can be resolved with the help of $AELS$ model in the repairing box. Once, the exploration problem is solved, the updated solution is sent back to the processing stage of the algorithm.

Similar procedure of the proposed repairing box (along with $AELS$) and the algorithm will be followed for the linear

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2934946, IEEE Access

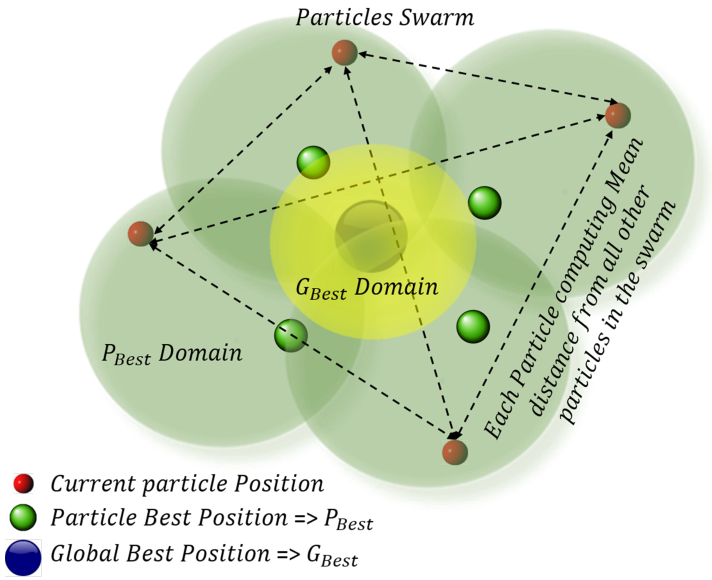


FIGURE 6. Mean Distance Computation between Swarm particles

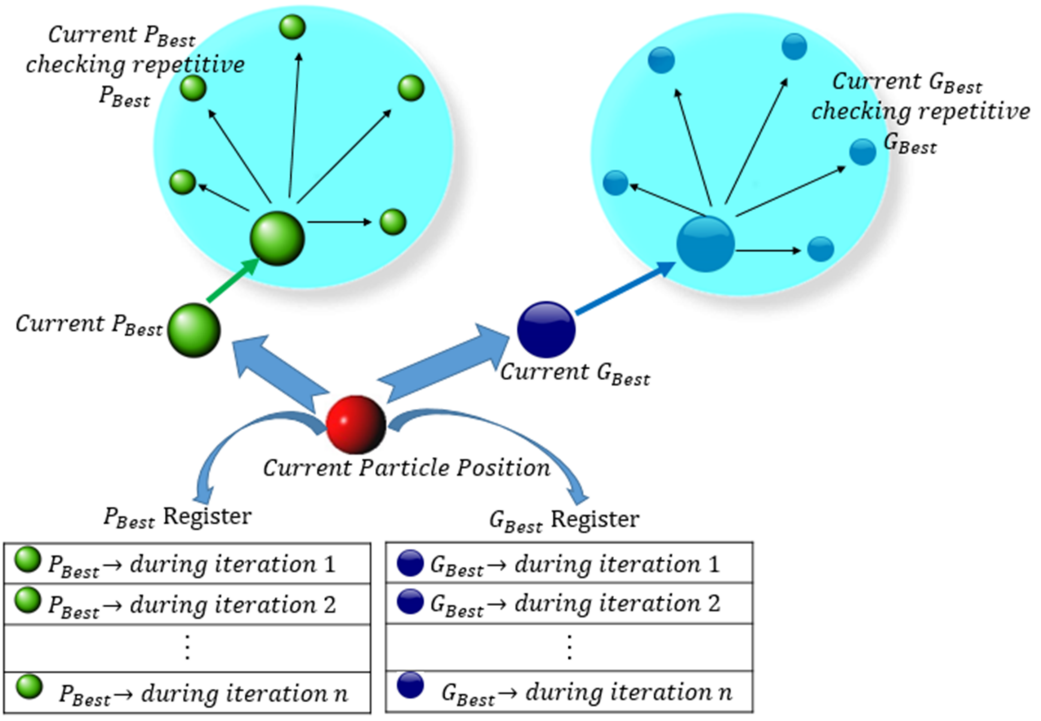


FIGURE 7. Each Particle Pbest and Gbest Registers formation

combination function ($init_{LC}$). If there is no local/global optima problem found, then Pareto based feasible solutions are stored the in the appropriate storage boxes of $init_1, init_2$ and optimal solutions in $init_{LC}$.
The pseudo code in the processing stage of the *PMLG-PSO* algorithm is given as follow.

C. POST PROCESSING PHASE

In the algorithm, two separate initializations ($init_1$ *and* $init_2$) are used to collect non-dominated solutions, in order to create the archive for the Pareto front. Whereas the algorithm's optimal solutions are stored in the form of G_{best} solutions from $init_{LC}$. These optimal solutions are a collection of connected links (paths). The G_{best} solutions are preserved in the form of an archive and stored in the G_{best} storing box, as a number of available optimal solutions, as it is shown in *Algorithm* 3.

Algorithm 2: Pseudo code for Processing stage of PMLG-PSO Algorithm

for both Objective functions **do**
  for $i = 1 : N$ **do**
    if $f_{xi_{present}} < f_{xi_{previous}}$ **then**
      update: P_{best}: $x_{i_r}, x_{i_l} \rightarrow P_{best_r}, P_{best_l}$
      store: $p^d register\ (P_{best})$ = empty
      store: $P^D register\ (P_{best})$ = empty
    else
      $P_{best_{previous}}$ = updated $-P_{best}\ (P_{best_r} OR\ P_{best_l})$
      evaluate: updated$-P_{best}\rightarrow$Repairing Box
      Repairing Box ($AELS$ method for P_{best}):
      if p^d=$p^d_{previous}$ and P^D=$P^D_{previous}$ **then**
        initialize: random p^d from P_{best} dimensions
        initialize : x^d_{max} and $x^d_{max}$from P_{best} dim.
        compute: P^D
      else
        initialize: random p^d from "$p^d\ \ Register$"
        initialize: random P^D from "$P^D\ \ Register$"
      end if
      update: p^d register (P_{best})
      update: P^D register (P_{best})
      update: $P_{best}\ (P_{best_r}, P_{best_l})$ register
    end if
    stored $P_{best}\ (P_{best_r}, P_{best_l})$
    update $G_{best}\ (G_{best_r},\ G_{best_l})$
    $P_{best}\ (P_{best_r}, P_{best_l}) \rightarrow G_{best_{previous}}\ (G_{best_r},\ G_{best_l})$
    if $G_{best_{present}} < G_{best_{previous}}$ **then**
      update G_{best}
      store : $G_{best}\ (G_{best_r},\ G_{best_l})$ in G_{best} Register
      store: $p^d\ (G_{best})$ = empty
      store: $P^D\ (G_{best})$ = empty
      $G_{best_{previous}}$ => updated $G_{best}\ (G_{best_r}\ , G_{best_l})$
    else
      evaluate: $G_{best} \rightarrow$ Repairing Box
      Repairing Box ($AELS$ method for G_{best})
      if $(p^d\ = p^d_{previous}$ and $P^D\ = P^D_{previous}$ **then**
        initialize : random p^d from G_{best} dimensions
        initialize : x^d_{max} *and* x^d_{max} from G_{best} dim.
        compute : P^D
      else
        initialize : random p^d from G_{best} Register
        initialize : x^d_{max} *and* x^d_{max} from G_{best} dim.
        compute : P^D
      end if
      update : p^d register (G_{best})
      update : P^D register (G_{best})
      update : $G_{best}\ (G_{best_r}, G_{best_l})$ register
    end if
  end for
  stored : $G_{best}\ (G_{best_r}, G_{best_l})$ register
  end for

Algorithm 3: Pseudo code for Post Processing Stage of the PMLG-PSO Algorithm

Input: G_{best} from objective functions
Output: Optimal path for the traffic request
for $i = 1$:maximum iteration for G_{best} **do**
  for $j = 1 : N$ **do**
    G_{best}: representing solutions of the problem
    select link with minimum value in G_{best} matrix
    initialize path from selected links
    path for requested traffic
  end for
  archive the selected links as the optimal
end for

V. EMPIRICAL ASSESSMENTS

To assess the effectiveness of the proposed *PMLG-PSO* algorithm, we have made two concise performance analysis investigations. The first evaluation appears in the outcomes of solving the MCOP optimization problem in MPLS/ GMPLS networks by using the Pareto Front graphs. The second investigation is a comparative study analysis; where the offered algorithm performance is compared against numerous meta-heuristic algorithms such as particle swarm optimization (PSO), adaptive PSO [33], [34], Bat algorithm [35] and Dolphin algorithm [36]. Both researches focus on an algorithm convergence ratio (particularly the trapping in local/global optima) to test how successfully the proposed algorithm fixed the exploration problem when compared to the other optimization algorithms. The MPLS/ GMPLS network-based topology is used for the experiments, where objective functions; resource provisioning costs and traffic load balancing costs are considered, as described in the problem formulation section. Demonstration of the conducted experiments along with results breakdown are discussed in the proceeding sections.

A. EXPERIMENTS FOR PARETO FRONT AND RESULTS ANALYSIS

For conducting simulations, we have used the *MATLAB R2016a* package to develop the Pareto based solutions as well as optimal solutions based on described algorithms.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2934946, IEEE Access

TABLE 4. PMLG-PSO Algorithm Versions

Version	Number of Particles	Number of Iterations
$PMLG-PSO_1$	20	
$PMLG-PSO_2$	30	20
$PMLG-PSO_3$	40	
$PMLG-PSO_4$	20	40
$PMLG-PSO_5$	20	
$PMLG-PSO_6$	40	60

The experiments were scheduled in such a manner that the proposed *PMLG-PSO* algorithm was categorized into six classes, such as *PMLG-PSO*$_n$ where n = 1, 2, 3, 4, 5, 6. In each version, various approaches are applied (in case of number of particles or number of algorithm iterations as it is labelled in Table 4). These classes are implemented for a predefined number of nodes for MPLS/ GMPLS networks P = 80, 90, 100 and 110 nodes. The purpose of making these variations is to inspect the success of the algorithm on different network generating optimal solutions. The basic structure of the algorithm (described in *Algorithm* 1, 2 and 3) towards obtaining a minimization of the objective functions, is the same for all network's scenarios. Fig. 8 portrays the results of the Pareto archive that has been collected in the post-processing stage, which produced Pareto front-based solutions from *PMLG-PSO*$_1$ to *PMLG-PSO*$_6$ for a different scale of MPLS/ GMPLS networks.

A set of Pareto solutions from the Pareto archive, for two objective functions, were collected and are presented in Fig. 8 (a) – (d). As formerly explained, for each *PMLG-PSO*$_n$ (*PMLG-PSO*$_1$ to *PMLG-PSO*$_6$) class, dissimilar values are presented as Pareto solutions. For the ease of understanding, the obtained non-dominated solutions are presented in separate figures for each case and are linked by lines to highlight different *PMLG-PSO*$_n$ classes results. Fig. 8 indicates that the proposed algorithm is eligible to provide feasible Pareto based solutions with the contrast objective functions. Fig. 8 also shows that once the traffic load balancing costs increases, the resource provisioning costs decreases and vice versa.

B. PMLG-PSO ALGORITHM 'S CONVERGENCE INVESTIGATION

To investigate the *PMLG-PSO* success ratio in terms of exploration problem, the results are collected as the optimal solutions from the linear combination fitness function ($init_{LC}$). To discover the algorithm convergence activity with various network scale, the algorithm has been run on 30, 60, 90 and 100 nodes network (see Fig. 9 (a, b, c and d)). The purpose of this investigation is to discover the algorithm's capability of solving local/global optima problem and create optimal solutions with smooth convergence without any interruption of the exploration problem during any iteration of the algorithm.

The results shown in Fig. 9 (a, b, c and d) display the convergence activity of *PMLG-PSO* and portray the local/global optima problem (if one existed) during the convergence. In our experiments, each version of *PMLG-PSO*$_n$ is simulated using various network sizes. From the results obtained in Fig. 9, it is found that the algorithm does not experience a local/ global problem and smoothly converges towards its optimal solutions. We can conclude that the algorithm successfully overcomes the exploration problem and produces optimal solutions with the coherent convergence. In its nature *PMLG-PSO* is a stochastic algorithm, therefore the Pareto curves change. To examine these statistical variations, the paper uses a non-parametric test, the Kruskal-Wallis Test [40]. The purpose of the Kruskal-Wallis test was to identify if at least one sample stochastically dominates over the others. Table 5 shows our simulation results for each version of *PMLG-PSO*$_n$ which were conducted for a different number of nodes, it also includes resulted H values. In our simulations, all *PMLG-PSO*$_n$ versions had the same number of iterations but a different number of particles. *Run*$_n$ represents a number of simulations for each *PMLG-PSO*$_n$ algorithm version. Using the Kruskal-Wallis test, we have tested six groups of data *PMLG-PSO*$_n$ (*PMLG-PSO*$_1$ to *PMLG-PSO*$_6$). Based on test rules, two H values i.e., H_0 and H_1 were considered. H_0 indicates there is no significant difference between *PMLG-PSO*$_n$ (*PMLG-PSO*$_1$ to *PMLG-PSO*$_6$) data, while H_1 indicates a major difference. In the Kruskal-Wallis test a value α = 0.05 is used for the identification of H_0 or H_1. The degree–of–freedom (*df*) for α value is calculated where $df = k - 1$ and k represents a number of data sets [40]. In our case study (*PMLG-PSO*$_n$ versions), k value is 6 (as n = 6). For these data sets, by taking into account α = 0.05 and df = 5 and applying a state decision rule we get 11.0705. This means, H values less than 11.0705 satisfy H_0 hypothesis. On other hand, H values more than 11.0705 satisfy H_1 hypothesis. Having this in mind, when examining Table 5, we can see that for each case study shown, the computed H value is less than 11.0705. Therefore, according to Kruskal-Wallis test, there is no statistically significant difference among solutions produced by individual *PMLG-PSO*$_n$ versions. The experiment thus provides a proof that even a stochastic nature of the *PMLG-PSO* algorithm can produce optimal solutions that will not stochastically dominate over each other solutions. By other words, the algorithm does not produce abrupt solutions.

C. COMPARATIVE STUDY OF PMLG-PSO WITH META-HEURISTIC ALGORITHMS

The purpose of making comparison with the other meta-heuristic based approaches is to scrutinize the *PMLG-PSO* efficiency and other optimization techniques. For this assessment, the paper uses particle swarm opti-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2934946, IEEE Access

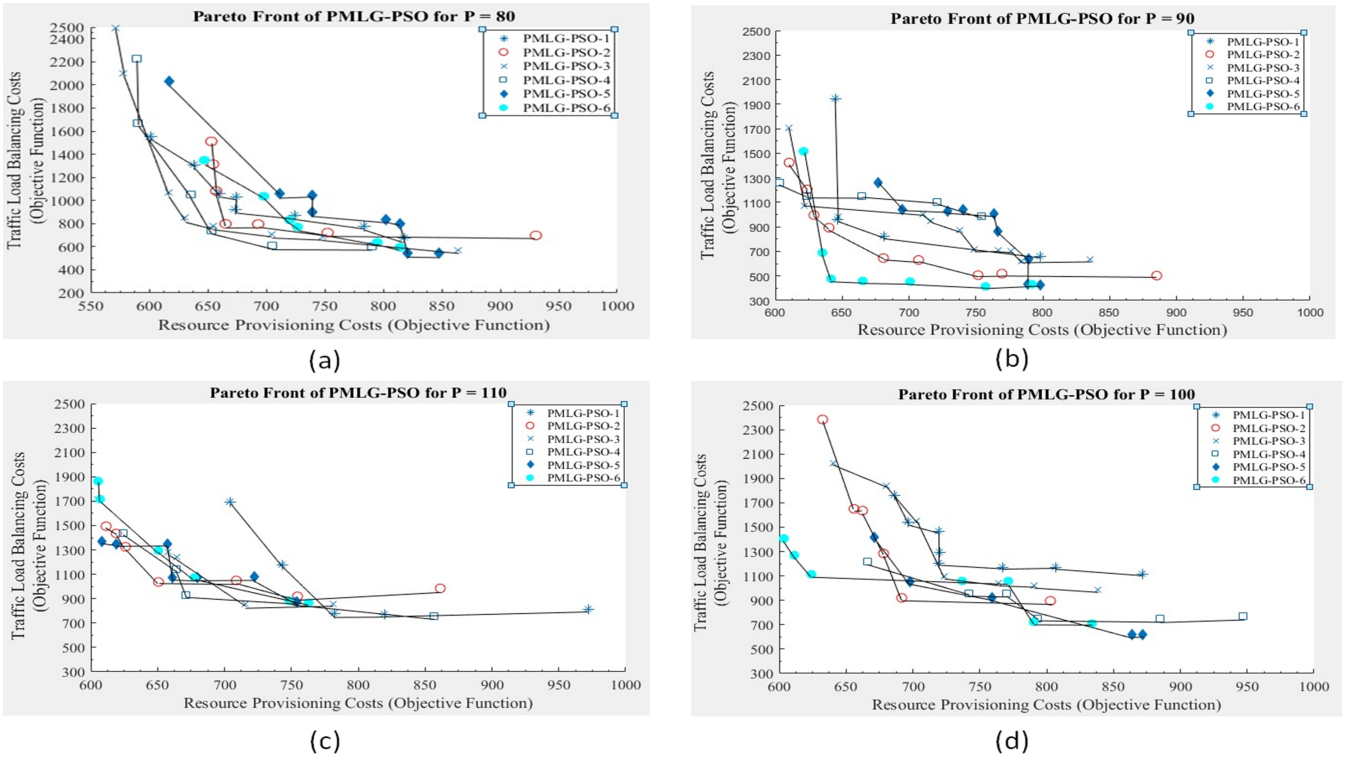


FIGURE 8. Pareto-Front Graphs, P=80, 90, 100, 110 Nodes Network

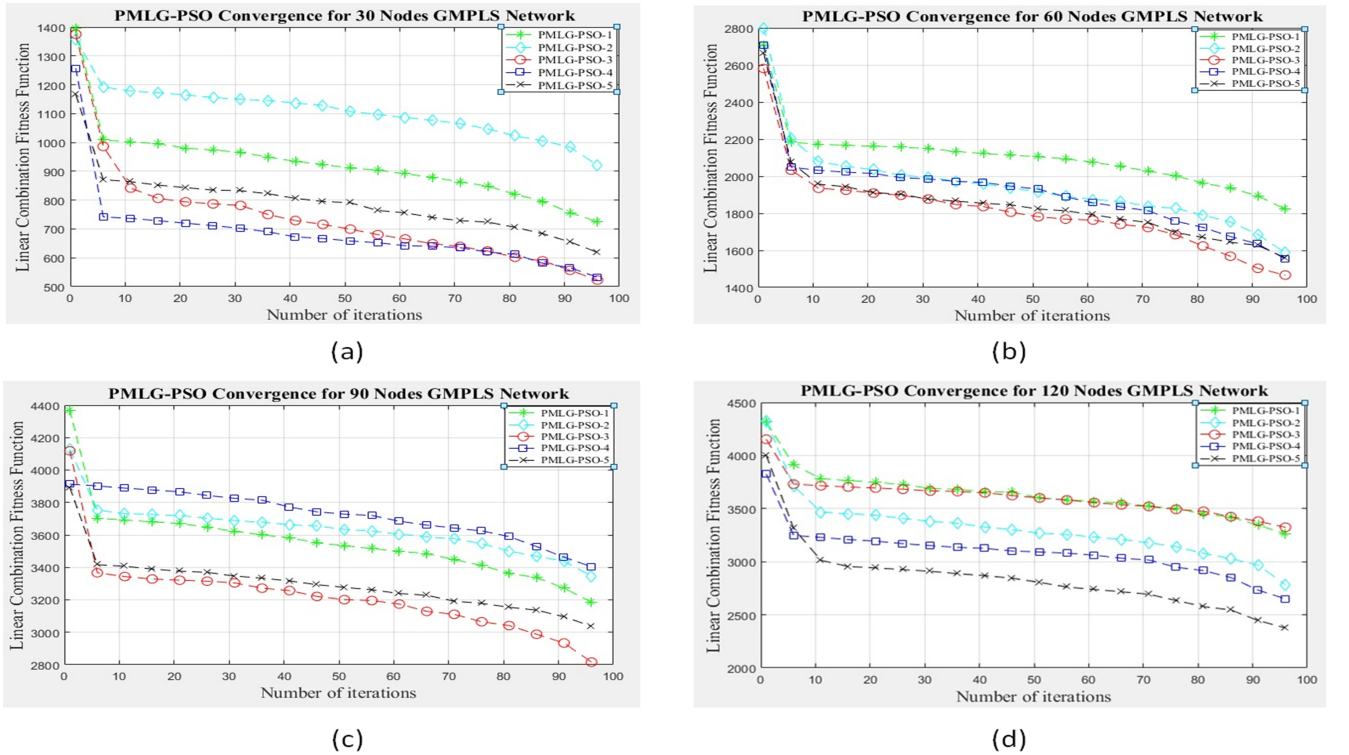


FIGURE 9. Analysis of Convergence of PMLG-PSO Algorithm

TABLE 5. Kruskal-Wallis Test results

$Nodes$	$PMLG-PSO\ Version$	Run_1	Run_2	Run_3	Run_4	Run_5	Run_6	H
80	$PMLG\text{-}PSO_1$	3.8881	5.8994	4.2613	0.0046	2.2354	0.9687	4.61
	$PMLG\text{-}PSO_2$	10.02	7.0455	2.126	3.4187	1.8147	4.1999	
	$PMLG\text{-}PSO_3$	5.9603	5.2494	6.908	1.1467	4.3732	1.7449	
	$PMLG\text{-}PSO_4$	4.0994	2.8054	0.5452	3.5105	1.0819	6.2051	
	$PMLG\text{-}PSO_5$	0.7129	0.489	0.802	3.992	7.6533	1.2520	
	$PMLG\text{-}PSO_6$	2.5965	4.064	2.8822	1.5261	2.3465	2.5549	
90	$PMLG\text{-}PSO_1$	3.2645	0.4765	4.5702	7.0599	0.0219	0.0532	2.25
	$PMLG\text{-}PSO_2$	2.0391	5.555	2.3688	1.6442	4.0108	4.5601	
	$PMLG\text{-}PSO_3$	1.0528	0.7591	4.0091	2.1252	1.1673	3.1011	
	$PMLG\text{-}PSO_4$	3.8526	2.8964	1.1428	4.9318	3.1934	0.6237	
	$PMLG\text{-}PSO_5$	1.1671	1.5639	3.4349	2.1304	3.4083	2.0638	
	$PMLG\text{-}PSO_6$	3.1653	3.089	6.4102	2.7346	0.7212	2.5105	
100	$PMLG\text{-}PSO_1$	0.0164	6.0832	4.7698	0.8023	5.8092	1.0933	1.75
	$PMLG\text{-}PSO_2$	2.0559	3.731	0.1134	3.7552	1.3694	7.1996	
	$PMLG\text{-}PSO_3$	0.1987	7.1935	0.6988	5.5008	4.5798	0.2009	
	$PMLG\text{-}PSO_4$	1.2743	1.8725	9.7466	2.2404	1.7511	3.5788	
	$PMLG\text{-}PSO_5$	2.6096	1.9812	2.7777	1.8384	3.5714	3.5314	
	$PMLG\text{-}PSO_6$	8.4271	4.3661	2.2835	2.6985	2.867	4.0630	
110	$PMLG\text{-}PSO_1$	0.1169	5.847	0.6445	3.7258	4.5795	0.6803	4.66
	$PMLG\text{-}PSO_2$	6.1731	3.8477	0.5514	0.3387	0.4063	1.4105	
	$PMLG\text{-}PSO_3$	1.7349	5.2295	6.4817	1.3506	5.1806	2.0496	
	$PMLG\text{-}PSO_4$	2.4049	1.6217	4.8202	3.3051	6.1764	1.9375	
	$PMLG\text{-}PSO_5$	3.3695	0.7787	0.9641	2.7674	2.1304	0.886	
	$PMLG\text{-}PSO_6$	2.5208	2.0252	0.849	4.4458	3.9305	1.8917	

mization (PSO), an advance version as adaptive particle swarm optimization (APSO), dolphin algorithm (DA) and bat algorithm (BAT). The experiments included a development of PSO, APSO, BAT and DA algorithms for the described MCOP optimization problem. Each algorithm is simulated using the same *MATLAB* version, and results are collected to figure out the exploration problem in the algorithms (if exists during the iterations). The results for both fitness functions (traffic load balancing and resource provisioning costs) were collected separately, as shown in Fig. 10 and Fig. 11, respectively. Fig. 10 and Fig. 11 show the convergence activity and how each algorithm suffers from the local/global optima problem. In Fig. 10, for all its cases (a, b, c, and d), algorithms (PSO, APSO, DA and BAT), apart from *PMLG-PSO*, had drastically failed to deal with the exploration problem and once their searching agents were trapped in their local/ global optima, the algorithms could not converge properly to find the optimal solutions for the resource provision fitness function. Consequently, these algorithms failed to find optimal solutions, and the result in this case is incomplete/ sub-optimal solution.

For example, in Fig. 10 (d), the figure shows that PSO, APSO, DA and BAT algorithms, are poorly deteriorated to converge properly until the end of iteration because of a local/global optima problem. As a reference, it is appeared that PSO and APSO have fallen to a local optima and cannot converge after the 14^{th} iteration. Likewise, for the traffic load balancing costs function, it can be seen in Fig. 11, that PSO, APSO, DA and BAT algorithms had generated sub-optimal solutions produced worse convergence activity with respect to the fitness function.

As a case study, let us consider Fig. 11 (a), where PSO algorithm exhibits an exploration problem and thus, cannot converge after 15^{th} iteration and repeats (produces) the same fitness function values. DA algorithm experiences the exploration problem in various stages of convergence during the iterations, such as 25^{th} to 30^{th}, 36^{th} to 60^{th}, and 66^{th} to 78^{th} iteration. While on the other side, proposed *PMLG-PSO* algorithm has better convergence rate when compared to other algorithms (PSO, APSO, DA and BAT) in forms of generating a number of optimal solutions. It can be concluded from Fig. 10 and Fig. 11 that the proposed *PMLG-PSO* has not only solved local/ global optima problem but also has fastest convergence rate, for different MPLS/GMPLS network sizes, compare to other mentioned algorithms (PSO, APSO, BAT and DA).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2934946, IEEE Access

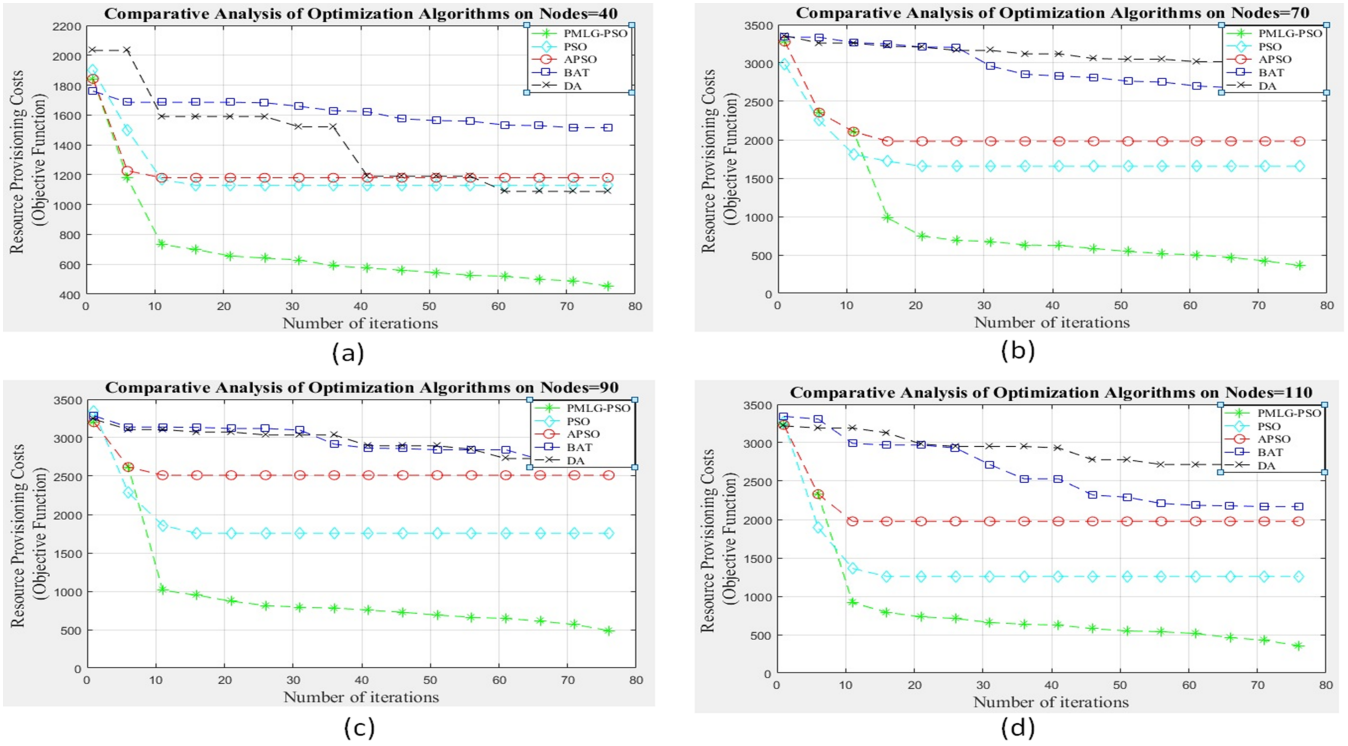


FIGURE 10. Comparison of Meta-heuristic algorithms w.r.t Convergence ratio for resource Provisioning Costs Function

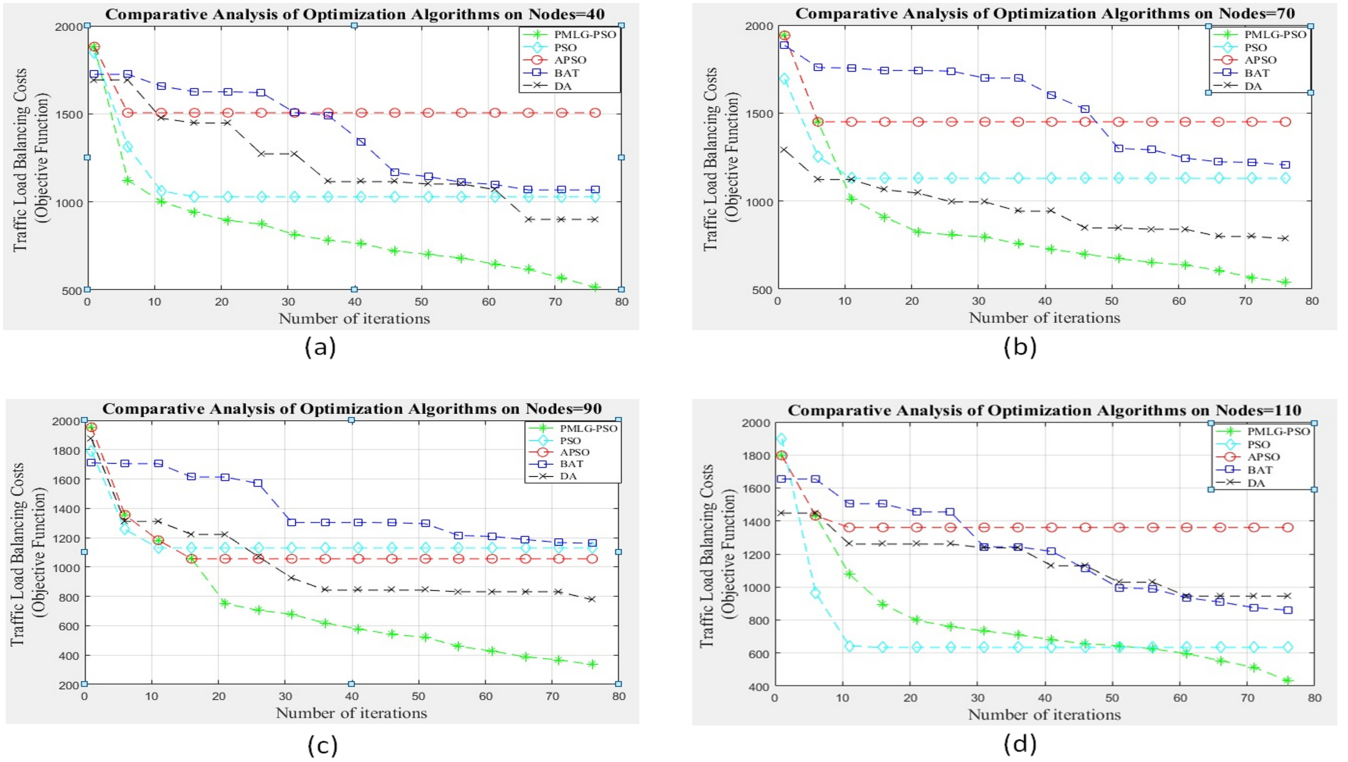


FIGURE 11. Comparison of Meta-heuristic algorithms w.r.t Convergence ratio for Traffic Load Balancing function

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2934946, IEEE Access

TABLE 6. Statistical based Comparative Analysis of 30 Nodes Network

Resource Provisioning Costs Objective Function					
	PSO	*APSO*	*BAT*	*DA*	*PMLG–PSO*
OS	228	168.3926	475.9140	537.79	66.3332
Mean	585.3900	758.9721	902.6723	615.2437	400.9275
SD	162.2130	318.7124	156.0202	49.6646	210.9559
Traffic Load Balancing Costs Objective Function					
	PSO	*APSO*	*BAT*	*DA*	*PMLG–PSO*
OS	50.2371	284.1536	65.8446	141.07	43.0134
Mean	771.5276	1168	1056.4	474.1919	467.4116
SD	233.5189	259.6976	348.5924	160.7355	224.6925

TABLE 7. Statistical based Comparative Analysis of 50 Nodes Network

Resource Provisioning Costs Objective Function					
	PSO	*APSO*	*BAT*	*DA*	*PMLG–PSO*
OS	644	448.7235	877.32	774.38	80.8595
Mean	1016.2	1381.8	1508.6	908.4148	465.0897
SD	221.678	507.7226	268.8695	68.1039	275.8044
Traffic Load Balancing Costs Objective Function					
	PSO	*APSO*	*BAT*	*DA*	*PMLG–PSO*
OS	295.02	880.3437	354.54	505.34	103.0016
Mean	1230.8	1968.1	1950.1	836.5914	490.1594
SD	423.7258	378.0275	511.4425	181.6391	254.2641

TABLE 8. Statistical based Comparative Analysis of 80 Nodes Network

Resource Provisioning Costs Objective Function					
	PSO	*APSO*	*BAT*	*DA*	*PMLG–PSO*
OS	1070	844.74	1570.2	1035.8	89.738
Mean	1761	2179.8	2555.7	1213.1	685.7291
SD	304.7261	827.6866	383.1460	97.4751	532.9176
Traffic Load Balancing Costs Objective Function					
	PSO	*APSO*	*BAT*	*DA*	*PMLG–PSO*
OS	430.44	1428.7	735.68	585.2	121.78
Mean	1890.8	3326.4	2679.2	907.0565	687.1046
SD	757.4162	627.0519	647.9020	184.7952	608.8075

TABLE 9. Statistical based Comparative Analysis of 100 Nodes Network

Resource Provisioning Costs Objective Function					
	PSO	*APSO*	*BAT*	*DA*	*PMLG–PSO*
OS	1585	1315.2	1786.6	1333.8	80.878
Mean	2230.5	2740.2	3066.1	1512.5	765.0592
SD	279.4764	1063.5	504.7764	108.3417	643.6658
Traffic Load Balancing Costs Objective Function					
	PSO	*APSO*	*BAT*	*DA*	*PMLG–PSO*
OS	634.44	2227	983.44	736.77	140.03
Mean	2226.9	2227	3634.7	1013.1	722.7989
SD	886.1651	641.8836	881.7565	192.8892	776.1407

D. STATISTICAL ANALYSIS

For further investigation, the same setup of the *PMLG-PSO* algorithm is run for 100 times to extract and then evaluate the overall performance of the algorithms using statistical parameters such optimal fitness function costs, MEAN value and Standard Deviation values. The output results are given in the following tables.

Table 6 to Table 9, represent the comprehensive comparative analysis of optimization algorithms over 30, 50, 80 and 100 nodes networks. While reviewing each algorithm output in the form of an optimal solution in the given tables, it can be concluded that for each network scale, the optimum solutions are obtained from the proposed *PMLG-PSO* algorithm. As the minimum function costs are obtained from the proposed algorithm and compare to other techniques. Similarly, for MEAN values, *PMLG-PSO* provides optimal MEAN values compare to other algorithms (PSO, APSO, DA and BAT). Here, minimum MEAN values are considered as the optimal values. As all mentioned algorithms have stochastic nature, therefore, the standard deviation parameters were used to monitor the deviation values of each run during all 100 iterations. For this case, DA algorithm has got the minimum standard deviation value but has sub-optimal values, whereas the other acceptable choice as for standard deviation values along with optimal solutions are from *PMLG-PSO* algorithm.

The tables (Table 6 to Table 9) give us conclusion that the proposed *PMLG-PSO* algorithm has maximum probability for generating optimal solutions even after implementing it into multiple numbers of simulations.

VI. CONCLUSION

The exponential increase in the data traffic demands substantial throughput improvements to avoid network congestion. Approaches with different algorithms were introduced to tackle this problem. The purpose of these algorithms is to distribute the network traffic that will assure the network efficiency and reliability. PSO is one of the many well-known meta-heuristic algorithms that are used to address the above challenge. However, PSO suffers from convergence problem. This is suggesting that a *standard* PSO may not be the best suitable algorithm for multi-objective-based approaches.

This paper has considered multi-path between routers of the MPLS/GMPLS networks and proposes a novel "*Pareto based Modified Local Global Particle Swarm Optimization (PMLG-PSO)*" algorithm for an optimization of two objective functions: the resource provisioning costs and traffic ink balancing costs. Our proposed

PMLG-PSO technique solved the exploration problem for the given constrained based multiple objective optimization (MCOP) problems. From the proposed algorithm, a Pareto based approach was adapted to create Pareto front graphs as well as optimal solutions.

To analyse the performance of our proposed *PMLG-PSO* algorithm, a number of experiments were conducted for obtaining: (i) a Pareto front for both objective functions; (ii) convergence graphs with different number of nodes, (iii) for comparing the convergence rate with other algorithms; (iv) statistical analysis of *PMLG-PSO* in comparison with the *standard* PSO, APSO, Bat and DA, and (v) the Kruskal-Wallis test.

While investigating the *PMLG-PSO* algorithm, we have shown that our proposed algorithm can successfully generate non-dominant solutions in the form of Pareto front, even when the MPLS/GMPLS network size varies, this is shown in Fig. 8. The algorithm can also effectively produce optimal solutions, as it is shown in convergence graphs in Fig. 9. By comparing *PMLG-PSO* with a *standard* PSO, APSO, Bat and DA applied to a different number of nodes, we have shown that PSO, APSO, DA, and BAT algorithms suffer from an exploration problem for the MCOP optimization and therefore cannot generate the optimal solutions (see Fig. 10 and Fig. 11). On the contrary, the presented *PMLG-PSO* can overcome the drawbacks of the former PSO algorithms, it converges smoothly, and successfully finds the optimal path that satisfies the minimization of both objective requirements and does it without any trade-off. Furthermore, the *PMLG-PSO* algorithm was compared to PSO, APSO, Bat and DA algorithms using statistical parameters such as MEAN, standard deviation, and optimal solutions (Table 6, 7, 8, and 9). The comparison has shown a dominance of the proposed *PMLG-PSO* algorithm over the other algorithms and its capability to produce better results. Because the *PMLG-PSO* algorithm is stochastic in its nature, the Kruskal-Wallis test was also conducted. From the test's results was concluded that the proposed *PMLG-PSO* algorithm does not suffer from 'generation of abrupt results that stochastically dominates each other, and therefore, proved its superiority producing betters results when compared to the *standard* PSO, APSO, Bat and DA algorithms.

REFERENCES

[1] R. G. Garroppo, S. Giordano, and L. Tavanti, "A survey on multi-constrained optimal path computation: Exact and approximate algorithms," Computer Networks, vol. 54, no. 17, pp. 3081–3107, 2010.

[2] M. Masood, M. M. Fouad, and I. Glesk, "A pareto based approach with elitist learning strategy for mpls/gmps networks," in 2017 9th Computer Science and Electronic Engineering (CEEC), pp. 71–76, IEEE, 2017.

[3] Y. Lee, Y. Seok, Y. Choi, and C. Kim, "A constrained multipath traffic engineering scheme for mpls networks," in 2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No. 02CH37333), vol. 4, pp. 2431–2436, IEEE, 2002.

[4] C. C. Coello, G. T. Pulido, and E. M. Montes, "Current and future research trends in evolutionary multiobjective optimization," pp. 213–231, 2005.

[5] G. M. Lee and J. K. Choi, "Multipath routing for traffic engineering with flow classification in gmpls network," in 2006 Asia-Pacific Conference on Communications, pp. 1–5, IEEE, 2006.

[6] Z. Du, S. Li, Y. Sun, and N. Li, "Adaptive particle swarm optimization algorithm based on levy flights mechanism," in 2017 Chinese Automation Congress (CAC), pp. 479–484, IEEE, 2017.

[7] C.-F. Wang and K. Liu, "A novel particle swarm optimization algorithm for global optimization," Computational intelligence and neuroscience, vol. 2016, p. 48, 2016.

[8] S. U. Khan, S. Yang, L. Wang, and L. Liu, "A modified particle swarm optimization algorithm for global optimizations of inverse problems," IEEE Transactions on Magnetics, vol. 52, no. 3, pp. 1–4, 2015.

[9] S. Zhang, C. K. Lee, H. K. Chan, K. L. Choy, and Z. Wu, "Swarm intelligence applied in green logistics: A literature review," Engineering Applications of Artificial Intelligence, vol. 37, pp. 154–169, 2015.

[10] A. Amuthan and K. D. Thilak, "Improved ant colony algorithms for eliminating stagnation and local optimum problemâĂŤa survey," in 2017 International Conference on Technical Advancements in Computers and Communications (ICTACC), pp. 97–101, IEEE, 2017.

[11] S. M. Mousavi, M. Moghadasi, and G. Fazekas, "Dynamic resource allocation using combinatorial methods in cloud: A case study," in 2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), pp. 000073–000078, IEEE, 2017.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization, doi: 10.1109/icnn. 1995.488968," in Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948, 1995.

[13] C. Akjiratikarl, P. Yenradee, and P. R. Drake, "Pso-based algorithm for home care worker scheduling in the uk," Computers & Industrial Engineering, vol. 53, no. 4, pp. 559–583, 2007.

[14] M. A. A. Aziz, M. N. Taib, and N. M. Hussin, "Assignments acceptance strategy in a modified pso algorithm to elevate local optima in solving class scheduling problems," in 2010 6th International Colloquium on Signal Processing & its Applications, pp. 1–5, IEEE, 2010.

[15] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," IEEE transactions on evolutionary computation, vol. 10, no. 3, pp. 281–295, 2006.

[16] C. Dong, G. Wang, Z. Chen, and Z. Yu, "A method of self-adaptive inertia weight for pso," in 2008 international conference on computer science and software engineering, vol. 1, pp. 1195–1198, IEEE, 2008.

[17] S. M. A. Salehizadeh, P. Yadmellat, and M. B. Menhaj, "Local optima avoidable particle swarm optimization," in 2009 IEEE Swarm Intelligence Symposium, pp. 16–21, IEEE, 2009.

[18] L. Barbulescu, J.-P. Watson, and L. D. Whitley, "Dynamic representations and escaping local optima: Improving genetic algorithms and local search," AAAI/IAAI, vol. 2000, pp. 879–884, 2000.

[19] K. E. Parsopoulos and M. N. Vrahatis, "Unified particle swarm optimization for solving constrained engineering optimization problems," in International Conference on Natural Computation, pp. 582–591, Springer, 2005.

[20] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," IEEE transactions on evolutionary computation, vol. 8, no. 3, pp. 204–210, 2004.

[21] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," in Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706), pp. 174–181, IEEE, 2003.

[22] P. J. Angeline, "Using selection to improve particle swarm optimization," in 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360), pp. 84–89, IEEE, 1998.

[23] H. Y. Cho, J. Y. Lee, and B. C. Kim, "Multi-path constraint-based routing algorithms for mpls traffic engineering," in IEEE International Conference on Communications, 2003. ICC'03., vol. 3, pp. 1963–1967, IEEE, 2003.

[24] U. F. Siddiqi, Y. Shiraishi, and S. M. Sait, "Multi-constrained route optimization for electric vehicles (evs) using particle swarm optimization (pso)," in 2011 11th International Conference on Intelligent Systems Design and Applications, pp. 391–396, IEEE, 2011.

[25] C. Jiuxin, S. Xuesheng, Z. Xiao, L. Bo, and M. Bo, "Efficient multi-objective services selection algorithm based on particle swarm optimization," in 2010 IEEE Asia-Pacific Services Computing Conference, pp. 603–608, IEEE, 2010.

[26] D. Tan, "Chaos particle swarm optimization algorithm for multi-objective constrained optimization problems," in Future Wireless Networks and Information Systems, pp. 469–476, Springer, 2012.

[27] L. Shi, Z.-H. Zhan, H.-Q. Yuan, J.-J. Li, and J. Zhang, "Distributed co-evolutionary particle swarm optimization using adaptive migration strategy," in 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–7, IEEE, 2017.

[28] A. Farrel and I. Bryskin, GMPLS: architecture and applications. Elsevier, 2005.

[29] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing ospf weights," in Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064), vol. 2, pp. 519–528, IEEE, 2000.

[30] F. Li and J. Chen, "Mpls traffic engineering load balance algorithm using deviation path," in 2012 International Conference on Computer Science and Service System, pp. 601–604, IEEE, 2012.

[31] E. K. Burke, G. Kendall, et al., Search methodologies. Springer, 2005.

[32] E. Zitzler, M. Laumanns, and S. Bleuler, "A tutorial on evolutionary multiobjective optimization," in Metaheuristics for multiobjective optimisation, pp. 3–37, Springer, 2004.

[33] Z. J. Zhan Z., "Adaptive particle swarm optimization," in Adaptive Particle Swarm Optimization, pp. 3–37, Springer, 2008.

[34] X.-F. Xie, W.-J. Zhang, and Z.-L. Yang, "Adaptive particle swarm optimization on individual level," in 6th International Conference on Signal Processing, 2002., vol. 2, pp. 1215–1218, IEEE, 2002.

[35] M. Masood, M. M. Fouad, and I. Glesk, "Proposing bat inspired heuristic algorithm for the optimization of gmpls networks," in 2017 25th Telecommunication Forum (TELFOR), pp. 1–4, IEEE, 2017.

[36] A. Kaveh and N. Farhoudi, "A new optimization method: Dolphin echolocation," Advances in Engineering Software, vol. 59, pp. 53–70, 2013.