Wireless Networks (2020) 26:4753–4766
https://doi.org/10.1007/s11276-019-01993-w

Nature-inspired meta-heuristics approaches for charging plug-in hybrid electric vehicle

Pandian Vasant[1,2] · Jose Antonio Marmolejo[3] · Igor Litvinchev[4] · Roman Rodriguez Aguilar[5]

Published online: 18 May 2019

Abstract

Currently, there is a remarkable focus on green technologies for taking steps towards more use of renewable energy sources within the sector of transportation and also decreasing pollution. At this point, employment of plug-in hybrid electric vehicles (PHEVs) needs sufficient charging allocation strategy, by running smart charging infrastructures and smart grid systems. In order to daily usage of PHEVs, daytime charging stations are required and at this point, only an appropriate charging control and a management of the infrastructure can lead to wider employment of PHEVs. In this study, four swarm intelligence based optimization techniques: particle swarm optimization (PSO), gravitational search algorithm (GSA), accelerated particle swarm optimization, and hybrid version of PSO and GSA (PSOGSA) have been applied for the state-of-charge optimization of PHEVs. In this research, hybrid PSOGSA has performed very well in producing better results than other stand-alone optimization techniques.

Keywords Nature-inspire metaheuristics · Hybrid optimization · Swarm intelligence · Artificial intelligence · State-of-charge optimization · Plug-in hybrid electric vehicle

1 Introduction

Researches on green technologies for transportation sector are gaining popularity among the research communities from different areas. In this wake, Plug-in hybrid electric vehicles (PHEVs) have great future because of their charge storage system and charging facilities from traditional grid system. Some researchers have shown that electrification of transport sector can cause a large amount of degradation in greenhouse gas emissions. Future transportation sector will depend much on the advancement of this emerging field of vehicle optimization. As a recent research interest regarding improving general fuel efficiency over a wider capacity battery system, the plug-in hybrid electric vehicles (PHEVs) can be charged thoroughly thanks to conventional power grid system. That also makes it possible the vehicles to be run in “all-electric-range” (AER) continuously. All-electric vehicles or AEVs is a kind of transport that use electric power as only sources to run the system. An improved adoption of PHEVs can play an important role over alternative energy integration into traditional grid systems because of that plug-in hybrid electric vehicles can employ all of these related strategies via just a connection to the smart grid [1]. It is needed to employ efficient mechanisms—functions and algorithms running in the context of smart grid technologies, for solving advanced problems such as cost reduction, energy management, efficient charging station via different objectives and also system constraints [2].

✉ Pandian Vasant
pandian.vasant@tdtu.edu.vn; vasantglobal@gmail.com

Roman Rodriguez Aguilar
rrodrigueza@up.edu.mx

1 Modeling Evolutionary Algorithms Simulation and Artificial Intelligence (MERLIN), Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam

2 Universiti Teknologi Petronas, Seri Iskandar, Malaysia

3 Facultad de Ingenieria, Universidad Panamericana, Augusto Rodin 498, 03920 Mexico City, Mexico

4 Nuevo Leon State University, Monterrey, Mexico

5 Escuela de Ciencias Económicas y Empresariales, Universidad Panamericana, Augusto Rodin 498, 03920 Mexico City, Mexico

It is important that about 62% of the vehicles in the United States (US) will comprise of PHEVs within the year 2050, according to the statistics provided by the Electric Power Research Institute (EPRI) [3]. Here, one remarkable objective is making the proper communication between the PHEV and the power grid more practical. For the maximization of customer contentment and minimization of burdens on the grid, a complicated control appliance will need to be addressed in order to govern multiple battery loads from a numbers of PHEVs properly [4]. The total demand pattern will also have an important impact on the electricity production owing to differences in the needs of the PHEVs parked in the deck at certain time [5]. Proper management can ensure strain minimization of the grid and enhance the transmission and generation of electric power supply.

According to the locations, control of PHEV charging can be in whether household charging or public charging. The introduced optimization here deals with the public charging station for plug-in vehicles. That's because most of PHEV charging is expected to take place in public charging location [6]. Wide diffusion of PHEVs in the market depends on a well-organized charging infrastructure. On the traditional power grid, there will be extra stress by the power demand from this new load [7]. Consequently, a good number of PHEV charging stations with suitable facilities are essential to be built for recharging electric vehicles, for this some strategies have been proposed by the researchers [8]. Charging stations are needed to be built at workplaces, markets/shopping malls and home. Boyle [9] suggested the necessity of building new smart charging station with powerful communication between utilities along with sub-station controlling infrastructure over proper energy utilization and grid stability. Also, sizeable energy storage, cost minimization; Quality of Services (QoS) and intelligent charging station for optimal power are underway [10]. In this stage, numerous techniques and methods were recommended for deployment of PHEV charging stations [11].

Among the research efforts, it is aimed to accomplish practical ways for proper interaction between the PHEV and the power grid. In this context, an advanced mechanism of control will be required to be designed for governing multiple loads of battery from a number of PHEVs appropriately, in the context of maximizing customer pleasure and minimizing the freightage on the grid [6].

Swarm intelligence came from the mimic of the living colony such as ant, bird, and fish in nature, which shows unparalleled excellence in swarm than in single in food seeking or nest building. Drawing inspiration from this, researches design many algorithms simulating colony living, such as ant colony optimization (ACO) algorithm [12], particle swarm optimization (PSO) algorithm [13], artificial bee colony (ABC) algorithm [14], and gravitational search algorithm (GSA) [15], which shows excellent performance in dealing with complex optimization problems [16]. The intrinsic characteristics of all the population-based metaheuristic algorithms like particle swarm optimization (PSO) and gravitational search algorithm (GSA) are to maintain a good compromise between exploration and exploitation in order to solve the complex optimization problems [17].

The performance of PHEV depends upon proper utilization of electric power which is solely affected by the battery state-of-charge (SoC). In Plug-in hybrid electric vehicles (PHEVs), a key parameter is the state-of-charge (SoC) of the battery as it is a measure of the amount of electrical energy stored in it. It is analogous to fuel gauge on a conventional internal combustion (IC) car [17]. State-of-charge determination becomes an increasingly vital issue in all the areas that include a battery. Previous operation policies made use of voltage limits only to guard the battery against deep discharge and overcharge. Currently, battery operation is changing to what could rather be called battery management than simply protection. For this improved battery control, the battery SoC is a key factor indeed [18].

A charging station is one way that the operator of an electrical power grid can adapt energy production to energy consumption, both of which can vary randomly over time. Generally, PHEVs in a charging station are charged during times when production exceeds consumption and are discharged at times when consumption exceeds production [19]. In this study, it is needed to perform an in-depth study for maximizing the average SoC for making intelligent energy allocation simpler for PHEVs in a charging station.

For covering the general objective and its sub-ones, four swarm intelligence-based techniques: particle swarm optimization (PSO), gravitational search algorithm (GSA), accelerated particle swarm optimization (APSO), and hybrid version of PSO and GSA (PSOGSA) were applied for solving the particular optimization problem hence presents comparative study on these four techniques. Results obtained thanks to this study are important findings for such an advanced real-world optimization problem of charging plug-in hybrid electric vehicle.

2 Mathematical formulation of the charging optimization

Assume that, we have a charging station employing a capacity of total power called as P. Total N numbers of PHEVs need to be served in a day (24 h). In order to make the system more effective, the suggested system should allow PHEVs for leaving the charging station before their expected leaving time. It is worth to mention that; each PHEV is regarded to be plugged-into the charging station once. The aim is to separate power intelligently for each PHEV coming to the charging station. The state-of-charge is the main parameter which needs to be maximized in order to allocate power efficiently. For this, the fitness function considered in this chapter is the maximization of average SoC, hence, allocates energy for PHEVs at the next time step. The constraints considered are: charging time, present SoC and price of the energy.

The fitness function is defined as:

$$\text{Max } J(k)=\sum_i w_i(k)SoC_i(k+1) \tag{1}$$

$$w_i(k)=f\left(C_{r,i}(k), T_{r,i}(k), D_i(k)\right) \tag{2}$$

$$C_{r,i}(k)=(1-SoC_i(k))\cdot C_i \tag{3}$$

where $C_{r,i}(k)$ is the battery capacity (remaining) needed to be filled for i no. of PHEV at time step k; C_i is the battery capacity (rated) of the i no. of PHEV; remaining time for charging a particular PHEV at time step k is expressed as $T_{r,i}(k)$; the price difference between the real-time energy price and the price that a specific customer at the i no. of PHEV charger is willing to pay at time step k is presented by $D_i(k)$; $w_i(k)$ is the charging weighting term of the i no. of PHEV at time step k (a function of charging time, price of the energy and also present SoC); $SoC_i(k+1)$ means the state for charge of the i no. of PHEV at time step $k+1$.

The weighting term considered here corresponds to a bonus, which is proportional to the attributes associated with a PHEV. As an example, if the considered PHEV comes with a lower initial SoC and less charging time (as remaining) and the driver is willing to pay a higher price, this particular PHEV battery charger will receive more power by the system:

$$w_i(k)\propto\left[C_{r,i}(k)+D_i(k)+1/T_{r,i}(k)\right] \tag{4}$$

The charging current is also assumed to be constant over Δt.

$$[SoC_i(k+1)-SoC_i(k)]\cdot Cap_i=Q_{\mathrm{i}}=I_i(k)\Delta t \tag{5}$$

$$SoC_i(k+1)=SoC_i(k)+\mathrm{I_i}(k)\Delta t/Cap_i \tag{6}$$

where the sample time Δt is defined by the charging station operators, and $\mathrm{I_i}(k)$ is the charging current over Δt.

The model of battery is considered as a capacitor cycle, where is the capacitance of battery (Farad). The model is briefly as:

$$C_i.\frac{dV_i}{dt}=I_i \tag{7}$$

Therefore, one can assume here that the change of voltage to be linear over a small time interval,

$$C_i.[V_i(k+1)-V_i(k)]/\Delta t=I_i \tag{8}$$

$$V_{\mathrm{i}}(k+1)-V_i(k)=I_i\Delta t/C_i \tag{9}$$

Because the decision variable is the power allocated to the vehicles, replacing $I_i(k)$ with $P_i(k)$,

$$\mathrm{I_i}(k)=P_i(k)/0.5\times[V_{\mathrm{i}}(k+1)-V_i(k)] \tag{10}$$

$$V_{\mathrm{i}}(k+1)=\sqrt{\frac{2P_i(k)\Delta t}{C_i}+V_i^2(\mathrm{k})} \tag{11}$$

Substituting (10) into (6) yields

$$SoC_i(k+1)=SoC_i(k)+\frac{P_i(k)\Delta t}{0.5.C_i.\left[\sqrt{\frac{2P_i(k)\Delta t}{C_i}+V_i^2(\mathrm{k})}+V_{\mathrm{i}}(k)\right]} \tag{12}$$

Finally, the objective function becomes

$$J(k)=\sum w_i.\left[SoC_i(k)+\frac{2P_i(k)\Delta t}{0.5.C_i.\left[\sqrt{\frac{2P_i(k)\Delta t}{C_i}+V_i^2(\mathrm{k})}+V_{\mathrm{i}}(k)\right]}\right] \tag{13}$$

There are two kind of inequality constraints used here to optimize the fitness function: (1) power from the charging station operator and (2) individual PHEV's state-of-charge (SoC). Power obtained from the utility ($\mathrm{P_{utility}}$) and the maximum power ($\mathrm{P_{i,max}}$) absorbed by a specific PHEV correspond to the constraints regarding primary energy. The overall charging efficiency of a particular charging station is described by η. From the system point of view, charging efficiency is supposed to be constant at any given time step. Maximum battery SoC limit for the i no. of PHEV is $\mathrm{SoC_{i,max}}$. When $\mathrm{SoC_i}$ reaches the values close to $\mathrm{SoC_{i,max}}$, the i no. of battery charger passes to a mode of standby. Here, limits by the constraint $\Delta\mathrm{SoC_{max}}$ is applied over the state of charge ramp rate.

Table 1 represents the fitness function parameters, which were adjusted—updated for the optimization process. There are total three (03) kinds of parameter: fixed, variables and constraints. Total charging time is fixed to 20 min and charging station efficiency assumed to be 0.9. The values are retrieved from various literatures [4, 28]. Moreover, State-of-Charge is in the range of 0.2–0.8 [29].

2.1 Swarm intelligence algorithms for charging optimization

The chosen four swarm intelligence algorithms for the charging optimization problem of this study can be explained briefly as follows, in order to enable readers to have enough idea about capabilities of the related algorithms regarding effective, intelligent optimization:

Table 1 Parameter settings of the objective function

Parameter	Values
Fixed parameters	Maximum power, $P_{i,max}$ = 6.7 kWh
	Charging station efficiency, η = 0.9
	Total charging time, Δt = 20 min
	Power allocation to each PHEV: 30 W
Variables	$0.2 \leq$ state-of-charge (SoC) ≤ 0.8
	Waiting time $\leq$ 30 min (1800 s)
	16 kWh $\leq$ battery capacity (C_i) $\leq$ 40 kWh
Constraints	$\sum_i P_i(k) \leq P_{utility}(k) \times \eta$
	$0 \leq P_i(k) \leq P_{i,max}(k)$
	$0 \leq SoC_i(k) \leq SoC_{i,max}$
	$0 \leq SoC_i(k+1) - SoC_i(k) \leq \Delta SoC_{max}$

2.1.1 Particle swarm optimization (PSO)

PSO is not only a technique, but also is known as a technique of evolutionary computation as introduced by Kennedy and his colleague Eberhart [15]. This technique was briefly an inspiration from social behavior shown by bird flocking. In detail, there are some particles (in other words, candidate solutions) in the algorithm structure and these particles fly within the search space for searching the best solution. In the meantime, they all are affected by the best solution (best particle) along their travel done. In other words, particles consider both best solutions, which are their own top solution and the best one, which has been detected so far. Generally, each of the particles should focus on its current velocity, current position, and position difference according to *pbest* and the *gbest* separately, for updating its position. PSO is briefly started to run over a randomly placed group of particles (solutions) and after that it runs over searching for optimum solution(s) by updating generations. In each iteration, all particles are updated by considering two "*best*" values, which are respectively the best solution (fitness) a particle has achieved so far. (This is called as "*pbest*". Also, the fitness value is stored). And the "*best*" value got so far by any particle in the related population (This is the global best value and called as the "*gbest*").

PSO was mathematically modelled as followed as:

$$V_i^{t+1} = wv_i^t + c_1.rand.\left(pbest_i - x_i^t\right) + c_2.rand.\left(gbest - x_i^t\right) \tag{14}$$

$$x_i^{t+1} = x_i^t + V_i^{t+1} \tag{15}$$

where v_i^t means velocity of the particle i at the iteration t. w is briefly a weighting function as follows:

$$\omega = \omega_{max} - \frac{w_{max} - \omega_{min}}{Itre_{max}} Itre \tag{16}$$

The principal steps in PSO can be summarized as follows (Fig. 1).

The values known as the most appropriate ones for ω_{min} and ω_{max} are respectively 0.4 and 0.9 [22]. For c_1 and c_2, it is 1–2 [7] while 2 is known as the most appropriate one in different cases [23]. On the other hand, rand means a random number as between 0 and 1 [7], x_i^t is the current position of particle *i* at iteration *t*, $pbest_i$ means the pbest of the agent *i* at the iteration *t*.

2.1.2 Accelerated particle swarm optimization (APSO)

Accelerated PSO was developed in order to accelerate the convergence of the algorithm is to use the global best only, by Yang [24] at Cambridge University in 2007. PSO and APSO-based optimizations have already been studied by the researchers for optimal design of substation grounding grid [25], performance analysis regarding MIMO radar waveform [26], designing frame structures [27], dual channel speech enhancement [28] and a faster path planner [29] etc.

In the algorithmic structure of the APSO, members of a population are called as particles while the population itself is the swarm. APSO briefly runs by initial adjustment as a randomly located population and here, each particle moves randomly according to some factors.

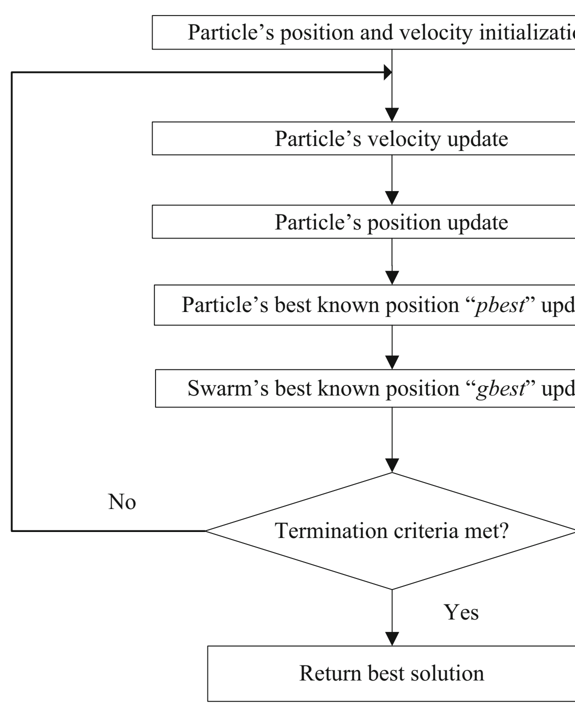

Fig. 1 Flowchart of particle swarm optimization (PSO)

A simplified version that could speed up the convergence of the algorithm is to use the global best only. So, in the APSO [94], the velocity vector is manufactured by a basic formula as where randn is drawn from (0, 1) to replace the second term. The update of the position is simply like-

$$V_i^{t+1} = V_i^t + \alpha.randn(t) + \beta.(g^* - x_i^t) \tag{17}$$

where *randn* is drawn from N (0, 1) and the update of the position is like the standard PSO method. The update of the position can be written in a single step in order to increase the convergence even further, as:

$$x_i^{t+1} = (1 - \beta)x_i^t + \beta.g^* + \alpha r \tag{18}$$

In our simulation, we use [30]:

$$\alpha = 0.7^t \tag{19}$$

Figure 2 shows the flowchart of APSO method.

The typical values for this accelerated PSO are $\alpha \approx$ 0.1–0.4 and $\beta \approx$ 0.1–0.7; nevertheless, $\alpha \approx 0.2$ and $\beta \approx 0.5$ are proposed [25]. In general, any evolutionary search algorithm shows improved performance with a relatively larger population. However, a very large population will cost more in terms of fitness function evaluations without producing significant improvements. In this simulation, the population size is set to 100. The parameter settings for APSO are demonstrated in Table 2.

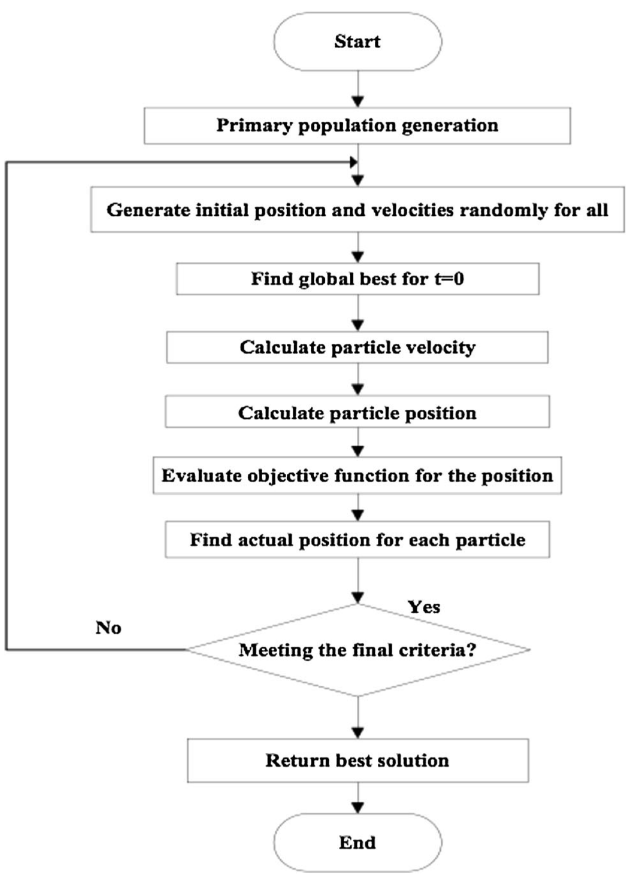

Fig. 2 Flowchart of accelerated particle swarm optimization (APSO)

2.1.3 Gravitational search algorithms (GSA)

GSA is briefly a population based optimization approach as developed by Rashedi et al. [17]. In this algorithm, there are total four specifications of each mass (or agent), which is position, inertial mass, passive gravitational mass and also active gravitational mass and. The position regarding the mass corresponds a solution of a particular problem. In this context, the masses (which are gravitational and inertial) are calculated through a fitness function. GSA briefly includes some agents (in other words, candidate solutions) and masses of these agents are proportional to their value calculated through fitness function. Along the appeared generations, all the masses attract each other by the gravity forces between them. Here, a heavier mass means higher force of attraction. In this context, the heavier masses seem close to the global optimum affects other masses, as proportional to their distances.

GSA-based optimization has already been used for economic dispatch with valve-point effects, optimal sizing and suitable placement for distributed generation (DG) in distribution system, post-outage bus voltage magnitude calculations, optimization of synthesis gas production [31], solving thermal unit commitment (UC) problem [32] and finding out optimal solution for optimal power flow (OPF) problem in a power system [33] etc. by the researchers. Specifically, we are studying the application of the Gravitational Search Algorithm (GSA) method for developing real-time and large-scale optimizations for allocating power.

The gravitational force is expressed as follows:

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}\left(x_j^d(t) - x_i^d(t)\right) \tag{20}$$

where M_{aj} is the active gravitational mass related to agent j, M_{pi} is the passive gravitational mass related to agent i, G (t) is gravitational constant at time t, ε is a small constant and $R_{ij}(t)$ is the Euclidian distance between two agents i and j. The G (t) is calculated as:

Table 2 APSO parameter settings

Parameters	Values
Size of the swarm	100
Maximum no. of steps	100
Alpha, α	0.2
Beta, β	0.5
Maximum iteration	100
Number of runs	50

$$G(t) = G_0 \times \exp(-\alpha \times iter/maxiter) \tag{21}$$

where α and G_0 are descending coefficient and primary value respectively, current iteration and maximum number of iterations are expressed as iter and maxiter. In a problem space with the dimension d, the overall force acting on agent i is estimated as following equation:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t) \tag{22}$$

where $rand_j$ is a random number with interval [0, 1]. From law of motion we know that, an agent's acceleration is directly proportional to the resultant force and inverse of its mass, so the acceleration of all agents should be calculated as follow:

$$ac_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{23}$$

where t a specific is time and M_{ii} is the mass of the object i. The velocity and position of agents are calculated as follow:

$$vel_i^d(t+1) = rand_i \times vel_i^d(t) + ac_i^d(t) \tag{24}$$

$$x_i^d(t+1) = x_i^d(t) + vel_i^d(t+1) \tag{25}$$

where $rand_i$ is a random number with interval [0, 1]. Moreover, the step involves in optimization using GSA is shown Fig. 3.

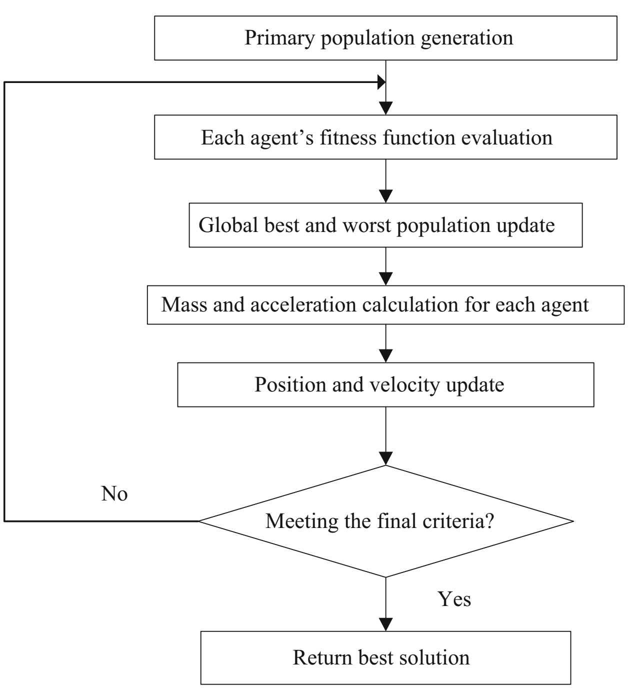

Fig. 3 Flowchart of gravitational search algorithm

In the technique of Gravitational Search Algorithm, the agents are firstly located via random values as each of them is accepted as a candidate solution. Following that, velocities of all agents are adjusted according to (23) while the gravitational constant, overall forces, and accelerations are set respectively by using the Eqs. (20) (21), and (22). The Eq. (24) is used for calculating positions of the agents. In this context, run of the GSA will be stopped when the stopping criterion of 100 iterations is met. Here, the parameter settings regarding GSA are shown in Table 3. The GSA parameters were selected: Primary parameter, G_0 = 100, Acceleration coefficient, α = 20 and no. of mass agents = 100. Since each agent could observe the performance of the others, the gravitational force is an information-transferring tool. The parameter settings for GSA are demonstrated in Table 3.

2.1.4 Hybrid particle swarm optimization and gravitational search algorithm (PSOGSA)

PSOGSA-based optimization has been employed already over the problem of economic load dispatch [34, 35], optimal static state estimation [36], dual channel speech enhancement [37], training feed-forward neural networks [38] and multi-distributed generation planning [39] etc.

The basic idea is to fit in the exploitation ability in PSO with the exploration ability in GSA to synthesize both algorithms' strength. The basic idea of PSOGSA is to combine the ability of social thinking (gbest) in PSO with the local search capability of GSA. In order to combine these two algorithms, velocity update is proposed as:

$$v_i(t+1) = w \times v_i(t) + \alpha' \times rand \times ac_i(t) + \beta' \times rand \times (gbest - x_i(t)) \tag{26}$$

where $v_i(t)$ means velocity of the agent i at the iteration t. On the other hand, w is the weighting factor, rand means a random number (between 0 and 1), ac_i(t) means the acceleration of the agent at the iteration t. Finally, gbest is

Table 3 GSA parameter settings [17]

Parameters	Values
Primary parameter, G_0	100
No. of mass agents, n	100
Acceleration coefficient, α	20
Constant parameter,ε	0.01
Power of 'R'	1
Maximum iteration	100
Number of runs	50

the best solution detected so far. Here, α' and β' are the weighting factors [40]. With adjusting α' and β', the abilities of global search and local search can be balanced. The position of the particle $x_i(t+1)$ in each iteration is updated using the equation:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{27}$$

The flowchart of hybrid PSOGSA method is shown in Fig. 4.

PSOGSA (adjusted according to the parameter setting in Table 4) was also run over the same fitness function and it was compared with the performance of gravitational search algorithm, by considering average best fitness. Maximum iterations and even swarm sizes were set exactly same to that of GSA and PSO techniques for achieving an objective comparison approach. The values of parameters c1, c2 and alpha were set as standard values, 0.5, 1.5 and 23 respectively [38]. The total number of runs remain 50 for the fare comparison purpose with other single techniques.

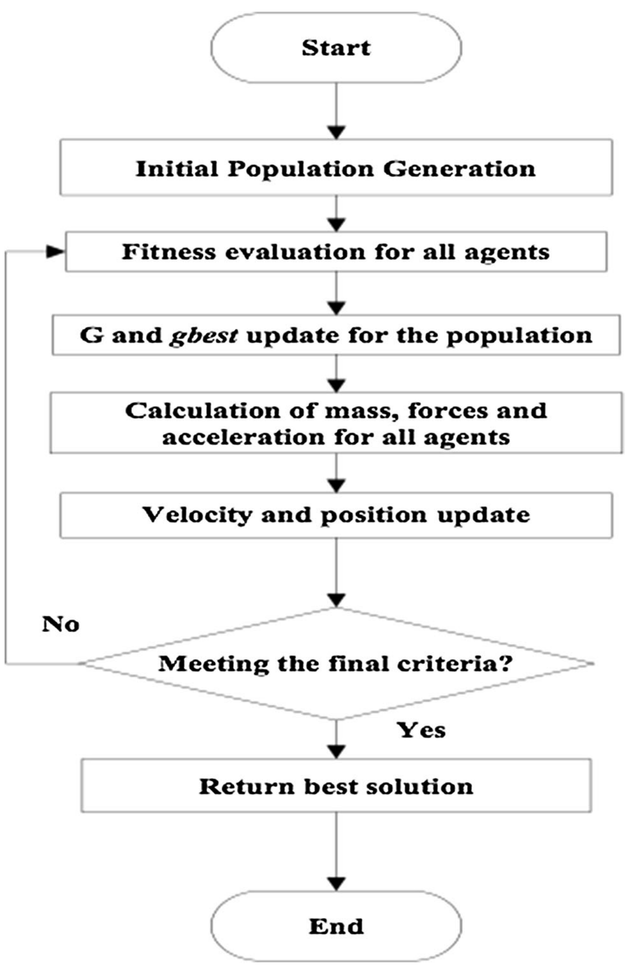

Fig. 4 Flowchart of PSOGSA

Table 4 Parameters settings of PSOGSA

Parameters	Values
Size of the swarm	100
Maximum iteration	100
PSO parameter, C1	0.5
PSO parameter, C2	1.5
Gravitational Constant, G0	1
GSA Constant parameter, α	23
Number of runs	50

3 Application details

In order to optimize state-of-charge with respect to charging time and present SoC, the explained four swarm intelligence-based methods (PSO, GSA, APSO, and PSOGSA) were applied considering the fitness function stated at the Eq. (13). All the optimization techniques were simulated to achieve the best fitness values by running them over the computer system having the configuration of CPU: Core^{TM} i5-3470 M, Processor: 3.20 GHz, RAM: 4.00 GB, with the MATLAB version-R2013a for the software environment of optimization experiments. A general flow of the application done in the study is presented in Fig. 5.

The related application process—comparison was done over five different problem scenarios over changing number of PHEVs: (1) 50 PHEVs, (2) 100 PHEVs, (3) 300 PHEVs, (4) 500 PHEVs, and (5) 1000 PHEVs. Findings and discussion moving from them are expressed under the next section.

4 Findings and discussion

Considering the application flow and five different problem scenarios, obtained findings can be expressed briefly as follows, over each of the employed swarm intelligence algorithms.

4.1 Findings with particle swarm optimization (PSO)

The algorithm was adjusted to run for a total of 100 iterations despite the fitness value converges before five iterations for all five scenarios and become stable. That's why an early convergence may cause the fitness function to trap into local minima.

For 50 PHEVs, the maximum best fitness and minimum best fitness were 469.7489 and 7.6478 respectively. The

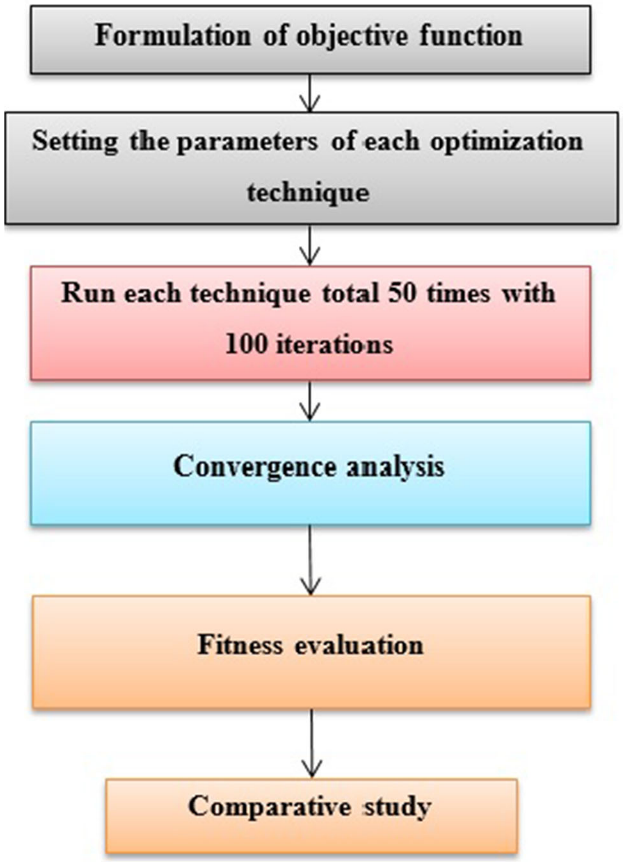

Fig. 5 Flowchart of the charging optimization application in the study

average best fitness is 165.9650. In 100 PHEVs, the maximum best fitness and minimum best fitness were 767.8722 and 9.5076 respectively. The average best fitness increases up to 182.9313. For 300 PHEVs, the maximum best fitness and minimum best fitness were 793.09 and 5.9631 respectively. The average best fitness is 197.5908. For 500 PHEVs, the maximum best fitness and minimum best fitness were 774.559 and 5.9631 respectively. The average best fitness is 197.5908. Finally, in 1000 PHEVs, the maximum best fitness and minimum best fitness were 678.9197 and 0.9963. The average best fitness decreases up to 172.4528.

Because PSO is a population-based optimization techniques and the fitness function is non-linear, so the fitness values fluctuates for each iteration [31–34]. Table 5 briefly presents the result. According to the table, it can be indicated that the average best fitness is in almost similar value scope for five different scenarios.

Table 6 shows the average computational time requirement for PSO method. The average computational time for 50 PHEVs is 1.620 s while for 1000 PHEVs it increases up to 2.328 s.

4.2 Findings with accelerated particle swarm optimization (APSO)

The algorithm was adjusted to run for a total of 100 iterations but the fitness value converges after 10 iterations and become stable. Consequently, there is an early convergence that may cause the fitness function to trap into local minima. This can be avoided by increasing the size of swarm hence the computational time will also be increased as well. For instant, a trade-off should be taken into consideration between the proper convergence and computational time.

Here, we applied each scenario 50 times for having idea about the performance and also superiority and efficiency level of the related algorithm.

In 50 PHEVs, the maximum best fitness and minimum best fitness were 469.7489 and 7.6478 respectively. The average best fitness is 165.9650. In 100 PHEVs, the maximum best fitness and minimum best fitness were 679.7151 and 9.5076. The average best fitness here is 182.9313. In 300 PHEVs, maximum best fitness and minimum best fitness were 541.4769 and 5.9631 respectively. The average best fitness is 197.5908. For 500 PHEVs, the maximum best fitness and minimum best fitness were 615.8314 and 5.9631. The average best fitness is 197.5908. In 1000 PHEVs, the maximum best fitness and minimum best fitness were 678.9197 and 0.9963. The average best fitness is 172.4528.

As APSO is a population-based optimization techniques and the fitness function is non-linear, so the fitness values fluctuates for each iteration [31–34]. However, the maximum best fitness remains in the range of 450–700 and the minimum best fitness remains in the range of 0.5–10. According to Table 7, average best fitness is in near value scopes for five different scenarios.

As APSO is a population-based optimization techniques and the fitness function is non-linear, so the fitness values fluctuates for each iteration [31–34]. However, the

Table 5 Fitness evaluation for PSO

Fitness function J (k)	50 PHEVs	100 PHEVs	300 PHEVs	500 PHEVs	1000 PHEVs
Max. best fitness	910.7513	767.8722	793.0902	774.559	697.1115
Avg. best fitness	142.839	171.102	169.3119	144.8008	156.8019
Min best fitness	4.8377	5.3836	5.219	7.1805	0.7301

Table 6 Average computational time for PSO

Number of PHEVs	Computational time (s)
50 PHEVs	1.620
100 PHEVs	1.669
300 PHEVs	1.764
500 PHEVs	1.953
1000 PHEVs	2.328

The term: exploration is used for defining the ability of an algorithm to spread out the problem in search gap whereas the term: exploitation is for the ability of detecting optimum solution, which is near to a favorable one [21].

In 50 PHEVs, 781.1267 and 0.2191 were found as the maximum best fitness and the minimum best fitness values. Here, the average value of best fitness is 158.8289. For 100 PHEVs, 872.648 and 1.005 were the maximum best fitness and the minimum best fitness values respectively. At this point, the average value of best fitness decreases up to 182.309. For 300 PHEVs, the maximum best fitness and minimum best fitness were 743.1251 and 2.3279 respectively. The average best fitness is 172.4296. In 500 PHEVs, the maximum best fitness and minimum best fitness were 836.2707and 0.9818. The average best fitness decreases up to 152.36437. For 1000 PHEVs, the maximum best fitness and minimum best fitness were 968.7652and 7.2747. The average best fitness decreases up to 161.52349.

Table 7 Fitness evaluation for APSO

Fitness function J (k)	50 PHEVs	100 PHEVs	300 PHEVs	500 PHEVs	1000 PHEVs
Max. best fitness	469.75	679.71	679.55	615.83	678.92
Avg. best fitness	162.70	168.23	147.42	184.15	171.16
Min best fitness	7.6478	3.46	3.54	5.9631	0.9963

maximum best fitness remains in the range of 450–700 and the minimum best.

Table 8 shows the average computational time requirement for APSO method. The average computational time for 50 PHEVs is 1.696 s and for 1000 PHEVs it increases up to 2.092 s.

APSO requires more adjustment of parameters, according to the standard form of PSO. But more number of PHEVs can be solved in less time by APSO, comparing to the PSO. By using the APSO, the velocity vector is effective for the capability of local exploitation. Also, APSO suffers early convergences in primary stages. In order to avoid premature convergence, Gravitational Search Algorithm (GSA) has been used in this study.

4.3 Findings with gravitational search algorithm (GSA)

Majority of the optimization algorithms comes with local searching techniques and these techniques may get stuck on the local maxima. Most search techniques make an effort for searching and detecting a global maximum in the presence by the local Maxima [20]. At this point, the GSA is good at performance shown during searching process.

Finally, Table 9 summarizes the result. From that it can be concluded that, average best fitness remain almost in similar pattern for five different scenarios.

Table 10 shows the Average computational time requirement for GSA method. The average computational time for 50 PHEVs is 2.72 s while for 1000 PHEVs it increases up to 2.092 s.

We performed each scenario 50 times to have idea regarding the performance and also superiority and efficiency of the related algorithm.

For 50 PHEVs, the maximum best fitness and minimum best fitness were 931.03 and 7.6478 respectively. The average best fitness is 165.9650. In 100 PHEVs, the maximum best fitness and minimum best fitness were 625.82 and 3.39 respectively. The average best fitness decreases up to 184.36. For 300 PHEVs, the maximum best fitness

Table 8 Average computational time for APSO

Number of PHEVs	Computational time (s)
50 PHEVs	1.696
100 PHEVs	1.706
300 PHEVs	1.761
500 PHEVs	1.832
1000 PHEVs	2.092

Table 9 Fitness evaluation of GSA

Number of PHEVs	Computational time (s)
50 PHEVs	2.720
100 PHEVs	4.439
300 PHEVs	11.279
500 PHEVs	18.165
1000 PHEVs	36.275

Table 10 Average computational time for GSA

Fitness function J (k)	50 PHEVs	100 PHEVs	300 PHEVs	500 PHEVs	1000 PHEVs
Max. best fitness	781.1267	872.6483	743.1251	836.2707	968.7652
Avg. best fitness	158.8289	182.309754	172.4296	152.36437	161.52349
Min best fitness	0.2191	1.0059	2.3279	0.9818	7.2747

and minimum best fitness were 434.16 and 7.43 respectively. The average best fitness is 181.03. In the case of 500 PHEVs, the maximum best fitness and minimum best fitness were 454.04 and 7.23 respectively. The average best fitness is 186.70. Finally, for 1000 PHEVs, the maximum best fitness and minimum best fitness were 740.40 and 0.17. The average best fitness decreases up to 185.16.

From the above numerical data, we can analyze the simulation behavior of APSO method. As it is a population-based optimization techniques and the fitness function is non-linear, so the fitness values fluctuates for each iteration. However, the maximum best fitness remains in the range of 400–950 and the minimum best fitness remains in the range of 0.1–8. Table 11 summarizes the result. From that it can be concluded that, average best fitness remain almost in similar pattern for four (05) different scenarios.

Table 12 shows the Average computational time requirement for APSO method. The average computational time for 50 PHEVs is 4.228 s while for 1000 PHEVs it increases up to 72.408 s.

4.4 A comparison among swarm intelligence techniques

In addition to the findings over each algorithm, it is also important to focus on a comparison including all algorithms. This sub-section deals with that in this manner. In detail, all of the four techniques were run on same computer along with same iterations (100) and total 50 independent runs in order to ensure the fare comparison [31]. The comparisons among applied swarm intelligence-based techniques are given below:

4.4.1 Stopping criteria

In any swarm intelligence algorithm there are some initial solutions from which candidate solutions are created. Next, each solution is evaluated and the algorithm choose the best solution. If the stopping criteria is met, the algorithm will produce final solution otherwise it will again search for best solutions from the initial step.

Table 12 Average computational time for APSO

Number of PHEVs	Computational time (s)
50 PHEVs	4.228
100 PHEVs	7.902
300 PHEVs	22.326
500 PHEVs	36.824
1000 PHEVs	72.408

4.4.2 Convergence analysis

Speed and rate of convergence to the optimal solution are significant when an algorithm discovers an ideal solution to a given problem [41]. Here, a trade-off is required to be done between computational time and the proper convergence. Table 13 shows the number of iterations needed to be converged for each algorithm for five different cases.

4.4.3 Fitness value

Fitness value presents the solutions for optimization technique applying to a particular objective function upon given constraints and set of parameters. The fitness value represents the strength of any optimization technique. In detail, maximum best fitness, average best fitness and minimum best fitness have been presented in order to evaluate the performance of the applied optimization techniques. PSOGSA shows best fitness values for all five cases (50, 100, 300, 500 and 1000 PHEVs). The fitness value comparison among all techniques are shown in Fig. 6. Single techniques like GSA and APSO show overall better result compared to PSO technique.

Table 11 Fitness evaluation for APSO

Fitness function J (k)	50 PHEVs	100 PHEVs	300 PHEVs	500 PHEVs	1000 PHEVs
Max. best fitness	931.03	625.82	434.16	454.04	740.40
Avg. best fitness	184.36	188.67	181.03	186.70	185.16
Min best fitness	3.39	3.71	7.43	7.23	0.17

Table 13 Convergence iterations

Number of PHEVs	Number of iterations taken to be converged			
	PSO	APSO	GSA	PSOGSA
50	< 10	< 10	35	< 5
100	< 10	< 10	35	< 5
300	< 10	< 10	15	< 5
500	< 10	< 10	40	< 5
1000	< 10	< 10	5	< 5

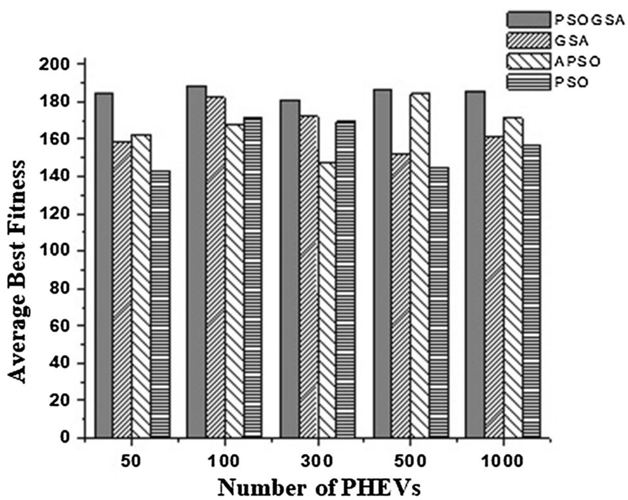

Fig. 6 Average best fitness versus number of PHEVs

4.4.4 Computational time

In order to maintain fare comparison, all the simulation runs on same computer as well as same swarm size (population) and iteration. Figure 7 shows the average

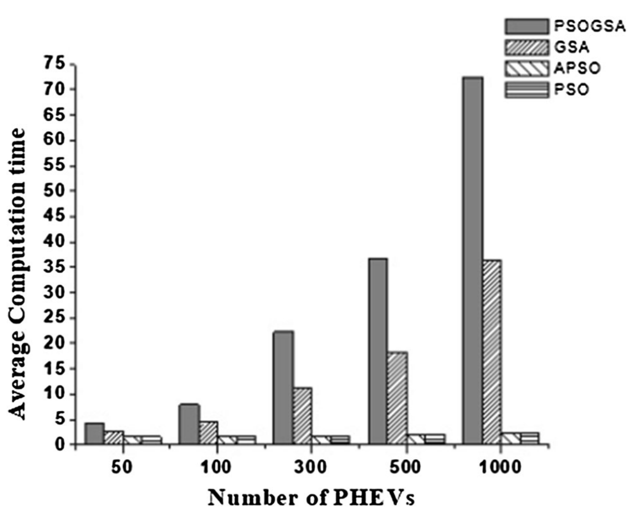

Fig. 7 Average best fitness versus number of PHEVs

computational time comparison of all the four optimization techniques considering five scenarios. PSOGSA, the hybrid technique takes more time to complete 100 iterations whereas both PSO and APSO techniques show better results in terms of computation time.

4.4.5 Robustness

Robustness is based on the ability of an optimization problem to perform well over a wide range of population [42]. Furthermore, optimization strategies and parameters must either remain constant over the set of problems or should be automatically set using individual test problems attributes. Among our optimization techniques, PSOGSA shows best robustness as the standard deviation is less compared to PSO, APSO and GSA. Table 14 shows the standard deviations of each techniques considering all five cases of PHEVs. According to the table, single methods like PSO, APSO and GSA show deviation in terms of average best fitness values. The standard deviation for each of these techniques is more than 10. Whereas, Hybrid PSOGSA shows the standard deviation 2.55.

5 Future research directions

Obtained results—findings in this study open the doors to further ideas. In this context, it is possible to think about some future research directions. Multi-objective capability should also be provided for multi-criteria optimization problems [43]. Some of remarkable ones can be explained as follows:

- *Ant colony optimization (ACO)* Ant colony optimization (ACO), introduced by Dorigo in his doctoral dissertation, is a class of optimization algorithms modeled on the actions of an ant colony. ACO is a probabilistic technique useful in problems that deal with finding better paths through graphs. Artificial 'ants' simulation agents locate optimal solutions by moving through a parameter space representing all possible solutions [12]. For solving a highly non-linear fitness function, ACO method can be applied in order to achieve high fitness value and also less computation time. This method is a member of Swarm Intelligence (SI) group. So, the similar results are expected like PSO and APSO. Researchers should apply ACO for solving charging problem of PHEVs and find the outcomes with extensive comparison with other methods.
- *Artificial bee colony (ABC) optimization* Artificial bee colony (ABC) algorithm is one of the most recently introduced swarm-based algorithms. In ABC, as a population based algorithm, the status of a food source

Table 14 Standard deviation for the applied techniques

Optimization techniques	Average best fitness for					Standard deviation
	50 PHEVs	100 PHEVs	300 PHEVs	500 PHEVs	1000 PHEVs	
PSO	142.84	171.10	169.31	144.80	156.80	11.83
APSO	162.70	168.23	147.42	184.15	171.16	11.95
GSA	158.83	182.31	172.43	152.36	161.52	10.62
PSOGSA	184.36	188.67	181.03	186.70	185.16	2.55

symbolizes a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution [44]. Future studies for solving smart charging problem of PHEVs should involve the ABC, too because of its proved success in advanced optimization problems.

- *Hybridization with local search method* Although swarm intelligence-based methods have established their capability to explore large search spaces, they are comparatively incompetent in fine-tuning the solution. Future studies can be carried out by hybridizing PSO or GSA (swarm intelligence-based algorithm) with local search method.

6 Conclusions

In this study, it was aimed to optimize state-of-charge, with respect to charging time, present SoC and in this context, the performed applications covered the related sub-objectives of optimizing the state-of-charge of Plug-in hybrid electric vehicle using swarm intelligence techniques and a general performance evaluation of swarm intelligence techniques in terms of fitness value and computational time. In order to achieve that, the study has employed four swarm intelligence techniques called particle swarm optimization (PSO) and gravitational search algorithm (GSA), accelerated particle swarm optimization (APSO) and hybrid version of PSO and GSA (PSOGSA), respectively. Because the results—findings that can be obtained with this study are important for an advanced real-world optimization problem like charging plug-in hybrid electric vehicle, necessary emphasis has given to different types of problem scenarios and the applied swarm intelligence research flows in this manner.

The future research works will be carried out with other meta-heuristics approaches such as Genetic Algorithms, Differential Evolution, Harmonic Search Algorithms, Ant Colony Optimization, Cuckoo Search Algorithms and hybrid optimization techniques of swarm intelligence and evolutionary computation.

Acknowledgements The authors would like to sincerely thank Mr. Imran Rahman, School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Malaysia for his great help and support in this research work. This research project also supported by Modeling Evolutionary Algorithms Simulation and Artificial Intelligence (MERLIN), Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam and Faculty of Science and Information Technology, Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perak, Malaysia.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Lund, H., & Kempton, W. (2008). Integration of renewable energy into the transport and electricity sectors through V2G. *Energy Policy, 36,* 3578–3587.
2. Hota, A. R., Juvvanapudi, M., & Bajpai, P. (2014). Issues and solution approaches in PHEV integration to the smart grid. *Renewable and Sustainable Energy Reviews, 30,* 217–229.
3. Soares, J., Sousa, T., Morais, H., Vale, Z., Canizes, B., & Silva, A. (2013). Application specific modified particle swarm optimization for energy resource scheduling considering vehicle-to-grid. *Applied Soft Computing, 13,* 4264–4280.
4. Su, W., & Chow, M.-Y. (2012). Computational intelligence-based energy management for a large-scale PHEV/PEV enabled municipal parking deck. *Applied Energy, 96,* 171–182.
5. Su, W., & Chow, M.-Y. (2011). Performance evaluation of a PHEV parking station using particle swarm optimization. In *Power and energy society general meeting, 2011 IEEE* (pp. 1–6).
6. Su, W., Eichi, H., Zeng, W., & Chow, M.-Y. (2012). A survey on the electrification of transportation in a smart grid environment. *IEEE Transactions on Industrial Informatics, 8,* 1–10.
7. Morrow, K., Karner, D., & Francfort, J. (2008). Plug-in hybrid electric vehicle charging infrastructure review. US Department of Energy-Vehicle Technologies Program.
8. Mayfield, D. (2012). Site design for electric vehicle charging stations, ver. 1.0. Sustainable transportation strategies.
9. Boyle, G. (2007). *Renewable electricity and the grid: The challenge of variability*. Earthscan: Routledge.

10. Hess, A., Malandrino, F., Reinhardt, M. B., Casetti, C., Hummel, K. A., & Barceló-Ordinas, J. M. (2012). Optimal deployment of charging stations for electric vehicular networks. In *Proceedings of the first workshop on urban networking* (pp. 1–6).
11. Li, Z., Sahinoglu, Z., Tao, Z., & Teo, K. H. (2010). Electric vehicles network with nomadic portable charging stations. In *2010 IEEE 72nd vehicular technology conference fall (VTC 2010-fall)* (pp. 1–5).
12. Dorigo, M. (2006). Ant colony optimization and swarm intelligence. In *Proceedings of the 5th international workshop, ANTS 2006, Brussels, Belgium, September 4–7, 2006* (Vol. 4150). Springer.
13. Eberhart, R. C., & Yuhui, S. (2001). Particle swarm optimization: developments, applications and resources. In *Proceedings of the 2001 congress on evolutionary computation, 2001* (Vol. 1, pp. 81–86).
14. Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39,* 459–471.
15. Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179,* 2232–2248.
16. Martens, D., Baesens, B., & Fawcett, T. (2011). Editorial survey: Swarm intelligence for data mining. *Machine Learning, 82,* 1–42.
17. Chiasson, J., & Vairamohan, B. (2005). Estimating the state of charge of a battery. *IEEE Transactions on Control Systems Technology, 13,* 465–470.
18. Piller, S., Perrin, M., & Jossen, A. (2001). Methods for state-of-charge determination and their applications. *Journal of Power Sources, 96,* 113–120.
19. Yang, J., He, L., & Fu, S. (2014). An improved PSO-based charging strategy of electric vehicles in electrical distribution grid. *Applied Energy, 128,* 82–92.
20. Ul-Haq, A., Buccella, C., Cecati, C., & Khalid, H. A. (2013). Smart charging infrastructure for electric vehicles. In *2013 international conference on clean electrical power (ICCEP)* (pp. 163–169).
21. Su, W., & Chow, M.-Y. (2011). Sensitivity analysis on battery modeling to large-scale PHEV/PEV charging algorithms. In *IECON 2011-37th annual conference on IEEE industrial electronics society* (pp. 3248–3253).
22. Wu, X., Cao, B., Wen, J., & Bian, Y. (2008). Particle swarm optimization for plug-in hybrid electric vehicle control strategy parameter. In *Vehicle power and propulsion conference, 2008. VPPC'08. IEEE* (pp. 1–5).
23. Soares, J., Morais, H., & Vale, Z. (2012). Particle swarm optimization based approaches to vehicle-to-grid scheduling. In *Power and energy society general meeting, 2012 IEEE* (pp. 1–8).
24. Yang, X.-S., Deb, S., & Fong, S. (2011). Accelerated particle swarm optimization and support vector machine for business optimization and applications. In *Networked digital technologies* (pp. 53–66). Springer.
25. Fergany, A. E. (2013). Accelerated particle swarm optimization-based approach to the optimal design of substation grounding grid. *Przegląd Elektrotechniczny, 89,* 30–34.
26. Reddy, B. R. (2012). Performance analysis of MIMO radar waveform using accelerated particle swarm optimization algorithm. arXiv preprint arXiv:1209.4015.
27. Talatahari, S., Khalili, E., & Alavizadeh, S. M. (2013). Accelerated particle swarm for optimum design of frame structures. *Mathematical Problems in Engineering*, *2013*, 649857. https://doi.org/10.1155/2013/649857.
28. Prajna, K., Rao, G. S. B., Reddy, K., & Maheswari, R. U. (2014). A new dual channel speech enhancement approach based on accelerated particle swarm optimization (APSO). *International Journal of Intelligent Systems and Applications (IJISA), 6,* 1.
29. Mohamed, A. Z., Lee, S. H., Hsu, H. Y., & Nath, N. (2012). A faster path planner using accelerated particle swarm optimization. *Artificial Life and Robotics, 17,* 233–240.
30. Gandomi, A. H., Yun, G. J., Yang, X.-S., & Talatahari, S. (2013). Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation, 18,* 327–340.
31. Ganesan, T., Elamvazuthi, I., Ku Shaari, K. Z., & Vasant, P. (2013). Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production. *Applied Energy, 103,* 368–374.
32. Roy, P. K. (2013). Solution of unit commitment problem using gravitational search algorithm. *International Journal of Electrical Power and Energy Systems, 53,* 85–94.
33. Duman, S., Güvenç, U., Sönmez, Y., & Yörükeren, N. (2012). Optimal power flow using gravitational search algorithm. *Energy Conversion and Management, 59,* 86–95.
34. Weber, G. W., Vasant, P., & Saucedo, J. A. M. (2018). Computer science and engineering. *Journal of Computational Science, 25,* 416–418.
35. Dubey, H. M., Pandit, M., Panigrahi, B., & Udgir, M. (2013). Economic load dispatch by hybrid swarm intelligence based gravitational search algorithm. *International Journal of Intelligent Systems and Applications, 5,* 21–32.
36. Mallick, S., Ghoshal, S. P., Acharjee, P., & Thakur, S. S. (2013). Optimal static state estimation using improved particle swarm optimization and gravitational search algorithm. *International Journal of Electrical Power and Energy Systems, 52,* 254–265.
37. Kunche, P., Rao, G. S. B., Reddy, K., & Maheswari, R. U. (2014). A new approach to dual channel speech enhancement based on hybrid PSOGSA. *International Journal of Speech Technology, 18,* 1–12.
38. Mirjalili, S., Mohd Hashim, S. Z., & Moradian Sardroudi, H. (2012). Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation, 218,* 11125–11137.
39. Tan, W. S., Hassan, M. Y., Rahman, H. A., Abdullah, M. P., & Hussin, F. (2013). Multi-distributed generation planning using hybrid particle swarm optimisation-gravitational search algorithm including voltage rise issue. *IET Generation, Transmission and Distribution, 7,* 929–942.
40. Vasant, P., Litvinchev, I., & Marmolejo-Saucedo, J. A. (2017). *Modeling, simulation, and optimization*. New York: Springer. ISBN 978-3-319-70541-5.
41. Beyer, H.-G. (2014). Convergence analysis of evolutionary algorithms that are based on the paradigm of information geometry. *Evolutionary Computation, 22,* 679–709.
42. Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G., & Stewart, W. R., Jr. (1995). Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics, 1,* 9–32.
43. Vasant, P., Litvinchev, I. S., & Marmolejo-Saucedo, J. A. (2019). *Innovative computing trends and applications*. New York: Springer. ISBN 978-3-030-03897-7.
44. Mernik, M., Liu, S.-H., Karaboga, D., & Črepinšek, M. (2015). On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation. *Information Sciences, 291,* 115–127.

Dr. Pandian Vasant is a senior lecturer at Universiti Teknologi Petronas, Malaysia and an Editor-in-Chief of International Journal of Energy Optimization and Engineering (IJEOE). He holds Ph.D. in Computational Intelligence (UNEM, Costa Rica), M.Sc. (University Malaysia Sabah, Malaysia, Engineering Mathematics) and B.Sc. (Hons, Second Class Upper) in Mathematics (University of Malaya, Malaysia). His research interests include Soft Computing, Hybrid Optimization, Innovative Computing and Applications. He has co-authored research articles in journals, conference proceedings, presentation, special issues guest editor, book chapters (248 publications indexed in Web of Science) and General Chair of EAI International Conference on Computer Science and Engineering in Penang, Malaysia (2016) and Bangkok, Thailand (2018). In the year 2009 and 2015, Dr. Pandian Vasant was awarded top reviewer and outstanding reviewer for the journal Applied Soft Computing (Elsevier). He has 27 years of working experiences at the universities. Currently he is an Editor-in-Chief of International Journal of Energy Optimization and Engineering, and Member of AMS (USA), NAVY Research Group (TUO, Czech Republic) and MERLIN Research Group (TDTU, Vietnam). H-Index Google Scholar = 30; i-10-index = 107. Professional references: https://scholar.google.com.my/citations?user=cOKj9goAAAAJ&hl=en; https://www.linkedin.com/in/vasant/.

Prof. Jose Antonio Marmolejo is a Professor at Panamerican University, Mexico. His research is on operations research, large-scale optimization techniques, computational techniques and analytical methods for planning, operations, and control of electric energy and logistic systems. He is particularly interested in topics related to artificial intelligence, digital twins, the Internet of things, Industry 4.0, Sustainable Supply Chains. He received his Doctorate in Operations Research (Hons) at National Autonomous University of Mexico. At present, He has the second highest countrywide distinction granted by the Mexican National System of Research Scientists for scientific merit (SNI Fellow, Level 2). He is a member of the Network for Decision Support and Intelligent Optimization of Complex and Large Scale Systems and Mexican Society for Operations Research. He has co-authored research articles in science citation index journals, conference proceedings, presentations, books, and book chapters.

Igor Litvinchev received his M.Sc. degree in Applied Mathematics from Moscow Institute of Physics and Technology (Fizteh), Russia; Ph.D. in Systems Theory and Operations Research and Dr.Sci. (Habilitation) in Systems Modeling and Optimization from Computing Center, Russian Academy of Sciences, Moscow. He has held long term visiting positions at universities in Brazil, Mexico, and Norway, as well as positions at various universities and research centers in Russia. He is an author of 4 books and numerous research papers focused on large-scale systems modeling, optimization, and control. His research was supported by more than 30 grants from NATO Scientific Affairs Division and European Community; ISF (USA) and RFBR (Russia); CNPq and FAPESP (Brasil); BRFBR (Belarus); CONACYT, PROMEP and PAICYT (Mexico). Professor Litvinchev is a member of Russian Academy of Natural Sciences and Mexican Academy of Sciences.

Roman Rodriguez Aguilar is a professor in the School of Economic and Business Sciences of the "Universidad Panamericana" in Mexico. His research is on large-scale mathematical optimization, evolutionary computation, statistical modeling, mathematical finance, competition and market regulation. He received his Ph.D. at School of Economics at National Polytechnic Institute, Mexico. A master's degree in Engineering from the School of Engineering at the National University of Mexico (UNAM), a master's degree in Administration and Public Policy in the School of Government and Public Policy at Monterrey Institute of Technology and Higher Education. A postgraduate in applied statistics at the Research Institute in Applied Mathematics and Systems of the UNAM and his degree in Economics at the UNAM. Prior to joining to Anahuac University, he has worked as a specialist in economics, statistics, simulation, finance and optimization, occupying different management positions in various public entities such as the Ministry of Energy, Ministry of Finance, and Ministry of Health. He has coauthored research articles in science citation index journals, conference proceedings, presentations and book chapters. He is a member of the National System of Researchers Level II of CONACYT.