Defence Science Journal, Vol. 73, No. 5, September 2023, pp. 564-571, DOI : 10.14429/dsj.73.5.18497

APSO Based Automated Planning in Constructive Simulation

Sanjay Bisht[#,*], S.B. Taneja[#], Vinita Jindal[!] and Punam Bedi[$]

[#]DRDO-Institute for Systems Studies & Analyses, Delhi - 110 054, India
[!]Keshav Mahavidyalaya, University of Delhi, Delhi - 110 034, India
[$]Department of Computer Science, University of Delhi, Delhi - 110 007, India
[*]E-mail: san_hem_bisht@yahoo.co.in

ABSTRACT

Constructive simulations are the applications used by the military for the training of their commanders in planning and analysis of various threats and Courses of Action. In the 'analysis wargames', there are need to automate many of the tasks of the commander which are carried out by subunit commanders on the ground. Deployment of defence units is one of such important decision making by commander. Deployments of units (and sub units) is dependent on multiple factors which needs to be satisfied/optimised for meeting the given objective of the unit. In this paper we have attempted to solve the multi criterion decision problem of optimal deployment of defence units in mountainous terrain using Particle Swarm Optimization(PSO) and Adaptive Particle Swarm Optimization(APSO). The algorithm has been tested with varied number of decision parameters and their weights using digital elevation and vector data of the terrain features. The auto deployment outcomes are found satisfactory. Our solution approach has potential in automated planning in constructive simulations.

Keywords: Particle swarm optimisation; Multi criteria; Heuristic optimisation; Genetic algorithm; Simulated annealing; Multi objective optimisation; Constructive simulation

1. INTRODUCTION

Constructive simulations are the applications used by the military for the training of their commanders in planning and analyses of various threats and courses of action. In these simulations, depending on the 'level of operations supported', 'resolution of combat entities', 'purpose (training/analysis)', etc., the combat & decision making process of the commanders are modelled. In the 'analysis wargames', there are need to automate many of the tasks of the commander which are carried out by subunit commanders on the ground. Deployment of defence units is one of the important decisions made by commanders. Deployments of units (and subunits) calls for consideration of multiple factors like the 'type of unit', 'terrain & environmental factors', threat, 'operation type (offensive/defensive'), etc. The final solution has to be optimal for the given operational objectives, constraints and relative weights to these factors. In our work, we have taken a specific case of auto-deployment of subunits (within the unit area) in the mountainous terrain. We have attempted to solve this problem using particle swarm optimisation.

Particle Swarm Optimization (PSO) is a bio-inspired stochastic evolutionary optimization algorithm of nature, which mimics the behaviour of a flock of birds[1] or a school of fish. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA)[2-4]. Like GA, PSO[5] searches the space globally and simultaneously. It is different from other optimization algorithms in a way that only the objective function is needed and it is not dependent on the gradient or any differential form of the objective. It also has very few hyper parameters. It is initialized with a pool/group of random solutions and searches for optima by updating through iterations. In PSO, the potential solutions, called particles, move through the problem search space by following the local best (cognition) solution and the current optimum particles of the swarm (social behaviour). In each iteration, all particle based on the value of individual cognition factor (C1) and social influence factor (C2) follows their local best solution and iterations global best solution respectively to converge quickly to an optimal solution.

The rate of the position change (velocity) is calculated with Eqn. (1) and parameters for the Eqn. (1) are described in Table 1.

$$V_{id} = W * V_{id} + c_1 * rand[0,1] * (P_{id} - X_{id}) + c_2 * rand[0,1] * (G_{id} - X_{id}) \quad (1)$$

Ming Cao[6], *et al.* in their paper demonstrate that PSO is capable of solving large-scale WTA problems efficiently. Hassan Haghighi[7] in their study used HPSOGA, a hybrid form of particle swarm optimization and genetic algorithm for optimal path planning in coverage missions by cooperated unmanned aerial vehicles. B. Abhisek[8] used Hybrid PSO-HSA (Harmony search) and PSO-GA algorithm for 3D path planning in autonomous UAVs for better exploratory and exploitative search. In other Military applications, Xuezhi Lei[9] used PSO form selecting the distribution centre's location in military

Received : 06 September 2022, Revised : 25 April 2023
Accepted : 28 April 2023, Online published : 31 August 2023

logistics. Can Gao[10] use the Hybrid Particle Swarm algorithm and hill-climbing method for solving the Location Problem of the Distribution Centre. Vinita Jindal[11], *et al.* in their paper used pre-emptive hybrid Ant Particle optimisation (HAPO-P) algorithm for smart transportation.

Table 1. PSO parameter description

W : Inertia Weight
V_{id} : Particle velocity
c1,c2 : constant where value of c1=1.2 and c2=2.0
$rand^{0,1}$: Random number ranging from 0 to 1
X_{id} : Current solution from each individuals
P_{id} : Personal Best, the best solution from each individuals
G_{id} : Global Best, the best solution from the whole population
The new position of the particle would be:
$X_{id} = X_{id} + V_{id}$
The MOE of the solution is a Fn of Xid
MOE = fn(Xid).
Basic example of PSO application
In a mountainous terrain since, the Objective is to find the optimal peak of greatest height , then there will be two parameter in Objective Fn (Xid, Yid) where,
Xid : Current Easting posn of each individuals
Yid : Current Northing posn of each individuals
Vidx : Particle velocity in easting direction
Vidy : Particle velocity in northing direction
Pidx : easting Posn of best solution of each individuals
Pidy : northing Posn of best solution of each individuals
Gidx : easting Posn of the best solution of the
Gidy : northing Posn of the best solution of the whole population
And now in each iteration velocity update for each particle will be as follows:
$Vidx=W*Vidx+c1*rand^{0,1}*(Pidx-Xid)+c2*rand^{0,1}*(Gidx-Xid)$
$Vidy=W*Vidy+c1*rand^{0,1}*(Pidy-Yid)+c2*rand^{0,1}*(Gidy-Yid)$
The new position of the particle would be:
Xid=Xid+Vidx
Yid=Yid+Vidy
MOE Fn for this will be written as:
Zid= MOE function (Xid, Yid)

PSO works well in early iterations but has issues in reaching the near-optimal solution. To solve this issue Y. Sh[12-13], *et al.* have employed methods in improving solutions. One of the strategies would be to linearly decrease the inertia weight as the generations are increasing. But instead of adjusting the PSO parameter as per the increasing generation, this paper uses effective adaptive strategies[14-15] at the swarm particle's level, which recommends replacing ineffective particles with fresh ones (by again randomizing their positioning in space) from the current generation by keeping track of the history of the improvement of each swarm particles. In each generation, ineffective particles are tracked according to some predefined rule for judging the particle's ineffectiveness in the current generation based on the particle's history of the rate of solution improvements. For this, a term (Tc) is used for each particle which describes the number of particle non-performance count in the past generation below a designated threshold value(e).

Our results suggest that the adaptive Particle swarm Optimization (APSO) outperforms standard PSO.

We have attempted here to address the class of problems for decision support/ automated planning for application to constructive simulation. We have implemented the PSO and APSO based algorithm for automated sub-unit deployment and compared the performance as MOE (Measure of effectiveness) as to how close the deployment is to the ideal.

The results generated by our algorithm were discussed with domain subject matter experts and found to satisfy the commander's intent.

2. PROBLEM DESCRIPTION

The deployment of units is one of the important factors which influence military commanders on the concept of operations in different areas. As per the higher commander's intentions and overall plan of operations, the tactical commander appreciates the likely deployment areas on the map board followed by ground reconnaissance. The process of selection of area for deployment is based on the appreciation of a particular commander for that scope of operations. It is not necessary that the areas identified would be the same for the different commanders in time and space for the operations. This increases the complexity of ideal locations for deployment. This process is tedious and time-consuming wherein the commander initially appreciates each location on the map board considering the advantages and disadvantages. This process is based on certain multi-criteria peculiarities, which need to be balanced as per role and tasking (Defensive & Offensive Ops), some of the aspects are (Table 2) Extent of area, Orientation towards the enemy, Deployment: Linear- Extended, Consideration Heights and Spur lines, Slope, gradient, Availability of axis, Line of Sight Profile, obstacle check (Rivers/ Streams/ Nallahs, along Valleys, Re-entrant/ Dead Ground, Reverse slopes, soil condition, vegetation etc. For computerised 'analysis wargames', rather than a manual selection of each location on

Table 2. MOE parameters for optimal infantry parent unit deployment in mountainous terrain

Parameter name	Parameter variable	Description
Sector-wise area representation	WT(1)	Each subunit position of the Parent Unit should be occupied by at least one subunit position of a particle solution
Deployment of subunits in height	WT(2)	The particle subunit position should be at a dominant height so that subunit can engage the incoming enemy effectively from its weapons. Sub position should be deployed in spur Lines of Mountain ridges so that they hide from enemy line of sight
Minimum LOS from enemy units position	WT(3)	Subunit should be deployed in such a way that there should be minimum LOS with the enemy
Maximum LOS from subunit road axes	WT(4)	Subunit should be deployed in such a way that there should be maximum LOS from subunit to own road axes to guard logistic supplies
Inter sub unit gap	WT(5)	Inter sub unit Gap of the PSO particle solution should be maximized so that subunits do not overlap boundaries

the map which is time-consuming, if the system pre-processes in auto-selection of optimal deployment areas before the actual simulation, this will help in the logical synthesis of data in a rationale way which is nearing to the military commander's way of appreciation. The system generated Automatic Deployment feature will aid commanders in planning the selection of ideal deployable areas in a particular terrain.

Table 3. Mathematical formulation of objective function (MOE)

	Mathematical formulation	Factors considered	Description	Remark
Objective Fn of MOE	u=(wi*F1+w2*F2+ w3*F3+w4*F4+ w5*F5)	Objective Fn	Where, each $w_k\,(0 \le w_k \le 1)$ is the weight for function $F_k(0 \le F_k)$ called the representative function.	w_k represent kth parameter initial weight given by the player. k: Index of parameters contributing to MOE
F_1	$F_1 = \frac{t}{m}$	Area representative function	Where, m: number of subunit posn in parent unit (here m = 3). t: number of sub unit posn (Plx^{i,j},Ply^{i,j}) actually occupied by any of particle Location (x_i,y_i) where, i ranging from 1.. m (number of sub unit posn in parent unit)	where, roundel (Plx^{i,j}, Ply^{i,j}), where i represent ith sub unit posn of parent unit and j represents no of peripheral points within ith sub units Polygon
F_2	$F_2 = H_d * S_R$ $H_d = \frac{1}{m}\left(\sum_{i=1}^{m}\frac{h_i}{h_g}\right)$ $S_R = \frac{1}{m*n}\sum_{i=1}^{m}\sum_{j=1}^{n}\Omega$	Height representative function	Where, (x_i,y_i) : ith position of particle representing a subunit Posn h_g : global max height of the terrain area under study h_i : height or elevation of the i location (x_i,y_i) of Particle H_d: Height Dominance Factor S_R : ratio of the points on a spur line that is at a lower elevation than the point (x_i,y_i) Ex(i,^{n}), Ey(i,^{n}) are periphery Location of (ellipse) generated by taking i Particle Location as centre as subunit frontage as major axes,subunit depth as minor axes.	Ω: =1 if Height of any jth peripheral point Ex(i,^{j}), Ey(i,^{j}) of ellipse of i Particle Posn < Height of Particle Posn (x_i,y_i), where j ranges from 1 to n. n is the total number of peripheral points generated for i particle position roundel
F_3	$F_3 = \frac{l_W}{q*m}$ $l_W = \sum_{i=1}^{m}\sum_{j=1}^{q}\delta$	Rd Axis representative function	Where, l_W : sum of the existence of LOS of all road axes locations Rdx(j), Rdy(j) with respect to the particle ith Location (x_i,y_i) for all sub polygons. Here we consider q random points on the own road Axis	where, δ = 1 if LOS exists between any pair of any jth own side Road axes location (Rdx(j), Rdy(j)) and ith particle Posn (x_i,y_i) of m points otherwise δ = 0 where, Rdx^q, Rdy^q : represents total q Points of own road axes
F_4	$F_4 = \frac{l_E}{q*m}$ $l_E = \sum_{i=1}^{m}\sum_{j=1}^{q}\omega$	Enemy Axis representative function	Where, l_E : sum of the ratio of the non-existence of LOS of all enemy axes (Enx(j), Eny(j)) location with respect to the point (x_i,y_i) for all sub polygons. Here we consider q random points on the Enemy Axis	where, ω = 1 if No LOS exists between any pair of any jth Enemy location (Enx(j), Eny(j)) and ith particle Posn (x_i,y_i) of m points otherwise δ = 0 where, Enx^q, Eny^q : represent total q Points of Enemy axes
F_5	$P = \sum_{i=1}^{m}\sum_{j=i+1}^{m} Dist\left((x_i,y_i),(x_j,y_j)\right)$ $S = \sum_{i=1}^{m}\sum_{j=i+1}^{m} Dist\left((x_{ci},y_{ci}),(x_{cj},y_{cj})\right)$ $F_5 = \min\left(\frac{P}{S},1\right)$	Inter sub unit gap representative function	where, P : sum of distance of all distinct pair (x_i,y_i), (x_j,y_j) (where, j > i) of the polygon with m points (x_i,y_i) random ith sub unit location generated by PSO. S: sum of distance of all distinct pair (x_{ci},y_{ci}), (x_{cj},y_{cj}) (where, j > i). (x_{ci},y_{ci}) is actual standard ith sub unit centre locations generated by algorithm using parent unit periphery polygon. Where, i ranging from 1..m m=number of total sub unit locations of parent units	

3. PROBLEM FORMULATION

In the following section, we have formulated the problem of deployment of unit (subunit) in the mountainous terrain with a few set of parameters (and their weights). The problem is formulated and solved using PSO and APSO. The solution can be extended to include more parameters (Table 2).

Problem of automated deployment of subunits within the unit area is characterised by multi criterion/factors. Relative importance of these factors are decided by the unit commander as per the operational situation and operational role of the unit. Problem addressed in this paper has taken 5 factors (F1-F5) in consultation with the military subject matter experts viz. Dispersion of subunits within the unit area, Height dominance within the given area of deployment, Exposure/visibility with own/enemy units, visibility to logistics supply axis and sub-unit dispersion/spread. These five operational domain factors contributing MOE are mathematically represented/formulated (Table 3) as F1-Area representative factor, F2-Height representative factor, F3-Enemy axis representative factor, F4-Road Axis representative factor and F5-Inter sub-unit gap representative factor respectively.

These factors (F1-F5) are dynamically evaluated for the given scenario. Scenario includes, Terrain Elevation data (DEM), Terrain features vector Data-Roads, Track, Spur lines, unit deployment area, Friendly/own and enemy units/entities with their locations and resources (equipment's/weapons etc.).

Objective function, MOE (Measure of Effectiveness) is taken as weighted sum of the factors (F1-F5). PSO and APSO based algorithm is implemented to maximise the MOE for the desired operational objective.

Random population of particles is generated. Each particle represents the set of locations of sub-units and MOE is computed based on the weighted sum of factors (Table 3). Factor F1(Sector-wise area representation) represents how many particles are within the sub unit area. It is computed by ratio of the number of particles within the sub-unit areas and the total number of the sub-unit areas. Factor F2 (Deployment of subunits in Height) is computed as sum of ratio of height of each particle position and maximum possible height in the given parent unit region. Factor F3 (minimum LOS from enemy) is computed as ratio of number of non-LOS of particle with all enemy locations and the total possible particle-enemy interactions (m*n, where m=number of sub units and n=no of enemy locations) Factor F4 (maximum LOS from own Road axes) is computed as ratio of total existence of LOS (between m PSO particle positions and r road axes location) and (m*r). Fifth MOE parameter F5 " Inter sub unit Gap " was introduced so that all PSO particle positions must have minimum sub unit inter distance. Although , some PSO solution are having high value in first four parameters, but if all PSO sub positions might fall on or near same location ,that may not be a good solution, therefore minimum inter sub unit distance has to be maintained.

4. IMPLEMENTATION

The following section details the implementation of the algorithm for automated deployment problem described and formulated in the previous section. Algorithm Pseudo code for PSO and APSO are placed in Table (4, 5). Input to the algorithm includes the scenario as described in previous section. The user (Unit commander in our case) assigns weights to the factors and marks the unit area polygon (Cx, Cy). Sub-unit areas ($Plx^{i,j}$, $Ply^{i,j}$) are generated (equal to the number of sub units of the parent unit, e.g 3 sub positions in our case of Coy Unit) within which the our objective is to automatically generate the sub-unit locations. Here i represents sub unit position of parent unit and j represents no of peripheral points within the i^{th} sub unit area position.

Table 4. Automated deployment algorithm (PSO)

```
For each particle in Population(Pop)
Initialize particle with random Posn(X3,Y3) within parent unit  Polygon
Calculate (MOE) fitness value of each particle
If the (MOE) fitness value is better than its personal   best
        set current value as the new pBest
Update global best particle  if pBest > global best
END For
Do for each iteration
For each particle in Population
    Calculate particle velocity according equation (2)
    Update particle position according equation    (3)
    Calculate (MOE) fitness value of each particle
    If the fitness value is better than its personal best
            set current value as the new pBest
   Update global best  particle if pBest > global best
  End
While (maximum iterations  or minimum error criteria is not attained)
```

Table 5. Automated deployment algorithm (APSO)

```
Initialize Tc=3
Initialize DelError=.0001
For each ith particle in Population(Pop)
Initialize particle with random Posn(X3,Y3) within parent unit Polygon
Initialize particle NonPerformCtr(i)=0
Calculate (MOE) fitness value of each ith particle
If the (MOE) fitness value is better than its personal   best
        set current value as the new pBest for ith particle
Update global best particle if pBest > global best
END For
Do for each iteration
For each ith particle in Population
   IF NonPerformCtr(i) > Tc
           NonPerformCtr(i)=0
Poly area
   End if
    Calculate particle velocity according equation (2)
    Update particle position according equation    (3)
    Calculate (MOE) fitness value of each ith particle
           Compute Relative Error Fn(REN)= abs( Particle(i).MOE-Global.
           MOE) /   abs(min(Particle(i).MOE,Global.MOE))
           If    REN < DelError

           NonPerformCtr= NonPerformCtr+1
        End If
        If the fitness value is better than its personal best
           set current value as the new pBest
   Update global best  particle if pBest > global best
  End
While (maximum iterations  or minimum error criteria is not attained)
```

The problem of optimal deployment of subunits of the parent unit is solved by using Particle Swarm Optimisation where each solution of PSO is a particle-containing 3 locations (x_i,y_i), i is ranging from 1 to 3. Each swarm contains a population of n particles. PSO randomly initialize all n Particle's three locations within the parents unit's area polygon (Cx, Cy). In every PSO iteration, each particle's MOE is computed (Table 3) and the particle's best cost is updated with the current best cost, if current MOE is better the particle Best MOE Cost. Also in each iteration if any particle MOE is better than the current Global Best, then Global Best is replaced with particle MOE.

Table 3 depicts the mathematical formulation of the MOE of each PSO Particle in detail.

MOE of each particle representing solution as (x_i, y_i) is given as:

$u = \sum_{k=1}^{n} w_k F_k$ where, $w_k \left(0 \le w_k \le 1\right)$ is the weight for function $F_k\left(\text{representative function}\right)\left(0 \le F_k\right)$.

This process is repeated until the std deviation of last n (say 20) iteration is less than required std error (.0001). F_k is value of k[th] representation Function, k ranging from 1.. n=5, u is MOE value ranging between $\left(0 \le u \le 1\right)$

In each iteration velocity update for each particle is:

$$V_{idx} = W * V_{idx} + c_1 * rand\left[0,1\right] * \left(BestPosnX\left(i,j\right) - X\left(i,j\right)\right) + + c_2 * rand\left[0,1\right] * \left(GlobalBestParticle.GX\left(j\right) - X\left(i,j\right)\right)$$

$$V_{idy} = W * V_{idy} + c_1 * rand\left[0,1\right] * \left(BestPosnY\left(i,j\right) - Y\left(i,j\right)\right) + c_2 * rand\left[0,1\right] * \left(GlobalBestParticle.GY\left(j\right) - Y\left(i,j\right)\right) \quad (2)$$

The new position of the particle would be:

$$X\left(i,j\right) = X\left(i,j\right) + V_{idx}$$

$$Y\left(i,j\right) = Y\left(i,j\right) + V_{idy} \quad (3)$$

where, j varies from 1:n sub unit Posn for each i[th] Particle of swarm, [X(i,j) , Y(i,j)] represent locations of j[th] sub unit for i[th] particle of swarm. BestPosnX(i,j), BestPosn(i,j) represent so far best location of j[th] sub unit for i[th] particle of swarm, GlobalBestParticleGX(i,j), GlobalBestParticleGY(i,j) represent so far Global best location of j[th] sub unit for i[th] particle of swarm. Other PSO value taken : W=1.0 , c1=1.5 , c2=2.0 and Tc=3.

In the APSO algorithm (Table 5), the particles which did not performed for Tc times are replaced with the randomly re-initialised particles within Parent unit. The non performance criteria is taken as when the relative error function Fi value goes below (e=10^-4).

Relative error function Fi is computed for each i th particle in each generation as follows:

Fi= (FgBest -Fi) / Min(Abs(Fi), Abs(FgBest))

where, Fi is MOE value for i[th] particle at iteration(it) and FgBest is so far best MOE of particle. The APSO algorithm is described in Table 5.

5. EXPERIMENTAL SETUP, RESULTS AND ANALYSIS

5.1 Experimental Setup

Algorithm is implemented in MATLAB. Terrain map data (vector, raster) and DEM data (Tiff) are loaded and displayed. Simulation front end facilitates marking of unit locations, roads and other vector features. Two scenarios were created with 5 factors to be addressed for deployments. User inputs included initial deployment area (unit polygon) of the unit on the desired location on the GIS map. User (unit commander) gives the weights to each of the five factors (F1-F5) to meet the operational objective. Scenario-1 gives equal importance to all the factors for the given operational need. In the Scenario-2, higher importance is given to few factors (like dominance to terrain height and inter subunit gap) as compared to other factors (like Line of Sight). System generates three initial sub

Table 6. Result analyses

Scenario	**1. Equal WTS to all factor 0.20**		**2. Variable WTS to all with dominent height and inter PL Gap** **WT(1) : Area Rep: 0.10** **WT(2) : HT Advantage: 0.40** **WT(3) : LOS With road axes: 0.20** **WT(4) : MIN LOS With enemy axes: 0.00** **WT(5) : Inter latoon Gap: 0.30**	
	PSO	APSO (Tc = 3, Error = .0001)	PSO	APSO (Tc = 3, Error = .0001)
POP	200	200	200	200
STD error for termination	StdDev = 0.0001	StdDev = 0.0001	StdDev = 0.0001	StdDev = 0.0001
Area Rep Val	1	1	1	1
PL HT Val	0.8033	0.8687	0.8386	0.8585
Road LOS Val	0.5833	0.5833	0.4167	0.4167
NO LOS With enemy	1	1	0.9167	1
Inter PL GAP Val	1	1	1	1
MOE Val	0.8773	0.8904	0.9188	0.9434
Iteration terminated	107	127	63	112
Exchange for best SOL	29	22	14	22
No of non performing particles	NA	26	NA	41

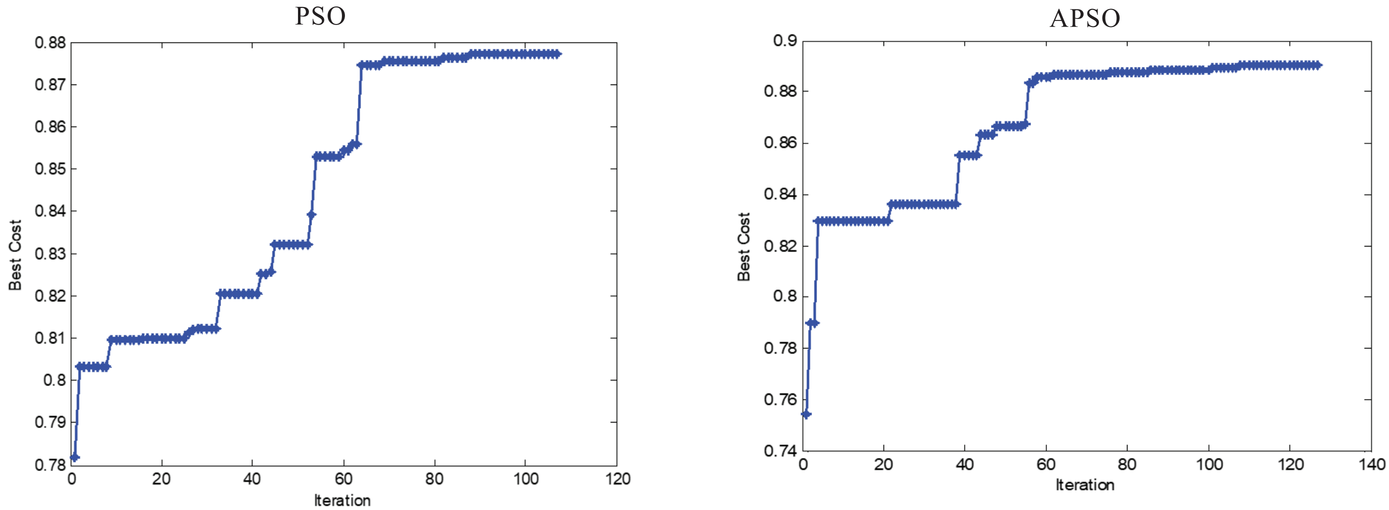

Figure 1. Performance graph (Scenario-1).

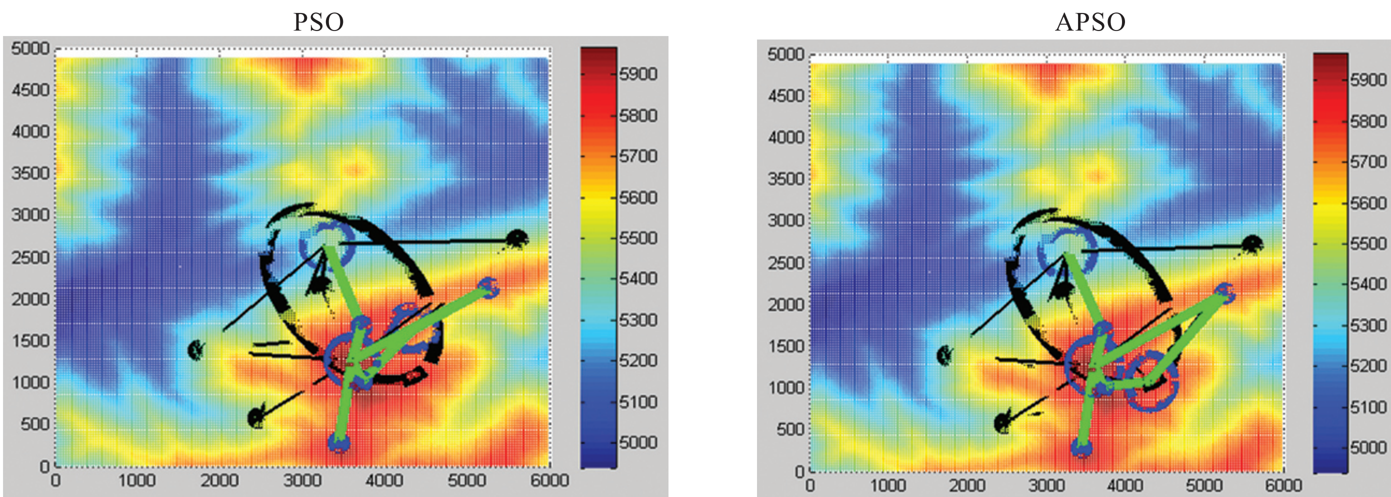

Figure 2. Automated deployment output (Scenario 1).

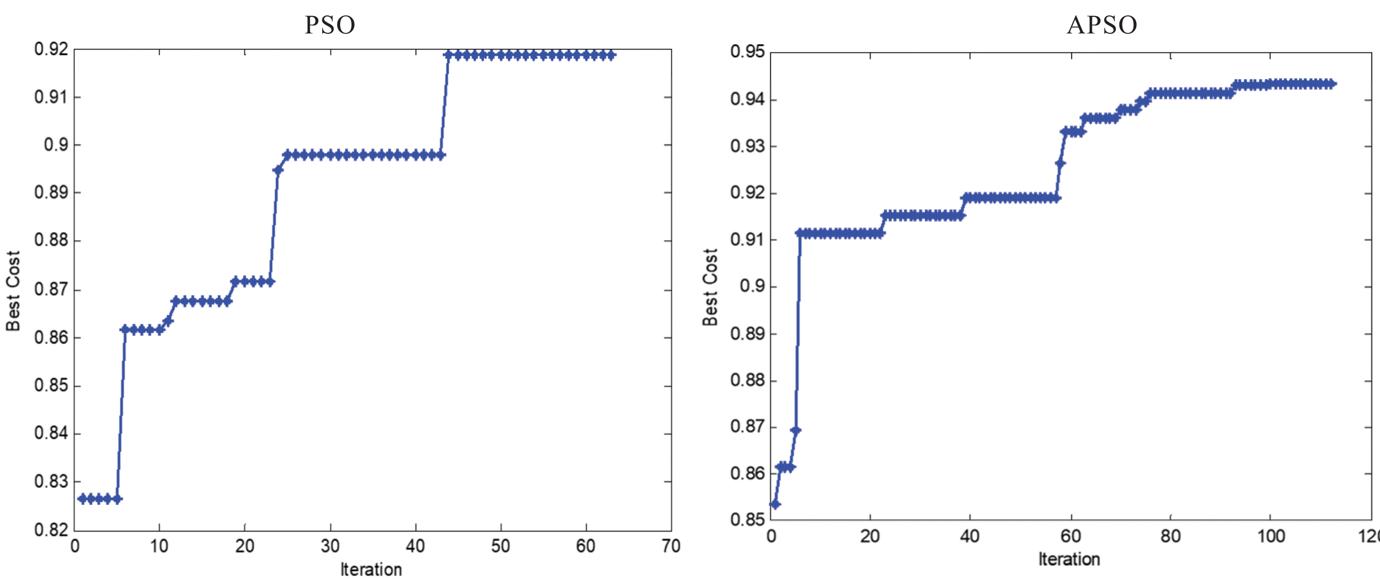

Figure 3. Performance graph (Scenario-2).

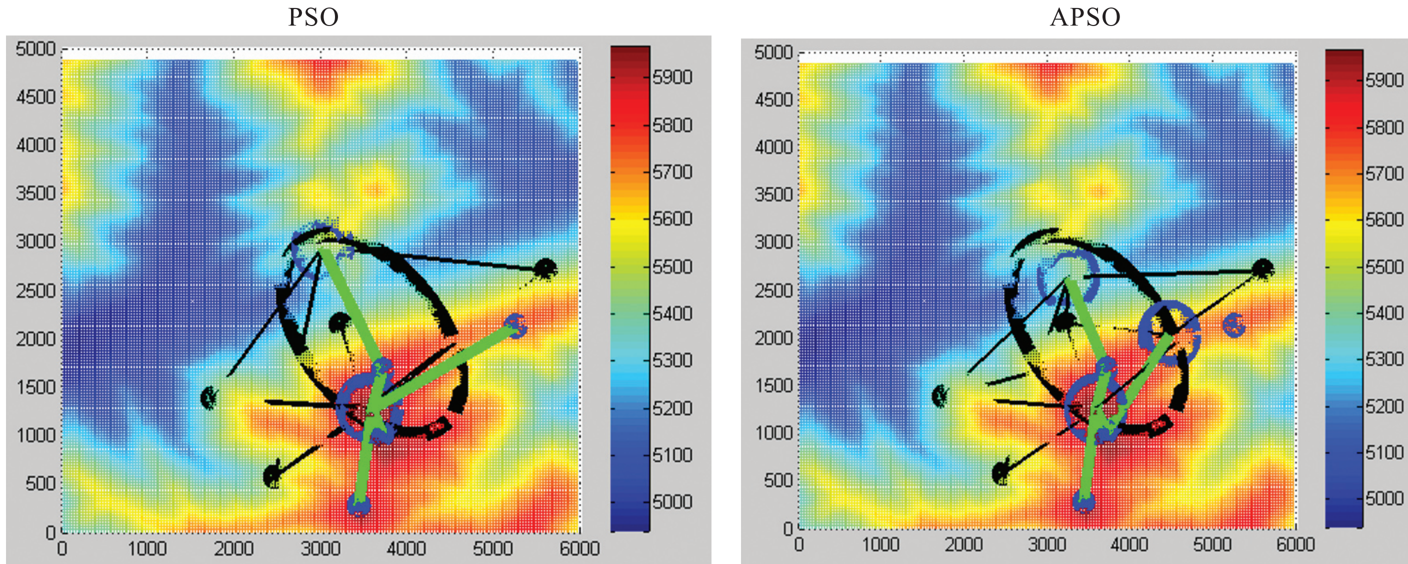

Figure 4. Automated deployment output (Scenario 2).

unit position and sub unit polygons, which are the input to our algorithm. Optimal population size has been taken as 200 by experimentation through multiple APSO simulation runs. Simulation was executed for both the scenarios with PSO and APSO algorithms. Termination criterion of the algorithm is when the standard deviation of last n (say 20) iteration goes below required std error (0.0001).

5.2 Result Analyses

The results of the two scenarios are listed in Table 6. Scenario 1 is with equal weights /importance to all the 5 multi criterion factors. Scenario 2 where the operational need dominated by relatively higher importance to achieve the height dominance with minimum from enemy locations and also archive the larger inter subunit gap to cover the maximum intended unit area. Automated deployment locations are computed through our PSO and APSO based algorithms.

Results show that the number of iteration with APSO algorithm is higher than PSO, but the MOE improvement is 3 % higher for scenario and 4 % higher for scenario2. The results shows that 13 % (scenario1) and 20 % (scenario2) particles outperformed and needed to be re-initialised resulting in improved MOE at the cost of increase of number of iteration by 18 % and 77 % respectively for scenario1 and scenario2. Also performance graph (Fig. 1) we can see that for scenario 1, PSO almost took 63 iteration to stabilize MOE value , where as APSO took almost 54 iteration to stabilize MOE. In performance graph (Fig. 4) ,we can see further that for scenario 2, PSO almost took 47 iteration to stabilize MOE value of 91 %, where as APSO took only 5 iteration to stabilize MOE value of 91 %. In scenario 2 PSO version after 50 iteration is almost showing no further improvement in solution , whereas in APSO version showing continuous gradual increment in MOE of solution.

6. CONCLUSIONS

The process of selection of area for deployment of units is one of the important factors in a commander's operational planning which is based on the appreciation of a particular commander for that scope of operations. It is not necessary that the areas identified would be the same for the different commanders in time and space for the operations. In the constructive simulation applications, there are requirements for automating many of the operational decision making to abstract the inputs to the desired level to meet the desired objective of the training/analysis. Automation of the tasks of the commanders below the specific hierarchy is an important aspect. Our approach to solving one such problem (deployment of subunits) as a case study has shown encouraging results. In our experimental set up, with GIS vector DEM data and 5 factors in two scenarios generated the Optimal different sub unit position within parent Unit. The deployments generated by our algorithms were discussed with military subject matter experts (SMEs) and were satisfactory.

Results of APSO have shown improvement as compared to standard PSO approximately by 3 - 4%. Results of different cases/scenarios illustrated in our case study have the potential in solving similar automated planning. The solution approach has also application in solving more complex nonlinear multi-objective planning problems in the military simulation domain.

REFERENCES

1. Reynolds, C.W. Flocks herds and schools: A distributed behavioural model. *Computer Graphics*, 1987, **21**(4), 25-34.
doi: 10.1145/37401.37406
2. Shahid, S. A comparative study of genetic algorithm and the particle swarm optimization. *Int. J. Electrical Eng.*, 2016, **9**, 215-223. ISSN 0974-2158.
3. Zaim, A.N. Genetic algorithm application in optimisation of wireless sensor networks. *Hindawi Publishing Corp. Sci. World J.*, 2014 , 1-15.
doi: 10.1155/2014/286575

4. Davis, L. Handbook of genetic algorithms, Van Nostrand Reinhold, New York, 1991.
5. Kennedy, J. & Eberhart, R. Particle swarm optimisation, Proceedings of ICNN', 95. *In* International Conference on Neural Networks, 1995, IEEE. doi: 10.1109/ICNN.1995.488968
6. Ming Cao, W.F. Swarm intelligence algorithms for weapon-target assignment in a multilayer defense scenario: Comparative study. *Symmetry*, 2020, **12**(5), 824. doi: 10.33 90/sym12050824.
7. Hassan, H.; Seyed. H.S.; Mehdi Dehghan, S.M.; & Karimi, Jalal. Hybrid form of particle swarm optimization and genetic algorithm for optimal path planning in coverage mission by cooperated unmanned aerial vehicle. *J. Aerospace Technol. Manage.,* 2020, **12**. doi:10.5028/jatm.v12.1169
8. Abhishek, B.; Ranjit, S. & Shankar, T. Hybrid PSO HSA and PSO GA algorithm for 3D path planning in autonomous UAVs. Springer Nature Switzerland AG, 2020. doi: 10.1007/s42452-020-03498-0
9. Lei, X. Selection of distribution center's location based on particle swarm optimisation in military logistics. *Comput. Sci., Comput. Eng. Design*, 2009, CORPUS id: 63837631
10. Gao, C.; Song,Y.; Shen, J. & Li, Y. Solving location problem of distribution center based on hybrid particle swarm algorithm" journal of computers. *J. Comput.,* 2018. doi:10.17706/jcp.13.10.1185-11912018
11. Jindal, V. A pre-emptive hybrid ant Particle optimization (HAPO-P) algorithm for smart transportation. *In* International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016. doi: 10.1109/ICACCI.2016.7732237
12. Y, Shi. Fuzzy adaptive particle swarm optimization. Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), IEEE, 2001, 101-106. doi: 10.1109/CEC.2001.934377
13. Y, Shi. Empirical study of particle swarm Optimization. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), IEEE, 1999. doi: 10.1109/CEC.1999.785511
14. Zhan, Z.H .; Zhang, J.; Yun Li, Y. & Shu-Hung Chung, H. Adaptive particle swarm optimisation. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), IEEE, 2009, **39**(6). doi: 10.1109/TSMCB.2009.2015956
15. Xie, X.F.; Zhang, W.J. & Yang, Z.L. Adaptive particle swarm optimisation on individual level. *In* 6th International Conference on Signal Processing, IEEE, 2002. doi: 10.1109/ICOSP.2002.1180009

ACKNOWLEDGEMENT

The authors would like to thank Sh JK Bhargava, Scientist 'G', Sh G.C. Brahma, Scientist 'E' and Col Vikram, ISSA for providing tactical domain inputs and support during implementation.

CONTRIBUTORS

Mr Sanjay Bisht obtained his M.Tech (Computer Technology and Application) from the Delhi Technological University, Delhi and working as Scientist 'F' at DRDO-ISSA, Delhi. His areas of interest are Wargame, Heuristic Optimisation, Genetic Algorithm, PSO, Simulated Annealing, and Intelligent Agent Technology.
He has contributed in domain setting to algorithm mapping, mathematical formulation, software implementation, testing.

Mr S.B. Taneja obtained his M.Tech (Computer Science) from the Indian Institute of Technology, Roorkee and working as Scientist 'H' at DRDO-ISSA, Delhi. He is involved in the design and development of Wargame Simulation systems. His areas of interest are Distributed simulation, Wargame simulation system design, Geographical information system, and agent-based models for constructive simulation.
He has contributed in tactical domain setting & model conceptualisation and validation.

Dr Vinita Jindal obtained her PhD in Computer Science from University of Delhi and working as a Professor in the Department of Computer Science, Keshav Mahavidyalaya, University of Delhi. She is working in the area of Artificial intelligence and networks. Her areas of interest include: Covert channels and their detection, Cybersecurity, Intrusion detection systems, Dark web, Deep learning, Recommender systems, Swarm intelligence and Vehicular networks.
She has contributed in PSO parameters tuning, draft checking, result validation and supervision and guidance .

Dr Punam Bedi obtained her PhD in Computer Science from University of Delhi and working as a Senior Professor in the Department of Computer Science, University of Delhi. Her research interests include Steganography and steganalysis, Cybersecurity, Intrusion detection systems, Recommender systems, Deep learning, Artificial intelligence for healthcare, and Artificial intelligence for agriculture.
She has contributed in algorithm performance validation, draft checking, supervision and overall guidance.