Joint 48th IEEE Conference on Decision and Control and
28th Chinese Control Conference
Shanghai, P.R. China, December 16-18, 2009

Discrete Invasive Weed Optimization Algorithm: Application to Cooperative Multiple Task Assignment of UAVs

Mohsen Ramezani Ghalenoei, *Student Member, IEEE*, Hossein Hajimirsadeghi, *Student Member, IEEE,* and Caro Lucas

Abstract— **This paper presents a novel discrete population based stochastic optimization algorithm inspired from weed colonization. Its performance in a discrete benchmark, time-cost trade-off (TCT) problem, is evaluated and compared with five other evolutionary algorithms. Also we use our proposed discrete invasive weed optimization (DIWO) algorithm for cooperative multiple task assignment of unmanned aerial vehicles (UAVs) and compare the solutions with those of genetic algorithms (GAs) which have shown satisfactory results in the previous works. UAV task assignment problem is of great interest among researchers and many deterministic and stochastic methods have been devised to come up with the problem. Monte Carlo simulations show successful results that verify better performance of DIWO compared to GAs in both optimality of the solutions and computational time.**

I. Introduction

INVASIVE weed optimization (IWO) is a continuous, stochastic numerical algorithm inspired from weed colonization which is proposed by Mehrabian and Lucas in [1]. IWO has shown successful results in many practical applications like optimization and tuning of a robust controller [1], developing a recommender system [2], design of encoding sequences for DNA computing [3], distributed identification and adaptive control of a surge tank [4], analysis of electricity markets dynamics [5], optimal positioning of piezoelectric actuators [6], and adaptive beamforming [7].

Many computational problems such as traveling salesman problem (TSP), vehicle routing problem (VRP), job scheduling, graph coloring, quadratic assignment, and routing for telecommunication networks are inherently discrete. Due to IWO's characteristics, e.g. remarkable outcomes in previous works and ease of implementation, we are encouraged to introduce a novel discrete version of IWO named as discrete invasive weed optimization (DIWO) and apply it to discrete combinatorial problems like time-cost trade-off (TCT) problem and cooperative multiple task assignment of unmanned aerial vehicles (UAVs).

UAVs are very prominent and practical nowadays because of their outperforming ability in both civilian and military applications like search, surveillance, rescue, information gathering, mapping buildings and facilities, fire detection and extinguishment, crop yield prediction [8], and other dangerous missions that would be risky for a human pilot. The lack of an on-board pilot, significant weight savings, longer endurance, and lower costs [9] are some of the other reasons that justify the emergence and prevalence of UAVs. Especially, increasing interest is devoted to investigation of collaborative behaviors of UAVs. For instance, protection of a ground convey of vehicles with help of a team of UAVs is discussed in [10].

Flight path planning, trajectory tracking, collision and obstacle avoidance, synchronization between cooperative tasks [11], motion control [8], wireless communication, visual sensing, and task allocation which is described and discussed more in this paper, are some of hot topics in UAV research area. Note that most of these topics are also common in the autonomous collective robotics [11]. The European project, civil UAV applications and economic effectively of potential configuration solutions (CAPECON), has been started in 2001 to identify and define civil UAV applications [12]. In the near future, a significant planning capability is required to execute UAVs missions without human interference [13]. Many of the state-of-the-art technologies and innovations such as sensor networks, and machine vision systems are developed in these vehicles for various applications.

UAV task assignment has been an active research area for the past few years [14]. This problem is a non polynomial-hard (NP-hard) combinatorial one and is similar and related to some common assignment and scheduling problems such as TSP, VRP, and dynamic network flow optimization (DNFO) [15].

There have been various evolutionary algorithms utilized to solve UAV task assignment problem. For example particle swarm optimization (PSO) algorithm is used in [15]. Other non-evolutionary techniques such as network flow optimization model, dynamic programming, branch and bound, decision tree [9], linear programming, binary linear programming (BLP), and mixed integer linear programming (MILP) are also exploited [14], [15]. In [14], a team theory approach is discussed and applied to this problem without any global communication among UAVs. Smith and Nguyen presented a fuzzy approach based on genetic programming for task scheduling of autonomous and cooperative vehicles in [16]. Task Assignment problem with balancing search is proposed and solved in [12] with predictive strategy. Gil *et al.* [11] solved task assignment problem coped with

M. Ramezani Ghalenoei and H. Hajimirsadeghi are with the Control and Intelligent Processing Center of Excellence for their M.S. at School of Electrical and Computer Engineering, University of Tehran, 14395–515, Tehran, Iran (phone: +98-912-2248494; emails: {moh.ramezani, h.hajimirsadeghi}@ece.ut.ac.ir).

C. Lucas is with the Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, and School of Cognitive Sciences, IPM, 14395 – 515, Tehran, Iran (phone: +98-21-887-166-24; fax: +98-21-887-250-29; e-mail: lucas@ipm.ir).

imperfect communication and arbitrarily finite delays. In [17], Shima *et al.* proposed a specific task assignment problem and resolved it with genetic algorithms (GAs). In this work, we take their assumptions [17] and compare our results with theirs. In addition, on closer inspection, we propose a modified GA (M-GA) which preserves a limited number of infeasible solutions in each generation, because it is possible that infeasible solutions may have more useful information than feasible ones [1].

The remainder of this paper is organized as follows. In Section II, DIWO is introduced while in Section III, we go through our first case study, the TCT problem, and compare the results with five other evolutionary methods. UAV task assignment problem is elaborated in Section IV and solved with DIWO in Section V where simulation results and analyses are given. Finally, conclusions are drawn, and future works are presented in Section VI.

II. Discrete Invasive Weed Optimization

Due to IWO's distinctive properties, its global and local abilities for exploration and exploitation, and also its successful results in a considerable number of applications after a short time of its development, we are motivated to introduce DIWO. DIWO is the modified version of IWO, suitable for discrete optimization problems like TCT and UAV task assignment. The framework of DIWO is the same as IWO's, but some considerations are taken for exploration in discrete search spaces. At first, we explain IWO briefly [1], and then transform it to the discrete version.

1) *Initialization:* A population of initial seeds (N_0) is randomly being dispread over the search space.
2) *Reproduction*: The individuals, after growing, are allowed to reproduce new seeds linearly depending on their own, the lowest, and the highest fitness of the colony (all of plants). The procedure is illustrates in Fig. 1 [1]. Note that maximum (S_{max}) and minimum (S_{min}) number of seeds are predefined parameters of the algorithm and adjusted according to structure of problem.
3) *Spatial Dispersal*: The generated seeds are being randomly scattered with a normal distribution over the search space. The mean of distribution is equal to the location of parent plant, but standard deviation (SD), σ, will be reduced from a specified initial value, $\sigma_{initial}$, to the final value, σ_{final}, according to (1).

$$\sigma_{iter} = \frac{(iter_{max} - iter)^n}{(iter_{max})^n}\left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final}. \qquad (1)$$

Where σ_{iter} is the SD at the present step, and $\sigma_{initial}$, σ_{final}, $iter_{max}$ (maximum number of iterations), and n (modulation index) are other parameters. This nonlinear modification has shown satisfactory performance in many simulations [1]. This assumption means that seeds will be randomly distributed such that they lie close to the parent plant [3].

4) *Competitive Exclusion*: When the maximum number of population in a colony is reached (P_{max}), each weed is allowed to produce seeds and spread them according to the mechanism mentioned in step 2 and step 3, respectively. After that, new seeds with their parents are ranked together with respect to their fitness. Next, weeds with lower fitness are eliminated to reach the maximum allowable population size in a colony.

   This mechanism by using the "survival of the fittest" idea [18] (a common concept in evolutionary algorithms) gives a chance to plants with lower fitness to reproduce, and if their offsprings have good fitness, they can survive in their offspring's existence [1].
5) *Termination Condition*: The whole process continues until the maximum number of iterations has been reached, and we hope that the plant with the best fitness is the closest one to the optimal solution.

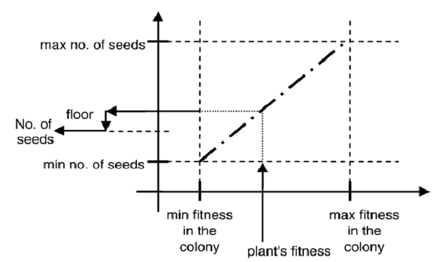

Fig. 1. Seeds reproduction procedure [1]

In DIWO algorithm, the processes for initialization, reproduction, competition exclusion, and also termination are completely the same as IWO. But spatial dispersal is modified to random selection of solutions from a neighboring hypercube in the discrete space of solutions around the plant with a normal distribution. The main challenges in this method are definition of the neighboring hypercube (i.e., a metric for distance between solutions), and then selection of an appropriate spatial dispersal approach providing the important attribute of original IWO which is "offspring seeds should lay near their parent plant." These issues are somehow heuristic and task dependent, varying from one problem to another. The pseudo code for DIWO is given in Fig. 2.

1. Genearte random population of N_0 individuals from the set of feasible solutions
2. i=: 1
3. do
   a. Compute maximum and minimum fitness in the colony
   b. For each individul $w \in W$
      i. Compute the number of seeds for w, corresponding to its fitness
      ii. Randomly select the seeds from the feasible solutions around the parent plant (w) in a neighborhood with normal distribution
      iii. Add the generated seeds to the solution set, W
   c. If $|W| = N > p_{max}$
      i. Sort the population N in descending order of their fitness
      ii. Truncate population of weeds with smaller fitness until $N = p_{max}$
   d. i=: i+1
4. Repeat 3 until the maximum number of iterations

Fig. 2. Pseudo code for DIWO algorithm

TABLE I
TEST PROBLEM FOR DISCRETE OPTIMIZATION

		Option 1		Option 2		Option 3		Option 4		Option 5	
Activity no.	Depends on	Duration (days)	Cost ($)	Duration (days)	Cost ($)	Duration (days)	Cost ($)	Duration (days)	Cost ($)	Duration (days)	Cost ($)
1	-	14	2400	15	2150	16	1900	21	1500	24	1200
2	-	15	3000	18	2400	20	1800	23	1500	25	1000
3	-	15	4500	22	4000	33	3200	-	-	-	-
4	-	12	45000	16	35000	20	30000	-	-	-	-
5	1	22	20000	24	17500	28	15000	30	10000	-	-
6	1	14	40000	18	32000	24	18000	-	-	-	-
7	5	9	30000	15	24000	18	22000	-	-	-	-
8	6	14	220	15	215	16	200	21	208	24	120
9	6	15	300	18	240	20	180	23	150	25	100
10	2, 6	15	450	22	400	33	320	-	-	-	-
11	7, 8	12	450	16	350	20	300	-	-	-	-
12	5, 9, 10	22	2000	24	1750	28	1500	30	1000	-	-
13	3	14	4000	18	3200	24	1800	-	-	-	-
14	4, 10	9	3000	15	2400	18	2200	-	-	-	-
15	12	12	4500	16	3500	-	-	-	-	-	-
16	13, 14	20	3000	22	2000	24	1750	28	1500	30	1000
17	11, 14, 15	14	4000	18	3200	24	1800	-	-	-	-
18	16, 17	9	3000	15	2400	18	2200	-	-	-	-

III. TIME-COST TRADE-OFF PROBLEM

In this section, we want to solve the time-cost trade-off (TCT) problem introduced in [19] and evaluate the performance of our proposed DIWO in comparison with five other evolutionary algorithms (EAs): particle swarm optimization (PSO), ant colony optimization (ACO), genetic algorithms (GAs), memetic algorithms (MAs), and shuffled frog leaping (SFL) using the simulation results provided in [20]. As it is mentioned in [20], it was tried to find the best possible parameters for the above-mentioned algorithms to have a reasonable comparison and providing guidelines for determining the best parameters for each algorithm. The detailed description of the TCT problem is characterized in Table I. The problem consists of 18 activities with a maximum of 5 options (methods of construction) for each one and indirect cost of \$500/day. Options are decision variables which should be determined so that the following cost function (J) is minimized:

$$J = T \times I + \sum_{i=1}^{n} C_{ij}, \tag{2}$$

where $n =$ number of activities; $C_{ij} =$ direct cost of activity i with option j; $T =$ total project duration; and $I =$ daily indirect cost. In this problem, the optimization process is said to be successful when the project is terminated by 110 days. Also the optimal cost for this problem is \$161,270.

For the process of seeds generation, a distance metric between two options j and j' is defined equal to $(Duration_{ij} - Duration_{ij'}) \times I + C_{ij} - C_{ij'}$, i.e., for activity i the distance between option j (parent plant's option) and the other options is calculated with this metric and then normalized. Next, one of the options is randomly chosen with a normal distribution according to its distance from j and the current value of SD (σ_{iter}). Simulation results for this problem by performing 20 trial runs are summarized in Table II. It can be observed that DIWO outperforms other algorithms in both percentage of success and average value of the cost function. The parameters of DIWO for this experiment are also listed in Table III. With this parameters setting, the average number of function evaluations is about 20000, addressing the computational complexity of DIWO in this problem. Note that in simulation of DIWO, the process stopped when maximum allowable iteration was reached, but for other EAs, a different termination criterion was used in [20].

TABLE II
RESULTS OF THE TCT PROBLEM FOR COMPARISON WITH 5 EAS

Alg.	Minimum project duration (days)	Average project duration (days)	Minimum cost ($)	Average cost ($)	%Success rate
DIWO	110	110	161,270	161,582	100
GAs	113	120	162,270	164,772	0
MAs	110	114	161,270	162,495	20
PSO	110	112	161,270	161,940	60
ACO	110	122	161,270	166,675	20
SFL	112	123	162,020	166,045	0

TABLE III
DIWO PARAMETER VALUES FOR TCT PROBLEM OPTIMIZATION

Symbol	Quantity	Value
N_0	Number of initial population	10
$iter_{max}$	Maximum number of iterations	400
dim	Problem dimension	18
P_{max}	Maximum number of plant	40
S_{max}	Maximum number of seeds	3
S_{min}	Minimum number of seeds	1
n	Nonlinear modulation index	3
σ_{init}	Initial value of standard deviation	1
σ_{final}	Final value of standard deviation	0.08

To illustrate the transient performance of DIWO algorithm, trace of mean, minimum, maximum, and standard deviation of the objective values for 20 runs are depicted in Fig. 3 and Fig. 4, respectively.

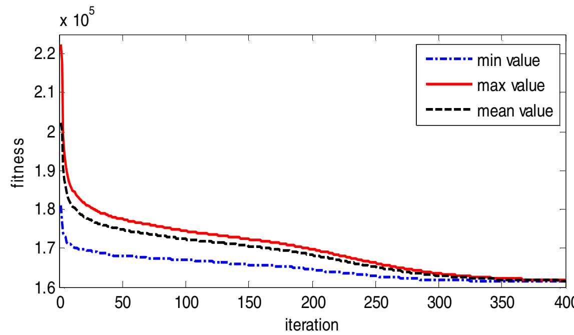

Fig. 3. Mean, max, min of cost functions in the TCT problem.

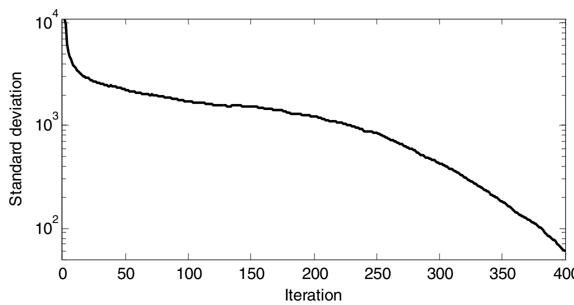

Fig. 4. Standard deviation of cost functions in the TCT problem.

IV. Cooperative Multiple Task Assignment of UAVs

In this section, the cooperative multiple task assignment problem (CMTAP) for a set of homogeneous UAVs is defined according to [17]. Let $T = \{1,2,\dots,N_t\}$ denotes the set of stationary ground targets, $V = \{1,2,\dots,N_v\}$ denotes the set of UAVs performing the tasks $M = \{1,2,\dots,N_m\}$, on the specified targets. For a common example, tasks can be categorized in the form of $M = \{Classify, Rescue, Verify\}$ for a generic surveillance and rescue mission. Task assignment is allocating UAVs (V) to perform various tasks (M) at several targets (T) so as to minimize an objective function (J) with respect to various constraints. In this problem the task precedence and coordination should be considered to make feasible solutions, i.e., in the above-mentioned mission, the procedure of rescue on a target can be performed exactly only after the classification process, and also before the verification maneuver on that target. More other constraints might be imposed on the problem, e.g., fuel consumption constraints, timing constraints and flyable trajectories. Note that in real time applications, predictions can be so inaccurate due to the nature of the problem and result in such poor decisions that it is simply better not to predict ahead [11].

The UAV CMTAP which is a well-known NP-hard combinatorial optimization problem in computer science can be represented by a tree which is helpful for problem visualization and encoding [17]. The nodes of tree show the assignment of a UAV $i \in V$ to a task $k \in M$ on a target $j \in T$ at a specific time, so the tree can represent both the decision and state spaces at the same time. The children of each node incorporate all of the possible assignments that can be made according to the remaining tasks. Hence, a branch of the tree, from a root node to a leaf node makes a feasible solution for the proposed problem (the length of this branch is equal to $N_c = N_t N_m$).

A number of performance criterions have been introduced for CMTAP; among them the cumulative distance (R) and the longest distance travelled by the UAVs are more popular. The former aims to minimize the use of overall group resources while the latter is searching for the minimum time for the team to terminate the whole mission. In this study, we use the cumulative form for the simulations, which is formulated in (3). Note that in simulations, Euclidean distance is used as an estimate of the flyable trajectory length.

$$J = \sum_{i=1}^{N_v} R_i \tag{3}$$

For encoding the solutions, the proposed genetic scheme in [17] is employed. In this method, unlike the conventional string encoding approach, an individual is represented by a matrix. The individual matrix has two rows and N_c columns. The columns are the cells which are equivalent to the nodes in the tree representation. The first row is dedicated for encoding the assignment of vehicles while the second row presents the sequence of tasks to be performed on the targets. Actually, the second row is encoded by the numbers in the set of targets T, and the ordering of the appearance of each target (from left to right) determines which task is being performed. Hence, to have a feasible solution, each target should be appeared in the second row for exactly N_m times. An example matrix for a problem with two UAVs, three targets, and three tasks is illustrated in Fig. 5.

Vehicle	2	1	1	2	2	1	2	2	1
Target	1	3	1	2	3	2	1	2	3

Fig. 5. Example of individual matrix for encoding the solutions

The number of feasible solutions for this problem can be used as a measure of the problem's computational complexity. In [17], it is proved that the number of feasible solutions (N_f) can be calculated by (4).

$$N_f = \frac{(N_t N_m)!}{(N_m)^{N_t}} {N_v}^{N_t N_m} \tag{4}$$

It is apparent that N_f and subsequently the computational complexity would increase very rapidly as the number of targets, tasks, or UAVs increases. So, devising an algorithm which can make a near optimal feasible solution with speed and accuracy balance [18] and monotonically improvement through the time is of great interest for this problem.

Evolutionary algorithms which have shown successful results to tackle with complexity of discrete combinatorial optimization problems might be the best choice to solve this problem. They may also offer the possibility of multiple parallel processors implementation on embedded systems [17].

V. SIMULATIONS AND RESULTS

As discussed in the previous section, the UAV cooperative multiple task assignment problem is a very complex combinatorial problem. In this section, we elaborate DIWO for UAV CMTAP in a centralized manner of control and compare it with GA which showed a good performance in [17]. Moreover, for inspection of trade off between feasibility and infeasibility of solutions, we also propose a modified GA (M-GA) to compare with two other algorithms. M-GA method preserves a limited number of infeasible solutions in each generation. It is possible that infeasible solutions may have more useful information than feasible ones and besides, the whole process might reach to its optimum point more easily if the procedure can cross an infeasible area.

All steps in our proposed DIWO are the same as the original IWO except the spatial dispersal step. The procedure of dispersal is as follows. At every iteration variable $L \in \{1, 2, \ldots, N_c\}$ which specifies the start cell of the dispersal is computed according to (5), and then cells $L, L+1, \ldots, and\ N_c$ are selected and changed randomly to make a new seed.

$$L(iter) = floor\left(\frac{N_c - 1}{\ln(iter_{max})} \times ln(iter) + 1\right) \tag{5}$$

Equation (5) has shown satisfactory performance in simulations, while provides a good balance between exploration and exploitation.

This new seed is established upon its parent and due to the trend of L which is evident in Fig. 6, their resemblance increase, as the simulation goes on. This specific trend of L has a similarity to SD in original IWO (1). Indeed, this method of spatial dispersal, in this specific problem, is consistent with the law that "offspring seeds should lay near the parent plant." With this formulation, the number of parameters in DIWO is five as: N_0 = number of initial population; $iter_{max}$ = maximum number of iterations; P_{max} = maximum number of plants; S_{max} and S_{min} = maximum and minimum number of seeds.

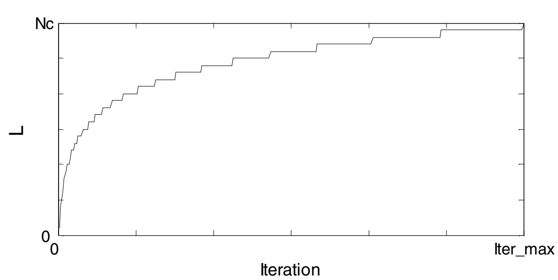

Fig. 6. L vs. iteration value

In the GA [17] and our DIWO algorithm, we enforce each solution to be a feasible one. But in the M-GA we let some infeasible solutions stay in the population. (We assign this parameter in this problem equal to 30) This approach consumes less time than GA and may result in better performance which is apparent in Table IV and Table V. Simulation results for scenario 1 and 2 by performing 20 Monte Carlo trial runs are summarized in Table IV and Table V, respectively. Scenario 1 is a small size problem consisting of 3 targets and 4 vehicles and scenario 2 is a large one with 10 targets and 8 vehicles. We use the Euclidean path as an estimate of the flyable trajectory for simulating the scenarios.

TABLE IV
RESULTS OF SCENARIO 1 MEAN OF 20 MONTE CARLO RUNS

Algorithm	Cost (Km)	Computational Time (s)
GA	1289.05	17.24
DIWO	1203.51	8.42
M-GA	1244.20	17.23

TABLE V
RESULTS OF SCENARIO 2 MEAN OF 20 MONTE CARLO RUNS

Algorithm	Cost (Km)	Computational Time (s)
GA	5847.59	74.73
DIWO	5703.75	54.91
M-GA	5872.88	74.62

It can be observed that DIWO outperforms two other algorithms in both the minimization of the objective function and computational time. The workstation used for simulations is the same for all of algorithms and the coding structures are tried to be efficient, so the computational time comparison is fair and reasonable.

The parameters of DIWO and GA (from [17]) are listed in Table VI. The values in parentheses are just related to scenario 1 and the others are the same for both scenarios. Note that in the simulations, the optimization process stopped when maximum allowable iteration was reached (the same for all of algorithms, i.e., N_g=$iter_{max}$).

TABLE VI
PARAMETER VALUES

Alg.	$Ns= N_0$	$Ng = iter_{max}$	Ne	Pm	Pc	$Pmax$	$Smax$	$Smin$
GA	200	300(200)	6	0.01	0.94	--	--	--
DIWO	200	300(200)	--	--	--	30	10(6)	1
M-GA	200	300(200)	6	0.01	0.94	--	--	--

In Table VI, N_s = number of offsprings in each iteration for GAs; N_g = number of generations which acts as a termination criteria; N_e = number of elite offsprings; P_m = Probability of mutation; and P_c = Probability of crossover. It is noteworthy that the initial populations in Monte Carlo runs are the same for all the algorithms to eliminate the effect of initialization. For a better comparison, the mean of 20 Monte Carlo simulation results for scenario 1 and 2 are depicted in Fig. 7 and Fig. 8, respectively.

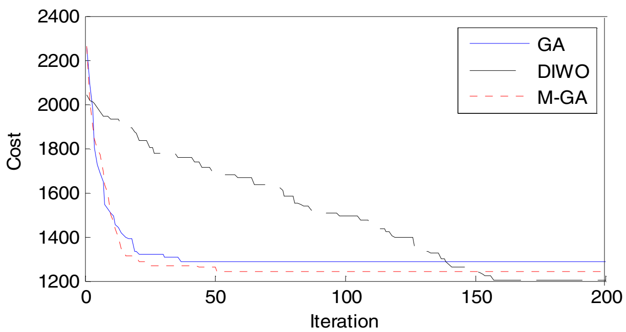

Fig. 7. Mean of 20 Monte Carlo runs, Results of scenario 1.

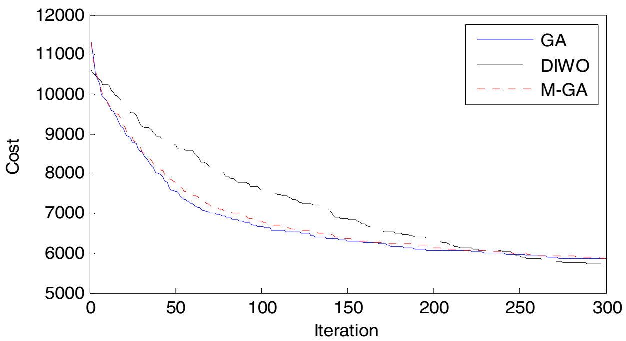

Fig. 8. Mean of 20 Monte Carlo runs, Results of scenario 2.

VI. CONCLUSION

Discrete Invasive Weed Optimization (DIWO) a novel Bio-inspired stochastic optimization algorithm was proposed in this work. DIWO framework is similar to IWO except for the dispersal spatial step which is somehow task dependent and heuristic. Performance of DIWO was evaluated and compared with five other evolutionary algorithms through a Time-Cost Trade-Off problem.

Also its application to UAV task assignment is investigated and compared with two different GAs. The cost function is assumed to be the total travelling path of all vehicles to fulfill the whole mission. Monte Carlo simulations verified DIWO capability for optimization by yielding better results (lower cost) in less computational effort. In practical cases, predictions can be very imprecise so if any uncertainty such as failure of one task happens, the optimization process should be performed again with new initialization. So, less computational time and ease of practical implementation on embedded systems make the proposed algorithm beneficial in real time decision-making situations. For further investigations, the effect of incorporating surveillance to the task assignment problem, communication imperfection and consideration of flyable path instead of Euclidean path can be surveyed.

REFERENCES

[1] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from invasive weed colonization," *Ecological Informatics*, vol. 1, pp. 355-366, 2006.

[2] H. Sepehri-Rad and C. Lucas, "A recommender system based on invasive weed optimization algorithm," in *Proc. IEEE Congress on Evolutionary Computation*, 2007, pp. 4297–4304.

[3] X. Zhang, Y. Wang, G. Cui, Y. Niu, and J. Xu, "Application of a novel IWO to the design of encoding sequences for DNA computing," *Computers and Mathematics with Applications* (2008), doi:10.1016/j.camwa.2008.10.038.

[4] H. Hajimirsadeghi and C. Lucas, "A hybrid IWO/PSO algorithm for fast and global optimization," in *Proc. EUROCON 2009,* St. Petersburg, RUSSIA, May 2009.

[5] M. Sahraei-Ardakani, M. Roshanaei, A. Rahimi-Kian, and C. Lucas, "A Study of Electricity Market Dynamics Using Invasive Weed Optimization," in *Proc. IEEE Symposium on Computational Intelligence and Games*, Perth, Australia, Dec. 2008.

[6] A. R. Mehrabian and A. Yousefi-Koma, "Optimal positioning of piezoelectric actuators of smart fin using bio-inspired algorithms," *Aerospace Science and Technology*, vol. 11, pp. 174–182, 2007.

[7] M. Roshanaei, C. Lucas, and A.R. Mehrabian, "Adaptive beamforming using a novel numerical optimization algorithm," *IET Microw. Antennas Propag.*, 2009, Vol. 3, Iss. 5, pp. 765–773.

[8] J. Azinheira and A. Moutinho, "Hover control of an UAV with backstepping design including input saturations," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 3, pp. 517-526, May. 2008.

[9] S. J. Rasmussen and T. Shima, "Tree search algorithm for assigning cooperating UAVs to multiple tasks," *Int. J. Robust Nonlinear Control*, vol. 18, pp. 135–153, 2008.

[10] S. C. Spry, A. R. Girard and J. k. Hedrick, "Convoy protection using multiple unmanned aerial vehicles: organization and coordination," presented at the 2005 American Control conference, Portland, OR, USA.

[11] A. E. Gil, K. M. Passino, S. Ganapathy, and A. Sparks, "Cooperative task scheduling for networked uninhabited air vehicles," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 44, no. 2, pp. 561-581, Apr. 2008.

[12] Y. Jin, Y. Liao, A. A. Minai, and M. M. Polycarpou, "Balancing search and target response in cooperative unmanned aerial vehicle (UAV) teams," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 36, no. 3, pp. 571-587, Jun. 2006.

[13] M. Alighanbari and J. P. How, "A robust approach to the UAV task assignment problem," *Int. J. Robust Nonlinear Control*, vol. 18, pp. 118–134, 2008.

[14] P. B. Sujit, A. Sinha and D. Ghose, "Multi-UAV task allocation using team theory," in *Proc. 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, Seville, Spain, pp. 1497–1502.

[15] F. Pan, X. Hu, R. Eberhart, and Y. Chen, "A new UAV assignment model based on PSO," presented at the 2008 IEEE Swarm Intelligence Symposium, St. Louis, MO, USA.

[16] J. F. Smith III and T. H. Nguyen, "Autonomous and cooperative robotic behavior based on fuzzy logic and genetic programming," *Integrated Computer-Aided Engineering*, vol. 14 pp. 141–159, 2007.

[17] T. Shima, S. J. Rasmussen, A. G. Sparks, and K. M. Passino, "Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms," *Computers & Operations Research,* vol. 33, pp. 3252–3269, 2006.

[18] K. M. Passino and T. D. Seeley, "Modeling and analysis of nest-site selection by honeybee swarms: the speed and accuracy trade-off," *Behavioral Ecology and Sociobiology*, vol. 59, pp. 427–442, 2006.

[19] C. Feng, L. Liu, and S. Burns, "Using genetic algorithms to solve construction time-cost trade-off problems," *J. Comput.Civil Eng.*, vol. 11, no. 3, pp.184–189, 1997.

[20] E. Elbeltagia, T. Hegazyb, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, pp. 43–53, 2005.