

CFO: A New Compact Swarm Intelligent Algorithm for Global Optimization and Optimal Bipedal Robots Walking

Lyes TIGHZERT

Laboratoire de Technologie Industrielle et de l'Information
(LTII), Faculté de Technologie,
Université de Bejaia,
06000 Bejaia, Algérie
ltighzert@gmail.com

Boubekeur MENDIL

Laboratoire de Technologie Industrielle et de l'Information
(LTII), Faculté de Technologie,
Université de Bejaia,
06000 Bejaia, Algérie
bmendil@gmail.com

Abstract—this paper introduces compact swarm intelligence. It presents a new estimation of distribution algorithm based on the swarm behavior of fireflies. The proposed algorithm is called the compact firefly optimizer (cFO). It uses a compact representation to store the population of the fireflies. Thus, the whole population is represented by a probabilistic distribution function. The compact swarm intelligence proposed allows the reduction of the computational capacity required to evolve the population toward the optimal solution. An important numerical test is used to evaluate the performance of the proposed algorithm. The results obtained show that the proposed compact firefly optimizer (cFO) is very competitive. Furthermore, the new concepts of compact swarm robotics are also introduced. The algorithm is applied to describe the first compact swarm robotics system. The objective is to realize an optimal stable walking by bipedal robot. Discussions, propositions and analysis let to think to a new important starting point for an emerging field of compact swarm intelligence, robotics, search and optimization.

Keywords—swarm intelligence; compact representation; optimization; Firefly algorithm; compact swarm robotics; compact swarm intelligence; bipedal walking; path planning.

I. INTRODUCTION

Swarm intelligence (SI) is the cooperative behavior of species living in groups [1]. Each swarm agent is capable of interacting with its environment and locally with the others particles in its side. The collective behavior of the decentralized swarm makes it self-organizing system. On 1989, Gerardo Beni and Jing Wang have introduced those concepts into artificial intelligence field [2]. Since then, researchers around the world have proposed the exploitation of the principle behind swarm intelligence to propose very efficient algorithms for difficult engineering problems. The proposed algorithms consist of population-based strategies. The particles ability, e.g. extracting information from environment and sharing them in the crowd, leads the whole population toward the optimal sub region in the search space. In the literature a very performant optimizers, based on swarm

intelligent systems, have been proposed by several authors. This includes ant colonies [3], birds [4], bees colonies [5], fish schooling [6] and fireflies [7] etc.

The firefly algorithm (FA) was introduced by Xin-She Yang in Cambridge University based on the modelling the brightness and attractiveness characteristics of the firefly swarm [7]. It is a population based algorithm. It uses the fireflies' position to represent the temporally found solutions. Each firefly is capable of radiating signals to attract the others toward its own position. The fireflies are assumed to be bisexual and the attractiveness is not for sexuality. To imitate the swarm behavior, we associate for each firefly a radiating signal to attract the others. If the x_i firefly is better than x_j , the latter will be attracted and moved toward x_i . Hence, the best fireflies will attract the others toward them. To sum up the algorithm steps with more simplicity we present it through three steps:

1- Brightness: depending on the distance between two fireflies and the atmospheric absorption coefficient, a lightning signal is associated to each one. The mathematical formula is given in (1)

$$I(r) = I_0 e^{-\gamma r^2} \quad (1)$$

Where I_0 is the intensity of the light source; γ is the absorption coefficient and r is the distance between the two considered fireflies.

2- Attractiveness: the attractiveness can be calculated as follow :

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (2)$$

Where β_0 is the attractiveness of the firefly when $r = 0$.

3- The moving step : For the entire population is moved toward the cost-efficient ones with the flowing model :

$$x_j^{t+1} = x_j^t + \beta_0 e^{-\gamma r^2} (x_i^t - x_j^t) + \alpha \cdot randn(-1,1) \quad (3)$$

Where α : is the mutation coefficient which is generally a self-adaptive parameter decreasing through iterations, $rand(-1,1)$ a normal randomized number between $[-1,1]$.

The firefly algorithm is known to be very performant [8]. An important comprehensive review on FA is given by Fister et al. [9]. FA is widely used in several real world applications. We can cite: clustering in wireless networks [10], power systems [11], database [12], etc. The association of swarm intelligence with robotics is called swarm robotics [13]. FA-based swarm intelligent robotics systems and approaches are also proposed [14].

In this paper, we propose a new swarm-based estimation of a distribution algorithm. The proposed algorithm is called the compact firefly optimizer (cFO). Therefore, it introduces compact swarm intelligence. Numerical experiments, tests, for convergence quality, accuracy, stability and speeders are provided. The cFO promises the reduction of computer devices required by others SI systems. Furthermore, the new concepts of compact swarm intelligent robotics is also introduced, evaluated and discussed. The results presented in this study, the concepts and paradigms used let us to think that a new interesting field is emerging. This dissertation gives a step toward compact swarm intelligence/robotics. The remaining of this paper is organized as follows: in the next section we introduce, analysis the proposed algorithm. In the third section: a set of benchmark test, including unimodal and multimodal functions, are used to analysis, compare the proposed algorithm to FA, DE, PSO, and compact differential evolution (cDE). The results shown that cFO is very competitive. In the fourth section, compact swarm robotics is introduced. We give application of cFO for optimal bipedal walking. The section five concludes this study.

II. THE COMPACT FIREFLY OPTIMIZER

In this section we present the proposed algorithm. It is directly inspired from the swarm behavior of the fireflies. The proposed algorithm combines the logic of compact evolutionary algorithms (cEAs) [15] and the swarm intelligence inspired from FA. First, we give a short review on the compact representation concepts and implementation.

The compact evolutionary algorithms were proposed by researchers to overcome the problem of population size and memory/capacity required by SIs and EAs [15, 26]. Harik et al. have proposed the first compact genetic algorithm [16] that uses binary representations. C. Wook introduced the Elitism-Based Compact Genetic Algorithms [17]. The later proposed cGA variants, consisting of non-persistent and persistent cGAs, that reconcile cGA with elitism. The real valued compact genetic algorithm (rcGA) was proposed by Ernesto Mininno et al. [18]. This variant encodes the population by mean of a predefined probability density function (PDF). The same paradigms have been expended to design the compact differential evolution (cDE) [19].

In this paper we propose compact swarm intelligence. The proposed algorithm, called cFO, introduces the compact representation into FA. Unlike other population-based (swarm) intelligent algorithms, the cFO uses a probability vector (PV) to represent the swarm. Like in (cDE), the PV

contains only the mean and the standard deviation of each parameter of the optimization problem. Hence, we use a probabilistic density function to estimate the repartition of the population. We sum up the different steps of cFO as follows: In each cycle, two individuals are generated from the virtual population defined by PV. After the evaluation of their fitness,

Algorithm 1: Compact Firefly Optimizer

```

1. N : Virtual population size
2.  $\lambda$  : Initial standard deviation
3. m : problem dimension
4.  $R_{dump}$  = Initialize the dumping ratio
5.  $R$  = Initialize the mutation coefficient
6. t=1 /* iteration index */
7. for j=1:m do /* Initialization of PV */
8.    $\mu_j^t = 0$ ;  $\sigma_j^t = \lambda$ ;
9. end
10. Elite=generate(PV) /* Assume Elite */
11. while Termination condition /* Main loop */
12.    $G_1$ =generate(PV(t))
13.    $G_2$ =generate(PV(t))
14. [winner, loser] = compete( $G_1, G_2$ )
15. Move the winner toward the elite
16. for i=1: m /* Crossover */
17.   if rand < pCr
18.     winner(i)=elite(i)
19.   endif
20. endfor
21. [winner, loser] = compete(winner, elite)
22. elite = winner /* Update elite */
23. for i=1 : m do /* updating PV */
24.    $\mu_i^{t+1} = \mu_i^t + 1/N \times (w_i^t - l_i^t)$ ;
25. /*  $l_i$  and  $w_i$  are the genes of the loser and winner */
26.  $[\sigma_i^{t+1}]^2 = [\sigma_i^t]^2 + [x_i^t]^2 - [x_i^{t+1}]^2 + \frac{1}{N} \times ([w_i^t]^2 - [l_i^t]^2)$ ;
27. endfor
28.  $R = R * R_{dump}$ 
29. while j <=10 /* refinement the elite */
30.   NewElite = (1-RL) * elite + RL * randn(-1,1)
31.   [winner, loser] = compete(NewElite, elite)
32.   elite = winner /* Update elite */
33. for i=1 : m do /* updating PV */
34.    $\mu_i^{t+1} = \mu_i^t + 1/N \times (w_i^t - l_i^t)$ ;
35. /*  $l_i$  and  $w_i$  are the genes of the loser and winner */
36.  $[\sigma_i^{t+1}]^2 = [\sigma_i^t]^2 + [x_i^t]^2 - [x_i^{t+1}]^2 + \frac{1}{N} \times ([w_i^t]^2 - [l_i^t]^2)$ ;
37. endfor
38.  $RL = RL * R_{Dump}$ 
39. j=j+1
40. end while
41. end while
42. return elite and its cost

```

Fig. 1. The pseudo code of the compact firefly optimizer

a brightness signal is associated to each one, including the best ever found and the two sampled individuals. The intensity of the brightness signal associated to each individual is calculated using the same formula presented in the previous section for FA. A tournament selection is then performed. We use the same attractiveness process used in the standard version of firefly algorithm. The pseudo code of cFO is presented in Fig.1.

After initialization step, the algorithm sampled an elite from VP. The sampling procedure uses a normal distribution function of means μ and standard deviation λ . This procedure is described in details in [18,19]. In the main loop, for each cycle, the algorithm generates from VP two individuals and uses a tournament selection. The winner of this tournament will be rewarded by allowing it to move toward the elite. e.g. one can propose to move the two sampled fireflies. The third step consists of crossing the moved firefly with the elite.

The probability of this crossover operator must be superior to 0.5 to assure good convergence. The fourth step is a competition between the elite and the new firefly obtained after the crossover operation. If the new produced firefly is better than the elite, we update the elite. The fifth step consists of adjusting the means and the standard deviation of the probability density function. This step is implemented using the same adaptation proposed in [18,19]. The sixth step is a local search strategy based on rate-learning refinement. This step is done using the equation (4)

$$x_{elite}^{t+1} = (1 - RL) * x_{elite}^t + RL * randn \quad (4)$$

Where RL is the rate-learning (memory percentage consideration). It is a self-adaptive parameter. When RL tends to 1, the algorithm randomizes and does not compute the next solution by taking great consideration of the actual position. In this case the algorithm does not use its memory consideration. When RL tends to 0, the next solution is computed depending on actual solution and the stochastic effect is small. In this stage, we decrease RL through iterations. The algorithm will learn to take consideration on the previous solution iteration after iteration. The algorithm starts with $RL=0.5$ and decreases it toward 0 using (5).

$$RL = RL * RL_{dump} \quad (5)$$

The objective is to give the algorithm more randomization in the first iterations and more memory consideration in the afterward iterations. This stage allows refinement, diversification and intensification. In the next section we provide a complete numerical investigation for the performance analysis of cFO and its validation. Maintaining the Integrity of the Specifications

III. NUMIRICAL EXPIRIMENTS

A. Test Functions

In order to demonstrate the performance of the proposed algorithm, a set of 12 famous benchmarks are used. Their mathematical forms are described in the sited references. The functions $f1-f5$ are unimodal and the remaining are multimodal. We test those functions for different dimensions: $D=10, 30$ and 50 . They are as fallow: Sphere function [20]; Schwefel's problem 2.22 [20]; Schwefel's problem 1.2 [20]; Schwefel's problem 2.21 [20]; Generalized Rosenbrock function [20]; Generalized Rastrigin function [20]; Generalized Griewank function [20]; Ackley function [20]; Zakharov function [21]; Alpine function [21]. Pathological function [21] and the Periodic problem [20].

B. Algorithms Comparison

In order to evaluate the performance of the proposed cFO, the results are compared to four optimization and search algorithms. The algorithms used are as fallow.

Algorithm 1, Compact Firefly Optimizer (cFO): is the algorithm proposed in this paper (Fig.2). It is inspired from the swarm behavior of fireflies. It uses a probability density function to encode the population. It requires less memory for its implementation and is suitable for real world applications.

Algorithm 2, the firefly algorithm (FA), it is the standard version of firefly algorithm [7].

Algorithm 3, Compact Differential Evolution (cDE): is the algorithm proposed in [19]. It can been seen as compact version of the DE. The cDE uses non-persistent elitism strategy. It uses the same schema differential evolution defined as the DE/best/rand/1.

Algorithm 4, Differential Evolution (DE): it the standard version by Storn and Prince [34].

Algorithm 5, Particle Swarm Optimization PSO: is the standard particle swarm optimization inspired from the social behavior of particle swarm. [see 35].

C. Algorithms' parameters

The parameters setting of the different algorithms are taken from their original literature. The maximum number of fitness evaluation is fixed to 50 000. For each algorithm, the parameters given bellow. The parameters of Firefly Algorithm are taken from [7], those of cDE from [19], those of DE from [36] and those of PSO from [23].

D. Simulation Results and Discussion

In this section we analysis and compare the cFO. In order to respect the maximum page for this conference, we give the most interesting results. The algorithms are compared in terms of ranking, convergence quality, accuracy, stability, and processing time.

1) Ranking: The results of the compared algorithms are ranked. Their ranking gives the rate of their success. The cFO gives a total of 51 % of good results. The FA gives 5%. The cDE gives 24%. The others gives 10% each one. The objective here is not demonstrate absolutely that cFO is more preferment then the others. We provide just a comparison study. The extension of this work, with more tests an experiments, can leads to a final conclusion on wish one is more performant.

2) Convergence Study: the compared algorithm are stochastic based swarm intelligent or evolutionary algorithms. As a consequence, their results differ from one run to another. Hence, each tested function, for the $D=10, 30$ or 50 , are repeated 50 times. Furthermore. To study the excellence of the compared algorithms and their accuracy, we compare the averaged optimal fitness over 50 independents runs. At each run, the optimal fitness is returned after 50000 iterations (fitness evaluations). The detailed results are summarized in tables II–VII. They give the mean and the standard deviation.

TABLE I
 Average (Aver.) and Standard Deviation (Sta. Dev.) of the Best Fitness Values Found by cFO, FA, cDE, DE, PSO for the set of functions F1 to F14

Function D		F1			F2			F3			
		10	30	50	10	30	50	10	30	50	
Algorithm	CFO	Aver.	5.09E-146	1.37E-144	3.8683E-7	3.8656E-5	0.03949	0.16635	3.552E-9	0.01525	0.089785
	Sta. Dev.	1.83E-146	4.58E-145	9.770E-7	2.3752E-4	0.03686	0.0576	2.1465E-8	0.005834	0.016794	
FA	Aver.	0.000122	0.0104564	0.0283076	0.0245640	0.0165435	0.2314348	0.002345	1.6345638	360.432568	
	Sta. Dev.	0.000934	0.002286	0.2434314	0.0075932	0.0164383	0.0034358	0.0147490	0.5335419	134.34258	
cDE	Aver.	0.8624	263.7583	423.4056	0.5003	0.908606	2.585880	1336.145	37350.97	100100.2	
	Sta. Dev.	5.5884	711.5054	914.5064	1.762067	2.036084	2.751408	1227.761	7711.845	17259.02	
DE	Aver.	2.044E-40	6.0404E-8	0.2461	1.778E-23	1.5574E-5	0.108288	0.285056	6883.577	31531.75	
	Sta. Dev.	3.4541E-40	2.3016E-8	0.0495	1.214E-23	2.9094E-6	0.01163	0.285646	1076.481	3751.417	
PSO	Aver.	2.666E-32	3.0137E-6	0.1444	3.908E-19	3.5163E-5	0.032358	6.006E-10	368.2142	7126.968	
	Sta. Dev.	7.157E-32	5.0858E-6	0.1265	8.446E-19	3.8750E-5	0.017076	2.264E-9	150.9673	1520.009	
Function D		F4			F5			F6			
		10	30	50	10	30	50	10	30	50	
Algorithm	CFO	Aver.	1.2685E-7	0.035561	0.074233	7.899338	27.76993	47.33945	5.131144	21.09080	11.65153
	Sta. Dev.	9.0539E-7	0.009287	0.012918	0.450653	0.280674	0.252148	7.034827	37.27785	47.12399	
FA	Aver.	0.0545368	0.053453	12.436471	3.232456	15.143526	46.020278	6.2015	50.64537	100.847363	
	Sta. Dev.	0.0020432	0.024362	10.32453	2.4352651	5.534262	100.64537	5.6785646	41.546370	54.6453738	
cDE	Aver.	100.0	100.0	100.0	42.00375	618.9767	2844.527	2.566987	12.88195	35.99267	
	Sta. Dev.	0.0	0.0	0.0	139.3829	1398.839	6102.021	3.910002	8.075877	11.72466	
DE	Average	2.761E-7	1.878261	22.62559	4.139185	17.47056	84.68554	0.0	26.99560	96.54391	
	Sta. Dev.	1.237E-7	0.253685	1.625283	1.788605	1.861150	16.45281	0.0	2.754542	6.670114	
PSO	Aver.	3.8479E-9	3.817139	13.76587	3.778750	37.08230	89.56919	2.865481	38.96080	91.74087	
	Sta. Dev.	5.4162E-9	0.843045	1.443106	0.524305	22.18057	35.28208	1.484323	10.08217	17.09201	
Function D		F7			F8			F9			
		10	30	50	10	30	50	10	30	50	
Algorithm	CFO	Aver.	0.124787	0.019794	1.5042E-5	2.6690E2	2.603E3	2.847E3	10.70671	8.285449	7.924328
	Sta. Dev.	0.317787	0.141329	1.0474E-5	1.905E1	1.722E2	2.031E2	10.08428	10.35054	10.32421	
FA	Aver.	12.546373	51.635373	54.645373	0.054638	0.143527	0.435232	585.66973	2417.6006	4032.08393	
	Sta. Dev.	9.6453733	67.645373	33.534262	0.0453621	0.326241	0.0736242	1634.6453	1721.3593	463.399233	
cDE	Aver.	1.246113	4.094425	10.51240	1849.759	560.5064	7661.705	5.603183	9.386307	11.56448	
	Sta. Dev.	6.789555	10.28160	18.86302	3864.991	9192.180	12266.23	6.136570	3.271249	2.191822	
DE	Aver.	0.0	4.685E-5	0.519167	1.272E-4	47.03910	4374.661	4.440E-15	7.831E-5	0.201510	
	Sta. Dev.	0.0	8.6375E-5	0.065155	1.272E-13	56.92969	275.9198	0.0	1.422E-5	0.026664	
PSO	Aver.	0.083944	0.011847	0.126878	570.0840	3418.602	7159.119	5.364E-15	5.679E-4	0.908231	
	Sta. Dev.	0.034021	0.012929	0.099374	210.7867	519.0818	925.4370	1.558E-15	6.122E-4	0.698934	
function D		F10			F11			F12			
		10	30	50	10	30	50	10	30	50	
Algorithm	CFO	Aver.	2.63E-145	0.034594	0.299229	1.14E-5	1.03E-4	1.51E-4	1.755634	9.514926	18.05094
	Sta. Dev.	1.90E-145	0.012772	0.067238	1.18E-5	4.77E-5	6.51E-5	0.520116	1.077688	1.511874	
FA	Aver.	0.0435285	0.1534149	0.222312	1.0286E-4	0.002623	1.077448	5.396E-6	1.2420E-5	4.95543E-5	
	Sta. Dev.	0.0153616	0.0147173	0.0519868	3.5923E-5	0.0000232	1.6275433	1.4652E-7	8.674E-6	9.591E-6	
cDE	Aver.	66.55966	685.8281	1500.600	0.00128	0.00186	0.00384	0.850309	3.244790	5.824111	
	Sta. Dev.	53.64350	135.5109	206.5784	0.00399	0.00447	0.00756	0.184848	0.503218	0.656763	
DE	Aver.	8.418E-4	72.92364	260.0215	3.617E-13	4.6373E-5	2.1424E-4	0.557888	2.444161	7.031129	
	Sta. Dev.	9.9220E-4	13.29351	31.16179	2.265E-12	5.4780E-6	1.1962E-5	0.134602	0.375145	0.322994	
PSO	Aver.	1.362E-15	2.198773	34.54683	1.834E-22	1.2529E-8	1.176E-5	1.352900	9.082079	17.68312	
	Sta. Dev.	2.538E-15	0.771874	7.011500	1.183E-21	8.996E-9	7.185E-6	0.324117	0.539834	0.745015	

TABLE II
 Ratio of the cFO time processing to others.

Algorithm Time Ratio	cFO/FA	cFO/cDE	cFO/DE	cFO/PSO	Algorithm Time Ratio	cFO/FA	cFO/cDE	cFO/DE	cFO/PSO
F1	0.1622	0.2535	0.1453	0.2024	F7	0.1813	0.2865	0.1680	0.2255
F2	0.1855	0.2958	0.1691	0.2302	F8	0.2093	0.3392	0.1909	0.2513
F3	0.7633	1.2727	0.7638	0.7130	F9	0.1920	0.3029	0.1745	0.2324
F4	0.1659	0.2609	0.1556	0.2120	F10	0.2050	0.3259	0.1848	0.2467
F5	0.2467	0.3977	0.2281	0.3005	F11	0.2776	0.4573	0.2593	0.3242
F6	0.2751	0.4374	0.2494	0.3214	F12	0.2076	0.3248	0.1882	0.2428

2) **Convergence Study:** the compared algorithm are stochastic based swarm and evolutionary algorithms. As a consequence, their results differ from one run to another. Hence, each tested function, for the $D=10, 30$ or 50 , are repeated 50 times. Furthermore. To study the excellence of the compared algorithms and their accuracy, we compare the averaged optimal fitness over 50 independents runs. At each run, the optimal fitness is returned after 50000 iterations (fitness evaluations). The detailed results are summarized in tables I. it gives the mean and the standard deviation. The algorithms are tested on different benchmarks. One can see that cFO surpasses the others.

3) **Algorithms Stability and accuracy:** The stability and the accuracy of the compared algorithm is measured based on the results' standard deviation. This later gives the repartition of the results around the averaged values found over 50 trials. From Table.1, we can see that the proposed cFO is more precise and more stable than the others.

4) **Algorithm speediness:** the time processing of an algorithm is a very important indicator to validate its performance. For real engineering problems we are searching for fast algorithms. The time processing of an algorithm depends on its strategy. In this study, we calculate averaged time of each algorithm for each function and then we calculate the ratio of the cFO time processing to other algorithms. We calculate T_{cFO}/T_{FA} , T_{cFO}/T_{cDE} , T_{cFO}/T_{DE} , T_{cFO}/T_{PSO} . If the results returned is superior to 1, this means that cFO is slower than the compared algorithm. In the inverse case, cFO is faster. The results are summarized in the Table II. We can see that cFO is faster than the others for all benchmark functions.

IV. APPLICATION TO OPTIMAL BIDEPDAL WALKING

In this section we propose, in the known of the authors, the first compact swarm robotic system. The swarm robotic is the utilization of swarm intelligence in robotic systems [13]. As described above, we have introduced in this paper the concept of compact swarm intelligence, we give here its application in robotics. But before this, some concepts are introduced and discussed.

we introduce the concept of compact swarm robotics. This later can be defined as the application of compact swarm intelligence to robotics. In compact swarm robotics, the variables and information that require excessive computational capacity, e.g. memory, CPU etc., are represented by a probabilistic density functions. This allows a good and fast evolution of the swarm intelligence embodied on the robot. The information are compacted in the memory and can be regenerated by means of the destitution function used to represent these information. Although this concept has no relation with fuzzy set, logic and reasoning, the compact swarm robotics leads to a like-fuzzy representation of the robot (agent) environment. As the solutions provided by the introduced compact swarm intelligence are not exact, but contrary more probabilistic and so then more flexible, the obtained intelligence is more malleable. In each time the information are sampled from their compact storage, the

adjustment mechanism will enhances and refines their representation. This motivates the robot (agent) at learning more, and at every time the given task is repeated. Moreover, the computational capacity reduction and the flexibility of the results of compact swarm intelligence are the most factors that have motivated us to propose the concepts of compact swarm robotics and its associated paradigms. Obviously, this approach is more benefic.

The compact swarm intelligence introduced here on cFO is very performant and it allows the reduction of the computer capacity required to find the optimal solution. Therefore, this have motivated us to introduce the compact swarm robotics based on the cFO. So, we propose the utilization of cFO in bipedal robotics. Making bipedal and humanoid robot at walking, is known to be one of the most difficult and challenging domains [22]. Despite this, optimal stable walking is realized using cFO.

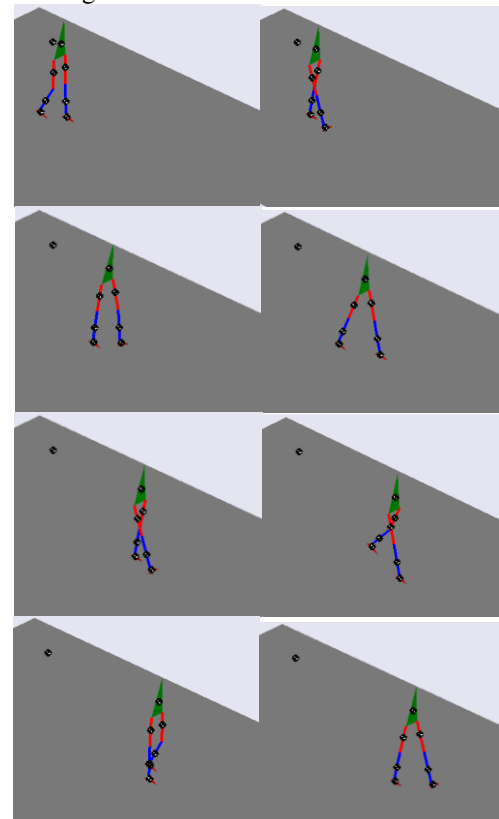


Fig. 2. Bipedal robot walking allure obtained with cFO

1) **Robot Model:** the bipedal robot in this experiment is the seven links bipedal robot described in [25]. In order to respect the maximum number of page for this conference. We present briefly the model. The seven link model is a planar bipedal robot with 9 degree of freedom. The detailed mathematical model, for kinematics, inverse kinematics, dynamics etc. can be found in the literature [25]

2) **Objective:** the objective is how to realize an optimal stable walking. The gait of each angle must follows an optimal path.

3) **Optimal path planning:** like in [25], each angle trajectory is defined using *spline* functions. To link the initial

position to the final position, we use three other intermediary points. The path linking the initial position of a given angle to its final position while passing through the intermediary via-points, is assumed to be a *spline path*.

4) *Results*: the parameters of the spline functions (coefficients) are calculated using the compact firefly optimizer. Optimization of the robot trajectory is also described in [25]. Each cubic spline function have 8 parameters and for each angle we have three point via for each step. Hence we have 448 parameters per steps. The results, walking allures, are presented in Fig.3. The model is created and simulated with *Sandscape/Matlab 2013*. The walking allure is presented in Fig.3. We not that we have not more space to provide supplementary results.

V. CONCLUSION

In this paper compact swarm intelligence (cSI) is introduced. A new intelligent compact swarm-based optimization algorithm is introduced. The proposed algorithm is called cFO for: *compact firefly optimizer*. In compact swarm intelligence only probability density function (PDF) are used to represent wide information, i.e population, swarm, etc., The proposed algorithm and intelligence allows the reduction of the memory device required by other evolutionary and swarm intelligent algorithms, hence compact swarm intelligence, i.g. like cFO, is very suitable for real world application. The proposed algorithm is tested under 12 mathematical benchmarks known to be difficult optimization problems. For the validation of cFO, we compare its performance to Firefly algorithm, compact differential evolution (cDE), differential evolution (DE) and PSO. The results obtained show that cFO surpasses the others in terms of convergence quality, accuracy, stability and speediness. The proposed algorithm is applied for bipedal walking. So, this study introduces the compact swarm robotics. The concepts and paradigms used, the precision of this investigation, the propositions given and analysis let the authors to think to a new important starting point for an emerging field of compact swarm intelligence, robotics, search and optimization. Before this, extension of this work to providing more tests and analysis must be done.

REFERENCES

- [1] H. Hamann; "State-of-the-Art in Modeling and Design of Swarms", Space-Time Continuous Models of Swarm Robotic Systems, pp. 27–39, Springer, 2010
- [2] Beni, G., Wang, J. Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30 (1989)
- [3] Z. Wang, H. Xing, T. Li, Y. Yang, R. Qu and Y. Pan, "A Modified Ant Colony Optimization Algorithm for Network Coding Resource Minimization", IEEE Transactions on Evolutionary Computation, vol. 20, no. 3, pp. 325–342, June 2016
- [4] S. Ullah, K. S. Yang, L. Wang and L. Liu, "A Modified Particle Swarm Optimization Algorithm for Global Optimizations of Inverse Problems", IEEE Transactions on Magnetics, Vol. 52, No. 3, March 2016
- [5] X. Ding, G. Liu, M. Du, H. Guo, C. Duan and H. Qian, "Efficiency Improvement of Overall PMSM-Inverter System Based on Artificial Bee Colony Algorithm under Full Power Range", IEEE Transactions on Magnetics, Vol. 52, No. 3, 2016
- [6] C. J. A. Bastos Filho; F. B. Lima Neto; M. F. C. Sousa; M. R. Pontes And S. S. Madeiro, "On the influence of the swimming operators in the Fish School Search algorithm", Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, Pages: 5012 - 5017, 2009,
- [7] Yang XS, "Nature-inspired metaheuristic algorithms", Luniver Press, UK, 2010.
- [8] S. Agarwal, A. P. Singh and N. Anand, "Evaluation performance study of Firefly algorithm, particle swarm optimization and artificial bee colony algorithm for non-linear mathematical optimization functions", Fourth Inter. Conf.e on Computing, Communications and Networking Technologies, pp. : 1 – 8, 2013,
- [9] I. Fister, I. Jr. Fister, X-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms. Swarm and Evolutionary Computation", 2013.
- [10] A. Nadeem, T. Shankar, R. K. Sharma and S. K. Roy, "An Application of Firefly Algorithm for Clustering in Wireless Sensor Networks", Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing, pp. 869-878, Springer 2016
- [11] T. S. Gorripotu, R. K. Sahu and S. Panda, "Application of Firefly Algorithm for AGC under Deregulated Power System", Computational Intelligence in Data Mining, pp 677-687, Springer 2014
- [12] M. Woźniak, M. Gabryel, R. K. Nowicki and Bartosz A. Nowak, "An Application of Firefly Algorithm to Position Traffic in NoSQL Database Systems", Knowledge, Information and Creativity Support Systems, Vol. 416 of the series Advances in Intelligent Systems and Computing pp 259-272, Springer 2016
- [13] Gerardo Beni, "From Swarm Intelligence to Swarm Robotics", LNCS in Swarm Robotics, Vol. 3342, pp 1-9, Springer 2005.
- [14] N. Rokbani, A. Casals and A. M. Alimi, "IK-FA, a New Heuristic Inverse Kinematics Solver Using Firefly Algorithm", Computational Intelligence Applications in Modeling and Control, Vol. 575, pp. 369-395, Springer 2014
- [15] C. Apomtewan and P. Chongstitvatana, "A hardware implementation of the compact genetic algorithm," in Proc. IEEE Congr. Evol. Comput., vol. 1, 2001, pp. 624–629.
- [16] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," IEEE Trans. Evol. Comput., vol. 3, no. 4, pp. 287–297, Nov. 1999.
- [17] D. Dumitrescu, B. Lazzerini, L. C. Jain, and A. Dumitrescu, "Elitism-based compact genetic algorithms" Evolutionary Computation. Boca Raton, FL: CRC Press, 2000.
- [18] Ernesto Miniño, Francesco Cupertino, and David Naso, "Real-Valued Compact Genetic Algorithms for Embedded Microcontroller Optimization" IEEE Trans. On Evo. Comp., Vol. 12, pp. 203-209, April 2008
- [19] E. Miniño, F. Neri, F. Cupertino and D. Naso, "Compact Differential Evolution" IEEE Trans. On Evo. Comp., Vol. 15, No. 1, pp. 32-54, Feb. 2011
- [20] X. Yao, Y. Liu, and G. M. Lin, "Evolutionary programming made faster," IEEE Trans. Evol. Comput., vol. 3, no. 2, pp. 82–102, Apr. 1999.
- [21] V. K. Koumoussis and C. P. Katsaras, "A sawtooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance," IEEE Trans. Evol. Comput., vol. 10, no. 1, pp. 19–28, Feb. 2006.
- [22] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," J. Global Optimiz., vol. 11, pp. 341–359, 1997
- [23] Zhi-Hui Zhan, Jun Zhang, Yun Li, and Yu-Hui Shi, "Orthogonal Learning Particle Swarm Optimization", IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 15, NO. 6, pp. 832-847, DECEMBER 2011
- [24] S. Wang, "Machine Learning Algorithms in Bipedal Robot Control", IEEE Trans. on Sys., Man, and Cyber. Part C, Vol. 42, Sep. 2012
- [25] M. Hobon, "Modélisation et optimisation de la marche d'un robot bipède avec genoux anthropomorphiques", PhD thesis, l'École Nationale Supérieure d'Arts et Métiers, French 2012.
- [26] P. Larrañaga and J. A. Lozano, "Estimation of distribution algorithms: A new tool for evolutionary computation", Dordrecht: Kluwer Academic Publishers, 2002.