International Journal of Advances in Computer Science and its Applications – IJCSIA
Volume 4 : Issue 1 [ISSN 2250 – 3765]
Publication Date : 09 Januarv 2014

VEDEPSO Algorithm for Designing Unique Library of DNA Strands

Krishna Veni Selvan, Mohd Saufee Muhammad and Sharifah Masniah Wan Masra
Department of Electronic Engineering
Faculty of Engineering
Universiti Malaysia Sarawak
94300 Kota Samarahan, Kuching, Sarawak, Malaysia

***Abstract*—Combinatorial optimization happens when there are two or more objectives to be optimized in solving a problem. The DNA words or oligonucleotides designing are one of a multi-objective combinatorial optimization problem. In this paper, the designations implied minimizations of H-measure, similarity, hairpin and continuity functions subjected to a predefine range of melting temperature and GC-content. A novel multi-swarm optimization approach is introduced to design a library of DNA strands. This approach is called vector evaluated differential evolution particle swarm optimization (VEDEPSO). The results obtained from the VEDEPSO algorithm is evaluated using Pareto dominance technique. A list of selected non-dominated solutions is shown as the final results from the research.**

Keywords-oligonucleotides; differential evolution; particle swarm optimization; Pareto

I. INTRODUCTION

DNA code words or oligonucleotides designing are a process of arranging the four DNA alphabets, *A*, *T*, *G* and *C* within a predefined length. These arrangements are then evaluated using some combinatorial constraints. The constraints used in this research are four objective functions, named H-measure, similarity, hairpin and continuity. Two other constraints, melting temperature and GC-content are applied to limit the chemical characteristics of the designed strands. Ensembles of DNA alphabets create a unique DNA library which is mainly used for molecular computing [1], DNA nanotechnology [2, 3], DNA tagging [4], DNA microarray [5], genetic engineering and other biotechnology applications.

Since previous researches [6, 7, 8, 9] that have been conducted using single swarm particle swarm optimization (PSO) had a major drawback, because single swarm does not have fair minimizations of all four objectives. Therefore in this paper, a multi-swarm optimizing technique with hybrid of differential evolution (DE) with PSO algorithm, named vector evaluated differential evolution particle swarm optimization is employed to designed better DNA libraries.

II. PROPOSED METHODOLOGY

In 2009, a novel optimizer known as vector evaluated differential evolution particle swarm optimization or VEDEPSO is proposed by Grobler and Engelbrecht [10]. It is a combination of vector evaluated particle swarm optimization (VEPSO) and vector evaluated differential evolution (VEDE) algorithms. As in [10] the performance of VEDEPSO algorithm is proven to be better than VEPSO and VEDE. The algorithm includes four swarms, S_1, S_2, S_3 and S_4 with each being assigned to minimize one objective function.

The minimizing assignments are: Swarm 1 (S_1) - H-measure fitness; Swarm 2 (S_2) – similarity fitness; Swarm 3 (S_3) – continuity fitness, and Swarm 4 (S_4) – hairpin fitness. Discrete search space is more suitable to be implemented into this research because DNA codes designing are a discrete problem. The fitness is calculated based on an average of each objectives minimized by its swarm. The formula and the parameters for each objective functions and the two constraints are referred to [11]. In each swarm 20 particles/individuals are randomly positioned. Each particle/individual has 280 binary data which is then encoded into seven DNA strings of 20-mer length each. In this algorithm, the binary representations of the four DNA alphabets are such as "00" for "*A*", "01" for "*C*", "10" for "*G*" and "11" for "*T*".

There is an information exchange among all four swarms, which is depicted in Fig. 1. Randomly, the best information, *best* particle/individual is been used by its own swarm and also been transferred to other swarms to ensure a balance simultaneous minimization between all the four swarms.

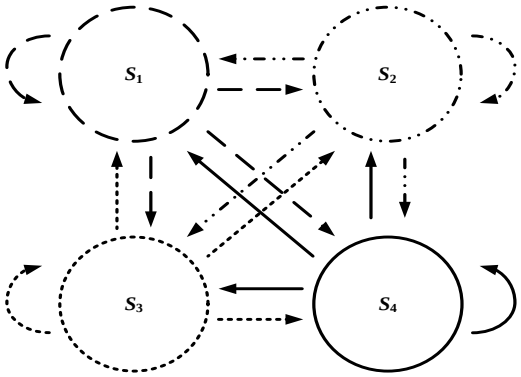

Figure 1. Information flow of*best* particle/individual in VEDEPSO.

In this approach, discrete PSO optimizer is used in the first and third swarms, S_1 and S_3 while discrete DE is applied in the second and fourth swarms, S_2 and S_4.

A. Discrete PSO

As in this VEDEPSO algorithm, the discrete PSO algorithm searches for particle with optimum fitness in S_1 and S_3. Discrete PSO is developed in 1997 by Dr. Kennedy and Dr. Eberhart [12]. The idea of PSO is derived based on bird flocking, fish schooling and swarming concepts. At each time, the lowest fitness value of each particle is memorized and noted as the *pbest* particle. As the particles move around the algorithm tracks the particle with an optimum fitness value and upgrade it as the *gbest* particle. The velocities and positions of each particle are formulated using (1) through (4).

$$\boldsymbol{V}_{ij}^{new} = \omega \boldsymbol{V}_{ij}^{old} + c_1 r_1 (\boldsymbol{pbest}_{ij} - \boldsymbol{X}_{ij}^{old}) + c_2 r_2 (\boldsymbol{gbest} - \boldsymbol{X}_{ij}^{old}) \quad (1)$$

$$\omega = \omega_{max} - ((\omega_{max} - \omega_{min}) / k_{max}) \times k \quad (2)$$

$$S(\boldsymbol{V}_{ij}^{new}) = 1/(1 + e^{-\boldsymbol{V}_{ij}^{new}}) \quad (3)$$

$$\boldsymbol{X}_{ij}^{new} = \begin{cases} 0, & \text{if } r_3 \geq S(\boldsymbol{V}_{ij}^{new}) \\ 1, & \text{if } r_3 < S(\boldsymbol{V}_{ij}^{new}) \end{cases} \quad (4)$$

$\boldsymbol{V}_{ij}^{new}$ and $\boldsymbol{X}_{ij}^{new}$ are the velocity and position of *j*th dimension of the *i*th particle respectively. r_1 and r_2 are random numbers between the interval [0, 1]. c_1, c_2 are the acceleration coefficients with a constant 0.5 value each. The inertia weight, ω decreases linearly between maximum inertia weight, ω_{max} and minimum inertia weight, ω_{min} using (2). ω_{max} and $\omega_{min}$are fixed with 0.9 and 0.4 values respectively. Current iteration and maximum number of iterations are represented by *k* andk_{max} respectively. *S*(.) is a sigmoid function and r_3 is a quasi random number uniformly distributed within the range of [0.0, 1.0]. The discrete search space implies a velocity limitations within a range of |-0.5, 0.5| for exploitations that directs the searching within feasible area.

B. Discrete DE

Swarms S_2 and S_4 utilize DE algorithm to search for optimum solutions. DE is a popular evolutionary algorithm technique developed by Storn and Price [13] in 1997. DE operates based on three main functions, mutation, crossover and selection. Discrete DE is proposed by Engelbrecht and Pampara [14] in 2007. *DE*/*best*/1/*bin* is selected and used in the VEDEPSO algorithm. Therefore the mutation and crossover function are calculated as in (5) through (7).

$$\boldsymbol{X}_{ij}^{mutant} = \begin{cases} \boldsymbol{gbest} + F(\boldsymbol{X}_{i1j} - \boldsymbol{X}_{i2j}), & \text{if } r_4 \leq CR \text{ or } j = j_{rand} \\ \boldsymbol{X}_{ij}^{old}, & \text{if } r_4 > CR \text{ or } j \neq j_{rand} \end{cases} \quad (5)$$

$$S(\boldsymbol{X}_{ij}^{mutant}) = 1/(1 + e^{-\boldsymbol{X}_{ij}^{mutant}}) \quad (6)$$

$$\boldsymbol{X}_{ij}^{new} = \begin{cases} 1, & \text{if } r_5 \geq S(\boldsymbol{X}_{ij}^{mutant}) \\ 0, & \text{if } r_5 < S(\boldsymbol{X}_{ij}^{mutant}) \end{cases} \quad (7)$$

where$\boldsymbol{X}_{ij}^{mutant}$ is the mutated individual of *j*th dimension of the *i*th individual. The mutations of each individual occur based on a condition that the parent individuals, $i \neq i1 \neq i2$. *F* is a scaling factor and *CR* is the crossover constant which are both fixed to 0.5 values. j_{rand} is a randomly chosen integer within [1, *D*] where *D* is the total number of dimensions in an individual. $\boldsymbol{X}_{ij}^{old}and\boldsymbol{X}_{ij}^{new}$are the previous mutated individual and the new offspring individual respectively of *j*th dimension of the *i*th individual. *S*(.) is a sigmoid function and r_4 and r_5 are quasi random numbers uniformly distributed within the range of [0.0, 1.0]. Fig. 2 explains pseudo code for the implementation of VEDEPSO for DNA words designing.

```
With each swarm, S1, S2, S3, S4
Randomly initialize velocities and positions of all particles and individuals in a swarm
    For swarm, S1, S3
        For every n particle, i1, i2, i3, i4, . . . ,in
            Calculate the fitness values
            If f (X_ij^new) <f (pbest_ij), Then pbest_ij = X_ij^new
        If f (pbest_ij) <f (gbest), Then gbest = pbest_ij
        If f (gbest) <f (best), Then best = gbest
        End For

        Update pbest_ij, gbest and best particles in an archive
        For every n particle, i1, i2, i3, i4, . . . ,in
Calculate particle's velocity using (1) and (2)
        Limit the velocity of particle by |-0.5, 0.5|
        Calculate particle's position according to (3) and (4)
        End For
    End For

    For swarm, S2, S4
        For every n individual, i1, i2, i3, i4, . . . ,in
Calculate the fitness values
        If f (X_ij^new) <f (X_ij^old), Then best individual = X_ij^new
        If f (best individual) <f (best), Then best = X_ij^new
        End For

Update best individual and best individuals in an archive
        For every n individual, i1, i2, i3, i4, . . . ,in
                If a random selection of i ≠ i1 ≠ i2 Then
                Calculate new offspring according to (5) through (7)
                End For
    End For
Until maximum number of iteration is achieved
Update non-dominated particles/individuals in an archive
```

Figure 2. Pseudo code of VEDEPSO for DNA words designing.

C. Pareto Dominance Concepts

In any multi-objective optimizations, a set of Pareto optimal solutions are produced based on Pareto dominance concept. This concept mentioned that a particle u_x dominated another particle v_x if and only if $f(u_x) \leq f(v_x)$ for all *x*-objectives or $f(u_x) < f(v_x)$ for at least one objective. Strictly, in the Pareto optimal solutions there should not have any other particle u_x' that has $f(u_x') \leq f(u_x)$ [15]. The non-dominated particles are selected as the solutions for the multi-swarm VEDEPSO algorithm.

III. Results and Discussions

The VEDEPSO algorithm runs for 10 times with each of 1000 iterations. The algorithm was developed using Microsoft Visual Basic 2008. Each designed DNA strands are limited within 30–80 °*C* and have 30–80 percents of GC-content. The parameters applied during the code words designing are shown in Table I. As a result, the algorithm attained an average fitness

svalue of 33.171 for the $S1$, 44.371 for the S_2 and 0.00 for both S_3 and S_4. The algorithm works very well to optimize each objective concurrently. Nevertheless, the Pareto distributions among swarms are not so balanced. Still major empty gaps occur in the fitness distribution. These indicate that the algorithm needs some improvement to minimize the distance among every particles' fitness. However, the information exchanging scheme of *best* particle and individual among all swarms for VEDEPSO works efficiently in directing every particle/individual towards global solutions.

TABLE I. PARAMETERS USED IN THE DNA CODE WORDS DESIGNING

	Number of runs	10
	Number of iterations	1000
DNA words	Number of sequences	7
	Length of each sequence	20-mer
	Range of *Tm*	30 – 80 ^oC
	Range of GC %	30 – 80 %
DE and PSO	Number of particles/individuals, i	20
	ω_{max}	0.9
	ω_{min}	0.4
	c_1	0.5
	c_2	0.5
	Random values: r_1, r_2, r_3, r_4, r_5	[0, 1]
	F	0.5
	CR	0.5
	V_{min}, V_{max}	[-0.5, 0.5]
T_m	Na^+	self complementary strands (1M); non- self complementary strands (1/20M)
	C_T	self complementary strands (10nM); non- self complementary strands (10/4nM)
H-measure, Similarity	H_{con}, S_{con}	6
	H_{dis}, S_{dis}	17%
Continuity	Threshold, t	2
Hairpin	Pair, p; Ring, r	6

Accurate parameter selections for c_1, c_2, CR and F also ensured quick redirections of new particles and individuals to better solutions in a swarm. Therefore it even reduces the computational time. The mutation and crossover operations implied also create diversity between every individual becausethe differences among all individuals are computed and minimized. The discrete search spaces allow the two searching agents, PSO and DE to have better explorations in the aim of obtaining their global solutions. Last but not least, 4 non-dominated particles and individuals are listed in Table II.

IV. CONCLUSIONS

Overall, the VEDEPSO algorithm successfully optimizes all objective functions simultaneously. Both PSO and DE own a simple and good optimizing mechanism, therefore hybridizing them together into an algorithm manage to minimize complex DNA words problem. Anyhow, the algorithm requires more improvement to reduce the fitness distribution among every swarm.

ACKNOWLEDGMENT

This research is supported financially by the Ministry of Higher Education (MOHE), Malaysia, under Fundamental Research Grant Scheme FRGS/02(11)/711/2009(27).

REFERENCES

[1] Adleman L M. Molecular computation of solutions to combinatorial problems. *Science*, 1994, 266(5187): 1021–1024.

[2] Seeman N C, Wang H, Yang X, Liu F, Mao C, Sun W, Wenzler L, Shen Z, Sha R, Yan H, Wong M H, Sa-Ardyen P, Liu B, Qiu H, Li X, Qi J, Du S M, Zhang Y, Mueller J E, Fu T J, Wang Y, and Chen J. New motifs in DNA nanotechnology.*Nanotechnology*, 1998, 9(3): 257–273.

[3] Winfree E, Liu F, Wenzler L A, and Seeman N C. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 1998, 394(6693): 539–544.

[4] Brenner S, and Lerner R A. Encoded combinatorial chemistry.*Proceedings of National Academy of Sciences USA*, 1992, 89(12): 5381–5383.

[5] Gerry N P, Witowski N E, Day J, Hammer R P, Barany G, and Barany F. Universal DNA microarray method for multiplex detection of low abundance point mutations. *Journal of Molecular Biology*, 1999, 292(2): 251–262.

[6] Khalid N K, Ibrahim Z, Kurniawan T B, Abidin M S Z, Khalid M, and Engelbrecht A P. DNA sequence optimization based on continuous particle swarm optimization for reliable DNA computing and DNA nanotechnology. *Journal of Computer Science*, 2008, 4(11): 942–950.

[7] Selvan K V, Muhammad M S, Masra S M W, Ibrahim Z, and Lim K S. DNA words based on an enhanced algorithm of multi-objective particle swarm optimization in a continuous search space. *2011 International Conference on Electrical, Control and Computer Engineering* (*INECCE*), 2011, pp. 154–159.

[8] Khalid N K, Ibrahim Z, Kurniawan T B, Khalid M, and Engelbrecht A P. Implementation of binary particle swarm optimization for DNA sequence design. *International Symposium on Distributed Computing and Artificial Intelligence* (DCAI'09), 10–12 June 2009, University of Salamanca, Spain.

[9] Muhammad M S, Selvan K V, Masra S M W, Ibrahim Z, and Abidin A F Z. An improved binary particle swarm optimization algorithm for DNA encodings enhancement. *2011 IEEE Symposium on Swarm Intelligence* (*SIS*), 2011, pp. 1–8, 2011.

[10] Grobler J, and EngelbrechtA P. Hybridizing PSO and DE for improved vector evaluated multi-objective optimization. *IEEE Congress on Evolutionary Computation*, 2009, pp. 1255–1262.

[11] Shin S Y, Lee I H, Kim D, and Zhang B T. Multiobjective evolutionary optimization of DNA sequences for reliable DNA computing. *IEEE Transactions on Evolutionary Computation*, 2005, 9(2): 143–158.

[12] Kennedy J, and Eberhart R C. A discrete binary version of the particle swarm algorithm. *IEEE International Conference on Systems, Man, and Cybernetics*, 1997, 5: 4104–4108.

[13] Storn R, and Price K. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, 11(1):341–359.

[14] Engelbrecht A P, and Pampara G. Binary differential evolution strategies. *IEEE Congress on Evolutionary Computation*, 2007, pp. 1942–1947.

[15] Lim K S, Buyamin S, and Ibrahim Z. Convergence and diversity measurement for vector evaluated particle swarm optimization based on

ZDT test problems. In *2011 Fifth Asia Modelling Symposium* (*AMS*), 2011, pp. 32–36.

TABLE II. DNA LIBRARY OF FOUR NON-DOMINATED PARTICLES/INDIVIDUALS OBTAINED FROMVEDEPSO ALGORITHM

Swarm	Particle/Individual	DNA Code Words	Melting Temperature	GC-content	H-measure	Similarity	Continuity	Hairpin
1	20	CTCTCTCTATGCTATGCTCT	38.40	45	27	184	0	0
		CTCTATGCTATGCTCTCTCT	41.01	45	27	192	0	0
		ATGCTATGCTCTCTCTATGC	40.85	45	32	184	0	0
		TATGCTCTCTCTATGCTATG	37.74	40	32	176	0	0
		CTCTCTCTATGCTATGCTCT	38.40	45	27	184	0	0
		CTCTATGCTATGCTCTCTCT	42.58	45	27	192	0	0
		ATGCTATGCTCTCTCTATGC	32.03	45	32	184	0	0
2	1	GTATACGGTAGACGAGCCCT	43.35	55	53	42	9	0
		GTCAGTCACAAGGAGAGGTT	52.67	50	45	44	0	0
		ACCTAAGTTGCGACTGTGGA	42.84	50	45	40	0	4
		TGCAAACGATTCGTAAAGGA	38.11	40	42	53	25	0
		CAATGGTCACCAGGACTGAA	45.01	50	45	47	18	0
		TGAGAAGAATTCACCGAATT	41.45	35	44	44	9	0
		CGGATGCACACGGCTCGTAG	41.52	65	54	48	0	0
3	15	TACTACTGATGCCTATTACT	35.00	35	55	135	0	4
		ACTGATTCCTATTACTACTG	36.90	35	58	144	0	0
		ATTCCTATTACTACTGATTC	33.76	30	52	154	0	0
		CTATTACTACTGATTACTGA	35.16	30	52	155	0	0
		TTACGATTACTACTGATCAC	38.55	35	54	150	0	0
		GATTACTACTGATCACGGTT	38.62	40	55	150	0	0
		ACTACTGATTACGGTTACTA	29.93	35	52	134	0	0
4	8	AGCAGATGAAACCTACCGTT	42.17	45	47	95	9	0
		GACAAGAACAGATAGTATAC	49.31	35	41	83	0	0
		AGTGGACAAGACCTGCCCGG	49.67	65	47	88	9	0
		ACCAAGCAGATGAAACCTAC	40.52	45	46	103	9	0
		CGTTGACAAGAACAGATAGT	33.91	40	41	85	0	0
		ATACAGTGGACAAACCGTGC	39.52	50	45	74	16	0
		CGGTACCAAGCAGATGAATC	34.60	50	39	88	9	0