International Journal of Current Engineering and Technology
ISSN 2277 - 4106

Available at http://inpressco.com/category/ijcet

Research Article

Performance Improvement of Differential Evolutionary Algorithm: A Survey

Vanita G. Tonge[a*] and P.S.Kulkarni[a]

[a]Department of Information Technology, RCERT, Chandrpur

Accepted 4 Nov.2012, Available online 27Dec. 2012, **Vol.2, No.4(Dec. 2012)**

Abstract

Differential evolution is one of the latest evolutionary optimization algorithm applied to continuous optimization problems. Differential evolution for solving flow shop problem that belongs to the class of scheduling problems. The scheduling problems arise in diverse areas such as manufacturing systems, production planning, computer design, logistics etc.. Only in very special cases there exist exact polynomial algorithms to reach optimal solution. To solve PFSP problem many researchers have used different EVs algorithm such as ACO, PSO, GA. Out of such evolutionary algorithm DE is the best solution.

***Keywords*:** Differential Evolution, recombination, PSO, ACO, perturbation, PFSP.

1. Introduction

Evolutionary Computation (EC) uses ideas inspired from the behavior of community-based animals such as ants, bees and birds. Furthermore researchers proposed algorithms with variants of DE. Differential Evolution (DE) algorithm has emerged as a very competitive form of evolutionary computing more than a decade ago.

First a donor vector ($V_{i,t}$) is created in step a and then a trial vector ($U_{i,t}$) is created in step b by stochastically combining elements from $X_{i,t}$ and $V_{i,t}$. This combination is commonly done using an exponential distribution with crossover factor of CR. If the newly created child $U_{i,t}$ is better compared to $X_{i,t}$ then $U_{i,t}$ is stored for updating $X_{i,t+1}$. It should be noted carefully that $X_{i,ts}$ are updated to $X_{i,t+1s}$ after entire set of $U_{i,ts}$ are created. Once the population is updated, the generation counter is incremented and a termination criterion is checked.

Following properties of this DE should be noted: (i) There is 'elitism' at an individual level i.e. if the newly created trial vector Ui,t is inferior compared to the individual then individual is preserved as a child for the next generation and Vi,t is ignored. (ii) The algorithm follows a generational model i.e. the current population is updated only after the entire offspring population is created. One of the noticeable features of standard DE is elitism at the individual level i.e. a child is compared with its base parent (i.e. the individual at the index corresponding to which child has been created), and only the better of the two survives for the next generation. We modified this Replacement scheme by always accepting the newly created child i.e. without carrying out the parent-child comparison. This resulted in significant performance degradation in all three test problems with respect to both the metrics, indicating that elitism in DE by parent-child comparison is key to its performance. (Nikhil et al,2010)

DE has several advantages: it can search randomly, requires only fewer parameters setting, high performance and applicable to high-dimensional complex optimization problems. But similar to PSO, DE has several drawbacks including unstable convergence in the last period and easy to drop into regional optimum. Compared with else evolutionary computation, DE and PSO have advantage respectively (Ying-Chih et al,2009).

Performance of DE is based on its control parameters. Researchers have provided the various techniques to improve the performance of DE. We will see it in next section. The remaining paper is organized as follows. Section 2 introduces the differential evolution (DE) algorithm. Section 3 present PFSP problems, Section 4 gives brief review on performance improvement of DE, proposed plan is discussed in Section 5.Finally, Section 6 summarizes the concluding remark.

2. Differential Evolution (DE)

Differential Evolution is announced in 1995 by Price and Storn, and its superior performance in solving complex problems.(D. G. Mayer et al,2005).DE is based on individual's difference, utilize random research in solution space, further utilize the mechanism "mutation", "recombination", "selection" to compute every individuals to obtain appropriate individual. DE used the information between the difference in individuals to lead to search, thus the result of search is more unstable(Z.-F. Hao et, 2007).The advantage and weakness of DE is given as follows.

* Vanita G. Tonge is M.Tech Scholar and Prof.P.S.Kulkarni is the Project Guide.

Table 1. Advantages and weakness of DE

Advantages	Weakness
Keep the multiplicity of population	The convergence is unstable
Enhance the capacity of local search	Easy to drop into the pbest

Of course DE has some limitations, which we attempt to address in this paper. It has been reported that DE performs poorly on problems that are not linearly separable because of inefficient exploitation during the differential mutation phase (D. G. Mayer et al, 2005).

Two hypotheses were explored by Sutton; when the crossover rate (*Cr*) is low, DE can exploit the separability of a function. When DE has a *Cr* of 1.0, DE becomes rotationally invariant and depends entirely on the differential mutation step. In order to efficiently solve non-separable problems, DE typically must depend more on mutation than crossover, although DE lacks selection pressure in the differential mutation step to make efficient progress. Furthermore, making *Cr* equal to 1.0 is not recommended as it reduces the number of trial vectors and can result in stagnation(Luis et al,2005). Finally, DE becomes highly dependent on population size in order to avoid stagnation when no crossover is employed.

DE is similar to Genetic Algorithm, and the main procedure is discussed in the following(Luis et al,2005).

Initialization : Setup the parameters and initialize the Target population.

$x^{o}_{ij} = x_{min} + (x_{max} - x_{min}) * r1$ (1)

where $x_{min} = -1.0$, $x_{max} = 1.0$ and r1 is a uniform random number between 0 and 1.To initialize the target population The random method used by considering boundary constraints.

Mutation : The common strategies of DE are via the formula of mutation vector to produce change.

They are such as the following table2. The strategy symbol is DE/x/y. DE stands for differential evolution algorithm, x represents a string denoting the vector to be perturbed, y is the number of difference vectors considered for perturbation of x, and z is the type of crossover being used (exp: exponential; bin: binomial)

Recombination : The donor vector will change the information with the target vector randomly. A new vector "trial vector" called $u_{i,G+1}$ will be generated after recombination. Use the following formulation to decide in the iteration *j* the component *i* compose from target vector *xi* or donor vector v_i.

$$u_{j,i,G+1} = \begin{cases} v_{ji,G+1} & \text{if rand} \le CR \\ x_{ji,G} & \text{if rand} > CR \end{cases} \quad (6)$$

Selection : After the above mechanism, compare with Trial Vector and Target Vector to select which vector can be reserved into the next generation.

Table 2. Strategies of DE

Symbol	Formula	Description
DE/rand/1	$Vi,G+1 = Xr1,G + F(Xr2,G - Xr3,G)$ (2)	For extending the search space, random selected three vectors *Xr1,G, Xr2,G, and Xr3,G* using equation to obtain Donor Vector as *Vi,G+1*
DE/best/1	$Vi,G+1 = Xbest,G + F(Xr2,G - Xr3,G)$ (3)	Based on the DE/rand/1, this strategy use the present best particle *Xbest,G* to replace *Xr1,G*.
DE/rand/2/bin	$x_{r5}^{(G)} + F . (x_{r1}^{(G)} + x_{r2}^{(G)} - x_{r3}^{(G)} - x_{r4}^{(G)})$(4)	five distinct vectors other than the target vector are chosen randomly from the current population. Out of these, the weighted vector difference of each pair of any four vectors is added to the fifth one for perturbation
DE/rand/1/exp	$v = x_{r1}^{(G)} + F . (x_{r2}^{(G)} - x_{r3}^{(G)})$(5)	In exponential crossover, the crossover is performed on the D (the dimension or number of variables to be optimized) variables in one loop until it is within the CR bound

$$X_{i,G+1} = \begin{cases} u_{i,G+1} & \text{if } F(u_{i,G+1}) \le F(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \quad (7)$$

The following steps are involved:

Step 1 : Initialization.

Step 2 : Apply formula (1) to select initial Target Vector $X_{i,G}$ randomly.

Step 3 : Evaluate fitness.

Step 4 : Mutation: Selected three vectors according to formula (2) or (3) or (5) to mutate to generate Donor Vector.
Step 5: Recombination: Change information between the Target vector and the Donor Vector by formula (6) and obtain the Trial Vector.
Step 6 : Compare with Target Vector and Trial Vector selected by fitness to determine which can be reserve into next generation.
Step 7 : If the termination condition is failed to reach, go back to step 2, or output the optimal solution.

Fig. 1: The procedure of Differential Evolution.

A. *Why DE?*
To solve permutation flowshop scheduling problem researches have proposed various evolutionary algorithms like PSO, ACO, GA. But according to the survey DE gives best result for numerical problem. Following table shows comparative study among evolutionary algorithm.

- Advantages of DE
  - Simple concept
  - Immediately accessible for practical applications
  - Simple structure
  - Ease of use
  - Speed to get solutions
  - Robustness
- DE technique has better convergence properties than all the GAs.
- DE technique could locate the current best solution with a higher frequency than the GAs in all cases.
- DE reproduces the same results over many trials, whereas the GA performance is more dependent on the randomized initialization of the individual parameters.
- DE is better than PSO in terms of solution quality, running time, and chance of reaching the best solution in a variety of problems.

Table 3. Comparison Between Evolutionary Algorithm.

Principle	Crossover	Velocity and position Adjustment	Ferromom Deposition	Difference of neighboring vector search
Searching capability	Global search is involved	Local & global search involved	Global search is involved	Global search is involved
Mechanism	Probablistic approach	Population based search	Probablistic approach	Greedy search with multiple vectors
Type of prob continuous/ discontinuous	Suitable for all types of problem	Suitable for all types of problem	Suitable for all types of problem	Suitable for all types of problem
Complexity	Large computation are involved	Computation are simpe &easy	Large computation are involved	Computation are simpe &easy

3.The permutaion Flowshop Scheduling Problem

A Permutation Flowshop Scheduling belongs to the class of scheduling problems. The scheduling problems arise in diverse areas such as manufacturing systems, production planning, computer design, logistics etc.. is a production planing process consisting of a set J = {J1, J2... Jn} of n jobs to be executed in a set M = {M1, M2... Mm} of m machines. In this process every job Jj is composed by m stages $O1_{,j}$,$O2_{,j}$, ...,$Om_{,j}$ named operations. Every operation $O_{i,j}$ has a non-negative processing time t_{ij}. The Job operation $O_{i,j}$ must be only executed on machine i. A machine cannot execute more than one operation per time. Operation $O_{i,j}$ can be executed only after operation $O_{i-1,j}$ have _finished. Preemption is not allowed, i.e. once an operation is started, and it must be completed without interruption. All jobs must be executed in the same order by every machine, depend by a permutation. The completion time of an operation $O_{i,j}$, denoted by $C_{i,j}$ is defined by the recurrence:

$$C_{i,\pi(j)} = \begin{cases} t_{i\pi(j)} & if\ i = 1\ and\ j = 1 \\ C_{i,\pi(j-1)} + t_{i,\pi(j)} & if\ i = 1\ and\ j > 1 \\ C_{i-1,,\pi(j)} + t_{i,\pi(j)} & if\ i > 1\ and\ j = 1 \\ \max(C_{i,,\pi(j-1)}, C_{i-1,,\pi(j)} + t_{i,\pi(j)} & if\ i > 1\ and\ j > 1 \end{cases}$$

Notationally, the problem is referenced by F/permu/Cmax. The completion time of a job Jj is $C_{m,j}$. The makespan of a permutation is the maximum completion time of a job. The objective of Permutation Flowshop Scheduling Problem (PFS) is to find a permutation that minimizes the makespan. It is well-known that, if m = 2, the classical permutation flow-shop problem can be solved in O(n log n) time by the algorithm of Johnson (S.M. Johnson et al,1954), while the three-machine classical permutation flow-shop problem has been shown to be strongly NP-hard. An example of a permutation flowshop problem schedule is shown in the below figure:

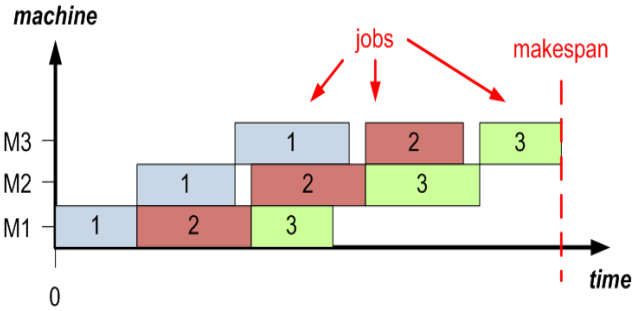

Fig. 2: Permutation Flowshop Scheduling Problem .

4. Literature Survey

Modified the Performance of Differential Evolution Algorithm with Dual Evolution Strategy

Ying-Chih Wu 1, Wei-Ping Lee 1, Ching-Wei Chien,proposed a proposed the structure of dual evolution to improve differential evolution, approach to sharing information increasing performance. During dual evolution, the performance of two algorithms against different functions needs to be compared and then determine which first carried out in accordance with. According to master-apprentice mechanism, selected better algorithm presently to share information to another algorithm, moreover confirm the accuracy of solving problems and increasing the convergence performance. Accordingly, information exchange enhances exploration and maintains diversity of the algorithm. Moreover, it decreases the probability of dropping into regional optimum (K. Price et al,1999).

Improving the Performance and Scalability of Differential Evolution

Antony W. Iorio and Xiaodong Li have addressed the stagnation issue of DE. in order to overcome stagnation in DE a very large population size had to be employed, or crossover was used in order to add more sampling diversity even though crossover is typically ineffective when optimization problems have many parameter interactions. Furthermore, rotationally invariant DE applied to non-separable problems is limited to rather low decision space dimensions and is highly dependent on population size (Antony et al, 2008)

5. Proposed Algorithm

Performance of DE is based on its control parameter and to get a better solution we have to experiment combination of CR and F factor.to overcome this drawback we proposed modified mutation factor for dynamically initialization of F (Ching-Wei et al,2011). Mutation factor (F) is a decisive factor to decide the ability of exploration (Global search) and exploitation (local search). Early studies main though experience or trial and error to find suitable value. In earliest version, these values are (0, 2] and . But the original authors found that the value of F in the range of 0.4 to 1 was an interval that could be considered, and recommend 0.5 as a good initial value. Scholars have suggested that the scope of F value should be set to [0.4, 0.8], and it is still suitable for use with some functions of the settings, rather than the common optimal value . The following is procedure of DE with modified mutation factor.

Step 1 : Initialization. Include the total number of solution vector; Maximum of generation; Mutation Factor; Crossover rate; define the solution space between the lower bound and upper bound. Initialize of each variable in its own range.

Step 2 : Generated initial solution vector randomly.

Step 3 : Evaluate the fitness value of the solution vector. If the fitness value doesn't improve, import deceleration factor into mutation. On the other hand, import acceleration factor upon improving the fitness value.

Step 4 : Mutation. Select several variable vectors and acquire their difference and multiply the *F* value from the *F* value function. Produce donor vector at the end.

Step 5 : Recombination. Swap donor vector and target vector by *CR* to produce the trial vector.

Step 6 : Selection. Compare target vector and trial vector to determine the one can be reserved.

Step 7 : if the termination conditions are unsatisfactory; go back to Step 2, or output the optimal solution

Conclusions

The permutation flowshop represents a particular case of the flowshop scheduling problem, having as goal the deployment of an optimal schedule for N jobs on M machines.DE is the best solution among the various evolutionary algorithm it gives better results. Many experiences in using different trial vector generation strategies and the DE control parameter settings have been studied. A comprehensive use of these experiences should be an effective way for improving the DE performance. The proposed dynamic generate function is a mechanism that dynamical improve the mutation factor during the generation to promote overall performance.

References

Nikhil Padhye, Piyush Bhardawaj and Kalyanmoy Deb(2010),Unifying Evolutionary Algorithms and Improving Differential Evolution, Proceedings of Simulated Evolution And Learning, *Springer LNCS*

Z.-F. Hao, G.-H. Guo, and H. Huang,(2007),A Particle Swarm Optimization Algorithm with Differential Evolution,*IEEE International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 1031-1035.

D. G. Mayer, and A. A. Archer,(2005),Differential evolution – an easy and efficient evolutionary algorithm for model optimisation, *Agricultural Systems*, vol. 83, pp. 315–328.

Luis Vicente Santana-Quintero and Carlos A. Coello Coello,(2005),An Algorithm Based on Differential Evolution for Multi-Objective Problems, *International Journal of Computational Intelligence Research*. ISSN 0973-1873 Vol.1, No.2), pp. 151–169

A. K. Qin, P. N. Suganthan,(2005), Self-adaptive Differential Evolution Algorithm for Numerical Optimization, 0-7803-9363-5/05/$20.00 ©2005 *IEEE*.

David Sotelo P. da Silva and Marcus V. S. Poggi de Aragão(2007), An Approximation Algorithm for Permutation Flowshop Scheduling Problem via Erdös-Szekeres Theorem Extensions, *Monogra_as em Ciência da Computação*, No. 28/07 ISSN: 0103-9741Editor: Prof. Carlos José Pereira de Lucena Dezembro.

Ying-Chih Wu 1, Wei-Ping Lee 1, Ching-Wei Chien 1(2011), Modified the Performance of Differential Evolution Algorithm with Dual Evolution Strategy, *2009 International Conference on Machine Learning and Computing IPCSIT vol.3 © (2011) IACSIT Press, Singapore*.

Antony W. Iorio and Xiaodong Li(2008),Improving the Performance and Scalability of Differential Evolution, X. Li et al. (Eds.): SEAL 2008, LNCS 5361, pp. 131–140, *Springer-Verlag Berlin Heidelberg*

Sutton, A.M., Lunacek, M., Whitley, L.D.(2007), Differential evolution and nonseparability: using selective pressure to focus search. In: GECCO: Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp.1428–1435

Lampinen, J., Zelinka, I(2000),On stagnation of the differential evolution algorithm. In: Proceedings of MENDEL, 6th International Mendel Conference on Soft Computing, pp. 76–83

Ching-Wei Chien 1, Zhan-Rong Hsu 1, Wei-Ping Lee 1(2011), Improving the Performance of Differential Evolution Algorithm with Modified Mutation Factor, *2009 International Conference on Machine Learning and Computing IPCSIT vol.3 © (2011) IACSIT Press, Singapore*.

M. Rickert, O. Brock and A. Knoll(2008),Balancing exploration and exploitation in motion planning, Robotics andAutomation, . ICRA 2008. *IEEE International Conference on*, pp. 2812-2817

Storn, R., Price, K.(1997),Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* vol. 11, pp. 41–359

K. Price(1999),An introduction to differential evolution, in: D.Corne, M. Dorigo, F. Glover (Eds.), *New Ideas in Optimization, McGraw-Hill, London*, pp. 79–108

M.R. Garey, D.S. Johnson and R. Sethi, (1997)The complexity of flow-shop and job-shop scheduling, Math. *Oper. Res*. 1 (1976) 117–129.

S.M. Johnson,(1954) Optimal two-and three-stage production schedules with setup times included, *Naval Res. Logist. Quart*. 1 ,61–68.

Xiao-lei DONG†, Sui-qing LIU, Tao TAO, Shu-ping LI, Kun-lun XIN,(2012), A comparative study of differential evolution and genetic algorithms for optimizing the design of water distribution systems,*Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering) ISSN 1673-565X (Print); ISSN 1862-1775 (Online), Dong et al. / J Zhejiang Univ-Sci A (Appl Phys & Eng) 13(9):674-68*

VU Truong (2012,)A Comparison of Partical Swam Optimization and Differential Evolutionary Algorithm,International, *Journal on Soft Computing* ,Vol. 03 No.