Empirical Analysis on the Population Diversity of the Sub-population in Distributed Differential Evolution Algorithm

Ramya Raghu* and G Jeyakumar**

Abstract: The Distributed Differential Evolution (dDE) algorithm is a natural extension of the Differential Evolution (DE) algorithm, which is a recent addition to the Evolutionary Algorithms (EAs) pool, in the Evolutionary Computing (EC) field of computer science. The algorithmic novelty of the dDE algorithm is well evident in the literature. However, the theoretical studies on the performance of the dDE algorithms are scarcely reported. This paper is an attempt to analyze the performance of the dDE algorithm with a theoretical study. A theoretical equation, to measure the population diversity of the sub-population of the dDE algorithm, after migration, is derived and the validity of the same is verified with a simple distributed framework of dDE with two sub-population.

Index Terms: Distributed Differential Evolution, Population Diversity, Theoretical Study of dDE, Population Variance, Distributed Framework.

1. INTRODUCTION

There are several optimization tasks around us. In order to solve these optimization tasks we have optimization techniques. One among these optimization techniques is Evolutionary Computation. The algorithms used in Evolutionary Computation are grouped and termed as Evolutionary Algorithms (*EAs*). The *EAs* relate the biological processes and optimization. There are many instances of Evolutionary Algorithms which include Evolutionary Strategies (*ES*), Evolutionary Programming (*EP*), Genetic Algorithm (*GA*) and Genetic Programming (*GP*).

Differential evolution, introduced by Storn and Price(1995,1997), is an optimization technique which can handle non-linear, non-differentiable and multimodal objective functions. *DE* inculcates the policy of survival of the fittest. It is a simple, robust and a powerful population based search algorithm. DE is said to have good convergence properties and it outperforms any other Evolutionary Algorithms.

In distributed differential evolution (*dDE*), the entire population is divided into *n* subpopulation and each of this subpopulation evolves independently. The subpopulation are interconnected by a topology. At every particular interval, known as the migration interval, each subpopulation sends their best candidate to the neighboring node which results in a cooperative co-evolution. Such continuous cooperative co-evolution leads to the success of distributed Differential Evolution. The novelty of distributed Differential Evolution lies in the exchange of information from one node to the other. This periodic exchange of information is termed as the process of migration. The process of migration is dependent on many factors which include the interval at which one node sends the information to the other (*migration interval*), number of migrants that are transferred from one node to the other (*migration rate*), the topology according to which the nodes are inter connected, number of islands in which each subpopulation is let to evolve independently and the

---
*   Dept of Computer Science and Engineering Amrita School of Engineering, Coimbatore Amrita Vishwa Vidyapeetham Amrita University
       India, *Emails: *ramyaraghu92@gmail.com, **g_jeyakumar@cb.amrita.edu*

migration policy. Migration policy includes the method of selecting the individual that is to be migrated to the other (selection policy) in order to replace another individual (replacement policy).

The impact of migration on the diversity of the population can be explained with the parameters of migration. As migration interval decreases, the population may be prone to genetic drift and finally loses the diversity very fast. If the migrations are rare, it leads to slow convergence as it gets stuck in the local minima, causing the algorithm to stop prematurely. The best performance is achieved with moderate migration intervals and small migration rates.

With migration, the exploration ability of *DE* increases as foreign candidates add diversity to the population. The population diversity should neither increase nor decrease too fast. In order to ensure that the algorithm is proceeding in the correct path, it is essential to maintain the diversity. If there is no balance between the exploration (mutation and crossover) and exploitation (selection) processes, the Evolutionary Algorithms lose their search capability, leading to premature convergence or stagnation (Zaharie, 2003, Zaharie and Zamfirache, 2006, Angela et al., 2008, Zaharie, 2001b). Measuring the population variance is a standard method to measure the population diversity.

The aim of this paper is to derive an equation to measure the changes in the population diversity of the sub-populations of the *dDE* algorithm and to verify the correctness of the equation. The experiments conducted and the results obtained are presented in this paper.

The remaining part of the paper is organized as: section II deals with related works, section III details the experiments conducted and their results and finally section IV concludes the paper.

2.   RELATED WORKS

In the literature of Evolutionary Algorithms (*EAs*), the theoretical studies are very limited compared to the empirical studies. This is true for *DE* algorithm also. However, the working behavior of any algorithm can be understood and further improved with suitable theoretical studies made on the algorithm. Few research attempts made in *DE* algorithm with theoretical foreground is presented below.

The evolutionary search dynamics of *DE* was analyzed theoretically and reported in (Dasgupta et al, 2008, 2009). Interestingly the runtime complexity analysis of *DE* algorithm with different stopping criteria was done by Zielinski et al, theoretically and empirically. The results are reported in (Zielinski et al., 2005). A mathematical modeling to convergence analysis of *DE* (but for continuous multi-objective optimization) was presented in (Xue et al., 2005a). This work was later extended for discrete case and reported in (Xue et al., 2005b).

Co-operative co-evolution was first put forward in (Potter and DeJong, 1994), for genetic algorithm (CCGA). This concept had been widely applied to all the EAs (Liu et al., 2001, Sofge et al., 2002, Bergh and Engelbrecht, 2004). It follows the Divide-and-Conquer strategy. The co-operative co-evolution architecture included three components viz. Problem decomposition, Subcomponent optimization and Subcomponent co-adaptation. The efficiency of one among the parallel models, i.e, island based distributed model, with cooperative coevolution concept was studied in (Skolicki and Jong, 2004, 2005) and (Cantu-Paz, 2001). The distributed models of *EAs* follow the exchange process called migration, to send and receive candidates among the populations. The migration process itself involves many parameters (Cantu-Paz, 2001, Skolicki and Jong, 2004).

In 2004, D.K Tasoulis et.al studied on how to parallelize Differential Evolution using ring topology in order to improve the speed and performance of DE. The different migration policies was experimented on ring topology and the performance was studied in (Jeyakumar and ShunmugaVelayutham, 2010b). The best and worst variants of DE were heterogeneously mixed together in different distributed frameworks (dDE framework) in (Jeyakumar and ShunmugaVelayutham, 2015) and the best variants were found to outperform the worst.

Zaharie (Zaharie, 2001a) made a significant contribution to the theoretical analysis of *DE* algorithm, by deriving an expression to measure the population diversity of the *DE* population, with the know control parameter values mutation rate (*F*) and cross over rate (P_c). This work was later extended in (Zaharie, 2002a) and (Zaharie 2002b). Zaharie also reported the impact of population variance in *DE* search for different mutation and crossover operators in (Zaharie, 2008). A theoretical insight about the influence of crossover operators of *DE* was provided in (Zaharie, 2009). By extending the theoretical work done by Zaharie (zaharie, 2001) for *DE/rand/1/bin* variant, in (Jeyakumar and ShunmugaVelayutham, 2010a) and (Thangavelu et al, 2015) the expressions to measure the expected population variance of few other variants of *DE* are presented.

But the idea of deriving a theoretical expression to measure the population variance at each generation of *dDE* is a new research direction. It is worth noting that this paper is an initial attempt in this direction of research. The authors derived the expression by considering the migration process of *dDE* and its validity is verified with suitable experiments.

3.　RESULTS AND DISCUSSION

The equation derived by Zaharie (Zaharie, 2001a), to measure the population variance of classical *DE* algorithm is not applicable to measure the population variance of the *dDE* algorithm, as this equation calculates the current population variance from the initial population variance of the population. The initial population variance of the classical *DE* algorithm is fixed. But in *dDE* algorithm, since the population variance of the sub-population changes after every migration between them, the initial population variance is not fixed. Hence, the equation derived in (zaharie, 2001a) is not applicable for *dDE*.

Thus the authors are proposing to derive a new equation to find out the population variance of each of the subpopulation taking into account the factor of migration. In the migration process of distributed Differential Evolution, the strategy chosen by the authors, the best candidate of each of the sub-population replaces a random candidate in the other sub-population. Consider a population of size 100 candidates is split into two subpopulations of size 50 candidates each. Thus there are two nodes S_1 and S_2, connected using ring topology, running each subpopulation. After a particular migration interval, each node transfers the best candidate to the next node and replaces a random candidate by the received candidate. This operational scheme of *dDE* is depicted in Figure 1.

Thus the migration process brings a change in the population diversity of each of the sub-population. The authors formulated an equation to measure the population diversity of the sub-population after the migration. The derived equation is shown in Equ (1).
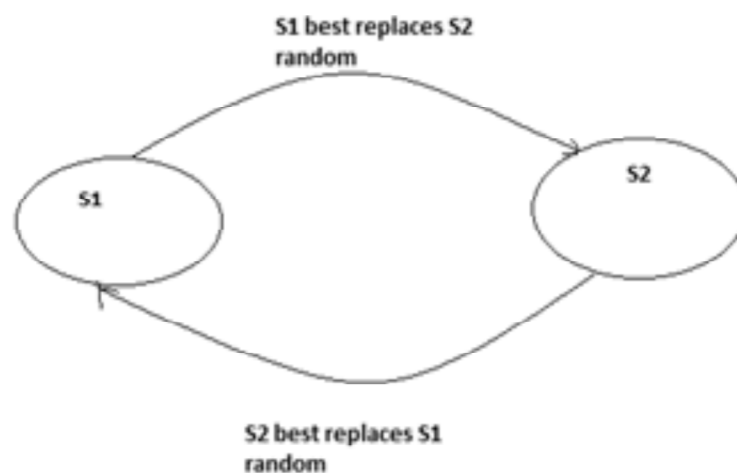

Figure1: Distributed Differential Evolution using two subpopulations

$$\sigma^2_{new} = \sigma^2_{old} + \frac{(X_{new} - \mu_{old})^2}{n-1} - \frac{(X_{old} - \mu_{old})^2}{n-1} - \frac{(X_{new} - X_{old})^2}{n(n-1)} \tag{1}$$

Where σ^2_{new} *is the new variance* (*after migration*), σ^2_{old} *is the old variance* (*before migration*), X_{new} *is the new* (*best*) *candidate*, X_{old} *is the old* (*random*) *candidate*, μ_{old} is the old mean, *n* is the total number of candidates in the population.

Two approaches are followed to verify the validity of the derived equation

(1) Sequential Approach: In a single population, at a certain interval, the best candidate is replaced by newly created random candidate. This experiment was conducted to analyze the validity of the derived equation (Equ (1)) in classical DE algorithm itself.

(2) Simulated Distributed Approach: The initial population is divided into two subpopulation of equal size, and they are allowed to evolve one by one in two different procedures. They also migrate at each generation. Since this approach simulates the distributed *DE* (*dDE*) where the subpopulation are evolved by independent processes, it verifies the validity of the equation in simulated distribute *DE* (*dDE*).

3.1. Sequential Approach

This experimental setup used a population of size 15 and dimension 4, *DE* variant *DE/rand/1/bin*, and the benchmark function '*Sphere Model*'. A random candidate of dimension 4 is generated and used to replace the best candidate of the population. The variance was calculated component wise and the average of all the variances thus obtained was observed as the total variance. This was made to repeat for 3000 generations. The algorithmic structure of this sequential approach is shown in Figure 2.

The statistical method used in this algorithm (Figure 2) for measuring the population variance uses the logic of finding the variance of the components of the population using the equation Equ (2) and finding their average as the population variance.

$$\text{var} = \frac{1}{n} \Sigma^n_{i=0} (X[1]_i - \mu)^2 \tag{2}$$

where $X[1]_i$ is the component [1] of the population

```
Algorithm:
    1.  Start
    2.  Initialize the population POP[NP][D],
        a random vector REPLACE[1][D],
        and set NP = 15, D = 4, Max_Gen = 3000.
    3.  Perform Mutation
    4.  Perform Crossover
    5.  Select the best candidate
    6.  Replace the best candidate by REPLACE[1][D]
    7.  Find the population variance using both the
        statistical method and by using the equation (Equ 1)
    8.  Continue until stopping criterion is met
```
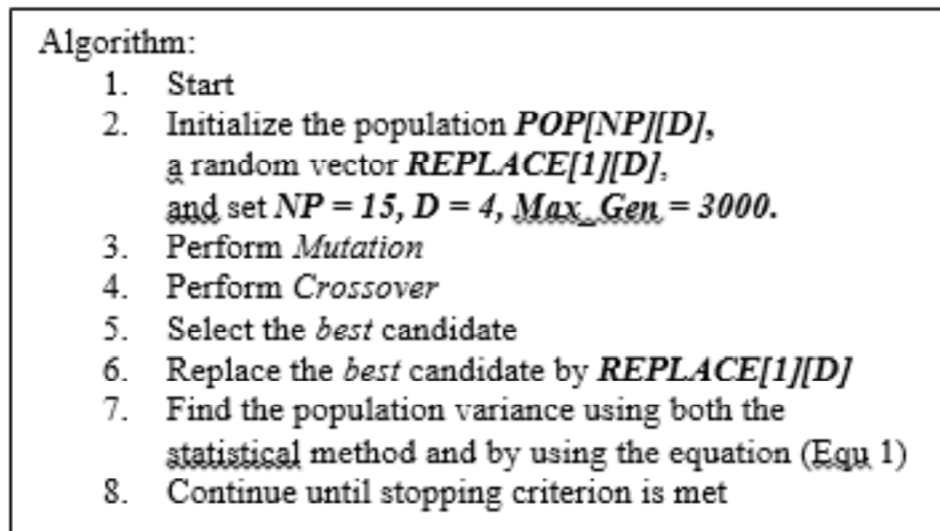
Figure 2: Algorithm for Sequential Implementation

and μ is mean of component[1]

The results obtained for the verification of the equation is presented in Table 1. The results includes the population variance values measured (for the first 150 generations) using the statistical equation (Equ (2)) and the by using the equation derived by the authors (Equ (1)). The Table 1 also shows the population variance measured without migration step. It is evident from the results that the population variance measured using the equation Equ (1) is matching with the values measured statistically, at all the generation. This proves the validity of the equation derive by the authors.

Table 1. Population variance measured for sequential implementation

G	Population Variance		
	Without Migration	With Migration	
		Statistical	Using Equ (1)
1	1852.80	1837/78	1837.78
10	134.47	211.38	211.38
20	4.64	87.98	87.98
30	0.97	84.87	84.87
40	0.07	83.43	83.43
50	0.01	83.92	83.92
60	0.00	83.84	83.84
70	0.00	83.86	83.86
80	0.00	83.85	83.85
90	0.00	83.86	83.86
100	0.00	83.86	83.86
110	0.00	83.86	83.86
120	0.00	83.86	83.86
130	0.00	83.86	83.86
140	0.00	83.86	83.86
150	0.00	83.86	83.86

3.2. Simulated Distributed Approach

Later, the above said experiment was extended to a population of 60 candidates and 30 dimension, for distributed *DE* with two sub-population. The population was divided into two subpopulation each having 30 candidates. Two different procedures are generated to evolve the subpopulation one by one. When one of the procedures is called, it performs mutation, crossover, selection and finds the best candidate in its subpopulation. Then this candidate is sent to the next procedure to replace a random candidate in the sub-population. Similarly, the second procedure also performs mutation, crossover, selection, finds best candidate and sends the best candidate to the first subpopulation. The algorithmic structure of this implementation is shown in Figure 3.

The results obtained after this implementation are presented in Table 2 for sub-population 1 and sub-population 2. The results are recorded for both the sub-populations, at every generation. The theoretical variance and the statistical variance after migration were found to be the same in every case. This reiterates the validity of the equation derived by the authors. Thus, the derived the equation is verified for its validity with two different experimental setups.

```
Algorithm:

    (i)     start
    (ii)    Initialize the population POP[NP][D], NP=60, D=30,
            Max_Gen = 300
    (iii)   Divide into two subpopulations of size 30 each
            for the procedures DE₁( ) and DE₂( )


DE₁( ) and DE₂( )
    (i)     Mutation
    (ii)    Crossover
    (iii)   Selection
    (iv)    Find the best candidate and Send it to other procedure
    (v)     Receive the best candidate from other procedure and
            Replace a random candidate by it.
    (vi)    Find_Variance ()


Find_Variance ( )
    (i)     Find the population variance using theoretical equation (Equ 1)
    (i)     Find the population variance using statistical equation (Equ 2)
```

Figure 3: Algorithmic Structure of Distributed Implementation

Table 2. Population variance measured for distributed implementation

G	Sub-Population 1		G	Sub-Population 2	
	Statiscial	*Using Equ (1)*		*Statiscial*	*Using Equ (1)*
1	2755/72	2755/72	1	3135.18	3135.18
2	2454/64	2454/64	2	2882.09	2882.09
3	2085.83	2085.83	3	2607.28	2607.28
4	1766/47	1766/47	4	2343.78	2343.78
5	1405.25	1405.25	5	2046.60	2046.60
6	1136.75	1136.75	6	1437.72	1437.72
7	816.92	816.92	7	922.19	922.19
8	610.55	610.55	8	683.48	683.48
9	476.84	476.84	9	557.16	557.16
10	321.72	321.72	10	362.21	362.21
11	155.01	155.01	11	201.60	201.60
12	95.85	95.85	12	117.21	117.21
13	38.81	38.81	13	79.79	79.79
14	18.19	18.19	14	33.02	33.02
15	8.93	8.93	15	19.75	19.75
16	3.94	3.94	16	7.17	7.17
17	2.14	2.14	17	2.75	2.75
18	1.10	1.10	18	1.41	1.41
19	0.36	0.36	19	0.53	0.53
20	0.11	0.11	20	0.17	0.17
21	0.03	0.03	21	0.06	0.06
22	0.00	0.00	22	0.01	0.01
23	0.00	0.00	23	0.00	0.00
24	0.00	0.00	24	0.00	0.00
25	0.00	0.00	25	0.00	0.00

4. CONCLUSION

This paper presented about the derivation of a theoretical equation to measure the population diversity of the sub-populations of the distributed differential evolution algorithm after the migration step. The validity of the derived equation is also verified with two different experimental approaches: sequential approach and simulated distributed approach. The experiments are conducted with the experimental setups suitably designed to implement classical *DE* and distributed *DE* (*dDE*) algorithms. The results obtained by the experiments proved that the derived equation to measure the population diversity as population variance of a population after migration of new candidates is valid.

References

[1]   Angela, A.R., Adriano, S., Andrade, O. and Soares, A.B. (2008), "Exploration vs exploitation in differential evolution", *Proceedings of the AISB 2008 Symposium on Swarm Intelligence Algorithms and Applications*, Aberdeen, Scotland, pp. 57-63.

[2]   Bergh, F.V.D. and Engelbrecht, A.P. (2004), "A cooperative approach to particle swarm optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 3, pp. 225-239.

[3]   Cantu-Paz, E. (2001), "Migration policies, selection Pressure and parallel evolutionary algorithms", Journal of Heuristics, Vol. 7, No. 4, pp. 311–334.

[4]   Dasgupta S., Das S., Biswas A., and Abraham A.,(2008) "The population dynamics of differential evolution: A mathematical model," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2008, pp. 1439–1446.

[5]   Dasgupta S., Das S., Biswas A., and Abraham A,(2009) "On stability and convergence of the population-dynamics in differential evolution," *AI Commun.*, vol. 22, no. 1, pp. 1–20, 2009.

[6]   Jeyakumar, G. and ShunmugaVelayutham, C. "A Comparative Study on Theoretical and Empirical Evolution of the Population Variance of the Differential Evolution Variants", *Lecture Notes in Computer Science* (*LNCS-6457*), IIT Kanpur, Springer-Verlag, pp. 75-79, (2010a).

[7]   Jeyakumar, G. and ShunmugaVelayutham, C.(2010) "Empirical study on migration topologies and migration policies for island based distributed differential evolution variants", *Swarm, Evolutionary, and Memetic Computing, Volume 6466 of the series Lecture Notes in Computer Science* pp. 29-37.(2010b).

[8]   Jeyakumar, G. and ShunmugaVelayutham, C. "Hybridizing Differential Evolution Variants Through Heterogeneous Mixing in a Distributed Framework", Hybrid Soft Computing Approaches, Studies in Computational Intelligence 611, Springer, PP. 107-151. 2015.

[9]   Liu, Y., Yao, D., Zhao, Q. and Higuchi, T. (2001), "Scaling up fast evolutionary programming with cooperative coevolution", *Proceedings of the IEEE Congress on Evolutionary Computation*, Vol. 2, pp. 1101-1108.

[10]   Potter, M.A. and DeJong, K.A. (1994), "A cooperative co-evolutionary approach to function optimization", *Proceedings of the 3rd International Conference on Parallel Problem Solving from Nature*, Springer-Verlag, pp. 249-257.

[11]   Skolicki, Z. and Jong, K.D. (2004), "Improving evolutionary algorithms with multi-representation island models", *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature – PPSN VIII, Lecture Notes in Computer Science*, Springer, Vol. 3242, pp. 420-429.

[12]   Skolicki, Z. and Jong, K.D. (2005), "The influence of migration sizes and intervals on island models", *Proceedings* of *the Genetic and Evolutionary Computation Conference*, pp. 1295–1302.

[13]   Sofge, D., DeJong, K.A. and Schultz, A. (2002), "A blended population approach to cooperative coevolution for decomposition of complex problems", *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 413-418.

[14]   Storn, R. and Price, K. (1995), "Differential Evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces", *Technical Report-95-012*, ICSI.

[15]   Storn, R. and Price, K. (1997), "Differential Evolution–A Simple and Efficient Heuristic Strategy for Global Optimization and Continuous Spaces". *Journal of Global Optimization 11(4), 341–359 (1997).*

[16]   Tasoulis, D.K., Pavliis, N.G., Plagianakos, V.P., Vrahatis, M.N.:(2004) *Parallel Differential Evolution. In: Congress on Evolutionary Computation* (CEC 2004), Portland (2004)

[17]   Thangavelu. S, Jeyakumar. G, Roshni M. Balakrishnan and Shunmuga Velayutham. C, (2015) "Theoretical Analysis of Expected Population Variance Evolution for a Differential Evolution Variant", *Computational Intelligence in Data Mining, Springer-Volume 2,* 403-416, January-2015.

[18] Xue F, Sanderson A. C., and Graves R. J., (2005a) "Modeling and convergence analysis of a continuous multiobjective differential evolution algorithm," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1. Sep. 2005a, pp. 228–235.

[19] Xue F, Sanderson A. C., and Graves R. J., (2005b) "Multiobjective differential evolution: Algorithm, convergence analysis, and applications," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1. Sep. 2005b, pp. 743–750.

[20] Zaharie, D. (2001a), "On the explorative power of differential evolution algorithms", *Proceeding of the 3rd International Workshop on Symbolic and Numeric Algorithms on Scientific Computing*, *SYNASC-2001*.

[21] Zaharie, D. (2001b), "Recombination operators for evolutionary algorithms", In-Ivanchev, D. and Todorov, M.D. (eds.), *Proceedings of the 27th Summer School of Applications of Mathematics in Engineering and Economics,* pp. 625-627.

[22] Zaharie, D. (2002a), "Critical values for the control parameters of differential evolution algorithms", *Proceedings of the 8th International Conference on Soft Computing*, pp. 62-67.

[23] Zaharie D., (2002b) "Parameter adaptation in differential evolution by controlling the population diversity," in *Proc. 4th Int. Workshop Symbolic Numeric Algorithms Sci. Comput.*, 2002b, pp. 385–397.

[24] Zaharie, D. (2003), "Control of population diversity and adaptation in differential evolution algorithms", In-Matousek, R. and Osmera, P. (eds.), *Proceedings of the Mendel's 9th International Conference on Soft Computing*, pp. 41-46.

[25] Zaharie, D. and Zamfirache, F. (2006), "Diversity enhancing mechanism for evolutionary optimization in static and dynamic environments", *Proceedings of the 3rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence*, pp. 460-471.

[26] Zaharie, D. (2008), "Statistical properties of differential evolution and related random search algorithms", In – Brito, P. (eds.), *Proceedings of the International Conference on* Computational *Statistics Porto*, pp. 473-485.

[27] Zaharie D., (2009)"Influence of crossover on the behavior of differential evolution algorithms," *Appl. Soft Comput.*, vol. 9, no. 3, pp. 1126– 1138, Jun. 2009.

[28] Zielinski K., Peters D., and Laur R., "Run time analysis regarding stopping criteria for differential evolution and particle swarm optimization," in *Proc. 1st Int. Conf. Exp./Process/System Modelling/Simulation/Optimization,* 2005.