Reducing the Required Experiments to Analyze the Performance of Metaheuristic Algorithms

Carlos A. Hernández Carreón[1], Héctor J. Fraire Huacuja[2], Karla Espriella Fernandez[1], Guadalupe Castilla Valdez[2], Juana E. Mancilla Tolama[1]

[1]IPN, E.S.I.M.E. Azcapotzalco
Av. de las Granjas No. 682 Col. Santa Catarina
02250-Azcapotzalco, México, D. F.
[2] Instituto Tecnológico de Ciudad Madero.
1o. de Mayo y Sor Juana I. de la Cruz S/N.
89440-Cd. Madero, Tamaulipas, México
cahc05@yahoo.com.mx, hfraire@prodigy.net.mx,
karla_rouge@yahoo.com, gpe_cas@yahoo.com.mx,
emtolama@hotmail.com

Abstract. When analyzing experimentally the performance of metaheuristic algorithms on a set of hard instances of a NP-hard problem, the required time to carry out the experimentation can be very large. A way to reduce the volume of experimentation is to incorporate variance reduction techniques in the computational experiments. The traditional approaches propose methods for the incorporation of these techniques which are dependent of the technique, the problem and the metaheuristic used. In this work we propose a new approach that consists in developing methods of general purpose which allow incorporating techniques of variance reduction, independently of the problem and of the metaheuristic used. To validate the feasibility of the approach, a method of general purpose is described which allows incorporating the technique of antithetic variables in computational experiments with metaheuristic algorithms. The experimental results show that with the proposed method a maximum reduction level of 55% in the number of experiments is achieved, with a minimum resources investment.

1. Introduction

Many real world applications require the solution of optimization problems which belong to a special class denominated NP-hard problems. Currently efficient algorithms to solve large instances for this kind of problems are not known, and one has the suspicion that it is not possible to build them. In [1, 2] it is indicated that the solution of large instances of NP-hard problems only can be solved simplifying the problem or using an approximate solution method. Nondeterministic heuristic methods of general purpose, denominated metaheuristics by Glover in [3], are currently considered very promising tools for the approximated solution of large instances of these problems.

Given the randomized characteristic of the metaheuristics, their performance is analyzed executing a series of computational experiments. In these experiments the

© G. Sidorov, B. Cruz, M. Martínez, S. Torres. (Eds.)
Advances in Computer Science and Engineering.
Research in Computing Science 34, 2008, pp. 25-34

Received 24/03/08
Accepted 26/04/08
Final version 04/05/08

behavior of the averages of the solution quality and the execution time of the algorithm are observed. When the variation of these averages is high, it is required a great quantity of experiments to determine a clearly tendency in the measurements carried out. In this work the problem of reducing this variation, and in consequence the number of experiments required to analyze the performance of the metaheuristic algorithms, is approached.

In the following sections the problem description, the related works, the proposed method and the experimental results are presented.

2. Problem Description

The goal of a simulation study is to determine the value of certain amount θ related to a stochastic process. A simulation produces a variate x_1 whose expected value is θ. A second execution of the simulation produces a new variate x_2; independent of the previous one, whose average also it is θ. This process continues until completing a total of n executions and producing n independent variates $x_1, x_2, \ldots x_m$ with $\theta = E(x_i)$ and $\sigma^2 = \mathrm{Var}(x_i)$. In order to estimate the mean (θ) and variance (σ^2) of the population, the sample mean $\bar{x} = \Sigma_i\, x_i / n$ and the sample variance $s^2 = \Sigma_i\, ((x_i - \bar{x})^2) / (n-1)$ are used. In [4] is indicated that to determine the goodness of the estimator $\bar{x}$ respect to θ, its quadratic error average $E[(\bar{x} - \theta)^2] = \mathrm{Var}(\bar{x}) = \mathrm{Var}(\Sigma_i x_i / n) = (\Sigma_i \mathrm{Var}(x_i)) / n^2 = \sigma^2/n = s^2/n$ must be used. This expression has two consequences: (a) that it is possible to increase the quality of the estimation of θ increasing the number of experiments and/or reducing to the variance sample and (b) that it is possible to reduce the number of experiments without affecting the quality of the estimation reducing the variance sample.

In [5] is indicated that to reduce the variance in computational experiments must be incorporating techniques of variance reduction. When it is analyzed the performance of metaheuristic algorithms experimentally on a group of hard instances of a *NP*-hard problem, the time required to carry out the experimentation is very long. A decrease in the number of experiments can be the only alternative to successfully obtain evidence enough to support the conclusions of the analysis of the algorithms performance.

Metaheuristic algorithms realize a search through the space of solutions making random decisions in order to avoid to get stuck on some local optima and to advance quickly toward the global optima. Typically in each computational experiment or execution of the algorithm an initial solution is randomly uniform generated and a local search is realized in the neighborhood of this solution to improve it. Once the possibility of improving it is exhausted, a new experiment begins and the process is repeated. In each experiment i the target output x_i is obtained. When all the experiments are executed, the mean and the variance of x_i are calculated. Given the randomness of the metaheuristic algorithms, their outputs depend on a set of uniform random numbers $\{u_i\}$ generated during the algorithm execution such way that $x_i = x_i (u_1, u_2, \ldots, u_l)$. A part of these numbers are binding with the decisions associated to the construction of the initial solution and others with the decisions in the local search process. Contrary to a simulation in a metaheuristic algorithm the amount of random decisions carried out in each experiment is variable; this constitutes the most important difficulty to apply the method of antithetic variates in the reduction of the variance in experiments with metaheuristic algorithms.

3. Related Work

Some of the most recent and relevant works related with the problem of incorporate the techniques of variance reduction in computational experiments with metaheuristic algorithms are the following one:

McGeoch is one of the pioneers in the development of methods of variance reduction in experiments with randomized algorithms. In [6, 7] is proposed the use of the techniques of common random numbers, antithetic and control variates, in the variance reduction in the experimental analysis of the performance of heuristic algorithms applied to the solution of the self-organized search problem. It establishes that, in the experimental analysis of the performance of algorithms exists many opportunities to apply techniques of variance reduction, because the algorithms are simpler and they have a more precise definition than the simulation problems. The main limitation of this approach is that the proposed methods are dependent of the technique, the problem and the algorithm used.

In [5, 8] the authors propose a method to reduce the variance in experiments with metaheuristic algorithms based on the technique of antithetic variates. The method is described in the context of the solution of the SAT problem with the threshold accepting metaheuristic. The method requires that the solutions of the problem be represented using binary vectors and that the metaheuristic generates an initial random solution.

As we can see the traditional approach consists on developing variance reduction methods that are dependent of the technique, the problem and the metaheuristic used. In this work a new approach is proposed, it consists on developing methods of general purpose that allow incorporating techniques of variance reduction, which they are independent of the problem and the metaheuristic used. In order to validate the proposed approach a method of general purpose based on the technique of antithetic variates is described and it is applied to different problems and metaheuristics.

4. Proposed Method

4.1. Main Idea

In order to develop a variance reduction general purpose method the basic idea consists on identifying the first random decision that always occurs (conditional independent) in each computational experiment or algorithm execution. This decision, which will be denominated base decision, it should be associated to the generation of alone one random number. Unlike the rest of the random numbers generated in the process, the random number associated to the base decision, must be stored so that it will be available in the following experiment. If the global variable $a \in (0,1)$ is used to store the random number associated to the base decision, before the number generation we must to verify if the number of the experiment currently executed is even or uneven. If the number of the experiment is uneven it must be generated a new random number and stored it in a, otherwise the complement of the random number currently stored in a must be stored in a. The idea is to use alone one of the random numbers generated in the algorithm process, to negatively correlate all the outputs of all experiments realized.

4.2. Method Description

The proposed method consists of the following four steps:

Step 1. Identify the base decision. Analyze the sequence of random decisions that occurs in each computational experiment. Determine all the decisions that not necessarily takes place in each experiment, these occurs if a conditional comes true. Identify those that always occur in the experiments, their occurrence is non-conditional dependent. From the set of decisions that consistently occurs, select the first one and set it like the base decision.

Step 2. Generate the random number associated to the base decision. Let a and c, the global variables in which the random number associated to the base decision and the current experiment number, respectively will be stored. Now before carrying out the base decision, if c is uneven a new random number must be generated and stored (a= random()), otherwise the complement of a must be stored (a=1-a). Once completed the previous process the variable a is used to carrying out the base decision. Now the pair of random numbers associated to the base decision in successive experiments (uneven and even) will be correlated negatively.

Step 3. Determine the outputs of each experiment. Execute all the experiments and get the values of the specified output t_i, for each experiment i =1,2..., NEXP.

Step 4. Determine sample mean and sample variance. The values of the variates x, y, and z are determined, using the following expressions:

$x_j(a) = t_{2j-1}$, $y_j(1-a) = t_{2j}$ y $z_j = (x_j(a) + y_j(1-a))/2$ $j=1,2,\ldots,NEXP/2$.

Calculate the mean and the variance of z.

By definition t_i, x_j, y_j, and z_j, are estimators of the expected value of variate t, and as $z = (x+y)/2$, then $Var(z) = Var((x+y)/2) = ¼ [Var(x) + Var(y) + 2 Cov(x,y)]$. Now as x and y were generated from negatively correlated inputs, then $Cov(x, y) < 0$. Then the method must produce a reduction of the sample variance of the estimator z_i with respect to any estimator z'_i obtained using values of t_i negatively noncorrelated.

This method can be used with any combination of a metaheuristic algorithm and NP_hard problem. The method is applicable to any kind of randomized algorithm, nevertheless in this work only are showed the results of experiments realized with metaheuristic algorithms.

4.3. Example

In this example the SAT problem instance f600 is used. The instance was solved executing 30 times the threshold accepting metaheuristic.

Table 1 shows the execution time t_i observed for each experiment i=1, 2,...,30, when the proposed method isn't applied. In this case, both the mean and variance of the variate t are directly calculated. However, the variates x, y, and z will be calculated to show the effects produced by the method application. Table 2 shows the x_j, y_j, and z_j values obtained without the method application. As we can see the x and y covariance is 0.0035 and the z variance is 0.0142.

Table 1. Execution times obtained without the proposed method application.

i	t_i	I	t_i	i	t_i	i	t_i	i	t_i	i	t_i
1	0.704	6	0.906	11	0.828	16	1.125	21	1.015	26	1.11
2	0.718	7	0.657	12	0.907	17	0.906	22	0.953	27	0.625
3	1.188	8	0.937	13	0.89	18	1.063	23	0.719	28	0.781
4	0.89	9	0.656	14	0.688	19	0.797	24	0.89	29	0.844
5	0.641	10	0.797	15	0.734	20	1.156	25	0.922	30	0.687

Table 2. x, y, and z values obtained without the proposed method application.

j	x_j	y_j	z_j	j	x_j	y_j	z_j
1	0.704	0.718	0.711	9	0.906	1.063	0.9845
2	1.188	0.89	1.039	10	0.797	1.156	0.9765
3	0.641	0.906	0.7735	11	1.015	0.953	0.984
4	0.657	0.937	0.797	12	0.719	0.89	0.8045
5	0.656	0.797	0.7265	13	0.922	1.11	1.016
6	0.828	0.907	0.8675	14	0.625	0.781	0.703
7	0.89	0.688	0.789	15	0.844	0.687	0.7655
8	0.734	1.125	0.9295				

Table 3 shows the execution time t_i observed when the proposed method is applied. In this case, both the mean and variance of the variate t are calculated using the method step 4. If the x_j, y_j, and z_j values are calculated, the x and y covariance is -0.00052 and the z variance is 0.00088. As we can see the proposed method application produces a variance reduction of 37%.

Table 3. Execution times obtained with the proposed method application.

i	t_i	i	t_i	t_i	t_i	i	t_i	i	t_i	i	t_i
1	0.704	6	0.641	11	0.688	16	0.922	21	0.922	26	0.703
2	0.687	7	0.922	12	0.734	17	0.782	22	1	27	0.797
3	1.00	8	0.812	13	0.907	18	0.843	23	0.875	28	0.891
4	0.75	9	0.641	14	0.64	19	0.844	24	0.86	29	0.625
5	0.734	10	1.109	15	1.031	20	0.609	25	0.937	30	0.781

5. Experimental Results

A set of experiments were executed to evaluate the variance reduction level when the proposed method is applied in experiments with metaheuristic algorithms. A set of instances were used and each one was solved with two implementations. For each algorithm an implementation incorporates the proposed method and the other not. Thirty experiments were executed with each instance and the specified outputs were measured. Finally the variances before and after the application of the method, were used to determine the variance reduction level.

To describe a typical set of experimental results it will be used three test cases that include three problems and three metaheuristics. A test case includes the SAT problem and the metaheuristics grasp [12] and tabu search [13], other includes the Lennard-Jones problem and two variants of the genetic algorithm described in [14], and the last one test case includes the hot rolling scheduling problem and a metaheuristic genetic algorithm based [15].

Table 4 shows the variance reduction level, of the solution quality average Avg(%E), obtained when the instances were solved with the metaheuristic grasp. The first column contains the instances used in the experiments. The columns VB and VA contain the variance obtained before and after the method application. The column %R shows the variance reduction percentage. Also, the table includes the global average of the reduction level reached.

Table 4. Variance reduction level observed with: the SAT problem and a grasp algorithm.

	Avg(%E)		
Instance	VB	VA	%R
uf200-0100	7.40221	1.809563	76%
uf250-01	3.68832	2.1144068	43%
uf250-010	4.80004	2.8571141	40%
F600	9.0066	10.409657	-16%
F1000	24.0414	9.4095563	61%
Global reduction avg.			41%

Table 5. Variance reduction level observed with: the SAT problem and a tabu search algorithm.

	Avg(%E)		
Instance	VB	VA	%R
uf200-0100	9.13128	4.12374	55%
uf250-01	15.0622	7.20976	52%
uf250-010	11.3515	3.35256	70%
F600	164.599	101.353	38%
F1000	256.397	101.923	60%
Global reduction avg.			55%

Table 6. Variance reduction observed with:
the Lennard Jones problem and a genetic algorithm (v1).

	Avg(%E)			Avg(NEOF)			Avg(NG)		
I	VB	VA	%R	VB	VA	%R	VB	VA	%R
15	0.05	0.03	40%	61,407	15,761	74%	1.08	0.54	50%
16	0.05	0.07	-40%	139,689	42,153	70%	3.46	1.27	63%
17	0	0	0%	9,760	3,789	61%	0.03	0.03	0%
18	0.06	0.02	67%	250,824	142,806	43%	5.1	3.31	35%
19	0.78	0.24	69%	294,092	210,221	29%	6.16	2.99	51%
20	0.49	0.29	41%	280,123	208,500	26%	4.64	2.36	49%
21	0.45	0.2	56%	411,777	160,780	61%	6.88	2.55	63%
22	0.82	0.28	66%	462,982	156,176	66%	6.22	1.98	68%
23	0.97	0.51	47%	338,332	285,224	16%	4.31	3.94	9%
24	0.46	0.16	65%	462,886	378,918	18%	5.79	5.37	7%
25	0.26	0.15	42%	198,652	357,526	-80%	3.21	5.3	-65%
26	0.81	0.19	77%	688,233	228,416	67%	8.72	2.62	70%
27	0.81	0.23	72%	535,909	129,526	76%	6.38	2.43	62%
28	0.64	0.27	58%	403,537	194,353	52%	4.99	3.04	39%
29	0.4	0.42	-5%	495,611	344,752	30%	4.88	3.88	20%
30	0.41	0.24	41%	1,341,653	112,631	92%	12.74	1.14	91%
Global reduction (Avg)			43%			38%			38%

Table 5 shows the variance reduction level, of the solution quality average Avg(%E), obtained with the metaheuristic tabu search. The Table 2 structure is similar to the structure of the Table 4.

Table 6 shows the variance reduction level observed in three random outputs. In this test case 15 instances of the Lennard Jones problem were solved with a genetic algorithm (v1). The random outputs were the average of the error percentage Avg(%E), the average of the evaluations number of the objective function Avg(NEOF) and the average of the generations number Avg(NG). The first column contains the instances used in the experiments, and the additional columns are grouped in three subtables, each one with three columns. The structure of the subtables is similar to the Table 4 structure.

Table 7 have the same structure that Table 6, and contains the reduction level observed when a variant of the genetic algorithm (v2) was used.

Table 8 shows the variance reduction level observed in two random outputs. In this test case 17 industrial instances of the hot rolling scheduling problem were solved with a genetic algorithm. The random outputs were the average of the rolling time Avg(%RT) and the average of the required time to obtain the best solution Avg(TB). The first column contains the instances used in the experiments, and the additional columns are grouped in two subtables, each one with three columns. The structure of the subtables is similar to the Table 4 structure.

Table 7. Variance reduction observed with:
the Lennard Jones problem and a genetic algorithm (v2).

	Avg(%E)			Avg(NEOF)			Avg(NG)		
I	VB	VA	%R	VB	VA	%R	VB	BA	%R
15	0.08	0	100%	175,755	71,702	59%	4.2	2.5	40%
16	0.03	0	100%	37,722	26,511	81%	4.2	0.83	80%
17	0	0	0%	4,647	2,253	52%	0	0	0%
18	0.11	0.04	64%	679,707	512,567	25%	14.7	10.8	27%
19	0.55	0.22	60%	953,268	397,323	58%	18.39	6.59	64%
20	0.68	0.21	69%	1,758,666	1,021,182	42%	28.48	16.7	41%
21	0.45	0.25	44%	916,672	997,527	-9%	15.93	13	19%
22	0.62	0.24	61%	1,547,998	750,757	52%	25.79	11.2	57%
23	0.33	0.25	24%	2,357,355	986,281	58%	29.86	13.1	56%
24	0.38	0.28	26%	873,663	1,758,849	-101%	11.43	26.8	-134%
25	0.29	0.1	66%	1,777,160	391,122	78%	21.61	6.85	68%
26	0.48	0.18	63%	2,334,290	1,020,252	56%	34.12	13.7	60%
27	0.54	0.39	28%	3,261,303	1,257,075	61%	44.78	14	69%
28	0.62	0.28	55%	2,197,110	597,272	73%	26.33	7.46	72%
29	0.49	0.22	55%	2,885,274	716,521	75%	31.06	8.7	72%
30	0.27	0.26	4%	4,187,946	722,313	83%	48.33	6.98	86%
Global reduction (avg)			51%			42%			42%

The experimental evidence shows that the proposed method can reduce the variance of the random outputs for almost all the considered instances and the reduction global levels go from 14% to 55%. The investment of resources required to obtain these reduction level is minimum, since only it is required to control one of the random numbers generated by the algorithm. An outstanding feature of the method is that it simultaneously produces a reduction in the variance of all the random outputs of the algorithm. In the Tables 3, 4 and 8 we can see that the method operates simultaneously on several outputs.

As we can se the proposed method allows to incorporate a technique of variance reduction in computational experiments, independently of the problem solved and of the metaheuristic used. Currently the most relevant open question is: why the reduction don't operate over several instances (see f600 and others).

6. Conclusions and Future Work

In this paper the problem of how to reduce the amount of required experiments to analyze the performance of metaheuristic algorithms was approached. Traditional approaches propose to apply variance reduction methods that are dependent of the technique, of the problem and of the metaheuristic used. The solution approach proposed in this work consists in developing reduction methods which they are independent of the problem and of the metaheuristic used.

Table 8. Variance reduction observed with: the hot rolling scheduling problem and a genetic algorithm.

Instance	Avg(RT) VB	VA	%R	Avg(TB) VB	VA	%R
hsm002.txt	0.000229	0.000188	17.90%	28.619	34.049	-18.97%
hsm003.txt	0.000186	0.000115	38.17%	3.591	2.812	21.69%
hsm004.txt	0.000180	0.000143	20.56%	4.901	2.275	53.58%
hsm005.txt	0.000183	0.000149	18.58%	5.089	10.467	-105.68%
hsm006.txt	0.000394	0.000312	20.81%	4.230	10.849	-156.48%
hsm007.txt	0.000395	0.000293	25.82%	4.807	3.265	32.08%
hsm008.txt	0.000472	0.000317	32.84%	4.840	3.031	37.38%
hsm009.txt	0.000302	0.000332	-9.93%	4.027	3.794	5.79%
hsm010.txt	0.000288	0.000196	31.94%	5.606	2.063	63.20%
hsm011.txt	0.000299	0.000161	46.15%	2.586	2.298	11.14%
hsm012.txt	0.000230	0.000154	33.04%	3.951	2.441	38.22%
hsm013.txt	0.000217	0.000129	40.55%	2.991	1.954	34.67%
hsm014.txt	0.000840	0.000626	25.48%	4.187	1.907	54.45%
hsm015.txt	0.000846	0.000624	26.24%	3.648	1.861	48.99%
hsm016.txt	0.000835	0.000661	20.84%	4.114	2.102	48.91%
hsm017.txt	0.000922	0.000494	46.42%	3.148	1.957	37.83%
hyl001.txt	0.001740	0.000503	71.09%	4.441	2.823	36.43%
Global reduction (avg)			29.79%			14.31%

In order to validate the proposed approach, a general purpose method based on the technique of antithetic variates is presented and it was applied to three *NP-hard* problems and three metaheuristics. The experimental evidence shows that the method has the capacity of simultaneously to reduce the variance of several random outputs of the used algorithm. On the other hand, the levels of global of reduction on the instances used in the test cases go from 14% to 55%. The propose method is one of the first general purpose methods that allows to incorporate a technique of variance reduction in computational experiments, independently of the problem solved and of the metaheuristic used. The method is applicable without greater modifications to any type of randomized algorithm. Currently the most relevant open question is: why the reduction don't operate over several instances

References

1. Garey, M.R., Johnson, D.S.: Computer and intractability: a Guide to the Theory of NP-Completeness. WH Freeman, New York (1979)

2. Papadimitriou, C., Steiglitz, K. Combinatorial Optimization: Algorithms and Complexity. Dover Publications (1998)
3. Glover, F.: "Future Paths for Integer Programming and Links to Artificial Intelligence". Comput. Oper. Res. Vol 13. (1986) 533-549
4. Barr, R.S., Golden, B.L., Kelly, J., Steward, W.R., Resende, M.: Guidelines for Designing and Reporting on Computational Experiments with Heuristic Methods. Proceedings of International Conference on Metaheuristics for Optimization. Kluwer Publishing, Norwell, MA (2001) 1-17
5. Johnson, D.S., McGeoch, L.A. "Experimental Analysis of Heuristics for the STSP. In: Gutin, G., Punnen, A. (eds.): The Traveling Salesman Problem and its Variations". Kluwer Academic Publishers, Dordrecht (2002) 369-443
6. Ross, S.M.: Simulación. Segunda edición. Prentice Hall (1999)
7. Fraire, H. J., Castilla, V. G., Hernández, R. A., Aguilar, L. S., Castillo, V. G., Camacho, A. C.: Reducción de la varianza en la evaluación experimental de algoritmos metaheurísticos. Proceedigns 13th International Congress on Computer Science Research CIICC06. ISBN 968-5823-33-2. (2006) 14-22
8. McGeoch, C.C.: "Analyzing Algorithms by Simulation: Variance Reduction Techniques and Simulation Speedups". ACM Computer Survey. Vol. 24 No. 2. (1992) 1995-212
9. McGeoch, C.C. Experimental Analysis of Algorithms. In: Pardalos, P.M., Romeijn, H.E. (eds.): Handbook of Global Optimization, Vol. 2 (2002) 489-513
10. McGeoch, C.C.: "Analyzing Algorithms by Simulation: Variance Reduction Techniques and Simulation Speedups". ACM Computer Survey. Vol. 24 No. 2. (1992) 1995-212
11. Fraire, H. J. Una metodología para el diseño de la fragmentación y ubicación en grandes bases de datos distribuidas. Tesis doctoral. CENIDET, Cuernavaca, México (2005)
12. González-Velarde, J.L.: GRASP. In A. Díaz, ed. Heuristic Optimization and Neural Networks in Operations Management and Engineering. Editorial Paraninfo, Madrid. (1996) 143-161
13. Glover, F., Laguna, M.: *Tabu Search*. Kluwer Academic Publishers. 1997
14. Romero, D., Barrón, C., Gómez, S. The Optimal Geometry of Lennard-Jones Clusters: 148-309.: Computational Physics Communications. Vol. 123 No. (1-3). (1999) 87-96
15. Espriella, F. K.: *Optimización mediante algoritmos genéticos: aplicación a la laminación en caliente*. Master Thesis. Instituto Politécnico Nacional, CICATA Unidad Altamira (2008).