Analysis of the Simulated Annealing Convergence in Function of the Standard Deviation and the Boltzmann Quotient for Scheduling Problems[1]

Martín G. Martínez-Rangel[1,2], Marco Antonio Cruz-Chávez[1], José Crispín Zavala-Díaz[2], David Juárez-Romero[1], Ocotlán Díaz-Parra[1]

[1]CIICAP-[2]FCAeI, Universidad Autónoma del Estado de Morelos
Avenida Universidad 1001. Col. Chamilpa, C.P. 62210. Cuernavaca, Morelos, México
{mmtzr,mcruz,crispin_zavala,djuarez,odiazp}@uaem.mx

Abstract. Simulated annealing (SA) converges by means of a probability of acceptance toward a minimum value of the cost function to a minimum temperature. When the cost function is very high, the probability of acceptance is minimum when temperature descends to a minimum value, for this, the probability is controlled for the temperature. An incorrect tuning of this parameter makes that the distribution of the probabilities of acceptance along the whole process of SA is slanted toward values very low or very high, what cause fall easily in local optimum. In this paper an analysis of correlation between the standard deviation and the distribution of probabilities of Boltzmann is made. The experimental results demonstrate that the standard deviation obtained through a sample of the solutions space of the problem, allow for a good tune of the initial temperature in SA.

1 Introduction

Many of the combinatorial optimization problems are classified like NP-hard, being JSSP (Job Shop Scheduling Problem) one of the most difficult problems to solve in this classification [2]. The required time to solve JSSP is increased exponentially according to the size of the problem. Benchmarks with 20 jobs and 20 machines are considered large because at the moment the solution is not know For great instances of these problems, there are no deterministic algorithms that can solve them. For this reason, they use a kind of metaheuristic nondeterministic that limits in polynomial time the approach to the global optimum in this set of problems [4].

Diverse metaheuristics have been proposed to the search of the global optimum. These include SA (Simulated Annealing) [5], [6], Tabu Search[7], [8], [9], Ant Colony [10] and Memetics Algorithms which are Genetic Algorithms used in a local search [10], [11], [12], [13] .The main characteristic that shares this algorithm is the great amount of time that us required to find or to approach a global optimum for very great instances by means of searches in neighborhoods of nondeterministic form. For JSSP, the metaheuristics work generates new solutions (Schedules) and review its

[1] This work was supported by Project 160 of the Fideicomiso SEP-UNAM, 2006-2007.

Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 282–293

Received 24/06/07
Accepted 31/08/07
Final version 30/09/07

quality by means of evaluation of its scheduling, which assigns the times of beginning and ending for each one of the operations involved in the instance of the problem in order to be able to evaluate the objective function that is desired, for example the Makespan. The Makespan is defined as the time taken to complete the last O_i operation assigned in the process and is equal to the sum of the time of beginning of O_i plus the time of processing of O_i.

In this document, a new approach tune the temperature of beginning of SA help to converge or to approach faster to the global optimum for JSSP. The more critical problem presented SA it has to do with the tuning of its parameters, specially the temperature of beginning that is considered within SA like a control parameter, of the algorithm depends. There are other parameters not less important as the area quality of the generated solutions, the probability distribution for metropolis for the acceptance of new solutions, as well as the speed of cooling there are diverse approaches to tune SA. In [15] an approach uses a method that consists of the selection of an upper level in the temperature for each restart of SA. This upper level is selected when random way solutions are generated to take the best solution found in a time defined by test and error, the value of the upper level obtained when evaluating the objective function of the chosen solution. This improves the SA efficiency. In [17] an approach that the temperature fluctuates between a maximum and a minimum (feedback), the change of temperature will depend on a variable that can take positive or negative values (of random way) during each execution from SA.

For more than half a century, the mathematicians Bienaymé y Chebyshev [19] examined separately the property of the variability of the data around the average. They found at least that, regardless how a data set is distributed the percentage of observations that is contained within distances of plus-less k Standard deviations around the average must be $(1-1/k^2)100\%$ (rule of Bienaymé y Chebyshev). By means of an analysis of the behavior of the process of SA with respect to the distribution of the data and the application of this rule, in this document it is considered that the initial temperature is equal to twice the Standard deviation of solutions generated randomly, and that the analysis of the Boltzmann function to accept or to reject a solution will depend if the new solutions are in a rank (upper limit and lower limit) determined by the average of the generated solutions plus or less twice the Standard deviation. In order to accept as a matter of principle probabilistic 75% of all the solutions generated in the SA process, that is equivalent to 95% in processes that follow a normal distribution.

This work is composed of the following sections. Section 1 introduces the subject. Section 2 describes a generalized scheme of the algorithm of simulated annealing and the use of parameters. Section 3 proposes a new approach to tune the temperature parameters and the probability Boltzmann distribution function for the acceptance or rejection of new neighbors using the Standard deviation of the quality of the generated solutions there are at quality is also important and shows the algorithm of Simulated Annealing using this approach. Section 4 presents the computational study and the experimental tests using instances of problems known for JSSP. Section 5 presents the conclusions.

2 Generalized Scheme of Simulated Annealing

Simulated Annealing is a technique of stochastic local search approximates the minimum value of the cost function $f : S \rightarrow \Re$ on a finite set of S. It is an iterative method that moves in the space of solutions using a function of neighborhood N(x). When generating a new solution x´ of x, the solution candidate x´ is accepted as the new solution if $f(x') < f(x)$ or if $f(x') > f(x)$ is rejected or accepted in grasp of the function of probability of acceptance of Boltzmann P(x), which involves the parameter of control T, and the difference of the values of the quality of the solution (x´-x). Initially T has very high values and according to the algorithm progress. T decreases and influences in the probability of acceptance of the solution x´. The general procedure of SA [5] is defined in the following form:

1. Select a value of high beginning to T_0, a limit T_f to decrement to T_0 and an initial state x_0

   $T_t \leftarrow T_0$, $x \leftarrow x_0$
2. For each iteration k, k=1....k_f to do the following:
   a. Repeat until the balance is reached:
      i. Calculate the value of state x by means of the cost function: $E_k \leftarrow f(x)$
      ii. Generate a new state x′ using a neighborhood function, $x' \leftarrow N(x)$
      iii. Calculate the value of the state x′ by means of the cost function, $E_k' \leftarrow f(x')$
      iv. Assign $x \leftarrow x'$ according to the probability determined by the function of acceptance P(x)
3. Reduce T to $k+1$ using a control factor γ, $T_{k+1} \leftarrow \gamma * T_k$, where $0 < \gamma < 1$.
4. When T_{k+1} is less than T_f, finish
5. Return the best found solution x and their value of cost E.

On the whole the method of Simulated Annealing consists of a system of states x and the relations between the following functions

1. $f(x)$: A function of costs to be minimized

2. N(x): A mechanism of generation of neighborhood (it generates new states)

3. P(x): An acceptance function that decides if the new state is accepted or is rejected
4. T(k): A parameter of control of the annealing

For problems of numerical optimization, x is defined as a vector of integer or real numbers, and the function of Boltzmann P (x) is used for the acceptance of new states.

The Boltzmann distribution function uses a probability density function of the Gaussian type. This is defined as:

$$P(x') = \begin{cases} 1 & \text{if} \quad f(x') <= f(x) \\ e^{(-f(x')-f(x))/T} & \text{otherwise} \end{cases} \tag{1}$$

For JSSP, a state x is defined by a solution S (schedule) of the problem. The cost function f (x) is defined in this work by makespan C_{max} (S). Neighborhood N (S) of S is defined as a set of feasible solutions that can be generated from S by means of only one step. This step is a disturbance of a pair of operations (i, j) assigned in a M_j machine.

3 Parameters for Simulated Annealing using a dispersion measurement

The algorithm of Simulated Annealing requires that the used parameters have certain values which are determined generally a priori. Within the required parameters are included the parameters of control T_0 y T_f, the factor of temperature γ that defines the speed of decrement of T, as well as the length of the Markov chain which defines the number of iterations to carry out in the algorithm of Metropolis [25] in which some trades before generating a decrement in T. Each problem of scheduling has different characteristics, and consequently different degrees of difficulty. For this reason it is necessary to find the suitable values of the parameters involved in the process of simulated annealing. In order to tune SA, it is required that the indices witch allow knowing throughout the process if the values of the used parameters are the correct ones. Two of the indices are the uniform distribution $P(x)$ of the probability of acceptance of the Boltzmann distribution function (equation 3) and the distribution $P(n)$ at random n number generated and evenly distributed between (0,1) to determine the acceptance or rejection of a solution regarding the Boltzmann distribution function. Equation 2 indicates that the sum of the distribution of the probability of acceptance of the Boltzmann function must approximately be equal to the sum of the probability distribution $P(n)$ generated randomly (See figure 1). The problem obtained to achieve the equality as far as the uniform distribution of probabilities is that $P(e^{(-f(x')-f(x))/T})$ has a Poisson distribution, whereas $P(n)$ has a normal distribution. By probabilistic principle it is accepted that $P(e^{(-f(x')-f(x))/T})$ has a greater dispersion, this by the own behavior of the phenomenon of SA ,that at first accepts great variations of energy (very high probabilities) when T is very high, and in agreement T decreases and tends to zero, the variation of accepting by $P(x)$ is mini-

mum. Therefore the equation (2) is taken like a valued parameter which is composed of two indices, to evaluate the SA behavior (to see figure 2).

$$\sum P(e^{(-f(x')-f(x))/T}) \cong \sum P(n) \tag{2}$$

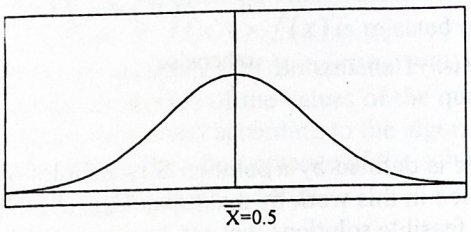

Fig. 1. Distribution of $P(n)$

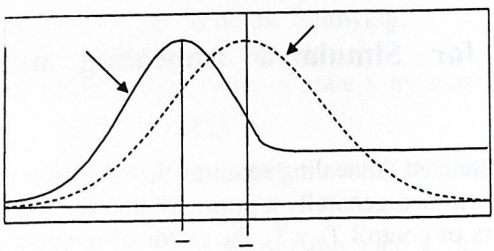

Fig. 2. Probability distribution of $P(e^{(-f(x')-f(x))/T})$ and $P(n)$

Aside from the equation 2 that shows the behavior of the SA process, it is another index that allows us to tune the parameters required in SA. This index is the standard deviation of the quality of the generated solutions that enter the process of SA within the procedure of Metropolis. This index shows us the dispersion degree that presents the variations in SA. In the statistic area, it is useful to know the degree dispersion that has the solutions in order to consider the degree of acceptance of these solutions. Regardless now a data set is distributed, the percentage of observations that are contained within distances $\pm$ k standard deviations around the average must be, at least:

$$(1-\frac{1}{k^2})100\% \tag{3}$$

Therefore, for the data whose polygons adopt any form, in the case of SA, at least [1 (1/22)]100%=75,0% of the observations must be contained within distances of 2 standard deviations $\pm$ around the average. 88,89% must be contained within distances plus or minus 3 standard deviations, and 93,75 standard deviations plus or less.

This is applicable when it is known that a particular random phenomenon did not follow the normal distribution pattern. In the case of SA, it follows a distribution pattern of Poisson.

In order to tune the degree of acceptance of a new solution, it is possible to chose between these three possibilities: for the SA process it is proposed to increase twice the standard deviation to the average value to obtain its upper bound and to diminish in the same proportion to obtain the lower bound (fig.3), which is equivalent to accept the 75,00% of the solutions in SA or the 95,44% in other processes with normal distributions. Table 1 shows the variation of the data to the around of the average according to the distribution pattern.

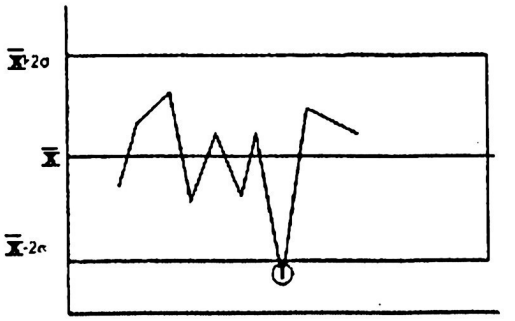

fig. 3. Minimum zone of occurrence to accept a 75% of the solutions for its evaluation in the proposed SA

Table 1. Variation of the data to the around of the average

Number of k units of standard deviation	Percentage of observations contained between average and k	
	Poisson distribution (Bienayme-Chebyshev)	Normal Distribution
1	No calculable	Exactly 68.26%
2	At least 75.00%	Exactly 95.44%
3	At least 88.89%	Exactly 99.73%
4	At least 93.75%	Exactly 99.99%

There are only accepted energy differences $\Delta E_k = -(f(x') - f(x))$, which are in the rank delimited by the $2*\sigma$, we can do that T_0 will be equal to the same $2*\sigma$ within the SA process ($T_0 = 2*\sigma$) in order that the distribution of $P(e^{(-f(x')-f(x))/T})$ will be uniform within SA. The importance to choose a good solution of beginning in SA to obtain good solutions already is mentioned in [15]. The values of each one of the parameters involved in the SA process based on the standard deviation are defined as follows:

1. For each one of benchmarks used in the tests, to generate a set of solutions Ω at random sufficiently to identify the standard deviation of the quality of the solutions that compose it, its average value, the minimum and maximum value found.
2. The Speed of cooling (length of the Markov chain), will be given by the double of large of the neighborhood of the problem [6] Ve = (n* (n-1))*2; where n is the number of jobs, which determines the cycle of Metropolis.
3. The value of *To* is equal to twice the Standard deviation ($2 * \sigma$) found for a set of solutions generated in point 1.
4. The decrement of the temperature is given by: $T \leftarrow \gamma * T$, γ is the coefficient of temperature that controls the cooling of the system.
5. A S′ solution of the problem could be evaluated to be accepted or to be rejected by means of the function of probability distribution of Boltzmann, as long as ΔE_k it will not be greater to twice the found standard deviation for the neighborhood upper bound and lower bound.

The previous parameters are used like an initial tuning to execute the process of simulated annealing. At a figure 4 shows the Algorithm of Simulated Annealing that allows to converge of faster way to optimal solutions for benchmarks of JSSP used like test, where $\Delta E_k = -(f(x') - f(x)) = \Delta S_k = -(f(S') - f(S))$ and $P(e^{(-f(x')-f(x))/T}) = P(e^{(-f(S')-f(S))/T})$.

```
1.  Input Data: S, σ
2.  Result: Makespan
3.  T = 2 *σ, Makespan = 10000;
4.  SizeNeighborhood ← Ve; //sampling size (n jobs)
5.  while (T > 0){
6.  Neighbor ← 0;
7.     while (NNeighbor< SizeNeighborhood){
8.        Generate a state S′ ∈ N ( S ) by means of a perturbation in S
9.        If (S' ≤ S)
10.       {
11.          S = S' ;
12.          if (Makespan > S) Makespan ← S;
13.          Neighbor++;
14.       } //end if
15.    else
16.       {
17.         ΔS ← S'-S
18.         if (ΔS ≤ 2 *σ) {
19.    With a randomly generated number α evenly distributed between (0,1)
20.            if ( α <exp (-ΔS /T))) Then {
21.               S ← S′ //accept new solution
```

```
22.                Neighbor++;
23.                } //end if
24.              else
25.                S ← S; //reject new solution
26.            }//end if
27.          else
28.              S ← S; //reject new solution
29.          } //end else
30.      } //end while
31.    T ← γ * T
32.  }// end while T_f > 0
```

Fig. 4. Simulated Annealing Algorithm with standard deviation

4 Computational Results

The proposed mechanism was implemented in C language in a PC with 2 GHz, and 1 Gb in RAM. In order to prove the efficiency of the proposed mechanism a set of benchmarks for JSSP of small size (Mt06, Mt10) [21], medium (La40) [22] and great (Yn1) [23] were used. The space of feasible solutions to be explored was generated by means of the neighborhood structure proposed in [20] and using the proposed mechanism of partial rescheduling in [24]. In table 2 there are the values of the standard deviation that were generated for the test instances. The set Ω is made up of 65.000 solutions for each one of the problems. In the reported results, the value of γ was fixed at 0.998, because it is the value at which the best results were obtained.

Table 2. Obtained standard deviation for Ω = 65000

Problem	Size	σ
Mt06	6x6	13.54
Mt10	10x10	153.63
La40	15x15	179.43
Yn1	20x20	108.91

The degree of dispersion of the solutions can be appreciated in figure 5. If to the obtained average they add plus or less twice for the σ, then we accept 75% of the solutions generated in the process of SA, that as it can be seen, the distribution pattern which it follows ΔE_k is similar to a Poisson distribution, where the greater number of optimal solutions is when ΔE_k it is approximated to zero (difference of energy between S' y S is minimum), reason why only good solutions when T_0 is minimum (it is approximate to zero) are accepted.

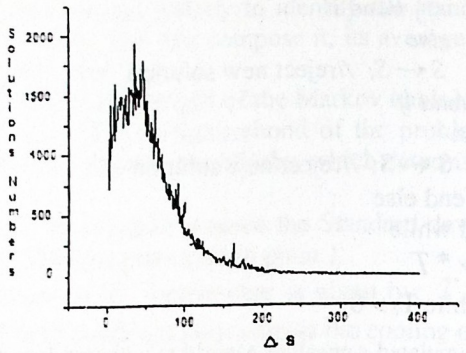

Fig. 5. Dispersion of the generated neighborhood for Yn1 20x20 problem

When limiting 75% of all the solutions within the space search of feasible solutions and tuning the initial temperature $T_0=2*\sigma$, it allows us to accelerate the convergence to a global optimum, when process of SA is controlled.

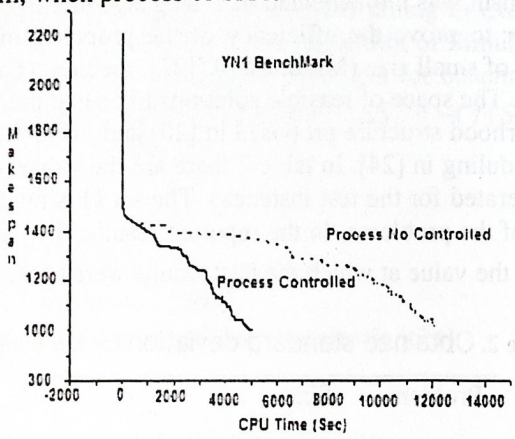

Fig. 6. Performance comparison using the Benchmark Yn1 20x20 problem

Figure 6 shows the results obtained for the Yn1 problem executing the algorithm of Simulated Annealing proposed using σ (controlled process) and to the algorithm of Simulated Annealing typical (without σ) or process no controlled, which initiates with an equal temperature to $2 * \sigma$ and accepting dependent solutions only by the criterion of Metropolis. The proposed algorithm manages to converge to makespan of 930 in 5526 seconds using like initial parameter of temperature $2 * \sigma$, and in addition, verifies that the new the proposed solutions as are within the fixed limits, that is to say, that $\Delta E_k \leq 2 * \sigma$. All the solutions were evaluated under this rule and her acceptance depended on the probability distribution of Boltzmann. The solutions that were outside the fixed limits were rejected automatically. In figure 6 one is in the controlled process that the time of convergence to obtain makespan of 1010 was of 4786 seconds, whereas in the no controlled process, the makespan was of 13960 seconds.

Table 3. Results for 4 instances of test for JSSP, using Controlled and Non-Controlled SA

Controlled SA						Non-Controlled SA		
Problem	Size Jobs x Machines	Best Know	C_{min}	RE	t_{min} Sec	C_{min}	RE	t_{min} Sec
Mt06	6x6	55	55	0.000	1200	57	3.6	5700
Mt10	10x10	930	930	0.000	2600	1010	8.6	7200
La40	15x15	1222	1240	1.4	4786	1278	4.8	12351
Yn1	20x20	886	930	4.9	5526	1010	13.99	13960

Table 3 show the best results of 5 executions for the 4 problems used as test, the data that appear corresponds to the relative error (RE) of the best solution (C_{min}) found and the required time to find this solution t_{min} for the four cases for the controlled and for non controlled process.

In this table it is observed that there is a great difference in the effectiveness of algorithms. The SA controlled in two small problems obtains optimal the global one and for the problems medium and great the relative error is of 1.4 and 4.9 respectively. These results contrast with the ones obtained by the non-controlled SA because the obtained relative error of 3.6, 8.6, 4.8 and 13.99 is much greater.

In order to show the efficiency of the controlled SA with respect to the non controlled SA, in figure 7, are shown the required times to obtain the same makespan in both processes. Analyzing the behavior of both SA one demonstrates that the efficiency of the SA is much better that in the no controlled SA, since little problems to larges problems (36 to 400 operations, for 6x6, 10x10, 15x15, and 20x20 problems).

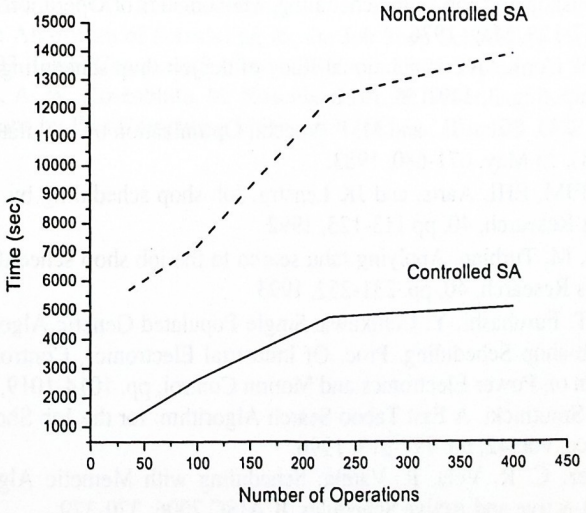

Fig. 7. Performance comparison using Mt06, Mt10, La40 and Yn1 problems

5 Conclusions

The tuning of the parameters used in SA represent the main challenge at the time of using this metaheuristics, The results observed in this document show that it is possible to tune the initial parameters for Simulated Annealing according to the degree of dispersion of the quality of the solutions, by means of the Standard deviation. Depending on the complexity of the problem, the values of the parameters change radically, reason why the Standard deviation turns out suitable like an average one to initialize the parameter of control for each kind of problem. The same Standard deviation is used to discriminate the solutions that will not be evaluated by means of the probability Boltzmann distribution function. The generated solution (S´) from a present solution (S) whose difference ΔS_k are greater to the $2^* \sigma$ is rejected and it is not evaluated as a feasible solution (this outside of range established by the inferior and the superior limit), this in order that $P(e^{(-f(S')-f(S))/T})$ is distributed of more uniform way. In the experimental test was observed that the mechanism, allows us to have a process of controlled SA, with a fast convergence to the global optimum.

References

1. A. Jain S. Meeran.: A State of the Art Review of JOBSHOP Scheduling Techniques. Technical Report. Department of Applied Physics, Electronic and Mechanical Engineering University of Dundee, Dundee, Scotland, K,DD1 4HN,1998.
2. M. R.Garey, D.S Jonson, and R. Shethi,; "The complexity of Flor Shop and Job Shop Scheduling, in Mathematics Of Operation research, vol. 1; No.2 (1976) pag. 117-129.
3. C H. Papadimitriou, K. Steigliths.: Combinatorial Optimization. Algorithms and Complexity. Dover Publications, Inc. 1998. [3] M.R. [3] Garey, D.S. Johnson and R. Sethi, The complexity of Flow shop and Job shop Scheduling. Mathematics of Operations Research, Vol. I, No 2, USA, 117-129, May, 1976.
4. D.Applegate, W. Cook, "A computational study of the job shop scheduling problem", ORSA Journal on Computing, 3, 149-156, 1991.
5. S. Kirkpatrick, S. D. Gelatt Jr., and M. P. Vecchi, Optimization by simulated annealing. Science, 220(4598), 13 May, 671-680, 1983.
6. V. Laarhoven PJM, EHL Aarts, and JK Lenstra. Job shop scheduling by simulated annealing. Operations Research, 40, pp.113-125, 1992.
7. M.D Amico M, M. Turbian. Applying tahu search to the job shop scheduling problem. Annual Operations Research, 40, pp. 231-252, 1993.
8. K. Morikawa, T. Furuhashi, Y. Uchikawa. Single Populated Genetic Algorithm and its Application to Job-shop Scheduling. Proc. Of Industrial Electronics, Control,Instrumentation, and Automation on Power Electronics and Motion Control, pp. 1014-1019, 1992.
9. E. Nowicki, C. Smutnicki. A Fast Taboo Search Algorithm: for the Job Shop Problem. Managemenr Science, vol. 42, pp. 797-813, 1996.
10. M. A. González, C. R. Vela, R. Varela: Scheduling with Memetic Algorithms over the Spaces of Semi-active and Active Schedules. ICAISC 2006: 370-379.
11. M. A. González, M. Sierra, C. R. Vela, R. Varela: Genetic Algorithms Hybridized with Greedy Algorithms and Local Search over the Spaces of Active and Semi-active Schedules. CAEPIA 2005: 231-240.

12.P. Moscato,. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms (Technical Report C3P 826). Pasadena, CA: Caltech Concurrent Computation Program, California Institute of Technology.
13.E.K. Burke and A.J. Smith, A Memetic Algorithm for the Maintenance Scheduling Problem, Proceedings of the ICONIP'97 Conference, Dunedin, New Zealand, 24-28 November 1997, (published by Springer), pp. 469-474.
14.P. J. Zalzala, and Flemming. Zalzala, A.M.S. (Ali M.S.),ed., Genetic algorithms in engineering systems /Edited by A.M.S. Institution of Electrical Engineers, London, 1997.
15.M. A. Cruz-Chávez, J. Frausto-Solís, D. J. Romero, Experimental Analysis in Simulated Annealing to Scheduling Problems when Upper Bounds are used, Transactions on Information Science and Applications, ISSN: 1790-0832, Vol. 2, Issue 5, pp. 581 - 586, 2005.
16.J. E. Beasley. OR-Library: Distributing test problems by electronic mail. Journal of the Operational Research Society,Vol. 41. No. 11, 1069-1072, 1990. Last update 2003.
17.Roy and Sussman, Les problemes d'ordonnancement avec contraintes disjonctives, Note D.S. no 9 bis, SEMA, Paris, France, December 1964.
18.M. Kolonko and M.T. Tran; Convergente of Simulated Annealing with Feedback Temperatura Schedule. Problem in The Engineeering and Informational Sciences, 11, 279-304, 1997.
19.M. G. Kendall, A. Stuart, The advanced Theory of Statistics, Vol.1 London: Charles Griffin 1958.
20.M. A. Cruz-Chávez, J.Frausto-Solís, J.R. Cora-Mora, Experimental Analysis of a Neighborhood Generation Mechanism Applied to Scheduling Problems, Proceeding of CERMA2006, IEEE-Computer Society, ISBN 0-7695-2569-7, pp 226-229, 26-29 September, México, 2006.
21.H. Fisher, and G.L. Thompson, Probabilistic Learning Combinations of Local Job Shop Scheduling Rules, In J.F. Muth and G.L. Thompson (Eds.), Industrial Scheduling, Prentice Hall. Englewood Cliffs. New Jersey 225-251. 1993.
22.S. Lawrence, Resource Constrained Project Scheduling: An Experimental Insvestigation of Heuristic Scheduling Techniques (Suplement). Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburg, Pensylvania, 1984.
23.T. Yamada R and Nakano, A Genetic Algorithm Aplicable to Large-Scale Job-Shop Instante Solving from nature 2, North-Holland, Ámsterdam 281-290, 1992.
24. M. A. Cruz-Chávez, M.G. Martinez-Rangel, J. A. Hernandez-Perez, J. C.-Zavala-Diaz, O. Díaz-Parra, An Algorithm of Scheduling for the Job Shop Scheduling Problem, Proceeding of CERMA2007, IEEE-Computer Society, ISBN, pp, 25-28 September, México, 2007.
25. N. Metropolis, A. W. Rosenbluth, M. Rosenbluth, A. H. Teller, and E. Teller. "Equation of State Calculations by Fast Computing Machines." *J. Chem. Phys.* **21**, 1087-1092, 1953.

Natural Language Processing