

Using Simulated Annealing with a Neighborhood Heuristic for Roll Cutting Optimization

Aplicando Recocido Simulado con Heurística de Vecindad a la Optimización de Cortes en Rollos

Horacio Martínez Alfaro and Manuel Valenzuela Rendón

Centro de Computación Inteligente y Robótica

Tecnológico de Monterrey

Monterrey, N.L. 64849 México

Ph. +52 81.8328.4381 F. +52 81.8328.4189

hma@itesm.mx; valenzuela@itesm.mx

Article received on July 15, 2008; accepted on April 03, 2009

Abstract

This article presents the use of the Simulated Annealing algorithm with a heuristic to solve the waste minimization problem in roll cutting programming, in this case, paper. Client orders, which vary in weight, width, and external and internal diameter, are fully satisfied. Several tests were performed with real data from a paper company in which an average of 30% waste reduction and 100% reduction in production to inventory are obtained compare to the previous procedure.

Keywords: Simulated Annealing, optimization, heuristics, cutting, paper rolls.

Resumen

Este artículo presenta el uso del algoritmo de Recocido Simulado con una heurística para resolver el problema de minimización de desperdicio en la programación de cortes en rollos, en este caso de papel. Las órdenes de los clients, que varían en peso, ancho, y diámetro interno y externo, se satisfacen al 100%. Se realizan varias pruebas con datos reales de una compañía en donde en promedio se obtiene un ahorro del 30% de desperdicio y 100% de producción a inventario comparado con el procedimiento anterior.

Palabras clave: Recocido simulado, optimización, heurísticas, corte, rollos de papel.

1 Introduction

Paper industry has a great product variety which can be classified in four manufacturing segments: packaging, hygienic, writing and printing, and specialties. Every segment use paper rolls as basic input for their processes. These rolls are called *master rolls* and they vary in their internal diameter or *center*, D_i , external diameter or simply *diameter*, D_e , and *width*, w , depending on the paper type and the process it will undertake. Figure-1(a) shows these characteristics.

Although there are several paper manufacturing segments for paper, in a same segment there are several paper types which basic difference is density, G .

Client orders received by a paper manufacturing company are classified by paper type, then by their diameter, and then by their center. Although, each order varies in width (cm) and weight (kg), the paper manufacturing company delivers to each client a certain number of rolls (with the diameter, center, and width requested) that satisfy the order.

Once the orders are classified into groups by paper type (paper density), diameter, and center, each group is processed in a cutting machine with a maximum fixed width, W_{MR} combining the order widths (w_i) to satisfy each order weight having as objective to minimize the unused width of the master roll (M in Figure-1(b)). The previous described minimization problem is a combinatorial one since we are interested in finding the best widths combination to cut in a master roll satisfying all orders. All cutting combinations are grouped in a production cutting schedule (PCS), as shown in Figure 1(b), having a total weight or, equivalently, a total number of rolls to manufacture for each combination or item in the cutting schedule.

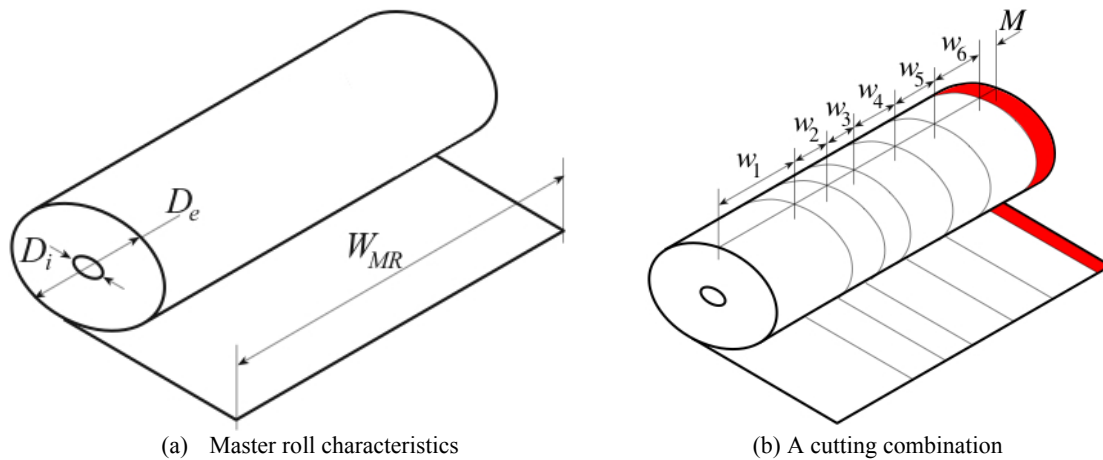


Fig. 1. Master roll characteristics

These are the characteristics/steps of the procedure performed in the paper manufacturing company where this research was carried out:

- Two or more people using a spreadsheet type application, i.e. MS Excel, perform the task.
- The process takes about 3 hours to obtain an acceptable total waste, which in this company was less than 10% of the total production weight, for four different width cutting machines using a very simple approximation iterative scheme.
- The smallest total waste achieved with this procedure is due to the production of some roll widths that go directly to inventory, i.e., certain commonly used roll widths are assigned to a “virtual” client’s order and created “on the fly.”

By performing a deep analysis of the problem, we found the main problem was in this last step of the procedure since the person in charge of generating the cutting schedule in the company was generating about 30% more of the total waste as inventory production. This means that if the total waste was 9,000 ton, additionally 12,000 ton were to inventory.

The process to obtain a cutting schedule has other characteristics:

- The required time of the procedure limits the number of times (two or three) it can be performed before deciding which cutting schedule is going to be used for production.
- The process considers a $\pm 10\%$ of the order weight (rarely, more) to help in generating the cutting schedule.
- The same person that generates the cutting schedule decides the order of each combination before sending it for production.
- Cutting blade movement is not considered in the process and it is performed by the cutting machine operator.

This article shows the development of an application that:

- automates a new generating procedure to obtain a cutting schedule,
- the objective function is total waste width, and
- eliminates the need of generating cuts to inventory

by using the simulated annealing algorithm (Laarhoven & Aarts, 1987; Malhotra, Oliver, & Tu, 1991; Martínez-Alfaro & Valenzuela-Rendón, 2004; Martínez-Alfaro & Flugrad, 1994; Martínez-Alfaro & Ulloa-Pérez, 1996;

Martínez-Alfaro & Flores-Terán, 1998; Martínez-Alfaro, Valdez & Ortega, 1998a, 1998b) and a very simple heuristic that simplifies work by Martínez-Alfaro & Valenzuela-Rendón-(2004) in the generation of a new state/solution for the simulated annealing algorithm.

2 Methodology

Knowing that the paper manufacturing company delivers paper rolls, first we compute the number of rolls that are equivalent to the each weight order. Master roll weight is given by the following equation (Martínez-Alfaro & Valenzuela-Rendón, 2004):

$$p_r = (D_e^2 - D_i^2)\pi W_{MR} G \tag{1}$$

where p_r is the master roll weight (in kg), D_e is the external diameter (in m), D_i is the center (internal diameter in m), W_{MR} is the master roll width (in m), and G is the paper density (in kg/m³). With this, we can compute the number of rolls for an order:

$$n_r = \left\lceil \frac{p_i/w_i}{p_r/W_{MR}} \right\rceil = \left\lceil \frac{p_i W_{MR}}{p_r w_i} \right\rceil \tag{2}$$

where n_r is the number of rolls for the i -th order, p_i (in kg) is the weight for the i -th order, w_i (in m) is the width of the i -th order, p_r (in kg) is the master roll weight, and W_{MR} (in m) is the master roll width.

2.1 Waste Optimization

The initial cutting schedule procedure is performed by selecting a random order width w_i from the order set to include in a combination, and it will continue until the sum of the included widths is greater or equal to the master roll width or the number of them is equal or greater to the maximum number of cuts, C_b (Figure-1(b) shows a six-cut combination), allowed for that cutting machine. Once one of these conditions is satisfied, another cutting combination is initiated. The process continues until all widths in the order set are included in a combination. Table-1 shows a set of orders where R is the equivalent number of rolls for the required width. This set of orders were processed on a cutting machine of $W=202.5$.

Table 1. Cutting orders (widths in cm)

ID	Width	R	ID	Width	R	ID	Width	R	ID	Width	R
0	55.0	6	10	51.0	105	20	68.5	47	30	64.0	37
1	145.0	6	11	70.0	8	21	79.0	8	31	181.0	5
2	50.0	8	12	70.0	8	22	69.0	8	32	181.0	18
3	150.0	2	13	69.0	5	23	79.0	8	33	201.0	14
4	135.0	6	14	72.0	5	24	83.0	7	34	195.6	16
5	80.0	11	15	59.5	16	25	91.5	23	35	200.0	117
6	105.0	6	16	87.0	4	26	85.0	11	36	201.0	63
7	90.0	5	17	87.0	6	27	81.0	30	37	181.0	35
8	100.0	5	18	64.5	24	28	24.0	12			
9	55.0	24	19	83.0	4	29	100.0	13			

Table 2 shows a list of cutting combinations or cutting schedule for a cutting machine that allows a maximum of $C_b=3$ cuts. wp_k is the total width for that combination, $w_i, i=1, \dots, C_b$ are the widths included for that combination. Figure-2 shows a plot of the combinations and its totals widths. Once the cutting schedule is generated, the waste for each combination is computed as follows: first, compute the unused width part of the master roll:

$$M_k = W_{MR} - wp_k \tag{3}$$

with

$$wp_k = \sum_{i=1}^{C_b} w_i \tag{4}$$

and then in weight

$$m_k = M_k \frac{p_r}{W_{MR}} \tag{5}$$

Table-2. A set of possible cutting combinations for a three-cut machine using set of orders shown in Table-1

<i>k</i>	<i>wp</i>	<i>w</i> ₁	<i>w</i> ₂	<i>w</i> ₃
1	100.0	100.0	100.0	
2	100.0	100.0	100.0	
3	100.0	100.0	100.0	
4	100.0	100.0	100.0	
5	181.0	181.0		
6	100.0	100.0	100.0	
7	100.0	100.0	100.0	
8	100.0	100.0	100.0	
9	201.0	201.0		
10	100.0	100.0	100.0	
11	100.0	100.0	100.0	
12	200.0	200.0		
13	100.0	100.0	100.0	
14	79.0	79.0	68.5	55.0
15	200.0	200.0		
16	100.0	100.0	100.0	
17	181.0	181.0		
18	79.0	79.0	79.0	
19	83.0	83.0	64.5	55.0
20	69.0	69.0	69.0	64.5

<i>k</i>	<i>wp</i>	<i>w</i> ₁	<i>w</i> ₂	<i>w</i> ₃
21	200.0	200.0		
22	181.0	100.0	81.0	
23	200.5	81.0	64.5	
24	195.0	81.0	64.0	
25	202.5	83.0	68.5	
26	200.0	200.0		
27	201.0	201.0		
28	181.0	181.0		
29	200.0	200.0		
30	181.0	181.0		
31	200.0	100.0	100.0	
32	201.0	201.0		
33	200.5	91.5	85.0	
34	201.0	201.0		55.0
35	200.0	200.0		
36	200.0	200.0		
37	181.0	181.0		
38	201.0	201.0		
39	201.0	201.0		55.0
40	200.0	200.0		64.5

where m_k is the waste for the k -th cutting combination, M_k is the master roll unused part for combination k , p_r is the master roll weight, and W_{MR} is the master roll width. The total waste M_T is the sum of each combination waste m_k :

$$M_T = \sum_k m_k \tag{6}$$

which is the objective function for our optimization problem.

Optimization is performed only with order widths which are feasible to combine, i.e., if there is an order width that satisfies:

$$w_i + w_{\min} > W_{MR} \tag{7}$$

where w_{\min} is the smallest order width, then, it is not considered for optimization since the waste generated by these orders is fixed with or without optimization. Once an initial cutting schedule is generated, the Simulated Annealing algorithm is used to perform the optimization.

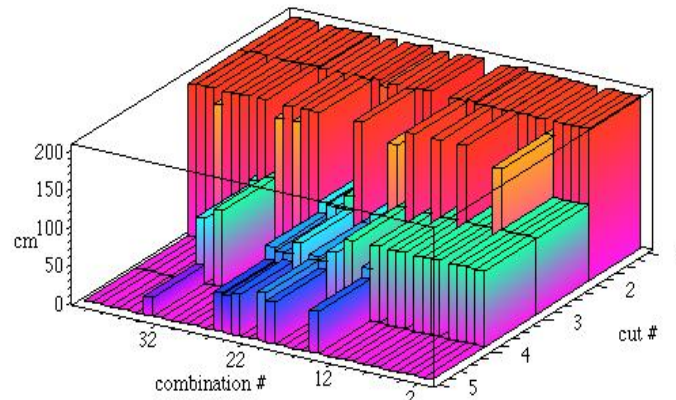


Fig. 2. A set of possible cutting combinations for a three-cut machine using set of orders shown in Table-1

The following section describes how to generate a new cutting schedule from this initial one by defining a neighborhood part. Simulated annealing algorithm has been successfully used in robotics (Martínez-Alfaro & Flugrad, 1994; Martínez-Alfaro & Ulloa-Pérez, 1996; Martínez-Alfaro & Valenzuela-Rendón, 2004) and scheduling (Martínez-Alfaro & Flores-Terán, 1998; Martínez-Alfaro et al., 1998a, 1998b) applications. The following section the algorithm is described and its requirements for implementation in our problem.

2.2 Simulated Annealing Algorithm

Simulated annealing is basically an iterative improvement strategy augmented by a criterion for occasionally accepting higher cost configurations (Rutenbar, 1989; Malhotra et al., 1991). Given a cost function $C(\mathbf{z})$ (analog to energy) and an initial solution or state \mathbf{z}_0 , the iterative improvement approach seeks to improve the current solution by randomly perturbing \mathbf{z}_0 . The Metropolis algorithm (Malhotra et al., 1991) was used for acceptance/rejection of the new state \mathbf{z}' at a given temperature T , i.e.,

- randomly perturb \mathbf{z} to obtain \mathbf{z}' , and calculate the corresponding change in cost $\delta C = \mathbf{z}' - \mathbf{z}$
- if $\delta C < 0$, accept the state
- if $\delta C > 0$, accept the state with probability

$$P(\delta C) = \exp(-\delta C / T) \quad (8)$$

this represents the acceptance-rejection loop or Markov chain of the SA algorithm. The acceptance criterion is implemented by generating a random number, $\rho \in [0, 1]$ and comparing it to $P(\delta C)$; if $\rho < P(\delta C)$, then the new state is accepted. The outer loop of the algorithm is referred to as the cooling schedule, and specifies the equation by which the temperature is decreased. The algorithm terminates when the cost function remains approximately unchanged, i.e., for n_{no} consecutive outer loop iterations.

Any implementation of simulated annealing generally requires four components:

1. **Problem configuration** - domain over which the solution will be sought.
2. **Neighborhood definition** - which governs the nature and magnitude of allowable perturbations.
3. **Cost function.**
4. **Cooling schedule** - which controls both the rate of temperature decrement and the number of inner loop iterations.

The domain for our problem is the set of cutting combinations. The objective or cost function is describe in the previous Section. The neighborhood function used for this implementation is the same used by Martínez-Alfaro & Flugrad-(1994) with the addition of a very simple heuristic.

Neighboring heuristic

Two cutting combinations, c_{n1} and c_{n2} , are randomly selected with the distance between them cooled, i.e. distance between them is decreased as the temperature decreases. Once selected the two cutting combinations, the heuristic consist of testing widths exchanges between them trying to decrease the average unused part of the master roll for both cutting combinations. This implies testing an exchange between width w_i in cutting combination c_{n1} with width- w_j in cutting combination- c_{n2} as shown in Figure-3. Cutting combinations may have at most C_b widths and, therefore, the exchange could be between an empty width in one combination and a non empty width in the other combination. Trivial exchanges (empty widths or equal widths in both combinations) are not performed.

The allowable perturbations are reduced by the following limiting function

$$\varepsilon = \varepsilon_{\max} \frac{\log(T - T_f)}{\log(T_0 - T_f)} \quad (9)$$

where ε_{\max} is an input parameter and specifies the maximum distance between two elements in a list, and T , T_0 , T_f are the current, initial, and final temperatures, respectively.

The cooling schedule in this implementation is the same hybrid one introduced by Martínez-Alfaro & Flugrad-(1994) in which both the temperature and the inner loop criterion vary continuously through the annealing process (Elperin,-1988). The outer loop behaves nominally as a constant decrement factor:

$$T_{i+1} = \alpha T_i \quad (10)$$

where $\alpha = 0.9$ for this paper. The temperature throughout the inner loop is allowed to vary proportionally with the current optimal value of the cost function. So, denoting the inner loop index as j , the temperature is modified when a state is accepted, i.e.,

$$N_{\text{in}} = N_{\text{dof}} \left[2 + 8 \left(1 - \frac{\log(T - T_f)}{\log(T_0 - T_f)} \right) \right] \quad (11)$$

where N_{dof} is the number of degrees of freedom of the system.

The initial temperature must be chosen such that the system has sufficient energy to visit the entire solution space. The system is sufficiently melted if a large percentage, i.e. 80%, of state transitions are accepted. If the initial guess for the temperature yields less than this percentage, T_0 can be scaled linearly and the process repeated. The algorithm will proceed to a reasonable solution when there is excessive energy; it is simply less computationally efficient. Besides the stopping criterion mentioned above, which indicates convergence to a global minimum, the algorithm is also terminated by setting a final temperature given by:

$$T_f = \alpha N_{\text{out}} T_0 \quad (12)$$

where N_{out} is the number of outer loop iterations and is given as data to our problem.

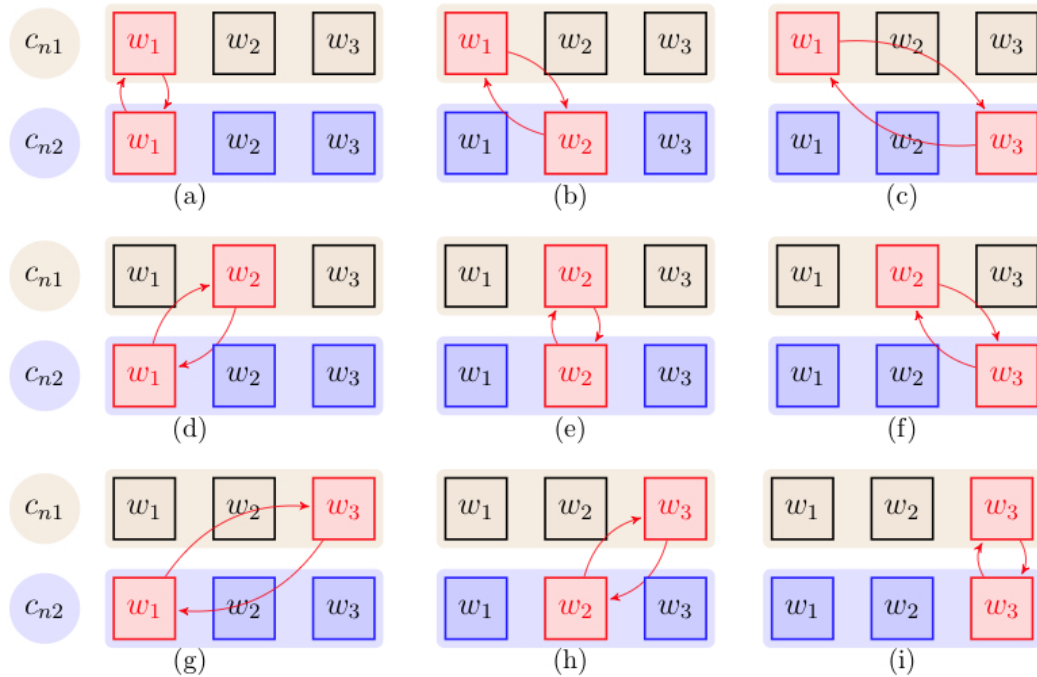


Fig. 3. Neighboring heuristic for three cut n_1 and n_2 combinations

3 Results

The source language for our implementation is in Python for Windows in a Centrino @1.69 GHz with 1 GB RAM. All tests were performed with real data given by the paper manufacturing company in which this research was carried out.

The initial data is shown in Table 1 and for a machine of 2.025 m maximum width and 3 cutting blade machine.

The present cutting schedule procedure generated a total waste of 17,628.5 kg and 16,742.0 kg to inventory and it took almost three hours. The system using the simulated annealing algorithm generated a total waste of 7,244.9 kg without the need of production cuts to inventory in about 40 seconds. The SA system obtained 430 cutting combinations. The sum of the required widths was 84,918.6-cm and since the master roll width was 202.5-cm, we would have needed 420 master rolls. The SA system generated 10 additional cutting combinations to the minimum number of them. The resulting waste is about 59% less than the manual procedure and considering production to inventory also as waste, it is about 79% smaller. The total production for this test was 211,317.3 kg. This means that the total waste, without inventory, compare to the total production decreased from 8.34% to 3.43%. If we include the inventory as waste, it went down from 16.26% to 3.43%. Now, if the paper production cost of 1,000 kg is \$400 USD, the SA system generated a total savings amount of \$4,154 USD and considering inventory it was \$10,850 USD. If we would know each machine production rate, the previous savings would be equivalent to \$2,549-USD/day and considering the inventory to \$6,659 USD/day.

Many tests were performed to test for robustness varying some parameters of the SA algorithm like the number of Markov chain ranging from 400 to 1400 and the initial temperature ranging from 400 to 4000 verifying total waste, total number of cutting combinations, the number of states, and the running time. The results showed that the total waste (which was part of the cost function) had a standard deviation which represented 0.01% of the mean; the number of cutting combinations had a standard deviation which represented 0.16% of the mean; and the number of

total states had a standard deviation of 15.9% of the mean which was very similar to the total running time which had a standard deviation of 15.39% of the mean.

The other of the SA algorithm also tested was the percentage of acceptance rate ranging from 80% to 70% obtaining practically the same results in the total waste and the number of cutting combinations; the number of total states and the running time decreased its standard deviation to 8.96% and 9.92% of their respectively mean.

Many other test were performed with orders for four different cutting machines, different maximum width or master roll width, and different number of cutting blades, for a period of four months obtaining verifying the previous tests for robustness. The average savings in total waste was 680,000 kg using the SA system which represented a 29% savings compared to the manual procedure. This represented a \$477,756 USD in savings. However, production that went directly to inventory was 12,000,000-kg which represented \$4.8 million USD. If we consider that the SA system does not generate production to inventory (unless it is specified), the total savings for four months with four cutting machines was more than five million USD.

4 Conclusions

All results were validated by people in charge of obtaining the cutting schedule at the paper manufacturing company. Savings generated by the SA system allows not only an optimization of the total waste but also in the elimination of production to inventory. The production to inventory resulted as the actual problem the company had. Although some inventory production could be sold, this production was generated to minimize the total waste during the generation of the cutting schedule. By using the system, it allows people from the company to spend more time in decision making, problem analysis and/or urgent orders since to obtain a cutting schedule takes less than a minute.

The simple neighboring heuristic allowed a faster and better solution compared to the previous results reported in Martínez-Alfaro & Valenzuela-Rendón (2004). Execution time was reduced to 1/10 and for higher number of orders, even more (1/50); and since the simulated annealing algorithm could explore more solutions of the domain, results are slightly better (absolute values).

The authors are currently working in a global optimal cutting schedule generation system that will allow the use of several cutting machines (different width) to process an order set to generate the cutting schedule.

References

1. **Burkard, R., & Rendl, F.** (1984). A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European J. of Oper. Res.*(17), 169–174.
2. **Elperin, T.** (1988). Monte carlo structural optimization in discrete variables with annealing algorithm. *Int. J. for Numerical Methods in Eng.*(26), 815–821.
3. **Gelfand, S. B., & Mitter, S. K.** (1985, December). Analysis of simulated annealing for optimization. In *Proc. 24th Conf. on Decision and Control* (pp. 779–786). Ft. Lauderdale.
4. **Gielen, G. G. E., Walscharts, H. C. C., & Sansen, W. M. C.** (1990, June). Analog circuit design optimization based on symbolic simulation and simulated annealing. *IEEE Journal of Solid-State Circuits* (25), 707–713.
5. **Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P.** (1983). Optimization by simulated annealing. *Science*, 220, 671–680.
6. **Kirkpatrick, S., & Toulouse, G.** (1985). Configuration space analysis of travelling salesman problems. *J. Physique* (46), 1277–1292.
7. **Laarhoven, P. J. M. van, & Aarts, E. H. L.** (1987). *Simulated annealing: Theory and applications*. Dordrecht, Holland: D. Reidel Pub. Co.
8. **Malhotra, A., Oliver, J. H., & Tu, W.** (1991). Synthesis of spatially and intrinsically constrained curves using simulated annealing. In *Asme Design Engineering Technical Conferences: Advances in Design Automation* (pp. 145–155).
9. **Martínez-Alfaro, H., & Flores-Terán, G.** (1998). Solving the classroom assignment problem with simulated annealing. In *Ieee Int. Conf. on Systems, Man, & Cybernetics*. San Diego, CA.

10. **Martínez -Alfaro, H., & Flugrad, D. R.** (1994, September). Collision-free path planning of robots and/or AGVs using B-splines and simulated annealing. In Asme 23rd Biennial Mechanisms Conference. Minneapolis.
11. **Martínez -Alfaro, H., & Ulloa-Pérez, A.** (1996). Computing near optimal paths in C-space using simulated annealing. In Asme Design Engineering Technical Conference/24th Biennial Mechanisms Conference. Irvine, CA.
12. **Martínez -Alfaro, H., Valdez, H., & Ortega, J.** (1998a). Linkage synthesis of a four-bar mechanism for n precision points using simulated annealing. In Asme Design Engineering Technical Conferences/25th Biennial Mecanisms and Robotics Conference. Atlanta.
13. **Martínez -Alfaro, H., Valdez, H., & Ortega, J.** (1998b). Using simulated annealing to minimize operational costs in the steel making industry. In Ieee Int. Conf. on Systems, Man, & Cybernetics. San Diego.
14. **Martínez -Alfaro, H., & Valenzuela-Rendón, M.** (2004). Using simulated annealing for paper cutting optimization. Lecture Notes In Artificial Intel ligenca Series: Lecture Notes In Computer Science, 11–20.
15. **Metropolis, N., Rosenbluth, A., Rosenbluth, M., & Teller, A.** (1953). Equations of state calculations by fast computing machines. J. of Chemical Physics (21), 1087–1091.
16. **Rutenbar, R.** (1989, January). Simulated annealing algorithms: An overview. IEEE Circuits and Devices, 19–26.
17. **Vanderbilt, D., & Louie, S.** (1984). A monte carlo simulated annealing approach to optmization over continuous variables. J. Comput. Phys.(36), 259–271.



Horacio Martínez Alfaro is with the Center for Intelligent Computing and Robotics at the Tecnológico de Monterrey (ITESM), campus Monterrey. He participates in the Research Chair on Evolutionary Optimization. He received a Postdoctoral certification from Tulane University (2009), a Ph.D. in Mechanical Engineering from Iowa State University (1993), the M.Sc. in Automatic Control from the ITESM, campus Monterrey (1986) and B.Sc. in Mechanical Engineering from the I.T. Saltillo (1984). His research interests include nature inspired techniques applied to optimization and forecasting in logistics, finance, automatic control, and dynamic systems.



Manuel Valenzuela Rendón is with the Center for Intelligent Computing and Robotics at the Tecnológico de Monterrey (ITESM), campus Monterrey. He holds the Research Chair on Evolutionary Optimization. He received a Postdoctoral certification from Tulane University (2009), a Ph.D. in Electrical Engineering from the University of Alabama (1981), the M.Sc. in Computer Science from the ITESM, campus Cuernavaca (1984) and B.Sc. in Electrical Engineering and Communications (1981) from the ITESM, campus Monterrey (1981). He is a member of the National Research System (SNI) of Mexico. His research interests include evolutionary computation, neural networks, and fuzzy logic applied to optimization and forecasting in logistics, finance, and automatic control.