Application of Simulated Annealing to a center-pivots scheduling problem.

Emerson Stiilpen Batista*, Heleno do Nascimento Santos* and Brauliro Gonçalves Leal†
*Federal University of Viçosa/Computer Science Dept.
36.570-000, Viçosa-MG, Brazil
Email: esb@dpi.ufv.br, hns@dpi.ufv.br
†Federal University of Viçosa/Agricultural Engineering Dept.
36.570-000, Viçosa-MG, Brazil
Email: brauliro@ufv.br

***Abstract*— This paper presents a way to solve a center-pivots scaling problem. A center-pivot is a large area irrigation system, which uses much water resource during its working. Its goal is to create a working scale to the center-pivots that act using the same water resource as a solution to avoid possible environmental damages, without compromising its efficiency. Due to the complexity of the problem, the use of metaheuristic Simulated Annealing was adopted in the searches of good solutions in a very small period of time. The application of the Simulated Annealing was created in a very simple manner, with the option of utilizing two different switch movements, to the generation of new neighbor solutions. The obtained results, with the algorithm, has generated excellent solutions which attended to the initial expectation.**

***Keywords*— Combinatorial Optimization; Scheduling; Simulated Annealing; Center-Pivot.**

I. Introduction

The center-pivot scheduling problem consists of the creation of service scales to center pivots during a period of time, by respecting operational and environment problems, and seeking to reach certain objectives. This schedule must find solutions that minimize the electric energy consume of the problem, in respect all restrictions that exist. It'll be discussed in this paper later.

According to [6], the center-pivot is an overhead irrigation very used in great planted areas and that it consists, basically, of a metallic piping, where sprinkers are installed. This piping receives the water under pressure of a central device, called point of the pivot, and it leans on in triangular metallic towers, mounted on tire wheels, as shown in the Fig. 1. The towers move continually, worked by electric or hydraulic devices, and it describe concentric movements around of the point of the pivot. The movement of the last tower begins a progress reaction in chain in a progressive way for the center.

In the aerial vision, shown by Fig. 2, it is possible to verify that the more large is the ray of the pivot, larger is the irrigable area and, consequently, larger is its profitability and its water consumption. If we consider the irrigation equipment as a type of high scale, and there isn't an effective irrigation way in this context, a great volume of water is necessary to the center-pivot working [26].

This way, if there is an area that possesses many pivots making use of a same common hydric resource (a river, for instance), it is possible to happen serious damages to the environment if all these pivots are working simultaneously without any control type. This risk generates the need to control an entire group of center-pivots, with the construction of an operation scale for all, so that the plantations and the ecosystem are not prejudiced [16].

Each center-pivot assists only a fixed customer. The customers are irrigable areas of plantations that have, every day, a different number of irrigation time through center-pivot (D_i). These hours are stipulated by the responsible agricultural engineer by each customer, with the objective of assisting the needs of the plantation. The number of each customer's hours can vary due to several external factors as the climate, the irrigated culture, the soil type, among other [26].

For we always work the time of irrigation in daily hours rates, all built solution is divided in 24 time windows, represented by the index j, that correspond at the hours of the day. This way, every center-pivot, represented by the index i, will have its work period distributed among the 24 time windows that composes the scale daily.

The main variable to be controlled in the problem is the volume of water used in each time window. Each center-pivot has a fixed water volume necessary for its operation (Q_i). So, the volumes sum of all the pivots that are working, in a certain time window, they

Fig. 1. Structures that compose the arm of a center-pivot.

Fig. 2. Aerial Vision of the area of covering of a center-pivot.

should always be smaller or equal to the water volume that is allowed to be removed of the hydric resource (Flow). This restriction is fundamental to avoid risks of environmental damages to the ecosystem.

The problem to stagger center-pivots has as goal the minimization of the energy consumption in the whole problem (C_{ij}); it can be obtained due to the cost of the electric power tariff, which is 60% cheaper in nocturnal periods than in diurnal periods.

The Problem of Stagger of center-pivots consists of the creation of scales of services to center-pivots for a certain period, respecting operational and environmental restrictions, and trying to reach certain objectives. It is a problem studied not enough and this study can avoid great financial and environmental damages in densely populated areas by center-pivots.

According to [18], the Whole Programming [44] has been used preferentially as a method of solution of stagger. However, the complexity of these problems usually requests the development of heuristic techniques [18] or the use of goal-heuristicses, like Simulated Annealing (SA) [36], Genetic Algorithms [1] and other goal-heuristicses [2].

The heuristicses and goal-heuristicses to solve this problem type are increasing significantly, because these methods, in spite of they guarantee not the obtaining of great solutions, they allow the insert of countless restrictions in a soft way and with fast results.

This work presents a simple and effective alternative to solve the problem in subject. A general mathematical model is proposed to represent the problem, and the goal-heuristics Simulated Annealing is used to treat it, in other words, to generate scales to be financially economic, and that satisfy to the customers' demand, and assist their needs and respect the limits of the hydric resources.

In the next section the developed modelling will be discussed, which represents the problem of stagger of center-pivots. In the Section 3 it will be discussed with larger details the Simulated Annealing heuristics proposed. The computing results accomplished are described in the Section 4. Finally, the conclusions are presented.

II. Data Modeling Proposition

The model proposed in this paper was created with the objective to validate all of the restrictions identified and, consequently, to be the sure we have the domain of all of the pertinent information about the problem.

The data modeling validation was made by using the LINGO software, which uses Linear Program (LP) and the Branch-and-Bound (B&B) algorithm. According to [22] LINGO it is a simple tool, which uses the power of linear and nonlinear math programming to formulate big problems, solve and analyze them.

Due to the dimensions of the question, which make it complex, it was decided the importance of adopting methods to find good solutions, demanding lower computational dedication and quicker processing time achieved by LP and B&B. However, with a reduced number of variables, it was reached very satisfactory results, making possible the validation of the model.

In this model, the first decision is the existence of n clients (center-pivots) which will have their working hours distributed in 24 time windows. To represent the problem, we have the following variables:

- n: represents the total number of center-pivots in the problem;
- i: represents each center-pivot;
- j: represents each time window;
- C_{ij}: represents the expense of the client i, by the pivot i, in the time window j;
- X_{ij}: it's a binary variable that, when it assumes the value 1, indicates that the pivot is working in the time window j;
- D_i: represents the number of water hours demanded by the client i;
- Q_i: represents the quantity of water used in the pivot.

The binary matrix X is the local where the scheduling is built. The time window in the question are shown in this matrix as being each line represented a different client i and each column j a different period of time. So, each position X_{ij} represents the window j of the client i, indicating if i can work in j, if it is opened, in other words, if X_{ij} = 1, or not, if it's closed, i.e., if X_{ij} = 0. Moreover, the result of the objective function and all restrictions of the model depend on the value restrained in X.

Each client must have the sum of opened windows equal to his/her time demand, i. e., the sum of time windows of each center pivot must be identical to each pivot necessity. So, this constraint guarantees the whole clients satisfaction.

Each time window must possess the sum of all needs of every opened pivots, equal or lower than the water volume needed, i. e., the total of water consumed in each time window must be equal or lower than the limit pre-established. This constraint guarantees water resources to each pivot without wasting.

The objective function wants to minimize the costs of all center-pivots working together, by adding the costs of attending each client in each opened window. There is the tendency of put center-pivots to work in night shift. This happens because the cost of electricity is 60% lower than diurnal period.

And from this, it was developed the following mathematic model:

Minimize:

$$\sum_i \sum_j C_{ij} X_{ij} \quad (1)$$

Subject to:

$$\sum_j X_{ij} = D_i, \forall i. \quad (2)$$

$$\sum_i Q_i X_{ij} \leq water, \forall j. \quad (3)$$

$$X_{ij} \in \{0, 1\}, \forall i, j. \quad (4)$$

$$i = 1, 2, ..., n. \quad (5)$$

$$j = 1, 2, ..., 24. \quad (6)$$

In the model, the function objective is represented by the equation (1). The restriction (2) guarantees the customer's total service and the restriction (3) guarantees that the hydric resource will be used inside of the stipulated limits. The restriction (4) indicates that the matrix X should just be composed by the values 0 (zero) and 1 (one). The restriction (5) represents the limits of the i index , that represents the center-pivots of the problem and it can assume any whole value between 1 and n (total number of center-pivots). The restriction (6) represents the limits of the j index, that represents the time windows and can assume any whole value in the interval from 1 to 24.

III. Simulated Annealing Application

The Simulated Annealing (SA) is an algorithm of local search capable to accept worse solutions than the existent to flee of great places. It was proposed firstly by [27], and it is based in an analogy with the thermodynamics, when it simulates the cooling of a group of warmed atoms [34]. To the SA use, a method should be defined for the generation of an initial solution S, a method for generation of the S' neighbor solutions (neighborhood structure), and a function objective $f(S)$ to be optimized.

The pseudo-code of Simulated Annealing's implemented algorithm is presented in the sequence, where S is the initial solution and S' is the neighboring solution of S. The implementation uses, as criterion of closing of the algorithm, an inferior limit (T_c) for the variable Temperature (T), that suffers constant decrement during the execution of the algorithm. So, when T assumes a smaller value or equal to T_c, the algorithm is contained.

Procedure Simulated Annealing(T_0, T_c, α):
    (T_0 is the initial temperature)
    (T_c is the final temperature)
    (α is the factor of decrement of the temperature)
    Begin
    $T \leftarrow T_0$;
    $S \leftarrow InitialSolution()$;
    While $T > T_c$ **do**
        To generate S' of S;
        $\Delta = f(S') - f(S)$;
        If $\Delta > 0$ **then**
            $S \leftarrow S'$;
        else
            To generate $r \in Unif[0,1]$
            If $r < e^{\frac{-(f(S')-f(S))}{T}}$ **then**
                $S \leftarrow S'$;
            end-if
        end-if
        $T \leftarrow T \times \alpha$;
    End-while
    Return S
End Simulated Annealing

A. Initial Solution

The method utilized to generate an initial solution goes by the principle of a greedy algorithm, and it create an orderly list of customers in a decreasing way, in agreement with the number of demanded hours of each center-pivot, and manipulate it one by one. So, it always accomplishes the choice that seems to be the best in the moment; it make a local great choice, in the hope that this light choice to the global great solution [13]. The pseudo-code of this method is presented in the sequence.

Procedure InitialSolution():
    Begin
    To create in the L list with empty clients;
    To add all clients of the question to list L;
    To order the L list in the decrescent order,
    by the number of irrigation working hours;
    While L is not empty **do**
        While the irrigation hours of first
        element are not attended **do**
            Open time window with lower;
        End-while
        Take out the first element of the list L;
    End-while
    Return S
End InitialSolution

By using the ordered list, the first element of the list is selected and its time windows start to be opened, always seeking to open time windows with the lowest water use at the moment. The process of window opening finishes when the number of opened windows is equal to the number of irrigation hours demanded by the client. In other more complex cases, the constraint (3) of the model is always respected. After the ending of the process of time windows opening of the first element, it is excluded of the list, and the whole process is repeated for the new first element of the list, until it is empty.

This method created to obtain an initial solution do not worry with generation an optimized solution in any moment, since its objective is only to create viable quick solutions.

According to the descript method, the pivots with the highest water use will always be the ones to be placed first, to make the water use in the lowest possible time windows. Therefore, as the pivots are excluded of the list, pivots of smaller consumption will be still in the list, and the windows will be always with a uniform consumption, and we can eliminate the possibility of finding a pivot of high use when the windows are close to its water use limit.

With this method, we also can modify the ordering rule list, if it is necessary, by replacing, for example, the number of demanded hours by the water volume.

The method of creating a neighbor solution, with the presented configuration, has shown to be perfect to solve our problem, and it has generating satisfactory results.

B. Neighborhood Structure

It was utilized two neighborhood structures, as movement changing types: *Modify Simple Window* and *Switch Clients*. These movements were developed considering the limitations required by the problem.

It is interesting to observe that these movements created do not make the solution inaccessible, they always respect the problem restrictions, as previously we has shown. If the solution becomes inaccessible, the utilized movement is automatically undone.

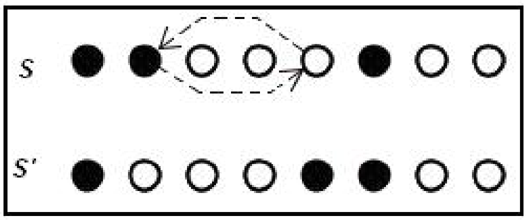

Fig. 3. *Modify Simple Window* movement.

The *Modify Simple Window* movement, as shown in Fig. 3, bases itself in doing the value substitution between two time windows with different values and that belongs to the same client. So, it was previously randomly selected a client to be modified, and then we also select randomly an opened time window of this same client to be closed. So, the client will need that one other window be closed in order that its time table be attended. The window to be opened is a nocturnal shift window, due to it is cheaper, but it also must respect the constraint to the water volume limitation.

The *Switch Clients* movement is a little more complex than the first one. It consists mainly in selecting randomly two clients. In sequence, it seeks a time window where one is opened and the other is closed. Finding that window, it replaces the values between them and checks if exists any problem with the water volume that will be used (Fig. 4a). If this constraint is disrespected, all the process is undone. Otherwise, it advances to next stage of the movement, which is to correct the number of irrigation hours of both clients, because after the switch, one of them is with one extra hour of what would be the correct value, and the other, one hour less. We will name those clients as $i+$ and $i-$ respectively.

The client $i+$ starts to search in its time windows, one that is opened to be closed, without worrying to the constraints, since the window closure will only reduce the water level. We shall name the chosen window index as j. After correcting the hours for client $i+$, it is time to correcting hours for client $i-$, which needs to open one of its time windows. The next step is to open the window j of the client $i-$ (Fig. 4b), and check if any constraints

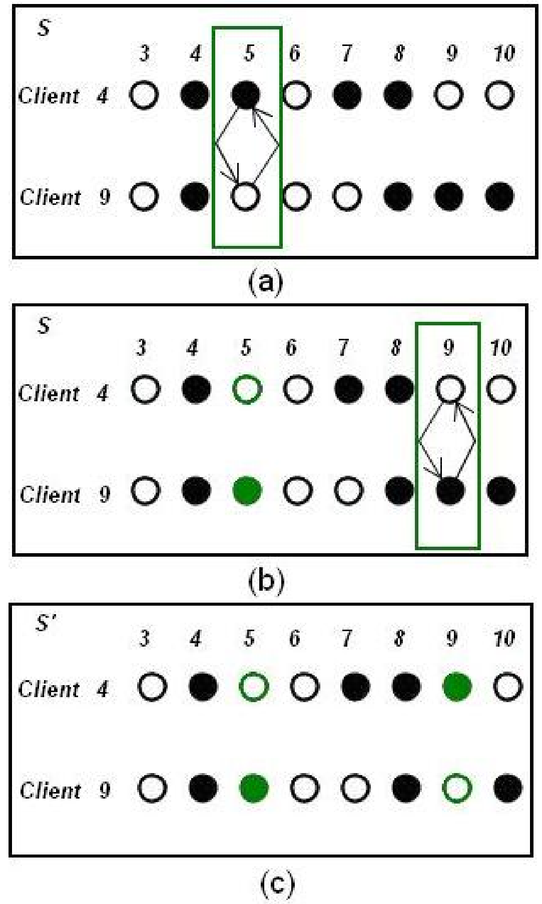

Fig. 4. *Switch Clients* movement.

was disrespected. In case of all constraints are respected, the movement is concluded (Fig. 4c). Otherwise, the window j of the client $i-$ will be closed again, and the search for a new window will be done randomly, until it finds a suitable time window. If all windows are tested and considered unsuitable, the whole movement is instantly canceled.

The neighborhood structures created were implemented in the algorithm SA in so that each neighbor solution is generated by only one of these movements, and its choice was made before the execution starts, and inserted by a parameter.

The objective function $f(S)$ used to evaluate the solutions is explained by the equation (1)(check Section III), and the restrictions presented in the shown model in the Section II, III and IV are attended in the construction function of a first solution and in the movement changes explained here.

IV. Computational Results

The developed system has presented quick and interesting results. If we compare the results provided by our system, with values found by LINGO with Linear

TABLE I

TEST RESULTS OF MODEL AND ALGORITHM VALIDATION.

Number of Center Pivots	Available Water Volume	Processing Time		Objective Function (monetary unit)	
		LINGO	SA	LINGO	SA
10	52.000 m^3/day	> 24 hr.	4 sec.	-	366.800
10	55.000 m^3/day	4 min. e 20 sec.	4 sec.	347.382	363.200
10	58.000 m^3/day	4 min. e 20 sec.	3 sec.	340.000	356.000

TABLE II

AGILITY TEST RESULTS.

Number of Center Pivots	Available Water Volume	Processing Time	Objective Function (monetary unit)
73	20.000 m^3/hour	17,235 sec.	1.748.000
158	35.000 m^3/hour	35,359 sec.	3.742.000
300	60.000 m^3/hour	70,859 sec.	7.572.800

Program, it is possible to notice the time gain obtained in a solution searching, without jeopardizing the solution.

To validate the model and the algorithm created, small tests were executed. In one of these tests, with 10 different center-pivots and an available water volume of 55.000 m^3/day, LINGO found a viable solution in 4 minutes and 20 seconds, and the algorithm would not achieve a optimal solution due to a lack of memory. With the SA, a good solution was found to the same problem, with less than 4 seconds of processing, by using a small quantity of memory.

For smaller water volume problems, but yet viable, like 52.000 m^3/day, for example, and the same 10 center-pivots, LINGO runs its algorithm for over than 24 hours, without finding a solution. The SA was capable of finding a good solution in less than 4 seconds, as represented in Table I.

With the objective of testing the SA with a big number of pivots, it was tested possibilities with 79, 158 and 300 center-pivots, with a lower water volume, testing the worst case, as we can see in Table II. The achieved results were satisfactory, because they were good solutions found in a very small time window, if we consider the dimension of the problem solved. Due to the fact that we do not know the optimal values of the problems, any comparative study was not done till now. However, according to specialist's opinion, the results were very good, they covered the necessity of quick and good results.

The used parameters by the SA, in all tests were done, T_0 = 1000000, α = 0.9975 e T_c = 0.01. All tests were done in a Sony Vaio fs-660/w laptop with Intel Centrino 1.73 GHz processor, 1 Gb RAM memory. The algorithm SA was developed in the C++ programming language.

V. CONCLUSIONS

This paper presented an approach to solving problem method to scaling center pivots. The model proposed was capable of representing the problem in all its aspects.

The use of Simulated Annealing was pointed as a good way to obtain good solutions in small periods of time and in a minimum computational demand, and it demonstrated a good option to the situation, because it reached all previewed objectives.

The obtained results (check in Section IV, Table I and II) show that the SA, combined with the proposed model (Section II) and the other methods descript in Section III, were capable of generating good quality solutions to all the instances in extremely short computational times.

The data structure used in the algorithm execution has also shown efficient, because it did not demand computational resources and, as consequence, it did not compromise the system performance.

ACKNOWLEDGEMENTS

The authors FAPEMIG (Minas Gerais State Foundation for Research Development) for the financial support.

REFERENCES

[1] U. Aickelin and P. White, "Building better nurse scheduling algorithms." *Annals of Operations Research*, vol. 128, pp. 159–177, 2004.

[2] E. Alba, *Parallel Metaheuristics - A new class of Algorithms.* Wiley & Sons., 2005.

[3] F. G. ao Avelino de Méllo, E. L. F. Senne, and L. A. N. Lorena, "Uma abordagem para o problema de escalonamento de motoristas," *Anais do XXVI Encontro Nacional de Engenharia de Produção*, Outubro 2006.
[4] E. Baskent and G. Jordan, "Forest landscape management modeling using simulated annealing." *Forest Ecology and Management*, vol. 165, pp. 29–45, 2002.
[5] T. Bck, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolucionary Computation.* Oxford University Press., 1997.
[6] S. Bernardo, A. A. Soares, and E. C. Mantovani, *Manual de Irrigação*, 8th ed. Editora UFV, 2006.
[7] C. Blum and A. Roli, "Metaheuristics in combinatorial optimisation: Overview and conceptual comparison." Universit Libre de Bruxelles, TR/IRIDIA/2001-13 13, 2001.
[8] K. Boston and P. Bettinger, "An analysis of monte carlo integer programming, simulated annealing, and tabu search heuristics for solving spatial harvest scheduling problems." *Forest Science*, vol. 45, no. 2, pp. 292–301, 1999.
[9] R. Burkard, S. Karisch, and F. Rendl, "Qaplib - a quadratic assignment problem library," *Journal of Optimization*, vol. 10, pp. 391–403, 1997. [Online]. Available: http://opt.math.tu-graz.ac.at/ karisch/qaplib/
[10] F. Busseti, "Simulated annealing overview." http://www.geocities.com/francorbusetti/, 2001.
[11] A. Camatta and A. R. S. Amaral, "Uma variante do simulated annealing aplicada a um problema de layout," *Anais do XXXVI Simpósio Brasileiro de Pesquisa Operacional*, Novembro 2004.
[12] V. Cerny, "Thermodynamical aproach to the traveling salesman problem: an efficient simulation algorithm." *Journal of Optimization Theory and Aplication*, vol. 45, no. 1, pp. 41–51, 1985.
[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press, 2002.
[14] M. V. da Silva, "Exploração do espaço de projeto em arquiteturas reconfiguráveis," Master's thesis, Universidade Federal de Viçosa - UFV, Agosto 2006.
[15] H. A. da Silva Gomes, "Utilização da metaheurística simulated annealing no problema de alocação de pessoal em empresas de transporte coletivo por ônibus," Master's thesis, Universidade Federal do Ceará, 2003.
[16] J. H. de Miranda and R. C. de Matos Pires, *Irrigação*, ser. Engenharia Agrícola. FUNEP, 2003, vol. 2.
[17] G. Eitzen, G. Mills, and D. Panton, "Multi-skilled workforce optimization." *Annals of Operations Research*, vol. 127, pp. 359–372, 2004.
[18] A. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier, "Staff scheduling and rostering: A review of applications, methods and models." *European Journal of Operational Research*, vol. 153, pp. 3–27, 2004.
[19] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, March 1995.
[20] F. Glover, "Future paths for integer programming and links to artificial intelligence." *Computers & Operations Research*, vol. 13, pp. 533–549, 1986.
[21] M. C. Goldbarg and H. P. Luna, *Otimização combinatória e programação linear: modelos e algoritmos*, 2nd ed. Editora Campus, 2005.
[22] A. C. Gomes and M. J. F. Souza, *Softwares de Otimização: Manual de Referência.*, Universidade Federal de Ouro Preto, 2004.
[23] A. Ignacio, V. Ferreira Filho, and R. Galvão, "Métodos heurísticos num entorno paralelo." *Anais do Simpósio Brasileiro de Pesquisa Operacional*, vol. 32, pp. 769–788, 2000.
[24] R. Jain, K. K. Ramakrishnan, and D. M. Chiu, "Congestion avoidance in computer networks with a connectionless network layer," Digital Equipment Corporation, MA, Tech. Rep. DEC-TR-506, Aug. 1987.
[25] N. Kahale and R. Urbanke, "On the minimum distance of parallel and serially concatenated codes," *IEEE Transactions on Information Theory*, submitted for publication.
[26] J. Keller and R. D. Bliesner, *Sprinkle and trickle irrigation.* Blackburn Press, 1990.
[27] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, 4598, no. 4598, pp. 671–680, May 1983.
[28] J. Klincewicz, "Avoiding local optima in p-hub location problem using tabu search and grasp," *Annals of Operations Research*, vol. 40, pp. 238–302, 1992.
[29] M. Laguna, T. Feo, and H. Elrod, "A greedy randomized adaptive search procedures for the 2-partition problem," Graduate School of Business and Administration, The University of Colorado at Boulder," tech. report., 1993.
[30] M. Laguna and J. Gonzlez-Velarde, "A search heuristic for just-in-time scheduling in parallel machines," *Journal of Intelligent Manufacturing*, vol. 2, pp. 253–260, 1991.
[31] Y. Li, P. Pardalos, and M. Resende, "A greedy randomized adaptive search procedures for the quadratic assignment problem," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 16, pp. 237–261, 1994.
[32] N. C. Loh, "High-resolution micromachined interferometric accelerometer," Master's thesis, Massachusetts Institute of Technology, Cambridge, 1992.
[33] G. R. Mauri and L. A. N. Lorena, "Simulated annealing aplicado a um problema de roteirização e programação de veículos," *Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional*, pp. 1054–1065, 2006.
[34] N. e. a. Metropolis, "Equation of state calculations by fast computing machines." *Jornal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.
[35] T. Noronha, M. M. da Silva, and D. Aloise, "Uma abordagem sobre estratégias metaheurísticas." *Revista Eletrônica de Iniciação Científica (REIC)*, vol. I, no. I, 2001.
[36] B. Owens, "Personnel scheduling using constraint logic programming." Computer Science and Software Engineering, Monash University, Australia, 2001.
[37] G. W. Pereira, "Aplicação da técnica de recozimento simulado em problemas de planejamento florestal multiobjetivo," Master's thesis, Universidade Federal de Minas Gerais, Maro 2004.
[38] G. W. Pereira and H. do Nascimento Santos, "Aplicação do simulated annealing na solução de problemas de planejamento florestal multiobjetivo," *Anais do XXXVI Simpósio Brasileiro de Pesquisa Operacional*, Novembro 2004.
[39] F. F. Pruski, R. del G. Rodriguez, L. F. de Novaes, D. D. da Silva, M. M. Ramos, and A. de F. Teixeira, "Impacto das vazões demandadas pela irrigação e pelos abastecimentos animal e humano, na bacia do paracatu," *Revista Brasileira de Engenharia Agrícola e Ambiental*, vol. 11, no. 2, pp. 199–210, 2007.
[40] C. R. Reeves, *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Press, 1993.
[41] E. O. Rodrigues, "Problema de especificação de células de manufatura na presença de restrições de projeto: Uma abordagem baseada em tabu search." Master's thesis, Universidade Federal do Espírito Santo - UFES, 1997.
[42] F. Rodrigues, "Metaheurística e sistema de suporte à decisão no gerenciamento de recursos florestais." Ph.D. dissertation, Universidade Federal de Viçosa, 2001.
[43] D. S. Vianna and J. E. C. Arroyo, "A grasp algorithm for the multi-objective knapsack problem," *SCCC*, vol. 00, pp. 69–75, 2004.
[44] L. Wolsey, *Integer Programming.* John Wiley and Sons., 1998.

[45] L. Yamamoto, "Uso de simulated annealing e algoritmo genético no problema da reconfiguração de uma rede de distribuição de enérgia elétrica," Master's thesis, Universidade Federal do Paraná, 2004.
[46] H. Youssef, S. M. Sait, and H. Adiche, "Evolutionary algorithms, simulated annealing and tabu search: a comparative study," *Engineering Applications of Artificial Intelligence*, vol. 14, no. 2, pp. 167–181, April 2001.