Solving Mixed Integer Nonlinear Chemical Engineering Problems via Simulated Annealing Approach

Y. Özçelik and Z. Özçelik
Ege University, Engineering Faculty,
Chemical Engineering Department, Bornova – Izmir, Turkey,
Tel: 90 232 388 40 00 / 1488; Fax: 90 232 388 76 00;
E–mail: yozcelik@eng.ege.edu.tr; zozcelik@bornova.ege.edu.tr

Original scientific paper
Received: September 5, 2003
Accepted: May 24, 2004

The optimization models of chemical engineering processes are generally nonlinear and mixed integer, nonlinear in structure, and contain several equality and inequality constraints and bounds on variables.

Mixed Integer Nonlinear programming problems (MINLP) can be solved using either gradient methods or stochastic methods. Gradient methods need to separate the problem to Mixed Integer Linear Programming (MILP) and Nonlinear Programming (NLP) problems and some special formulations where the continuity or convexity has to be imposed.

In this work, an algorithm (SARAN) that was based on a simulated annealing algorithm of *Corana*[1] was developed to solve MINLP problems. Then the algorithm and some alternatives for the basic steps of a simulated annealing were tested for 100 sequences of pseudo random numbers using 11 MINLP test problems. Finally the results were compared with the results of the M–SIMPSA[2] for small and medium scaled problems.

Key words:
Mixed Integer Nonlinear Programming, Nonlinear Optimization, Simulated Annealing

Introduction

There are two main approaches to solve NLP and MINLP problems.

– Gradient Approach

– Stochastic Approach

Various gradient algorithms have been described in the literature. The common property of deterministic algorithms[3,4] is to separate the original MINLP problems into sub–problems. On the other hand stochastic methods do not necessarily separate the problem into sub–problems. However, various problem–independent heuristics related to search interval compression and expansion and to shifting strategies are required for their effectiveness. Also these methods require the application of successive relaxations, which may substantially increase the effort in identifying feasible regions and obtaining the global optimum.

The simulated annealing algorithm is also a stochastic method based on stochastic generation of solution vectors and employ similarities between the physical processes of annealing and the optimisation problems.

The physical process of annealing

The aim of the process is to find the atomic configuration that minimizes internal energy. For a given configuration, a random move is carried out by randomly picking a molecule and moving it in a random direction for a random distance. The new configuration is then accepted or rejected according to an acceptance criterion, based on the Boltzman function, (1):

$$\left(p = \mathrm{e}^{\frac{-\Delta E}{k_{\mathrm{B}} T}} \right) \tag{1}$$

where ΔE is the change in the energy of the configurations, k_{B} is the Boltzman constant, and T is the temperature of the system. The acceptance of a new configuration depends on ΔE and temperature. T is reduced to reach a lower energy state. The low temperature is not a sufficient condition to find ground states of matter. The cooling process should be accomplished slowly, otherwise the resulting crystal will have many defects or the substance may form a glass with no crystalline order.

Optimisation by simulated annealing

In simulated annealing, the value of the objective function is analogous to the energy of the system and the aim is to minimize the value of the objective function, where the values of the continuous and discrete variables represent a particular configuration of the system. The behaviour of the system,

subject to such a neighbourhood move is determined from the value of the two successive values of the objective function.

There are two important approaches in the application of the simulated annealing to the continuous variables.

– The hybrid of random search and acceptance–rejection criteria of simulated annealing algorithms.

– The hybrid of simulated annealing algorithm and nonlinear simplex algorithm

Both of these two approaches have the following critical steps.

– Estimation of the initial temperature of the system.

– Acceptance of points generated.

– Cooling Schedule

– Termination of the constant temperature process and termination of the cooling process

– Dealing with constraints

Simulated annealing algorithms can be developed using alternative schemes for these critical steps.

Initial value of temperature

An initial value of the annealing temperature for a physical process of annealing and optimization algorithms are based on annealing as an effective parameter for the optimisation path. The value of the initial temperature for a physical process depends on the properties of the system and can be estimated experimentally. Analogously, the initial value of the temperature for an optimisation problem is estimated by the observation of the behaviour of the problem using a specified time, determined experimentally. The alternative schemes are available to estimate the initial temperature. Some of them were used in this work.

– *Prosenjit* and *Diwekar*[5] proposed any temperature value that satisfies the following inequality (2) as the initial temperature:

$$e^{\frac{-\Delta F}{k_B T}} \leq 0.8 \qquad (2)$$

The vector ΔF is obtained by evaluating the change in the objective function value for a large number (e.g., 100) of neighbourhood moves. This procedure has been applied to the optimal design problem of heat exchangers.

– *Aarst* and *Korst*[6] and *Aarst* and *Van Laarhoven*[7] proposed any temperature value that satisfies the following equality (3) as the initial temperature.

$$0.95 = \frac{m_1 + m_2 e^{\left(\frac{-\Delta f^+}{T_i}\right)}}{m_1 + m_2} \qquad (3)$$

m_1 and m_2 are the number of successful and unsuccessful moves, respectively. Δf^+ is the average increase in the objective function value for m_2. The total number of moves to determine an initial temperature is proposed as $100 \cdot N$.

For example, (10^5) may be proposed as a starting point of the temperature to determine the initial temperature.

Acceptance criteria

Two types of acceptance criteria have been proposed in literature.

– Metropolis algorithm accepts all downhill moves. It accepts uphill moves with a probability p which depends on temperature and the change of the objective function values evaluated in successive iterations (4). Metropolis accepts all the moves with a probability of 1 at high temperatures.

$$\begin{aligned} \Delta F \leq 0 \qquad & p = 1 \\ \Delta F > 0 \qquad & p = e^{\frac{-\Delta F}{T}} \end{aligned} \qquad (4)$$

– The method of a nonequilibrium Metropolis accepts uphill moves with a probability p which depends on temperature and the change of the objective function values evaluated in successive iterations similar to the Metropolis algorithm. Additionally, the inner loop of the annealing algorithm is terminated and cooling is applied when a downhill move occurs in the objective function.

– The Glauber algorithm accepts all moves at a high temperature with 0.5 probabilities (5). If the temperature is reduced, the probability acceptance of downhill moves remains constant at 1 in the Metropolis algorithm. The value of probability increases from 0.5 to 1 in the Glauber algorithm. On the other hand, the acceptance probability of an uphill move decreases from 0.5 to 0 in the Glauber algorithm,

$$p = \frac{e^{\frac{-\Delta F}{T}}}{1 + e^{\frac{-\Delta F}{T}}} \qquad (5)$$

Annealing schedule

Two types of annealing schedules have been proposed.

– The exponential type–cooling schedule,[8] where a constant multiplicative factor is employed, can be used to obtain a new temperature (6).

$$T^{k+1} = \alpha T^k \tag{6}$$

α is a temperature decrement factor which lies in the range of 0–1.

– *Aarst* and *Van Laarhoven*[7] proposed an alternating cooling schedule. (7)

$$T^{k+1} = \frac{T^k}{1 + \dfrac{T^k \ln(1+\delta)}{3\sigma(T^k)}} \tag{7}$$

$\sigma(T^k)$ is the standard deviation of the objective function at T^k. δ is the speed parameter lying on the range of 0–1. The value of δ varies depending on the equilibrium or the nonequilibrium application of Simulated Annealing with an order of 10^3 (0.01, 10).

Termination criteria for constant temperature step and cooling step

The constant number of moves or nonequilibrium comes together when constant number of moving approach can be used for the constant temperature step. In this approach, if the new function value is lower than the previous one, temperature is reduced without reaching equilibrium; otherwise the approach of constant number of moves is applied.

The process of cooling can be terminated using the following criteria

– The constant number of temperature levels

$$T^k < 1 \tag{8}$$

$$\frac{\mathrm{d}F(T)}{\mathrm{d}T}\frac{T}{F(T)} \leq \in \tag{9}$$

– ΔC can't be improved for several temperature levels.

Dealing with constraints

– The Cardosa Approach. All infeasible points are penalized assuming a very large positive value for a minimization problem or a very large negative value, otherwise.[9] The method is an infeasible path method. This means that infeasible points may be replaced by the new infeasible points, except that they are now centred on the best vertex and obey all bound constraints.

– A feasible path method may be used. This means, that feasible points may be replaced by feasible points, and the objective function is evaluated only for the feasible points. This approach may lead to inaccurate results.

Procedure and results

In this work, an algorithm based on the algorithm of *Corona*,[1] was developed. The algorithm of Corona was used to solve unconstrained problems. This algorithm was modified to solve constrained MINLP problems, and 11 MINLP problems were used to test the criteria that were used in the basic steps of the algorithm. Then these problems were utilised to compare the performance of the algorithm with the algorithm of M–SIMPSA[2] for MINLP problems. The 9 test problems of *Cardosa* et al[2] and 2 test problems of *Duran*[4] and the initial values for these problems, are given below.

1. (*M. F. Cardosa* et al.[2] and is also given in *Kocis* and *Grossmann*[3], *Floudas* et al.[10] and *Ryoo* and *Sahinidis*[11]). There is a local optimum at {y, x, Z} = {0, 1.118, 2.236} and nonconvexities arise in the first constraint. The global optimum is at (y, x, Z} = {1,0.5,2}

Minimize $Z = 2x + y$

Subject to:

$$1.25 - x^2 - y \leq 0 \qquad x + y - 1.6 \leq 0$$

$$\boldsymbol{L} = 0 \qquad \boldsymbol{L} \leq x \leq \boldsymbol{U} \qquad y = \{0,1\}$$

2. (*M. F. Cardosa* et al.[2] and is also given in *Kocis* and *Grossmann*[3] and *Salcedo* et al.[12]). The global optimum is at (y, x_1, x_2, Z} = {1, 1.375, 0.375, 2.124}.

Minimize $Z = -y + 2x_1 + x_2$

Subject to:

$$x_1 - 2\mathrm{e}^{(-x_2)} \leq 0 \qquad -x_1 + x_2 + y \leq 0$$

$$\boldsymbol{L}^T = (0.5,0) \quad \boldsymbol{U}^T = (1.4,10) \quad \boldsymbol{L} \leq x \leq \boldsymbol{U} \quad y = \{0,1\}$$

3. (*M. F. Cardosa* et al.[2] and is also given in *Floudas* et al.[10]). There are nonconvexities because of the first constraint. The global optimum is at (y, x_1, x_2, Z} = {1, 0.94194, –2.1, 1.07654}.

Minimize $Z = -0.7y_1 + 5(x_1 - 0.5)^2 + 0.8$

Subject to:

$$-\mathrm{e}^{(x_1 - 0.2)} - x_2 \leq 0$$

$$x_2 + 1.1y_1 + 1 \leq 0$$

$$x_1 - 1.2y_1 - 0.2 \leq 0$$

$$\boldsymbol{L}^T = (0.2, -2.22554) \qquad \boldsymbol{U}^T = (1, -1)$$

$$\boldsymbol{L} \leq x \leq \boldsymbol{U} \qquad y = \{0,1\}$$

4. (*M. F. Cardosa* et al.[2] and is also given in *Kocis* and *Grossmann*[3], *Floudas* et al.[10], *Salcedo*[12] and *Ryoo* and *Sahinidis*[11]). There are nonconvexities because of the equality constraints. The global optimum is at (y_1, y_2, y_3, Z} = {0, 1, 1, 7.667180}

Minimize $Z = 2x_1 + 3x_2 + 1.5y_1 + 2y_2 - 0.5y_3$

Subject to:

$$x_1^2 + y_1 = 1.25$$

$$x_2^{1.5} + 1.5y_2 = 3$$

$$x_1 + y_1 - 1.6 \le 0$$

$$1.333x_2 + y_2 - 3 \le 0$$

$$-y_1 - y_2 + y_3 \le 0$$

$$\boldsymbol{L}^T = \{0,0) \qquad \boldsymbol{L} \le x \quad y = \{0,1\}^3$$

5. (*M. F. Cardosa* et al.[2] and is also given in *Kocis* and *Grossmann*[3], *Diwekar* et al.[13] and *Diwekar* and *Rubin*[14]). It is related to the selection among two candidate reactors for minimizing the cost of producing a desired product. The global optimum is at $(y_2, x_1, x_2, x_3, Z\} = \{0, 1, 1, 7.667180\}$

Minimize $Z = 7.5y_1 + 5.5y_2 + 7x_1 + 6x_2 + 5x_3$

Subject to:

$$y_1 + y_2 = 1$$

$$x_4 = 0.91 - \mathrm{e}^{0.5x_1})x_6$$

$$x_5 = 0.81 - \mathrm{e}^{0.4x_2})x_7$$

$$x_6 + x_7 - x_3 = 0$$

$$x_4 + x_5 = 10$$

$$x_1 - 10y_1 \le 0$$

$$x_2 - 10y_2 \le 0$$

$$x_6 - 20y_1 \le 0$$

$$x_7 - 20y_2 \le 0$$

$$\boldsymbol{L}^T = (0;0;0;0;0;0;0)$$

$$\boldsymbol{U}^T = (10;10;40;20;20;10;10)$$

$$\boldsymbol{L} \le x \le \boldsymbol{U} \qquad y = \{0,1\}^2$$

6. (*M. F. Cardosa* et al.[2] and is also given in *Salcedo*[12]). This represents a quadratic capital budgeting problem. It has four binary variables and features bilinear terms in the objective function. The global optimum is at $(y_1, y_2, y_3, y_4, Z\} = \{0, 0, 1, 1, -6\}$

Minimize

$$Z = (y_1 + 2y_2 + 3y_3 - y_4)(2y_1 + 5y_2 + 3y_3 - 6y_4)$$

Subject to:

$$y_1 + 2y_2 + y_3 + 3y_4 - 4 \le 0 \qquad y = \{0,1\}^4$$

7. (*M. F. Cardosa* et al.[2] and was also given in *Yuan* et al.[15], *Floudas* et al.[10], *Salcedo*[3] and *Ryoo* and *Sahinidis*[11]). It has nonlinearities in, both, the continuous and binary variables. The global optimum is at $(y_1, y_2, y_3, y_4, x_1, x_2, x_3, Z\} = \{1, 1, 0, 1, 0.2, 0.8, 1.907878, 4.579582\}$

Minimize $Z = (y_1 - 1)^2 + (y_2 - 2)^2 + (y_3 - 1)^2 - \log(y_4 + 1) + (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2$

Subject to:

$$y_1 + y_2 + y_3 + x_1 + x_2 + x_3 - 5 \le 0$$

$$y_3^2 + x_1^2 + x_2^2 + x_3^2 - 5.5 \le 0$$

$$y_1 + x_1 - 1.2 \le 0 \qquad y_2 + x_2 - 1.8 \le 0$$

$$y_3 + x_3 - 2.5 \le 0 \qquad y_4 + x_1 - 1.2 \le 0$$

$$y_2^2 + x_2^2 - 1.64 \le 0 \qquad y_3^2 + x_3^2 - 4.25 \le 0$$

$$y_2^2 + x_3^2 - 4.64 \le 0$$

$$\boldsymbol{L}^T = (0,0;0) \qquad \boldsymbol{U}^T = (1.2;1.8,2.5)$$

$$\boldsymbol{L} \le x \le \boldsymbol{U} \qquad y = \{0,1\}^4$$

8. (*M. F. Cardosa* et al.[2] and is also given in *Berman* and *Ashrafi*[16]). The global optimum is at $(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, Z\} = \{0, 1, 1, 1, 0, 1, 1, 0, 0.93634\}$

Minimize $Z = -x_1 x_2 x_3$

Subject to:

$$y_1 + y_2 + y_3 - 1 \ge 0$$

$$y_4 + y_5 + y_6 - 1 \ge 0$$

$$y_7 + y_8 - 1 \ge 0$$

$$3y_1 + y_2 + 2y_3 + 3y_4 + 2y_5 + y_6 + 3y_7 + 2y_8 - 10 \le 0$$

$$x_1 = 1 - 0.1^{y_1}\ 0.2^{y_5}\ 0.15^{y_3}$$

$$x_2 = 1 - 0.05^{y_4}\ 0.2^{y_5}\ 0.15^{y_6}$$

$$x_3 = 1 - 0.02y_7\ 0.06^{y_8}$$

$$\boldsymbol{L}^T = (0,0;0) \quad \boldsymbol{U}^T = (1;1;1) \quad \boldsymbol{L} \le x \le \boldsymbol{U} \quad y = \{0,1\}^8$$

9. (*M. F. Cardosa* et al.[2] and is also given in *Wong*[17]). The global optimum is at $(y_1, x_1, x_3, Z\} = \{78, 27, 27, 32217.4\}$

Maximize $Z = -5357854x_1^2 - 0.835689y_1x_3 - -37.29329y_1 + 40792.141$

Subject to:

$$x_4 = a_1 + a_2x_3y_2 + a_3x_2y_1 - a_4x_1x_3$$

$$x_5 = a_5 + a_6x_3y_2 + a_7y_2y_1 + a_8x_1^2 - 90$$

$$x_6 = a_9 + a_{10}x_3x_1 + a_{11}x_1y_1 + a_{12}x_1x_2 - 20$$

$$\boldsymbol{L}^T = (27,27,27)\ \boldsymbol{U}^T = (45,45,45)\ \boldsymbol{L} \le x_1,x_2,x_3 \le \boldsymbol{U}$$

$$\boldsymbol{L}^T = (78,78) \qquad \boldsymbol{U}^T = (102,102) \qquad \boldsymbol{L} \le y \le \boldsymbol{U}$$

10. (*Marco A. Duran* and *Ignacio E. Grossmann*[4]). It is related to a synthesizing process. It is the one of simultaneously determining the optimal structure and operating parameters for a chemical process.

Minimize $Z = 5y_1 + 6y_2 + 8y_3 + 10x_1 - 7x_3 + +10x_1 - 7x_3 - 18\ln(x_2+1) - 19.2\ln(x_1 - x_2 + 1) + 10$

Subject to:

$$-0.8\ln(x_2+1) - 0.96\ln(x_1 - x_2 + 1) + 0.8x_3 \le 0$$

$$-\ln(x_2+1) - 1.2\ln(x_1 - x_2 + 1) + x_3 + 2y_3 \le 0$$

$$x_2 - 2y_1 \le 0 \qquad x_2 - x_1 \le 0$$

$$x_1 - x_2 - 2y_2 \le 0 \qquad y_1 + y_2 - 1 \le 0$$

$$\boldsymbol{L}^T = (0,0,0) \quad \boldsymbol{U}^T = (0,0,0) \quad \boldsymbol{L} \le y \le \boldsymbol{U} \quad y^0 = (1,0,1)$$

11. (*Marco A. Duran* and *Ignacio E. Grossmann*[4]). This is related to a synthesizing process.

Minimaze $Z = 5y_1 + 8y_2 + 6y_3 + 10y_4 + +6y_5 + 7y_6 + 4y_7 + 5y_8 - 10x_1 - 15x_2 + 15x_3 + +80x_4 + 25x_5 + 35x_6 - 40x_7 + 15x_8 - 35x_9 + e^{x_1} + +e^{(x_2/1.2)} - 65\log(x_3 + x_4 + 1) - 90\log(x_5 + 1) - -80\log(x_6 + 1) + 120$

$$-x_4 + x_7 + x_9 \le 0 \qquad -0.4x_5 - 0.4x_6 + 1.5x_8 \le 0$$

$$0.16x_5 + 0.16x_6 - 1.2x_8 \le 0 \qquad x_3 - 0.8x_4 \le 0$$

$$-x_3 + 0.4x_4 \le 0 \qquad e^{x_1} - 10y_1 - 1 \le 0$$

$$e^{x_2/1.2} - 10y_2 \le 0 \qquad x_7 - 10y_3 \le 0$$

$$0.8x_5 + 0.8x_6 - 10y_4 \le 0 \quad 2x_4 - 2x_7 - 2x_9 - 10y_6 \le 0$$

$$x_5 - 10y_6 \le 0 \qquad x_6 - 10y_7 \le 0$$

$$x_3 + x_4 - 10y_8 \le 0 \qquad y_4 + y_5 - 1 \le 0$$

$$y_3 - y_8 \le 0 \qquad x_1 = (1.25 - y_1)^{0.5}$$

$$x_2 = (3 - 1.5y_3)^{(1/1.5)} \qquad \boldsymbol{L}^T = (0;0;0;0;0;0;0;0;0)$$

$$\boldsymbol{U}^T = (2;2;1;2;2;2;2;1;3) \qquad \boldsymbol{L} \le x \le \boldsymbol{U}$$

Discussion

M–SIMPSA is a hybrid of the simulated annealing and nonlinear simplex and has a potential risk to reach the global optimum. For, the simplex moves may actually destroy the global convergence properties of simulated annealing. Taking this risk into account the proposed algorithm can be handled as an alternative approach to solve constrained MINLP problems.

The basic steps of the algorithm that was shown schematically in figure 1 are explained below:

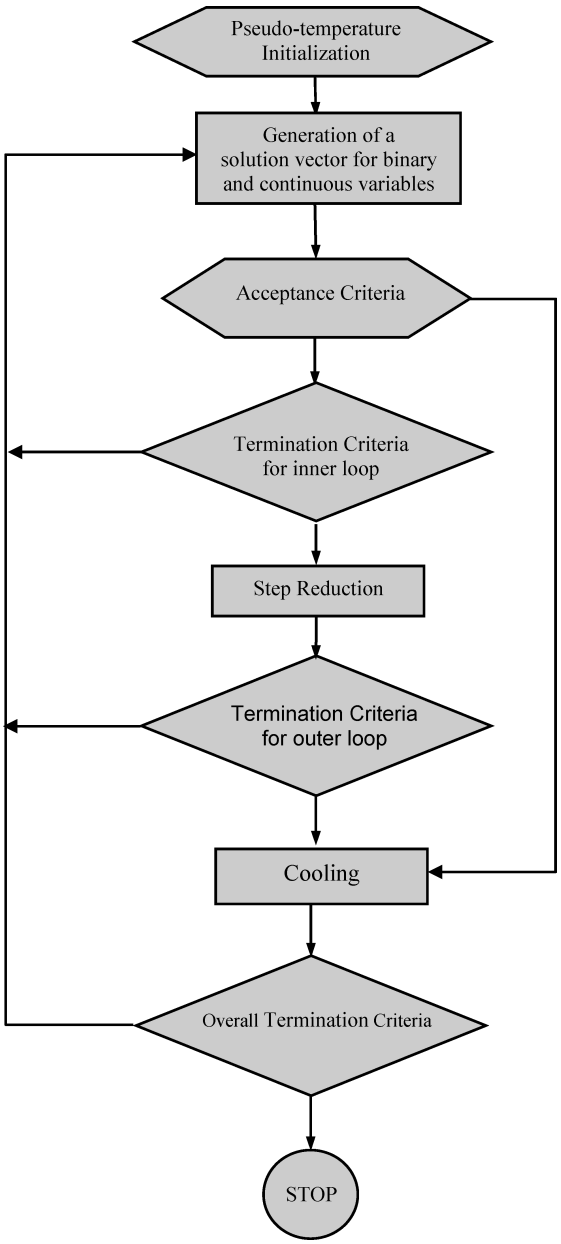

Fig. 1 – *The flowsheet of the proposed algorithm*

1. **The initial value of the pseudo temperature:** The initial value was determined by using the method of *Aarst* and *Korst*.[6] The performance of this method was compared with the method of *Prosenjit* and *Diwekar*,[5] and the calculated ratios between the initial pseudo temperatures, the number of functions evaluated, and the error % com-

pared with the values of the objective function in global minimum, are given in table 1. The initial values of the pseudo temperatures calculated by the method of *Aarst* and *Korst*[6] were greater than the others. Therefore, this method causes the evaluations of more functions. If only these parameters are considered, the method of *Prosenjit* and *Diwekar*[5] has an advantage over the method of *Aarst* and *Korst*.[6] But the low initial values of pseudo temperatures caused to converge to a local minimum (Problem 6).

Table 1 – *The ratios for initial temperatures and number of functions evaluated and errors compared with the values of the objective function in global minimum*

Problem	T_{iA}/T_{iP}	N_A/N_P	Error e_A / %	Error e_P / %
1	1.00	1.33	0.05	0
2	95.36	1.46	0.02354	0.02354
3	71.11	0.49	0.386423	0.590782
4	4842931937	1.00	0.000261	0.000261
5	104.29	3.54	2.338343	24.98217
6	90.91	1.15	0.008565	0.262395
7	100.06	1.00	0	0
8	26.09	2.56	0.402176	1.236314
9	60.99	1.70	0.00466	0.05458
10	100.39	1.00	0.288589	0.288589
11	25.1	1.87	0.0048	0.056

2. **Generation of a new point:** In the second step of the algorithm, the test points were generated using the following relations.

$$x_{i+1} = x_i + r\boldsymbol{v}_{\mathrm{mh}} e_{\mathrm{h}}$$
$$y_{i+1} = \mathrm{LB} + \mathrm{INT}(u\,\boldsymbol{sr}^1 + v) \tag{10}$$

x_i is the current optimum for the continuous variables, r and u are random numbers in the intervals $\lfloor -0.5, 0.5 \rfloor$ and 0, 1, respectively. e_h is the vector of the h^{th} coordinate direction and v_{mh} is the component of the step vector ($\boldsymbol{v}_{\mathrm{m}}$) along the same direction. LB and ***sr*** is the vector of the lower bounds and search region for discrete variables.

3. **The acceptance of the point generated:** The method of nonequilibrium Metropolis was preferred to decide the acceptance of the new point generated. The nonequilibrium Metropolis algorithm prevented the computational burden of the algorithm for test problems. Although the nonequilibrium Metropolis algorithm can lead to local optima this effect has not been observed for the test problems in this work.

4. **Dealing with constraints:** The constrained problems are transformed into the unconstrained ones using the following penalizing scheme,[18]

$$F(x, M) = f(x) + M\left\{\sum_{i=1}^{m}[\max(0, g_i(x))]^2 + \sum_{j=1}^{n} h_j(x)^2\right\} \tag{11}$$

M is a positive penalty parameter. The value of the penalty parameter can ideally affect the convergence. M = 100 gave good results for the test problems. The algorithm is an infeasible path method and uses all of the perturbed function values.

5. **Termination criteria:** The constant number of moves and constant number of step adjustment approaches, that were used, depend on the number of variables used for constant temperature search.

The number of moves = 5 * the number of variables

The number of step adjustment = 5 * the number of variables

Otherwise, the relation (9) was used to terminate the cooling process.

The method of Corona was used to adjust the value of the step reduction at the end of inner loop.

6. **Annealing scheduling:** The method of *Aarst* and *Van Laarhoven*[6] (7) was used to reduce the pseudo temperature of the problem.

Conclusions

An algorithm was proposed for MINLP problems. It was based on the algorithm of Corona that can solve only unconstrained continuous problems. The nonequilibrium Metropolis method[9] and a penalizing[11] scheme were also adapted to prevent computational burden and to improve the reliability of the algorithm, respectively.

The proposed algorithm was applied to solve several MINLP test functions published in literature. The performance of the algorithm is comparable with the performance of the M–SIMPSA for all ill conditioned problems. The average number of function evaluations for 100 runs, and success percentages for eleven ill-conditioned MINLP test problems, are given in table 2.

Table 2 – *The fraction of successes and the average number of function evolutions (The success fractions (%) were given for feasible path and infeasible path alternatives of M–SIMPSA)*

Problem	M–Simpsa			Saran	
	N_F	success		N_F	success
1	16282	99	100	5671	100
2	14440	83	100	8873	100
3	38042	0	100	10920	100
4	577	100	—	2581	100
5	42295	100	100	113252	90
6	4477	100	—	568	100
7	63751	60	97	40316	70
8	15462	100	—	49813	100
9	33956	87	95	26743	94
10	257536	–	–	225436	100
11	831149	–	–	825154	95

The value for speed parameter was a constant of 0.011 for all problems. This speed value provides convergence to problems 5, 7, 9 to a local minimum in 10, 30 and 6 times respectively per 100 runs. The convergence times were increased to 92, 100, 100 for these problems reducing the value of speed parameter to 0.0007, 0.0011 and 0.0006, respectively.

The computer program developed in this work, besides running the proposed algorithm based on mentioned criteria, is open in the user interface to try different criteria for the basic steps and compare with the structure given in this paper.

Nomenclature

e_A – Accuracy of convergence to the global minimum for Aarst and Korst[5] criteria, %

e_D – Accuracy of convergence to the global minimum for Diwekar method, %

e_{NEM} – Accuracy of convergence to the global minimum for nonequilibrium method, %

MINLP – Mixed Integer NonLinear Programming

N – Number of variables

N_A – The number of function evaluations in *Aarst* and *Korst*[5] procedure in determining initial annealing temperature

N_{Aarst} – The number of function evaluations in *Aarst* and *Korst*[5] procedure in reducing annealing temperature.

N_{CR} – The number of function evaluations in constant region reduction method

N_D – The number of function evaluations in the method of Diwekar to determine initial values of annealing temperatures

N_F – The number of function evaluations

$N_{Feasible}$ – The number of function evaluations in the feasible path procedure

$N_{Infeasible}$ – The number of function evaluations in the infeasible path a procedure

$N_{KirkPatrick}$ – The number of function evaluations in the method of Kirkpatrick in reducing annealing temperatures

N_M – The number of function evaluations in the method of Metropolis in accepting the points generated

N_{NEM} – The number of function evaluations in the method of nonequilibrium Metropolis in accepting the points generated

NESA – Nonequilibrium Simulated Annealing

NLP – Nonlinear Programming

T_iA – The initial value of Pseudo annealing temperature when the method of *Aarst* and *Korst*[6] was applied

T_iD – The initial value of Pseudo annealing temperature when the method of Diwekar was applied

ΔE – The change in the energy of the system

ΔF – The change in the value of objective function

References

1. *Corona, A.* et al., ACM Transactions on Mathematical Software, **13** (3) (1987) 262.
2. *Margarida F. Cardosa* et al., Computers and Chem. Engng., **21** (12) (1997) 1349.
3. *Kocis, G. R.*, *Grossmann, I. E.*, Ind. Engng. Chem. Res., **27** (1988) 1407.
4. *Marco A. Duran*, *Ignacio E. Grossmann*, Mathematical Programming **36** (1986) 337.
5. *Prosenjit D. Chaudhuri*, *Urmila M. Diwekar*, Ind. Engng. Chem. Res., **36** (1997) 3685.
6. *Aarst E.*, *Korst*, Simulated Annealing and Boltzmann Machines-a Stochastic Approach to Combinatorial Optimisation and Neural Computers. Wiley, New York, 1989.
7. *Aarst E. H. L.*, *Van Laarhoven P. J. M.*, (1985), Philips J. Res., **40** (1985) 193.
8. *S. Kirkpatrick* et al., Science, **220** (4598) 671.
9. *Margarida F. Cardosa* et al., Optimisation Computers Chem. Engng. **20** (9) (1996) 1065.
10. *Floudas* et al., Computers and. Chem. Engng. **13** (10) (1989) 1117.
11. *Ryoo, H. S.*, *Sahinidis. N. V.*, Computers and Chemical. Engineering, **19** (5) 551.
12. *Salcedo, R. L.*, Ind. Engng. Chem. Res., **31** (1) 262.
13. *Diwekar, U. M.* et al., Ind. Engng. Chem. Res. **31** (1992) 313.
14. *Diwekar, U. M., Rubin, E. S.*, Ind. Engng. Chem. Res. **32** (9) (1993) 2006.
15. *Yuan, X.* et al., Une Methode D'optimization Nonlineaire en Variables Mixtes Pour la Conception de ProcPdes, RAIRO–oper. Res. 1989.
16. *Berman, O., Ashrafi, N.*, IEEE Trans. Soft. Eng. **19** (1993) 1119.
17. *Wong, J.*, COED J. **10** (1990) 19.
18. *Yiping W., Luke E.K.*, Computers and. Chem. Engng. **26** (2002) 1415.