Using Particle Swarm Optimization for Mixed-Integer Nonlinear Programming in Process Synthesis

Bi Rongshan, Yang Xia
(Institute of Computer and Chemical Engineering, QingDao University of Science and Technology, QingDao City, ShanDong Province, China 266042)

Abstract: Process synthesis problems can be mathematically represented as mixed-integer nonlinear programming (MINLP) models, which are often irregular, large and non-convex and difficult to get the overall optimum by traditional method. In this paper, a new method named particle swarm optimization (PSO) is used to solve MINLP problems. By introduced penalty function and used sigmoid function, the PSO algorithm, which originally can only used in continues variables with max/min limitations, can solve the MINLP problems with equations and inequalities constraints. A few examples used to demonstrate the validity of the method and compared with the results of other methods, the results show that the PSO algorithm is a efficient method to deal with MINLP problems.
Key words: process synthesis, mixed-integer nonlinear programming, particle swarm optimization

Introduction

Synthesis problems in process engineering often give rise to mixed-integer nonlinear program (MINLP) problems, which when represented in algebraic form, correspond to mixed-integer optimization problems that have the following form:

$$
\begin{aligned}
\min \quad & f(X,Y) \\
s.t \quad & h(X,Y) = 0 \\
& g(X,Y) \le 0 \\
& X \in \{\ X \mid X \in R^n, X^L < X < X^H\ \} \\
& Y \in \{\ Y \mid Y \in \{0,\}^m \}
\end{aligned}
\qquad (1)
$$

where $f(x, y)$ is the objective function (e.g. cost), $h(x, y) = 0$ are the equations that describe the performance of the system (mass and heat balances, design equations), and $g(x, y) <= 0$ are inequalities that define the specifications or constraints for feasible choices. The variables x are continuous and generally correspond to the state or design variables, while y are the discrete variables, which generally are restricted to take 0-1 values to define the selection of an item or an action.

Major methods for MINLP problems include first Branch and Bound (BB), Generalized Benders Decomposition (GBD), and Outer-Approximation (OA). All these methods assume convexity to guarantee convergence to the global optimum. In recent years a new trend that has emerged in the formulation and solution of discrete/continuous optimization problems through a model that is known as Generalized Disjunctive Programming (GDP) . The basic idea in GDP models is to use boolean and continuous variables, and formulate the problem with an objective function, and subject to two three types of constraints: (a) global inequalities that are independent of discrete decisions; (b) disjunctions that are conditional constraints involving an OR operator; (c) pure logic constraints that involve only the boolean variables. Türkay and Grossmann proposed a logic-based Outer-Approximation algorithm. This algorithm is based on the idea of extending the Outer-Approximation algorithm by solving NLP subproblems in reduced space, in which constraints that do not apply in the disjunctions are disregarded, with which both the efficiency and robustness can be improved. All these methods are still in developing.

Non-rigorous techniques such as simulated annealing and genetic algorithms become popular in recent years, which do not make any assumptions on the functions. Though these methods cannot guarantee rigorous solutions, they are easy to use and if only given enough iterations, they can give the approximate optimum. These methods do not formulate the problem as a mathematical program since they involve procedural search techniques that in turn require some type of discretization.

In 1995 Kennedy and Eberhart developed a new method named particle swarm optimizer(PSO), this method is based on the simulation of a block of birds and be quickly used in the optimization problems and control problems, in this paper the PSO method is used to solve the MINLP problems.

The Method of PSO

Particle swarm optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart in 1995, which is similar to other population-based evolutionary algorithms in that the algorithm is initialized with a population of random solutions, and it is unlike most of other population-based evolutionary algorithms, however, in that PSO is motivated by the simulation of social behavior instead of survival of the fittest, and each candidate solution is associated with a velocity. The candidate solutions, called particles, then "fly" through the search space. The velocity is constantly adjusted according to the corresponding particle's experience and the particle's companions' experience. It is expected that the particles will move towards better solution areas.
The steps of applying PSO method as follows:

1) Initialize a population (array) of particles with random positions and velocities in the problem space.
2) Calculate the fitness function and set the values to the pbest for each particles, and set the best value of all the particles to gbest.
3) Change the velocity and position of the particle according to equations (2) and (3), respectively:

$$v_{id}^{k+1} = w * v_{id}^{k} + c1 * rand1 * (p_{id} - x_{id}^{k}) + c2 * rand2 * (p_{gd} - x_{id}^{k}) \qquad (2)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad (3)$$

4) Calculate the fitness again and compare particle's fitness evaluation with particle's pbest. If current value is better than pbest, then set pbest value equal to the current value, and the pbest location equal to the current location in d--dimensional space.
5) Compare fitness evaluation with the population's overall previous best. If current value is better than gbest, then reset gbest to the current particle's array index and value.
6) Loop to step 3) until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

Adapting the PSO to MINLP

PSO Method originally can only used to continuous variables with the explicit constraints as the following form in (4):

$$\begin{aligned} &\min && f(X) \\ &s.t && X \in \{\ X \mid X \in R^n, X^L < X < X^H\ \} \end{aligned} \tag{4}$$

As to the MINLP problem, there have both implicit constraints and discrete variables, so two methods used to deal with these situation.

1) Penalty function being used to eliminate the implicit constraints.
2) Introduced a sigmoid ([]) function to deal with the discrete variables. In process synthesis problem, the discrete variables are always 0-1 variables, so the sigmoid functions is used to constrain the variables within [0,1], then compare the sigmoid and a random value within [0,1] to determine the discrete value.

$$\begin{aligned} &sig(v_{id}^{k+1}) = \frac{1}{1+\exp(-v_{id}^{k+1})} \\ &x_{id}^{k+1} = \begin{cases} 0 & sig(v_{id}^{k+1}) \le \rho \\ 1 & sig(v_{id}^{k+1}) > \rho \end{cases} \end{aligned} \tag{5}$$

With these two methods, the original MINLP problem (1) now being converted to the following form of equation (6):

$$\begin{aligned} \min \quad & F = f(X,Y) + \mu * \left(\sum h^2(X,Y) + \sum g^2(X,Y) * \varphi(X,Y)\right) \\ & \varphi(X,Y) = \begin{cases} 0 & g(X,Y) \le 0 \\ 1 & g(X,Y) > 0 \end{cases} \\ & X \in \{\ X \mid X \in R^n, X^L < X < X^H\ \} \\ & Y \in \{\ Y \mid Y \in \{0,\}^m \} \end{aligned} \tag{6}$$

Where μ is a very big positive number.

Now, The steps of applying PSO method in MINLP problems as follows:

1) Introduced penalty function and sigmoid function to convert the original MINLP problem of (1) to the form(6).
2) Initialize a population (array) of particles with random positions and velocities in the problem space, as to discrete variables, the position and velocities are 0 or 1.

3) Calculate the fitness function and set the values to the pbest for each particles, and set the best value of all the particles to gbest.
4) For continuous variables, Change the velocity and position of the particle according to equations (2) and (3); As to the discrete variables, update the values according to the equations (2) and (5).
5) Calculate the fitness again and compare particle's fitness evaluation with particle's pbest. If current value is better than pbest, then set pbest value equal to the current value, and the pbest location equal to the current location in d--dimensional space.
6) Compare fitness evaluation with the population's overall previous best. If current value is better than gbest, then reset gbest to the current particle's array index and value.
7) Loop to step 3) until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

Determine the validity of the PSO

In this chapter, two examples are used to prove the validity of the PSO by compare the results of PSO and results of others algorithms.

Example 1: this is a NLP problem with five continuous variables, six inequality constraints and ten max-min constraints.

$$\begin{aligned}
\min \quad & 5.3578547x_3^2 + 0.8356891x_1 x_5 + 37.293239x_1 - 40792.141 \\
s.t. \quad & 0 \le 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5 \le 92 \\
& 90 \le 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \le 110 \\
& 20 \le 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \le 25 \\
& 78 \le x_1 \le 102,\ \ 33 \le x_2 \le 45,\ \ 27 \le x_3 \le 45,\ 27 \le x_4 \le 45,\ 27 \le x_5 \le 45,
\end{aligned}$$

Table 1 show the result of PSO and results of others algorithms.

Example 2: this example has nine discrete variables, one continuous variable and two inequality non-linear constraints.

$$\begin{aligned}
\min \quad & -\sum_{i=1}^{10} x_i \\
& (x_1 + x_2 + 1)^2 (x_3 + 2x_4 + 2x_5 + 3x_6 + 3x_7 + 3x_8 + 4x_9 + 10x_{10}) - 160.5 \le 0 \\
s.t. \quad & 0.5 - x_1 + x_4 - x_5 + x_6 - x_7 + x_8 - x_9 \le 0 \\
& x_i \in \{\ 0,1\ \}, i = 1,2,\ldots,9, \qquad x_{10} \ge 0
\end{aligned}$$

Literature [4] gives the sub-optimal of –17.75 with (0,0,1,0,1,0,0,0,0,15.75), literature [11] give the global optimal of –20.55 with (0,0,1,1,1,0,1,1,1,14.55)，using PSO, we get the other global optimal of –20.44 with（0,0,1,1,1,1,1,0,1,14.55）.

From the two example can see that the PSO algorithm is a valid method to deal with NLP and MINLP problems, it produce better results than GA and GRG algorithms and almost same good with the ASA algorithm, but PSO algorithm is easy to code and use compared with GA, GRG and ASA algorithms.

Table 1：Results Compare of Example 1 with different a algorithm

Project	Referrence	GA	GRG	ASA[10]	PSO
F(x)	-30665.5	-30182.269	-30373.95	-31023.78	-31025.54
x_1	78.00	81.49	78.62	78.00	78.00
X_2	33.00	34.09	33.44	33.0016	33.00
X_3	29.995	31.24	31.07	27.0923	27.071086
X_4	45.00	42.20	44.18	44.9971	45.00
X_5	36.776	34.37	35.22	44.9106	44.9691

A example of process synthesis

This is a process synthesis problem from literature[3], during the production of chloric ethane there have byproduct of chlorohydrin being produced synchronously. Three kinds of solvent can be used to deal with chlorohydrin, which solvent should be chose with the minimal pollution to the environment and cost, this is a process synthesis problem. The mathematic form as follows of this problem:

$$\min \quad 18921.6\cdot k_1\cdot w_1 + 11826\cdot k_2\cdot w_2 + 23652\cdot k_3\cdot w_3$$

$$s.t.\quad \begin{aligned} & 0.180y_1 - 6.03 = 0.06(z_5 - k_3w_3) \\ & y_1 - 5z_6 = 0 \\ & 2y_2 + z_2 = 2(y_1 - k_1w_1) \\ & y_2 = 0.1(y_1 - k_1w_1) \\ & 2y_3 + z_4 = 2y_2 \\ & y_3 = 0.1y_2 \\ & 2z_5 = (z_2 - k_2w_2) + z_4 \\ & w_i \geq 0 \\ & \sum_{i=1}^{3} k_i = 1 \qquad k_i \in \{\ 0,1\ \} \\ & z_6 \leq 7 \qquad\qquad y_3 \leq 0.5 \end{aligned}$$

This is a MINLP problem with three 0-1 variables, ten continuous variables, eight equality constraints and three max-min constraints. The results of PSO algorithm and other algorithms shown in table 2. From table 2, it can be seen that the PSO result better than both that of in literature[3] and ASA algorithm.

Conclusion

A new method named PSO algorithm used to process sythesis problems. First introduced the panelty function to convert implicit constraints MINLP problems to explicit MINLP problems, then used sigmoid function to deal with 0-1 discrete variable, with these two steps the PSO algorithm can been used in ordinary MINLP problems. Two examples used to validate the PSO algorithm and a chemical process synthesis problem is solved bye the PSO algorithm, the results show PSO algorithm can give better results than usual algorithms such as GA, GRG and ASA.

Table 2: Results compare of case study of chlorohydrin with different algorithm

	Literature[3]	ASA[10]	PSO
k1	1	1	1
w1	30.45455	30.4495	30.4193
K2	0	0	0
W2	0	0	0
K3	0	0	0
W3	0	0	0
Y1	35	35	35
Y2	0.4545455	0.45505	0.4581
Y3	0.04545455	0.045505	0.0460
Z2	8.181818	8.1909	8.2453
Z4	0.8181818	0.81909	0.8241
Z5	4.5	4.50499	4.5347
Z6	7	7	7
Total yearly operating cost ($\$.a^{-1}$)	576,248.7	576,153	575,980

Conference

[1] Kocis G R. Grossmann I E. Global optimization of nonconvex mixed-integer nonlinear programming MINLP problems in process synthesis. Industrial & Engineering Chemistry Research, 1988,27(8):1407-1421.

[2] Metin TURKAY，Ignacio E. Grossmann. Logic-based MINLP algorithms for the optimal synthesis of process networks [J] Computers chem.. Engng. 1996,20(8):959-978.

[3] EI-HalwagiM M, Hamad A A, Garrison G W. Synthesis of waster interception and allocation network [J]. AICHE J, 1996,42(11):3087-3101.

[4] Han-Lin Li. An approximate method for local optima for nonlinear mixed integer programming problems. Computers & Operations Research, 1992,19(5):435-444.

[5] Juan M. Zamora, Ignacio E. Grossmann. A comprehensive global optimization approach for the synthesis of heat exchanger networks with no stream splits. Computers chem.. Engng. 1997,21(suppl.):s65-s70.

[6] Grossmann I E, Z. Kravanja. Mixed-integer nonlinear programming techniques for process systems engineering. Computers chem.. Engng. 1995, 19(suppl.):s189-s204.

[7] Viswanathan J., I.E. Grossmann. A combined penalty function and outer approximation method for MINLP optimization. Computers chem.. Engng. 1990, 14:1165-1184.

[8] V. Dua, E.N. Pistikopoulos, An Outer-Approximation Algorithm for the Solution of Mutiparametric MINLP problems. Computers chem.. Engng. 1998,22(suppl.)s955-s958.

[9] Wang Kefeng, Yao Pingjing, Yuan Yi, Application of genetic algorithm in mixed integer nonlinear programming(MINLP) of process system synthesis. Journal of Chemical Engineering of Chinese Universities. 1997, 11(4):400-406.

[10] Du Hongbin, Xue Dongfeng, Yao Pingjing, Improved adaptive simulated annealing algorithm

for MINLP problems in process synthesis. Journal of Chemical Engineering of Chinese Universities. 2002, 16(1):107-110.

[11] Yang Ruoli, Wu Cangpu, A simulated annealing algorithm for mixed-integer nonlinear programming problems. Control and Decision. 1996(1):63-67.

[12] Eberhart, R. C. and Kennedy .J, A new optimizer using particle swarm theory," Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995:39-43.

[13] J. Kennedy, R. Eberhart, A discrete binary version fo the particle swarm optimization algorithm, Proc. Of the 1997 conference on systems, Man, and Cybernetics (SMC'97), 1997:4104-4109.

[14] Y. Shi, R. Eberhart, A modified particle swarm optimizer, Proceedings fo IEEE International Conference on Evolutionary Computation(ICEC'98), 1998: 69-73.

[15] XUAN Guang-nan, CHENG Run-wei. Genetic Algorithm and Engineering Design [M], Beijing: Science Press, 2000.