A soft approach for solving mixed optimization models

Chunhui Xu
Dept. of Management Information Science
Chiba Institute of Technology
Chiba 275-0016, Japan
xchunhui@yahoo.co.jp

Abstract

The primary purpose of the present paper is to introduce a soft approach for solving mixed optimizations models wherein discrete and continuous variables are included.

We advocate in recent years a soft approach for solving complicated optimization models, this paper follows this approach to suggest a method for solving general mixed optimization models.

Key words: Mixed optimization models, Optimal Solution, Soft Approach, Sampling.

1 Introduction

In modeling real optimization problems, continuous and discrete variables are usually necessary in the model, such a model is called a mixed optimization model.

It is known that mixed optimization models are hard to solve except certain special cases, how to solve a mixed model without special structures has been a challenging problem.

In recent years, we are advocating a soft approach for solving complicated optimization models, see Xu(2003), Xu and Ng(2006) for details about the soft approach. This paper will employ this approach to solve mixed optimization models.

We use the following formulation for general mixed optimization models,

$$min \quad J(x, y) : (x, y) \in Z \subset X \times Y \tag{1.1}$$

where $x = (x_1, \cdots, x_{n_1}) \in R^{n_1}$ is a vector of continuous variables, $y = (y_1, \cdots, y_{n_2}) \in R^{n_2}$ is a vector of discrete variables, while $J(x, y)$ is an objective function and Z is a set of feasible solutions.

X and Y are called the domains of x and y, respectively. Let the domain of x_i be $X_i = [a_i, b_i] \subset R^1$, the domain of y_j be $Y_j = \{c_{j1}, \cdots, c_{jk_j}\}$, then the domain x and y can be expressed as $X = \Pi_{i=1}^{n_1} X_i$ and $Y = \Pi_{j=1}^{n_2} Y_j$.

The feasible set Z is usually formed by a set of equality and inequality constraints as follows,

$$Z = \{(x, y) \in X \times Y | g_j(x, y) \leq 0, h_k(x, y) = 0, j = 1, \cdots, m_1, k = 1, \cdots, m_2\} \tag{1.2}$$

The contents of this paper are arranged as follows: Section 2 introduces the soft approach for solving complicated optimization models and a two-stage process for implementing the soft approach.

Section 3 gives an algorithm for solving mixed optimization models, Section 4 concludes this paper and presents issues for further study.

2 An introduction to the soft approach

This section will first introduces the ideas in the soft resolution approach and a two-stage process for implementing it.

2.1 Ideas in the soft approach

The soft approach softens the goal in "solving" an optimization model. Among several goal softening methods, we use the one suggested in ordinal optimization, which was proposed for solving complicated stochastic optimization models, refer to Lau, Ho(1997) for details on ordinal optimization.

This goal softening method softens the goal of solving an optimization model in two dimensions:

1. Instead of looking for an optimal solution, it seeks a good enough solution.

   To define a good enough solution, we don't use the cardinal performance but the order of a solution in the feasible set. For instance, the top $k\%$ solutions may be taken as good enough solutions, where k is set by the requirements in solving a model.

2. Instead of requiring the solution to be an optimal solution for sure, it accepts a solution if it is high likely to be a good enough solution.

   When the probability with which an alternative is a good enough solution is larger than some value, say 95%, this alternative can be taken as a solution.

In other words, the soft approach agrees that a model is " solved "if an alternative which is high likely a good enough solution is obtained.

The notions of good enough and high probability are set by the requirements in solving a model. To focus on the main issues, let's take the top 1% solutions as good enough, and 95% as high probability. Therefore, when a solution is founded to be in the top 1% with a probability bigger than 95%, the model is solved.

The good enough solutions usually form a subset of Z, seeking any point of a set is much easier than seeking exactly one point, it is easy to understand that "solving" an optimization model under the soft approach should be easier. In addition, the soft approach allows failure, i.e., we do not require the solution found to be a good enough solution for sure, it is reasonable to believe that a large class of models are expected to be solvable under the soft approach.

2.2 A two-stage process

We suggest to implement the soft approach in the following two stages.

- Stage 1: Sample the feasible set Z to generate a finite subset S.

  As for the sample set S, we require the probability with which S contains at least one good enough solution to be high. Let $G \subset Z$ be a set of good enough solutions, say the top 1% solutions, we require the following condition to be satisfied:

$$p\{|S \cap G| \geq 1\} \geq q\% \tag{2.1}$$

  Consequently, the best alternative in S will be a good enough solution with a probability not smaller than $q\%$.

- Stage 2: Select the best alternative z^* from the sample set S.

  Then z^* is the solution we are looking for.

This two-stage process is illustrated in Figure 1.

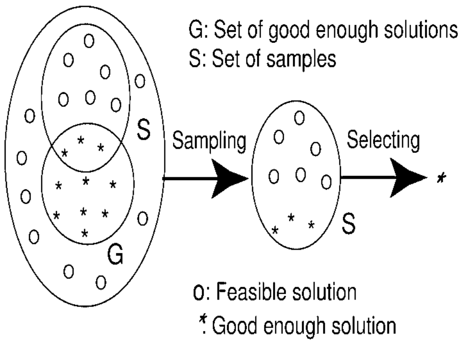

Figure 1: A two-stage process

Since S is a finite set, there is no conceptual difficulty in Stage 2. However, Stage 2 will be time consuming if S is too big. Fortunately, we will show below that a small sample set is enough to satisfy condition (2.1), no need to generate a big sample set, so Stage 2 will not take much computation time.

Suppose that we generate S by uniformly taking 1000 samples from Z, then the probability that none of the good enough solutions is included in S is $p = (1-1\%)^{1000} < 0.1\%$, consequently, $p\{|S \cap G| \geq 1\} = 1 - p > 99.9\%$. That is, a sample set with 1000 uniform samples can satisfy (2.1).

In other words, if we generate 1000 uniform samples from Z in Stage 1, and choose the best sample in Stage 2, then the best sample is one of the top 1% solutions with a probability not smaller than 99%, so the model is solved.

2.3 Features of the soft approach

From the two-stage process, we can easily see that the soft approach has the following two distinguished features.

- **No assumption about the objective function is needed.**

The objective function is used only in Stage 2. Since S is a finite and small set, we can evaluate all samples in S to get the best one, so no assumption about the objective function is needed.

- **Time for solving a model is controllable.**

We can estimate the computation time needed for solving an optimization model given the requirements in solving the model, and quantify the quality of the solution obtained within a given time limit.

Let t_1 be the time for taking one feasible sample, and t_2 be the time for evaluating one sample, then the time for solving a model, denoted by T, is equal to

$$T = |S|(t_1 + t_2) \tag{2.2}$$

where $|S|$ is the number of samples taken in Stage 1, which should be determined by the requirements in solving an optimization model and the sampling method used in Stage 1.

Suppose it is required that the solution should be one of the top $k\%$ alternatives with a probability not smaller than $q\%$, and uniform sampling is used in Stage 1, then $|S|$ can be obtained by solving the following equation.

$$p\{|S \cap G| \geq 1\} = 1 - (1 - k\%)^{|S|} = q\% \tag{2.3}$$

T is known by substituting $|S|$ into (2.2). So we can estimate the time needed for solving a model when requirements of solving a model are given.

On the other hand, we can quantify the quality of the solution obtained given a time limit for solving a model.

Let T_0 be the time limit given, then the number of samples we can take in Stage 1 is

$$|S| = \frac{T_0}{t_1 + t_2} \tag{2.4}$$

When Stage 1 takes samples with some uniform sampling method, substituting (2.4) into the following equation

$$p\{|S \cap G| \geq 1\} = 1 - (1 - k\%)^{|S|} \tag{2.5}$$

we can know the probability with which the solution is one of the top $k\%$ solutions.

Hence the soft approach not only gives a solution within a give time limit, but also tells the quality of the solution obtained.

3 An algorithm for solving mixed programming

This section presents an algorithm for solving mixed programming following the two-stage process.

To complete Stage 1, we need to decide the number of necessary samples and a method for taking these samples.

Suppose that the requirement for solving a model is: seeking one of the top $k\%$ solutions with a probability not small then $q\%$. Then the sample set S need to satisfy the following condition

$$p\{|S \cap G| \geq 1\} \geq q\% \tag{3.1}$$

Suppose that no information is available about the distribution of top $k\%$ solutions, and Stage 1 uses some uniform sampling method. Then the minimal number of necessary samples is determined by solving the following equation with $|S|$ as the variable:

$$1 - (1 - k\%)^{|S|} = q\% \tag{3.2}$$

In order not to impose any restriction on our method, we suggest to use rejection method for uniformly sampling a set described by (1.2).

The rejection method takes samples from $X \times Y$ uniformly and accepts samples if they are also in Z and reject otherwise. Because taking uniform samples from the domain is not difficult, the rejection method applies to sampling any a bounded set.

To take a uniform sample from X_i, we generate a number θ uniformly in $[0, 1]$, and let $x_i^{\prime} = \theta(b_i - a_i)$, then $x_i^{\prime}$ is a uniform sample in X_i. After generating one uniform sample for each component of x, a uniform sample of X is obtained.

To take a uniform point from Y_j, we first divide segment $[0, 1]$ into k_j equal parts, length of each part is $\frac{1}{k_j}$. And then we generate a number β uniformly in $[0, 1]$, and choose one point from Y_j in the following way,

$$y_j^{\prime} = c_{jt}, if \beta \in (\frac{t-1}{k_j}, \frac{t}{k_j}] \tag{3.3}$$

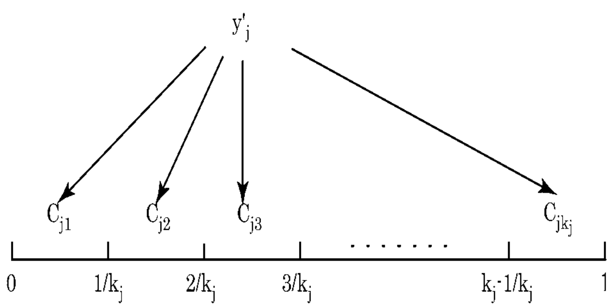

Figure 2: Uniform sampling of Y_j

As shown in Figure 2, when β is generated uniformly over [0,1], $y_j^{\prime}$ picked according to (3.3) is a uniform sample of Y_j. We can get a uniform sample of Y when one uniform sample for each component of y is picked this way.

To check if a picked sample is a feasible solution, we need only to check if all constraints are satisfied, which is easy to do. This algorithm will generate uniform samples of Z, because it does not pick any a point in $X \times Y$ more likely than other points.

Figure 3 shows a simple case of Z with one discrete variable x and one continuous variable y. $X = [a,b], Y = \{c_1, c_2, \cdots, c_5\}$. The feasible set is consisted by the four segments within the polyhedron ABCDEF.

When uniformly sampling Z with the rejection method, all points in the five segments P_1Q_1, P_2Q_2, P_3Q_3, P_4Q_4 and P_5Q_5 may be picked, but only these points within the polyhedron ABCDEF, indicated by back circles in Figure 3, will be kept, other points, indicated by white circles in Figure 3, will be rejected.

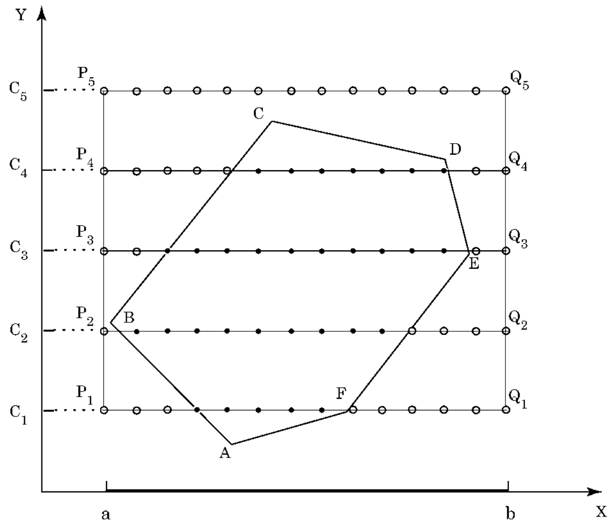

Figure 3: Uniform sampling: 2-dimension case

When necessary samples are obtained using the above sampling algorithm, we get the best alternative by evaluating all samples in S, leading to one alternative which meets the requirements in solving the model.

Remark 1 *The rejection method applies to sampling any a bounded set uniformly, the drawback is its efficiency. When the feasible set Z is a small part of the domain $X \times Y$, a lot of points will be rejected before getting a feasible point, so its efficiency will be very low.*

To uniformly sampling a set, there proposed several methods in the literature, see Rubinstein(1982), and Smith(1984) for details.

We will illustrate the above algorithm with an example.

Example Solve the following mixed optimization model with the soft method,

$$Min \quad J = (y_1y_2 + 2y_2y_3 + 3y_1y_3)\frac{e^{x_2^2} sin(10\pi x_1) + x_1^2 sin(20\pi x_2)}{1 + y_1 + 2y_2 + 3y_3} \tag{3.4}$$

where the feasible set is given by

$$Z = \{(x_1, x_2, y_1, y_2, y_3) | x_1 + x_2 \leq 1, y_1 + y_2 + y_3 \leq 2\},$$

and $x_i \in [0,1], y_j \in \{0.01, 0.02, \cdots, 0.99, 1\}, i = 1,2, j = 1,2,3$.

We solve this model with the soft method ten times, 10,000 uniform samples are taken in each time, the solutions produced are summarized in Table 1 below.

No	x_1	x_2	y_1	y_2	y_3	J
1	0.16	0.82	1.00	0.07	0.75	-1.01
2	0.14	0.85	0.69	0.26	0.94	-1.02
3	0.15	0.84	0.89	0.27	0.77	-1.16
4	0.15	0.78	0.98	0.13	0.81	-1.08
5	0.14	0.81	0.86	0.15	0.94	-1.04
6	0.15	0.82	0.75	0.66	0.59	-1.04
7	0.15	0.84	0.99	0.01	0.91	-1.15
8	0.15	0.84	0.88	0.23	0.76	-1.10
9	0.15	0.85	0.79	0.21	0.79	-1.06
10	0.15	0.84	0.83	0.10	0.90	-1.06

Table 1: Results of ten computing experiments

The best result in the ten experiment is $J = -1.16$, which is produced in the third experiment.

We plot the values of the objective function in these experiment in Figure 4.

As shown in Figure 4, all results are different, but theoretically they all are highly likely within the top 1%.

4 Conclusions

This paper proposed a soft method for solving mixed optimization models, the proposed method does not impose any assumptions on the models hence can be applied to solving various mixed optimization models.

Although the proposed method does not lead to an optimal solution, the quality of the result produced can be quantified, which differs essentially from stochastic and heuristic methods.

Further research can be conducted in the following directions:

(1) Efficient sampling methods. The rejection method applies to sampling any bounded sets, but its efficiency may be low, designing more efficient sampling methods is the key for getting a better result.

(2) Biased sampling methods. When there is some information about the distribution of good enough solutions, concentrating on the areas with a high density of good enough solutions in sampling will produce better results than uniform sampling, how to design such sampling methods is an interesting issue.

Acknowledgement

This work was supported partially by the Japan Society for the Promotion of Science under grant 17500184.

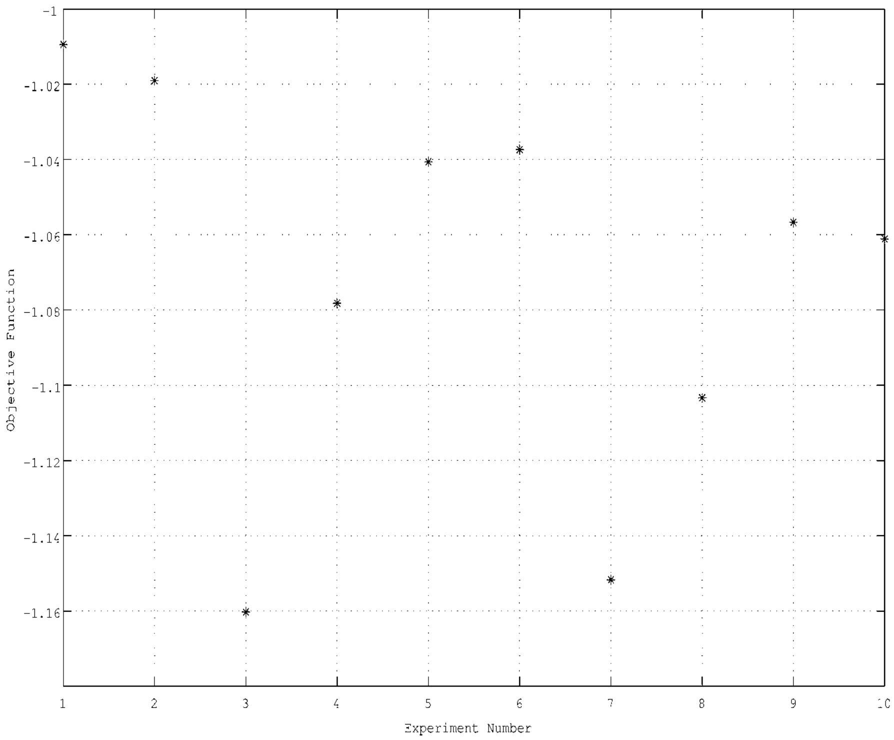

Figure 4: Results of ten computing experiments

References

[1] T.W.E. Lau, Y.C. Ho(1997), *Universal Alignment Probabilities and Subset Selection for Ordinal Optimization*, JOTA, 93(3), 455-489.

[2] R.Y. Rubinstein(1982), *Generating Random Vectors Uniformly Distributed Inside And On The Surface Of Different Regions*, European Journal Of Operational Research,10 (2), 205-209.

[3] R.L. Smith(1984), *Efficient Monte-Carlo Procedures For Generating Points Uniformly Distributed Over Bounded Regions*, Operations Research, 32 (6): 1296-1308.

[4] C. Xu(2003), Soft Approach for Solving Hard Optimization Problems, *Proc. of the Fall National Conference of Japan Society for Management Information*, 74-77.

[5] C.Xu, P.Ng(2006), *A soft approach for hard continuous optimization*, European Journal of Operational Research, Vol.173, No.1, 18-29, 2006.