Neighborhood Search Method for Non-binary ILP Problem

천 영 수*(Chun, Yung Soo)

〈개 요〉

이 논문은 정수선형계획법(ILP) 문제의 해를 구하는 새로운 기법을 소개하고 있다. 이 기법을 여기에서는 "인근탐색기법"이라고 명명하였다. 이 분석기법은 완화된 (즉, 정수 제한을 없앤) ILP문제의 최적해을 둘러싼 인근영역에 존재하는 정수해를 탐색함으로써 정수 ILP문제에 대한 준(準)최적해를 구하는 방법을 제공한다. 이 탐색과정은 원래의 ILP문제를 계산부담이 적은 이진 ILP문제로 전환한다. 그러나 인근영역이 실현가능한 정수해를 내포하지 않을 수도 있다. 그러므로 인근영역이 실현 가능한 정수해를 내포할 확률을 높일 수 있는 조치가 필요하다. 그리고 여기에서 제시된 규칙들은 이 기법의 계산 효율을 높일 수 있다.

Ⅰ. Introduction

Consider the following integer linear programming problem;

maximize $f = cx$ subject to $Ax \leq b$, $x \geq 0$, integers. ……………………………………………(P)

Assume the optimal solution to this ILP problem exists. Think of the relaxed LP problem where the integrality condition is relaxed. Let x' be the optimal solution to the relaxed LP problem, $f' = cx'$

As Fig. 1 conspires, it is fairly likely that some integer-valued vertices of the size-one cube which contains x inside are feasible. One of them would have the highest objective value. Let's call this point a "near-optimal solution in this cube", and denote it by x^*. x^* may happen to be the optimal solution to the original ILP problem or have an objective

1) 경상대학 경영학과

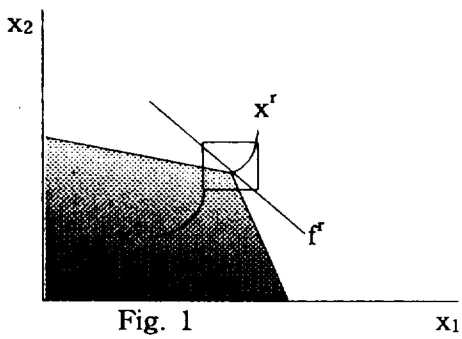

Fig. 1

value close to the optimal value. As a matter of course, it is also possible none of the vertex is feasible.

Ⅱ. Terminologies and notations

x^r=r−optimal solution: optimal solution to the relaxed problem,

x^{**}=optimal solution: optimal solution to the original ILP problem,

x^*=n−optimal solution: the best solution to the original ILP problem among the solution points inside a specified neighborhood.

N1(x^r)=the first neighborhood of x^r: the set of points whose i−th element has the value [x^r] or [x^r] + 1,

N2(x^r)=the second neighborhood of x^r: the set of points x with xj = [xj^r] −1, [xj^r], [xj^r] +1, or [xj^r]+2,

r−feasible region: set of feasible solution points to the relaxed LP problem,

[xj^r]=the largest integer ≤ xj

xj^r=[xj^r]+ej where 0 ≤ ej < 1,

[x^r] = ([$x1^r$], [$x2^r$], 7 · · · · · , [xn^r])^T

e =(e1, e2, · · · , e_n)^T

1 =(1, 1, · · · , 1)^T

y =(y_1, y_2, · · · , y_n)^T where yj = 0−1 variable.

Now we can use the following short expressions.

$x^r=[x^r]+e$

$N_1(x^r)=\{x \mid x=[x^r]+y\}$

$N_2(x^r)=\{x \mid x=[x^r]-1+y_1+2y_2+3y_3\}$.

In case of 2−dimensional points, neighborhoods of x^r can be represented as shown in Fig.2 and 3. Because we are interested only in the neighborhoods of x^r, dropping off (x^r) from the notations, $N_1(x^r)$ or $N_2(x^r)$, will cause no confusion. Also, N may be used as a short notation of N1 or Nk for any k.

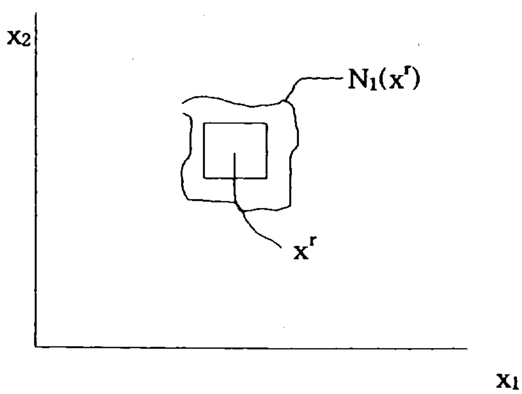

Fig. 2

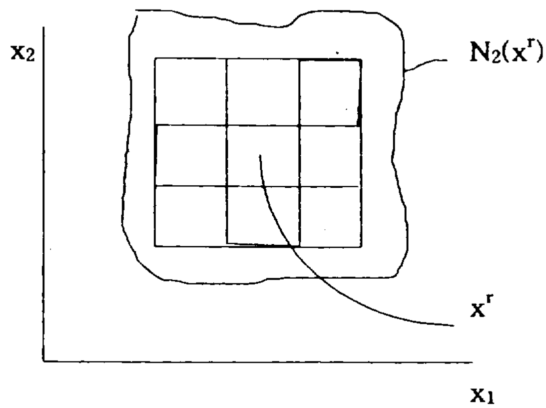

Fig. 3

Ⅲ. Reduction to Binary Problems

It seems sensible to try to search for x^* in N_1 first. In other words, we are looking for the best feasible solution to the original ILP problem (P) among the points $x=[x^r]+y$. Therefore, search for x^* in N_1 results in the problem;

maximize $f=cx=c([x^r]+y)=c[x^r]+cy$

subject to $Ax=A([x^r]+y)=a[x^r]+Ay \leq b$.

Since $[x^r]$ is s constant vector in this problem, we get the following problem;

maximize $\Delta f=cy$ ·· (B_1)

subject to $Ay \leq b'$ where $b'=b-A[xr]$

This problem is amazingly similar to the original problem. Non−binary integer variables in the original problem is substituted by binary variables. All the parameters moved into this derived problem unchanged except b. Calculating b' might not be computationally burdensome. If we find the optimal solution y^* to the problem (B1), our desired near optimal solution will be;

$$x^* = [x^r] + y^* \text{ and } f^* = c[x^r] + cy^*$$

Similarly, search for x^* in N_2 results in the problem;

$$\text{maximize } f = cy_1 + 2cy_2 + 3cy_3 \quad \cdots\cdots \quad (B2)$$

$$\text{subject to } Ay1 + 2Ay_2 + 3Ay_3 \leq b' \text{ where } b' = b - A([x^r] - 1)$$

Provided the optimal solution $y1^*$, $y2^*$ to problem (B_2) is found,

$$x^* = [xr] - 1 + y_1^* + 2y_2^* + 3y_3^*$$

$$f^* = c([x^r] - 1 + y_1^* + 2y_2^* + 3y_3^*).$$

Note that problem (B_2) has 3n variables and m constraints.

Ⅳ. Properties of the neighborhood

We understand that the solution method for binary problems is well established, and is computationally much less burdensome than that for non−binary ILP problems is. Therefore, we can turn to the problem (B_1) as a quick way to obtaining an approximate optimal solution to our original problem (P).

But, as pointed out at the outset, we cannot remove the possibility that N_1 fails to contain any feasible solution. Also, there's no guarantee $x^* = x^{**}$. From theoretical point of view, these two kinds of problems cannot be a source of nuisance because we can always find $x^* = x^{**}$ in N by extending the size of N large enough. But, extending the size of N gets computationally unattractive very quickly as the number of variables increases. We need to utilize the properties of neighborhood which determine the capability of neighborhood to contain feasible solutions with high objective values.

Generally speaking, the values of parameters in the problem P play all the role in determining the above−mentioned capability of the neighborhood. We also know certain structures of the values of parameters determine the geographical properties of the r-feasible region and the plane associated with some value of the objective function, which in their turns, determine the capability of the neighborhood.

Now, we need more definitions for a concise explanation.

Objective plane $=\{x \mid cx = f$ for a certain value of $f\}$

Type I failure: the event that the concerned neighborhood doesn't contain any feasible solution,

Type II failure: the event that $x^* \neq x^{**}$.

The terms "Type I success" and "Type II success" can be used to express the complementary events.

4.1 Factors determining Type I failures

4.1.1. Location of xr relative to integer points

If the optimal corner point xr lies so that ej $\fallingdotseq 0$ for some j, the possibility of Type I failure would be higher than otherwise.

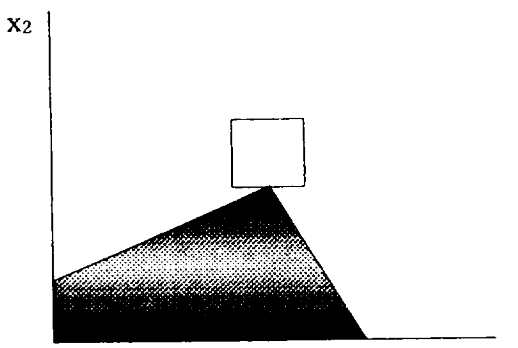

Fig. 4

4.1.2. Angle of the r-feasible region at x'

If there's a pair of binding constraints which make an acute angle of relaxed feasible region, the possibility of type I failure becomes high. (Only binding constraints are related to the angle at x')

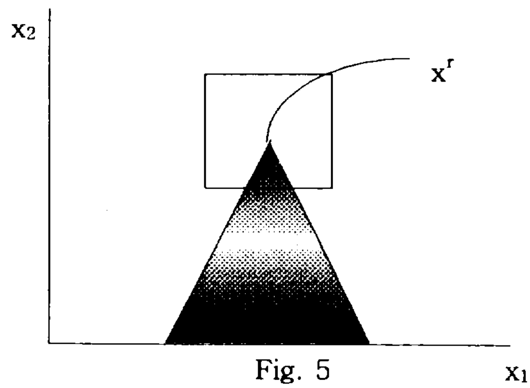

Fig. 5

4.1.3. Thickness of r−feasible region at x^r

If r−feasible region is thin at xr with respect to some constraint plane, then possibility of type I failure increases.

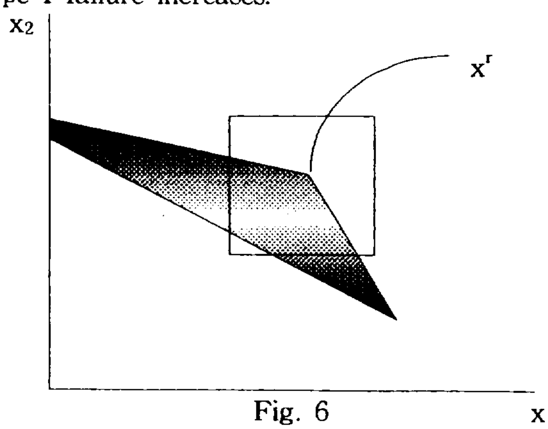

Fig. 6

Here, thickness means the distance from x^r to constraint plane along the normal line to that plane. (Only nonbinding constraints are related with thickness.)

4.2 Factors determining type II failure

− Parallelity of the objective plane with respect to a constraint plane −

A necessary condition for a large discrepancy between xj** and xj* is that the objective plane is almost parallel to some constraint plane(s).

Above statement is derived from geometrical insights. Analytical proof seems to be impossible because, after all, both type of failure is determined by the value of parameters. Never-

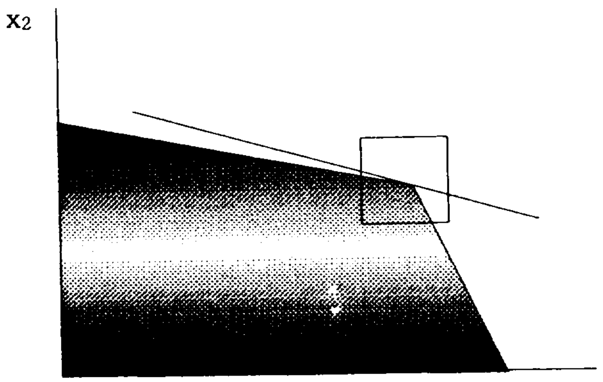

Fig. 7

theless, above considerations lead to the rules which will be used to raise the possibility of both type of success of the neighborhood.

V. The Heuristic to Raise the Success of Neighborhood

5.1. Shift of Neighborhood

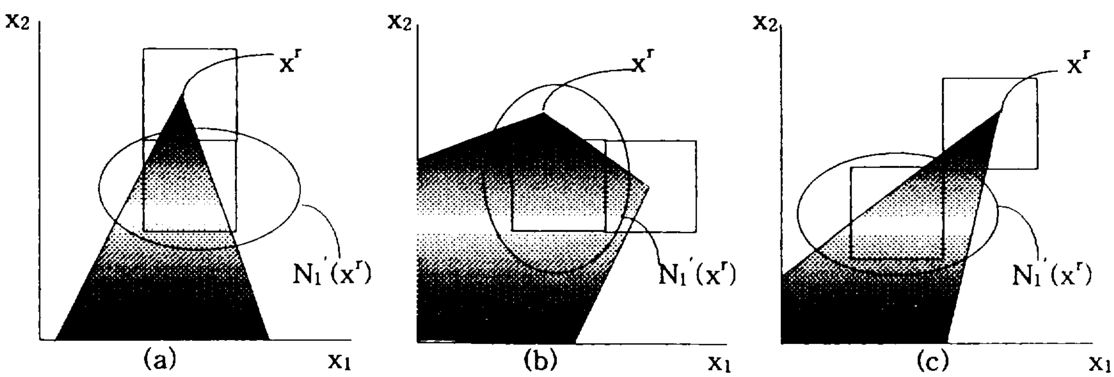

Fig. 8

Step 1 : Find n adjacent corner points d^k, $k = 1, \cdot \cdot \cdot, n$.

Step 2 : Compare with $\min_k\{dj^k\}$ and $\max_k\{dj^k\}$

Step 3 : Search for xj^* among $[xj^r]+1$ and $[xj^r]+2$ if $xj^r \leq \min_k\{dj^k\}$

Search for xj^* among $[xj^r]$ and $[xj^r]-1$ if $xj^r \geq \max_k\{dj^r\}$

Because r−feasible region is a convex set, xj^r will be the smallest feasible value of xj if all the adjacent points lie to the direction of greater xj, i. e. $xj^r \leq dj^k \ \forall k$, i.e., $xj^r \leq \min_k\{djk\}$. There's no use of searching for x^* where $xj < xj^r$. Reversed case can be proved similarly.

The definition of neighborhood set earlier no longer holds. But we'd like to retain the terminology "neighborhood" with the adjective "shifted" in front of it.

The benefit of shifting neighborhood region is in increasing the chance of type I success r ather than in saving efforts which might be spent on the search in a wrong region otherwise. This shifting alleviates the trouble caused by acute angles at x^r

This rule becomes completely ineffective when all the coefficients in binding constraints are >0, since every pair of constraints form an obtuse angle at x^k in such situations.

Modifications of the binary problem due to shifting

The points x to search among becomes

$x = [x^r] + \delta + y$

where $\delta = (\delta 1, \ldots, \delta n)^T$

with $\delta = 1$, if $xj^r \leq dj^k \ \forall k$

-1, if $xj^r \geq djk \ \forall k$

0, otherwise

So, we get the problem for the search in N_1' as;

maximize $f = cy$

subject to $Ay \leq b''$ where $b'' = b - A([x^r] + \delta)$

5.2. Removal of redundant constraints (Rule 2)

Drop off the nonbinding constraint, if $\sum_{j \in J_i^+} a_{ij} \leq b_i = (b - A[x^r])_i$

where $J_i^+ = \{j \mid a_{ij} > 0\}$

Proof 1

$$\sum_{j \in J_i^+} a_{ij} \leq b'$$

$$\Rightarrow \sum_{j \in J_i^+} a_{ij} y_j \leq b'$$

$$\Rightarrow \sum_{j\in J_i^+} a_{ij}y_j + \sum_{j\in J_i^-} a_{ij}y_j \le b_i$$

$$\Rightarrow \sum_j a_{ij}y_j = a^i y = (Ay)_i \le b_i \quad \forall y$$

That is, the constraint i is always satisfied if $\sum_{j\in J_i^+} a_{ij}y_j \le b$.

Proof 2

Consider Fig. 9. Let x^m = the point in N_i which is most likely to dissatisfy constraint i and x^f = the foot of the line drawn from x^r perpendicularly to get g_i−plane. Then, from geometrical insight, we know that.

$$x_j^f \le x_j^r \Rightarrow x_j^m \le x_j^r \Rightarrow x_j^m = [x_j^r].$$

$$x_j^f > x_j^r \Rightarrow x_j^m > x_j^r \Rightarrow x_j^m = [x_j^r] + 1.$$

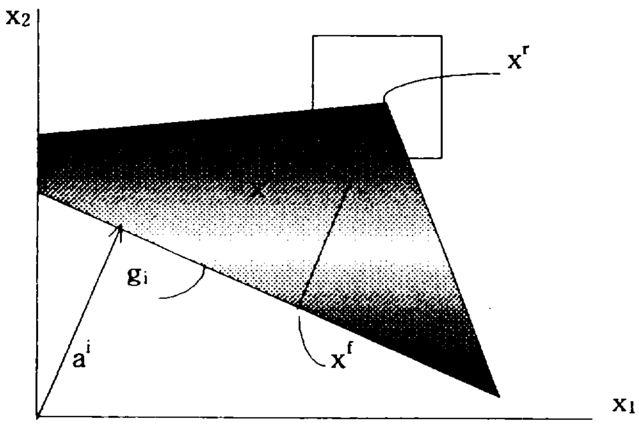

Fig. 9

Therefore, $x^m = [x^r] + \delta$ where $\delta = (\delta_1, \cdots, \delta)^T$ with $\delta_i = 1$, if $x_j^f > x_j^r$ 0, otherwise

Since both $x^f - x^r$ and a^i are perpendicular to g_i−plane, $x - x^r // a^i$

Hence $x^f - x^r = k(a^i)^T$ ………………………………………………………………………………………… (1)

Since x^f is on the g^i−plane, $a^i x^f = b^i$ ……………………………………………………………… (2)

Fome (1) and (2), $a^i(k(a^i)^T + x^r) = b_i$.

$$\therefore k = \frac{b_i - a^i x^r}{a^i(a^i)^T} = \frac{S_i}{a^i(a^i)^T} > 0 \text{ since } S_i > 0,\ a^i(a^i)^T > 0$$

From (1), we lnow that $x^f - x^r$ has the same direction as $(a^i)^T$, since $k>0$.

$$\therefore\ a_{ij} > 0 \Leftrightarrow x_j^f - x_j^r > 0 \Leftrightarrow \delta_j = 1,$$

$$a_{ij} \le 0 \Leftrightarrow x_j^f - x_j^r \le 0 \Leftrightarrow \delta_j = 0.$$

By definition of x^m, constraint i will be inactive if $a^i x^m \le b_i$

$$\Rightarrow a^i([x^r]+\delta) \le b_i$$

$$\Rightarrow a^i\delta \le b_i - a^i[x^r] = (b - A[x^r])_i = b_i'$$

$$\therefore a_i\delta = \sum_{j\in J_i^+} a_{ij}(1) + \sum_{j\in J_i^-} a_{ij}(0) = \sum_{j\in J_i^+} a_{ij}$$

because $\sum_{j\in J_i^+} a \le b_i$ implies constraint i is inactive

Implication of Rule 2.

As geometrical proof shows, rule 2 is related with the thickness of r-feasible region at x'. If r-feasible region is thick enough at x' with respect to a nonbinding constraint i, the whole neighborhood will lie in the area satisfying constraint i. We can stop paying attention to constraint i by removing it.

5.3. Reduction of the number of variables (Rule 3)

Set $y_j = 0$ if $a_{ij} > b_j' - \sum_{j\in J_i^-} a_{ij}$ for some i,

Set $y_j = 1$ if $a_{ij} < -(b_j' - \sum_{j\in J_i} a_{ij})$ for some i.

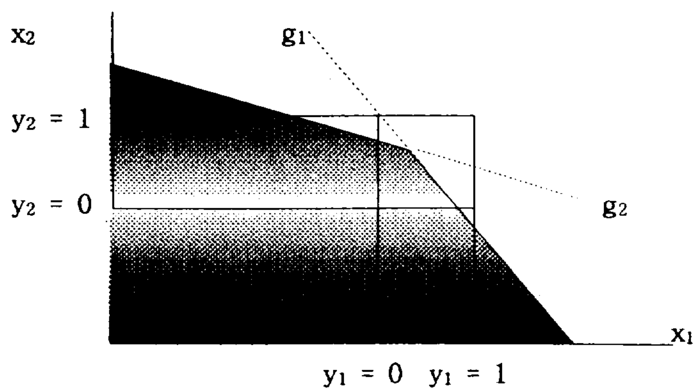

Fig. 10

It's possible one side of the neighborhood cube lies out of the r-feasible region by going beyond a certain constraint plane. In each cases, there's no reason to waste effort in searching for x* among the points on this side because it is sure that x* exists on the opposite side. So, we can fix the value of the variable which determine this pair of sides. In Fig. 10, we can set $y_1 = 0$ since the whole side determined by $y_1 = 1$ lies beyond g1 − plane. We can set $y_2 = 0$ for a similar reason.

Derivation of the rule

Set $y_k = \sim 0$, if (1) the best y with $y_k = 1$ violates constraint i, and (2) setting yk = 0 helps y satisfy constraint i. From (2), we know a_{ik} should be>0. Constraint i can be rewritten in the form of $a_{ik}y_k + \sum_{j\in J_i^+, j\neq k} a_{ij}y_j + \sum_{j\in J_i^-} a_{ij}y_j \leq b_i'$. Obviously, the best y will be such that $y = 0\ j\neq k\ \forall\ j\in J_i^+$ and $y = 1\ \forall\ j\in J_i^+$.

So, we can set $y_k = \sim 0$ if (1) the best y with $y_k = 1$ violates constraint i and (2) setting $y_k = 0$ helps y satisfy constraint i. From (2). a_{jk} should be > 0. Constraint i can be rewritten in the form of

$$a_{ik}y_k + \sum_{j\in J_i^+, j\neq k} a_{ij}y_j + \sum_{j\in J_i^-} a_{ij}y_j \leq b_i'.$$

Obviously, the best y will be such that

$y_j = 0\ j\neq k\ \forall\ j\in J_i^+$ and $y_j = 1\ \forall\ j\in J_i^+$

So, we can set $y_k = 0$ if $a_{ik} + \sum_{j\in J_i^+} a_{ij} > b_i'$.

Therefore $a_{ik} > b_i' + \sum_{j\in J_i^+} a_{ij}$ for some i.

Similarly, set $y_k = 1$, if (1) the best y with violates constraint i and (2) setting $y_k = 1$ helps y satisfy constraint i. From (2) a_{ik} should be < 0. Constraint i can be expressed as

$$a_{ik}y_k + \sum_{j\in J_i^+} a_{ij}y_j + \sum_{j\in J_i^-, j\neq k} a_{ij}y_j \leq b_i'.$$

The best y is such that $y_j = 0\ \forall\ j\in J_i^+$ and $y_j = 1\ \forall\ j\neq k, \in J_i^+$

So, we can set $y_k = 1$ if $\sum_{j\in J_i^-, j\neq k} a_{ij} > b_i'$

i. e., $\sum_{j\in J_i^+} a_{ij} - a_{ik} > b_i'$

i. e., $a_{ik} < -(b_i' - \sum_{j\in J_i^+} a_{ij})$ for some i.

5.4. Choice of the size of the neighborhood

All the explanations and rules presented thus far on the use of N_1 are applicable to the search in N_2. The only difference is the heavier computational burden for the search in a larger neighborhood. We tried to raise the capability of the neighborhood in an attempt to make N_1 more satisfactory. But use of N_2 may become inevitable sometimes. We can consider searching in an even larger neighborhood. The decision on the size of N may be determined by the properties of the problem and the user's attitude toward the capability to hold a near optimal solution and computational burden related to the neighborhoods of different size.

It would not be unusual our original ILP problem itself contains few binary variables. Such variables would reduce the advantage of problem (B_1) over problem (P). Anyhow, we don't need to spread out the region to be searched, since $x_i^{**}=0$ or 1 obviously. Therefore, if we decide to use N_2 for the ILP problem with a few binary variables, then the neighborhood to be searched will have the shape of rectangular cube instead of square cube.

It is also possible that a few variables have relatively large rates of contribution to the objective value. For such variables, we may want to search for the near−optimal value over a rather broad range of their integer values. Once again, we don't need to stick to square−cube neighborhood region.

5.5. Proper Sequence of Decisions

The appropriate sequence of decisions would be;

a. decision on the size of N

b. decision on the location of N (Rule 1)

c. removal of inactive constraints (Rule 2)

d. reduction of the number of variables (Rule 3)

VI. Evaluation of the Neighborhood Search Method

The usefulness of the Neighborhood Search Method cannot be commented on without the information on the possibility of type I and II successes of the neighborhood. We believe that only experience can provide this information. But we can tell roughly about the possibilities of both types of successes in case of a certain category of problems.

1) Type-I Failure

As stated earlier, the chance of type-I failure is dependent on the location of the relaxed solution x^r, and the angles and thickness of r-feasible region at x^r. Shifting the neighborhood will reduce the possibility of type I failure related to the unfortunate location of x^r.

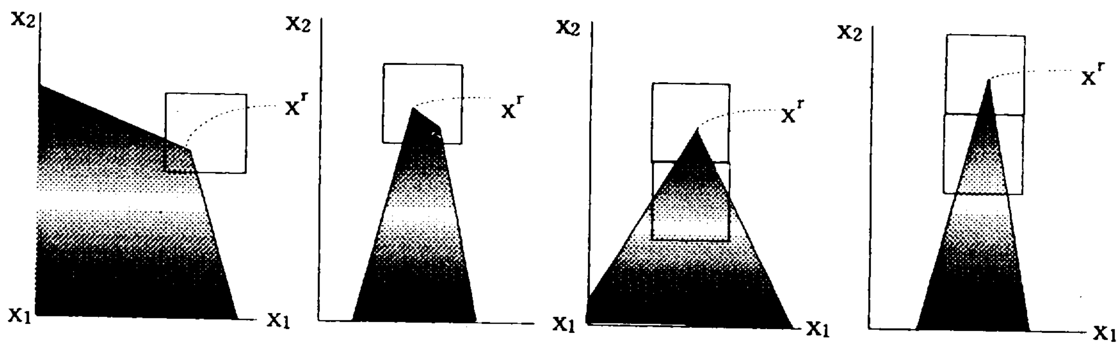

Fig. 11

As shown in Fig. 11(a), we don't need to worry about the location of x^r if (1) xj^r is not an extreme feasible value of xj for all j and (2) all the adjoint points lie outside N. Of course, Rule 1 doesn't lead to a shift of N in this situation. As shown in Fig. 11(b), the fact that xj^r is not an extreme feasible value of xj for all j doesn't guarantee that N contains at least a feasible solution. As shown in Fig. 11(c), shifting N helps N contain a feasible integer solution. But shifting N cannot remove the chance of type I failure completely (See Fig. 11(d)). After all, shifting N helps a lot unless r-feasible region is acute at xr and the r-feasible region is directioned to unfortunately avoid integer points in N.

The thickness of r−feasible region at xr is utilized to render a rule about the removal of inactive constraints. But the hazard of type I failure due to thinness is not cured. We just rely on luck.

The angles of r−feasible region formed by binding constraints are calculated by the formula

$$\cos\theta_{ik} = \frac{a^i(a^k)^T}{||a^i|| \cdot ||a^k||}$$

where θ_{ik}= the angle formed by constraint i and k.

Notes: a) $0<\cos\theta_{ik}<1$ ⇔ constraint i and k form an obtuse angle.

b) $\cos\theta_{ik}<1$ ⇔ constraint i and k form an perpendicular angle.

c) $-1<\cos\theta_{ik}<0$ ⇔ constraint i and k form an acute angle.

d) $\cos\theta_{ik}=0$ or 1 is incompatible with x^r because two parallel constraints cannot be binding at the same time.

Some problems such as transformation problems have only positive constraint coefficients making the angle at xr obtuse. For such problems, the possibility of type I success is likely to be high

2) Type II Failure

Let's all xj to be stable if $x_j^* - x_j^{**}$, and unstable if $|\ x_j^* - x_j^{**}\ | >> 0$. We know that as the objective plane comes to be more parallel to axis x_j, x_j becomes more unstable and chance of type II failure increases. Let θ_j be the angle between axis j and the normal line to the objective plane. Then

$$\cos\theta_j = \frac{Ce_j}{||a^i|| \cdot ||a^k||} = \frac{C_j}{||C||}$$

Therefore, Cj/||C|| → 0, i. e., $\theta_j \rightarrow \pi/2$ implies

f−plane gets more parallel to axis j.

⇒ x_j becomes more unstable

⇒ chance of type II failure increases.

On the contrary, $C_k = 0\ \forall k \neq j \Rightarrow |C_j|/||C|| = 1$ ⇒ f-plane ⊥ axis j ⇒ x_j becomes stable.

As the number of variables increases, the weight of Cj relative to ‖C‖ tends to decreases making xj more and more unstable. So, we can generally say that the chance of type II failure increases as the size of the problem increases. But, the sacrifice caused by accepting x^* as an approximation to x^{**} can sometimes be ignored considering the small contribution of xj to be objective value, |Cj|/||C||.

3) Test for the Optimality of x^*

Step 1 : Add the constraint $Cx >= f^* = Cx^*$ to problem (P). Remove all the nonbinding constraints in the final tableau.

Step 2: Find all the adjoint corner points of xr which bump into the plane $Cx = f^*$. Call them $c^k = 1, \cdots, n$.

Step 3: Think of the set

$$Z = \{z \mid z = (z_1, \cdots, zn)^T, z_j \text{ integer } \min_k\{d_j^k, x_j^k\} \leq z_j \leq \max_k\{d_j^k, x_j^k\}\}$$

If $Z \subset N$, then $x^* = x^{**}$.

We know x^{**} should satisfy the condition $Cx^{**} >= Cx^* = f^*$. So, $Cx >= f^*$ forms a constraint on the set of candidate points for x^{**}. As illustrated in Fig. 12, x^{**} must lie in the cross-shaded region (polygon BCDFG), and thus, if more loosely confined, in the shaded region(△ACE). The region represented by △ACE is the polytope with vertex x' and d^k, $k=1, \cdots, n$.

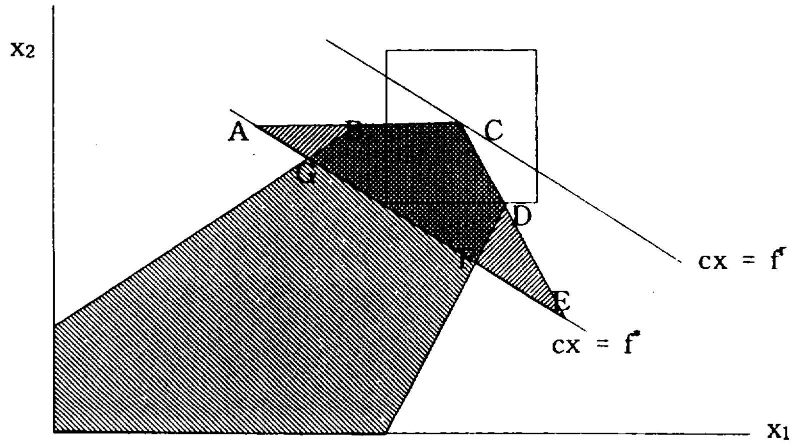

Fig. 12

Therefore x^{**} should satisfy $\min_k \{d_j^k, x_j^k\} \le x_j^{**} \le \max_k \{d_j^k, x_j^k\}$ for j = 1, ⋯ ,n. If all the integer points satisfying this condition are located in N, then $x^* = x^{**}$ because x^* is the best integer feasible point in N (Note: This test criterion may be powerful. It is only of theoretical interest)

7. Conclusion

As shown already, we can predict that the chance of type II failure increases as the number of variables increases. Therefore, we better use the Neighborhood Search Method as a quick method to obtain an approximate optimal solution x^* and f^*. If f^* is close enough to fr, we can either (1) take x^* as our final decision or (2) use f^* as the initial lower bound for the branch and bound method.

It sounds very good. But final evaluation should be reserved until a fair amount of experience produces relevant statistics regarding the type I and II successes and on the power of the rules which consist a part of this method. Statistics of interest should include (1) ratio of computation time taken by the solution of problem B_1 to that corresponding to problem P, (2) relative frequency of both type I and II success, and (3) the ratio of f^*/f^{**}

References

1. F. S. Hillier & G. J. Liebermann, "Introduction to Operations Research 3rd Ed.", Holden-day Inc., Sanfrancisco, 1980
2. H. A. Taha, "Operations Research, An Introduction, 3rd Ed." MacMillan, New York, 1982
3. H. A. Taha, "Integer Programming", Academic Press, New York, 1975
4. D. R. Plaine & C. McMillan Jr., "Discrete Optimization", Prentice Hall, New Jersey, 1971
5. G. Strang, "Linear Algebra and its Applications 2nd Ed.", Academic Press, New York, 1980.
6. A. Schrijver, "Theory of Linear and Integer Programming", John Wiley, New York, 1998
7. L. A. Wolsey, "Integer Programming", John Wiley, New York, 1998
8. G. L. Nemhauser & L. A. Wolsey, "Integer and Combinatorial Optimization", John Wiley, New York, 1988
9. J. E. Beasley, "Advances in Linear and Integer Programming", Oxford, London, 1996