

都市 I の訪問順番は 1 つに限られるので

$$\sum_{k=1}^n U(I, K) = 1 \quad (I=1, 2, \dots, n)$$

訪問順が K 番目になる都市は 1 都市に限られるので

$$\sum_{I=1}^n U(I, K) = 1 \quad (K=1, 2, \dots, n)$$

$U(I, K) = 1, U(J, K+1) = 1$ ならば $Y(I, J) = 1$ であるから

$$U(I, K) + U(J, K+1) - Y(I, J) = 1 \quad \begin{matrix} (I=1, 2, \dots, n) \\ (J=1, 2, \dots, n) \\ (K=1, 2, \dots, n-1) \end{matrix}$$

$Y(I, J)=1$ のときに限り $U(J)=U(I)+1$ であるから

$\left. \begin{matrix} U(J) > U(I)+1 \\ U(J) < U(I)+1 \end{matrix} \right\}$ の不等号の大なる方に $\text{bigM} \cdot \{1 - Y(I, J)\}$ を加えて

$$U(J) > U(I)+1 + \text{bigM} \cdot \{1 - Y(I, J)\} \quad (I=1, 2, \dots, n)$$

$$U(J) + \text{bigM} \cdot \{1 - Y(I, J)\} < U(I)+1 \quad (J=1, 2, \dots, n)$$

ただし、 $U(1)=1, J \neq 1$ とする

上下界

bounds

変数 $Y(I, I)$ を作ってしまったときは、 $Y(I, I) = 0 \quad (I=1, 2, \dots, n)$ を書く。

$Y(I, I)$ は不要な変数であるから最初から作らない方がよい。

変数型

binary

$$Y(I, J) \quad (I=1, 2, \dots, n)$$

$$(J=1, 2, \dots, n)$$

~~$$U(I, K) \quad (I=1, 2, \dots, n)$$~~

~~$$(K=1, 2, \dots, n)$$~~

general

$$U(I) \quad (I=1, 2, \dots, n)$$

終端

end

以上の定式化データを整数計画法プログラムにインプットすると、都市数 10 個程度の小さい問題ならば正解が得られる。正解が得られるということは、制約条件は正しく、これで十分あり、論理的にはこれ以上追加すべき条件はないはずであるが、整数計画法プログラムの特徴として、さらに制約条件を追加すると計算時間が大幅に変わる。制約を増加すると時間短縮になるとは限らないが、間違いなく時間短縮できるのは、出発都市を決めてやることである ($U(1, 1)=1$ など)。最終的に輪になって繋がった解が得られるので、出発地はどこでもよいはずで、実際どこにしても同じ解が得られる。

この他に往復経路の禁止 $Y(I, J) + Y(J, I) = 1$

三角経路の禁止 $Y(I, J) + Y(J, K) + Y(K, I) = 2$

四角経路の禁止など考えられるが、時間短縮になるか、逆効果となるかは、やってみないと分からない。

手順 2 ; 各 1 行の黒丸と黒丸の間の空欄に、白丸を書き込む。

表 - 2

	E01	E02	E03	E04	E05	E06	E07	E08
N01								
N02								
N03								
N04								
N05								
N06								
N07								
N08								
N09								
N10								
N11								
N12								
丸の数	4	7	10	9	7	7	6	4

手順 3 ; 黒白関係なく、各列の丸の数を数えて最下欄に書き込む。

これが、このエレメントのときのウェーブフロント数である。

解析モデルのウェーブフロント数という場合、最大値（上表であれば E03 のときの 10）を指すことが多いが、一般には、エレメント番号毎に決まるものである。

ウェーブフロント数の最適化というのは、エレメント番号を付け直し（すなわち、上表の列を入れ替えて）白丸の数をなるべく減らそうとするものである。実際に上表を列に沿ってハサミで切り、並べ替えてみたとき、黒丸は消せないが、並べ替えて黒丸に挟まれなくなった白丸は消すことができるので、白黒色を区別しているのである。

この例は、エレメントが三角と四角しかないが、二角でも二十角でも、三次元要素でも数え方は同じ手順でよい。一般にウェーブフロント数は FEM とは関係なく、点集合 N とその部分集合群 E が定義されていれば発生する問題である。

それでは、この最適化を整数計画法で定式化してみよう。

定数データ (エLEMENT数 m 、ノード数 n とする)

ノード・ELEMENT接続表 $d(I,J)$ ($I=1,2,\dots,n$)
 ($J=1,2,\dots,m$)

表 - 1 の黒丸を 1、空欄を 0 としたもの。

各「行」の黒丸の合計 $M(I)$

変数

$U(J,K) = 1$ 元のエLEMENT J が K 番目と付番されたとき
 $= 0$ " K 番目でないとき

$Z(I,K)$: $d(I,J)$ 表 1 の列に、ある並べ替え $U(J,K)$ をしたときの黒丸表
 ($I=1,2,\dots,n$)
 ($K=1,2,\dots,m$)

$W(I,J)$: $Z(I,K)$ の黒丸と黒丸の間に白丸を書き込んだ変数。黒または白い丸のところが 1、その他は 0 である。

黒丸表 $Z(I,J)$ が与えられたとき、白丸を書き込んだ表の変数 $W(I,J)$ を作り出すのは意外と難しい。次のような補助変数を使うことにする。

	$X(I,J)=0$	$X(I,J)=1$							
第 I 行									
					$Y(I,J)=1$		$Y(I,J)=0$		

すなわち、 $Z(I,J)$ の最初の黒丸の手前までは 0、それ以後を 1 とした変数 $X(I,J)$ と最後の黒丸より右は 0、それ以前は 1 とした変数 $Y(I,J)$ を用意する。

以上の準備により、

目的関数

~~WF 数の最大値を最小にするのは難しいので総和を最小化することにする。~~

minimize WFN

制約式

subject to

$$\sum_{I=1}^n W(I,J) = WFN \quad (J=1,2,\dots,m)$$

(FEM 解析が目的の WF 数最小化には総和を最小化の方が実用的である。)

エレメント J には、どれか 1 つの番号が振られるので、

$$\sum_{K=1}^m U(J, K) = 1 \quad (J=1, 2, \dots, m)$$

番号 K が振られるのは、どれか 1 エレメントに限られるので

$$\sum_{J=1}^m U(J, K) = 1 \quad (K=1, 2, \dots, m)$$

U(J,K)変換により、d(I,J)の黒丸は Z(I,K)に移るので

$$\sum_{J=1}^m U(J, K) \cdot d(I, J) = Z(I, K) \quad (I=1, 2, \dots, n)$$

$$(K=1, 2, \dots, m)$$

X(I,J) は、Z(I,J) より大きく、右側へ単調増加だから

$$X(I, J) \geq Z(I, J) \quad (I=1, 2, \dots, n)$$

$$(J=1, 2, \dots, m)$$

$$X(I, J) \geq X(I, J-1) \quad (I=1, 2, \dots, n)$$

$$(J=2, 3, \dots, m)$$

Z(I,J) の K + 1 列より後の列の和が M(I)ならば X(I,K)=0 でなければならないから

$$\sum_{J=K+1}^m Z(I, J) + X(I, K) = M(I) \quad (I=1, 2, \dots, n)$$

$$(K=1, 2, \dots, m)$$

Y(I,J) も同様にして

$$Y(I, J) \geq Z(I, J) \quad (I=1, 2, \dots, n)$$

$$(J=1, 2, \dots, m)$$

$$Y(I, J) \geq Y(I, J-1) \quad (I=1, 2, \dots, n)$$

$$(J=2, 3, \dots, m)$$

$$\sum_{J=1}^{K-1} Z(I, J) + Y(I, K) = M(I) \quad (I=1, 2, \dots, n)$$

$$(K=1, 2, \dots, m)$$

X(I,J) = 1、Y(I,J) = 1 のときに限り W(I,J) = 1 だから

$$X(I, J) + Y(I, J) - W(I, J) = 1 \quad (I=1, 2, \dots, n)$$

$$W(I, J) \geq X(I, J) \quad (J=1, 2, \dots, m)$$

$$W(I, J) \geq Y(I, J)$$

上下界

bounds

端のエレメントを 1 番と決めてやると計算が速いがなくてもよい。

変数型

binary

Z(I,J) (I=1,2,.....n)
(J=1,2,.....m)
U(I,K) (I=1,2,.....n)
(K=1,2,.....m)
W(I,J) (I=1,2,.....n)
(J=1,2,.....m)
X(I,J) (I=1,2,.....n)
(J=1,2,.....m)
Y(I,J) (I=1,2,.....n)
(J=1,2,.....m)

general

WFN

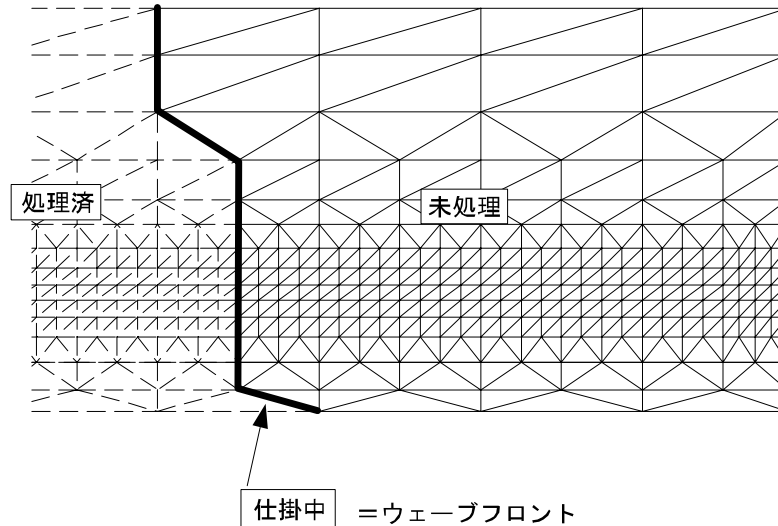
終端

end

以上の定式化により、エレメント数 10 個程度の解析モデルなら、WF 数を最適化できる。そんな小さなモデルなら、WF 数を最適化する必要は全くないので、整数計画法は、役に立たないという結論になる。

3. WF 数の最適化の難しさ

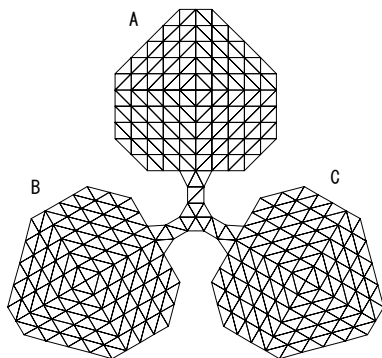
ウェーブフロント数の最適化とは、連立一次方程式のソルバーがエレメントをエレメント番号順に処理してゆくとき出来る、下図のような処理済部分と未処理部分の境界線をできるだけ短くしようとするものである。



WF 法ソルバーは、WF 数を最適化しなくても動くものであるから、「最適」には意味が無く、ある程度小さく纏められていればよいものである。上図のように平べったいモデルであれば、見た目どおり（すなわちノード座標に頼って）左から右へ上から下へ、仕掛中の境界線がなるべく鉛直線になるように付番してゆくのよいのは明らかである。

WF 数というのは、本来座標とは全く関係のない概念であるから、純粋数学的にはインチキであるが、実用上は優れた方法である。

ただし、計算機で座標に頼った optimizer を作ってしまうと、下図のようなとき、A の葉



から順に、上から下へ水平線に沿ってエレメント番号を付けて行くと、やがて B、C の葉を一緒に処理することになり、仕掛線は広がってしまう。真ん中の柄の細い部分まで来たとき、B を先にするか C を先にするか決めて、一枚ずつ処理した方が WF 数は明らかに小さい。一般の任意の形状に対し、この判断を計算機にやらせるのは非常に難しい。WF 数の最適化は、人間が目で見えて付番するのが一番よい。

2013. 11. 21

参考文献

岡本吉央「整数計画法によるパズル解法」

www.is.titech.ac.jp/~okamoto/PDF/2007/puzzleip.pdf