

n 個の 1 を有する 2 進数の昇順順序付け

九州共立大学大学院 学生員 王 克 成
九州共立大学大学院 正会員 三 原 徹 治
九州共立大学工学部 学生員 黒 木 英 文
第 一 復 建 (株) 正会員 千々岩浩巳

1. はじめに

n 個の 1 を有する m bit の 2 進数は $m \geq n$ の条件下で無数に存在する. これらを何らかのルールで順序付けすることができれば遺伝的アルゴリズム GA での最適化に用いる線列構成に有利となることが予想される.

本研究では, そのような 2 進数を昇順に順序付けすることができる変換 ${}_nT(s)$ を提示し, その算出手法について概説する.

2. 2 進数数列の順序付けのための変換

(1) ${}_nT(s)$ の定義

n 個の 1 を有する 2 進数から順序付けに必要な変換 ${}_nT(s)$ を「 $\text{"}1\text{"}$ の個数 = n の場合の $\text{"}0/1\text{"}$ 組合せを 2 進数数列と解釈し, その値を昇順に並べたとき s 番目の $\text{"}0/1\text{"}$ 組合せが表現する 10 進数値.」と定義する.

(2) $n=1$ のときの数列のバリエーション

表-1 に $n=1$ のときの数列のバリエーションの一部として $m=1,2,\dots,7$ の m 値, 2 進数数列, ${}_1T(s)$ 値および s 値を示す.

この場合, m bit の 2 進数は 1 種類しかないので 2 進数の bit 数を示す m 値と ${}_1T(s)$ 値の昇順順序を示す s 値は当然のことながら一致する.

(3) $n > 1$ のときの数列のバリエーション

表-2 に $n > 1$ の一例として $n=3$ のときの数列のバリエーションの一部を示す.

$n \leq m$ でなければ所望の組合せが物理的にありえないので $m \geq 3$ であり, $m=1,2$ はないが, ここでも $n=1$ のときと同様に ${}_3T(s)$ が昇順に並ぶように順序数 s を与えることができる. このように並べたとき, m の値ごとに $\square$ で囲っているような規則的な並びが現れる. しかもその並びは $n=2$ のときの数列に一致する.

(4) ${}_nT(s)$ の一般的な性質

$n=1$ および $n > 1$ の数列のバリエーションの観察から ${}_nT(s)$ の一般的な性質として次の 2 つの関係が得られる.

$$\text{性質 1: } {}_1T(s) = 2^{s-1} \quad (1)$$

$$\text{性質 2: } m \in {}_{m-1}C_n < s \leq {}_mC_n \text{ なる } m \text{ に} \\ \text{対して}$$

$${}_nT(s) = 2^{m-1} + {}_{n-1}T(s - {}_{m-1}C_n) \quad (2)$$

表-1 $n=1$ のときの数列のバリエーションの一部

m	64	32	16	8	4	2	1	${}_1T(s)$	s
1							1	1	1
2						1	0	2	2
3					1	0	0	4	3
4				1	0	0	0	8	4
5			1	0	0	0	0	16	5
6		1	0	0	0	0	0	32	6
7	1	0	0	0	0	0	0	64	7

表-2 $n=3$ のときの数列のバリエーションの一部

m	64	32	16	8	4	2	1	${}_3T(s)$	s
3					1	1	1	7	1
4				1	0	1	1	11	2
				1	1	0	1	13	3
				1	1	1	0	14	4
5			1	0	0	1	1	19	5
			1	0	1	0	1	21	6
			1	0	1	1	0	22	7
			1	1	0	0	1	25	8
			1	1	0	1	0	26	9
			1	1	1	0	0	28	10
6		1	0	0	0	1	1	35	11
		1	0	0	1	0	1	37	12
		1	0	0	1	1	0	38	13
		1	0	1	0	0	1	41	14
		1	0	1	0	1	0	42	15
		1	0	1	1	0	0	44	16
		1	1	0	0	0	1	49	17
		1	1	0	0	1	0	50	18
		1	1	0	1	0	0	52	19
		1	1	1	0	0	0	56	20
7	1	0	0	0	0	1	1	67	21
	1	0	0	0	1	0	1	69	22
	1	0	0	0	1	1	0	70	23
	1	0	0	1	0	0	1	73	24
	1	0	0	1	0	1	0	74	25
	1	0	0	1	1	0	0	76	26
	1	0	1	0	0	0	1	81	27
	1	0	1	0	0	1	0	82	28
	1	0	1	0	1	0	0	84	29
	1	0	1	1	0	0	0	88	30
	1	1	0	0	0	0	1	97	31
	1	1	0	0	0	1	0	98	32
	1	1	0	0	1	0	0	100	33
	1	1	0	1	0	0	0	104	34
	1	1	1	0	0	0	0	112	35

3. 算出手順と算出例

(1) ${}_nT(s)$ 値の算出手順

${}_nT(s)$ の性質 1,2 から ${}_nT(s)$ 値は次の手順で算出することができる.

- ① 求める ${}_nT(s)$ の n, s を与え, $t=0$ とおく. ② $m \in {}_{m-1}C_n \leq s \leq {}_mC_n$ の関係を満たす m を求め, $t \leftarrow t+2^{m-1}$.
 ③ $s \leftarrow s - {}_{m-1}C_n$, $n \leftarrow n-1$. ④ $n > 1$ ならば②へ戻る. $n=1$ ならば⑤へ. ⑤ $t \leftarrow t+2^{s-1}$, 求める ${}_nT(s)=t$ として終了.

(2) $n=4, s=53$ のときの ${}_4T(53)=178$ の算出例

$$\text{【1】 } {}_7C_4 = 35 \leq 53 \leq 70 = {}_8C_4 \Rightarrow m=8, 53 - {}_7C_4 = 53 - 35 = 18 \Rightarrow {}_4T(53) = 2^{8-1} + {}_3T(18)$$

$$\text{【2】 } {}_5C_3 = 10 \leq 18 \leq 20 = {}_6C_3 \Rightarrow m=6, 18 - {}_5C_3 = 18 - 10 = 8 \Rightarrow {}_3T(18) = 2^{6-1} + {}_2T(8)$$

$$\text{【3】 } {}_4C_2 = 6 \leq 8 \leq 10 = {}_5C_2 \Rightarrow m=5, 8 - {}_4C_2 = 8 - 6 = 2 \Rightarrow {}_2T(8) = 2^{5-1} + {}_1T(2) = 2^{5-1} + 2^{2-1}$$

$$\therefore {}_4T(53) = 2^{8-1} + 2^{6-1} + 2^{5-1} + 2^{2-1} = 128 + 32 + 16 + 2 = 178,$$

$${}_4T(53)=178 \text{ に対応する } 0/1 \text{ 線列 } = 10110010$$

4. 変換手順の問題点と対処方法

(1) 問題点と対処方法

- i) 手順②, ③を $(n-1)$ 回繰返す必要がある. $\cdots$ 提案手順のベースが ${}_nT(s)$ の性質 2 である限り有効な対処方法は見当たらない.
- ii) 手順②において $m \in {}_{m-1}C_n < s \leq {}_mC_n$ の関係を満たす m を求める場合に, $n \leq m$ の条件から $m=n, n+1, n+2, \cdots$ と条件適合を検討し, はじめて $s \leq {}_mC_n$ の関係を満足する m を求める m とする方法は, もれもなく確実であるが多くの検討数を必要とする. $\cdots$ 「必要な m を比較的効率的に求める方法」を導入して計算量の縮小を図る.

(2) 必要な m を比較的効率的に求める方法

ある s に対して

$$s = {}_mC_n = \frac{\prod_{i=1}^n (m+1-i)}{\prod_{i=1}^n i} = \frac{\prod_{i=1}^n (m+1-i)}{n!} \quad \text{つまり} \quad s \cdot n! = \prod_{i=1}^n (m+1-i)$$

となるような m が整数として得られることは一般にまれであるが, 右辺に着目すると相加平均・相乗平均関係から次の関係が導かれる.

$$\frac{1}{n} \sum_{i=1}^n (m+1-i) = m - \frac{n-1}{2} \geq \sqrt[n]{\prod_{i=1}^n (m+1-i)} \quad \therefore \prod_{i=1}^n (m+1-i) \leq \left(m - \frac{n-1}{2}\right)^n \quad (3)$$

式(3)より m に関する不等式が式(4)として得られ, 整数関係に直すと式(5)が得られる.

$$\sqrt[n]{s \cdot n!} + \frac{n-1}{2} \leq m \quad (4) \quad m^L \leq m \quad \text{ただし, } m^L = \text{INT} \left[\sqrt[n]{s \cdot n!} + \frac{n-1}{2} \right] + 1 \quad (5)$$

式(5)の関係を用いれば $m = m^L$ から昇順に順次検討すれば良いことになり, $m = n, n+1, n+2, \cdots$ と条件適合を検討する方法に比較して効率的に必要な m を求めることができる.

5. おわりに

本報告では, n 個の 1 を有する 2 進数を順序付けするルールのひとつとして ${}_nT(s)$ を提示した. まず, ${}_nT(s)$ の定義, バリエーションの観察から ${}_nT(s)$ の一般的性質を導いた. 次に, この一般的性質を利用した ${}_nT(s)$ 値の算出手順を構築し, その妥当性を検証した. 最後に ${}_nT(s)$ 値の算出に必要な $m \in {}_{m-1}C_n < s \leq {}_mC_n$ の関係を満たす m を比較的効率的に求める方法も示した.

提示した ${}_nT(s)$ の算出には「算出の骨幹部分を $(n-1)$ 回繰返す必要がある」という難点はあるが, n 値と順序数 s だけあればそれに対応する 0/1 線列が一意に決定されるという他に類をみない特長がある. ${}_nT(s)$ の応用を参考文献 1) に示す.

参考文献 1) 黒木英文, 三原徹治ほか: 調整対象作業の分割を考慮した需要平滑化を指向する工程計画変更案探索への GA の適用, 平成 19 年度土木学会西部支部研究発表会, 2008.3.