分割化ラインバランシング問題への分岐限界法の適用*

倉 重 賢 治** 柳 川 佳 也*** 宮 崎 茂 次***

Application of Branch and Bound Method in Divided Line Balancing Problem

Kenji Kurashige, Yoshinari Yanagawa and Shigeji Miyazaki

In large scale problems of line balancing, a weak point is to spend very effort of calculation when branch and bound method is adopted to solve the result. Therefore, in this paper, they propose the method which can seek the satisfactory result by split the problem into partition. Firstly, problem is divided into several parts, in each part the optimal line balancing is investigated, and then, the results of all parts are jointed together to search the trial solution for the whole part. Lastly, in order to improve the trial solution, line balancing of the union of partitional result is conducted again to search for the satisfactory result. Furthermore, computer is used for the result calculation, the result is compared with the conventional method to find the efficiency of this proposed method.

Key words: line balancing problem, branch and bound method

1. は じ め に

ラインバランシング問題のような組合せ問題の代表的な解法として分岐限界法が存在し、文献 1) ではワークステーション数最小化とステーションタイムの平滑化を同時に目指した評価値および下界の計算式が提案されている。しかし、分岐限界法を用いる場合、問題数の増加にしたがって、計算時間が指数的に増大し、大規模問題の場合、計算時間がかかりすぎるといふ欠点が生じる。

そこで本研究においては、問題をいくつかに分割する方法で、満足解を求める方法を提案する。まず最初に与えられた問題をいくつかの部分問題に分割する。次にそれぞれの部分問題の中で最適なラインバランシングを行い、それらの解をつなぎ合わせ全体の試行解を求める。最後にそれぞれの部分問題の解の結合部分に対して、再びラインバランシングを行い満足解を求める。また、実際にコンピュータによって計算を行い、その結果を従来法と比較し提案法の有効性を示す。

2. 本研究の記号とモデル

2. 1 記 号

n : 総要素作業数

L : サイクルタイム

p_k : 第 k ワークステーションのステーションタイム

K : ワークステーション数

R : 先行関係の強さ

文献 1) に用いられており $0 \sim 1$ までの値を取る。

R の値が大きい程、先行関係の強い問題である。

g : 問題を分割する場合の部分問題数

G_i : 第 i 番目の部分問題 ($i=1, \dots, g$)

2. 2 モ デ ル

総要素作業数、各要素作業の先行関係、および作業時間、サイクルタイム、部分問題数がそれぞれ与えられている。文献 1) と同様にワークステーション数最小化とステーションタイムの平滑化を同時に指向したバランス指標 BI

$$BI = \sqrt{\sum_{i=1}^K (L - p_i)^2}$$

を評価値として、ラインバランシングを行う。

3. 解法の手順

本解法は、以下の手順で求められる。

- (1) すべての要素作業を g 個のグループに分割し、それらのグループを部分問題とする。
- (2) 各部分問題に対してラインバランシングを行い、それらの解をつなぎ合わせて、全体の試行解とする。
- (3) 各部分問題の解の結合部分に対して再びラインバランシングを行い、全体の満足解を求める。

4. 問題の分割化

4. 1 分割化の手順

問題の分割化によって、部分問題 G_i の区切りとなる末端作業番号を N_i とし、次の二つの条件に基づき、各部分問題における最適な N_i を求める。

- (I) 各部分問題の作業数の差が、ある一定の範囲内に収まっている。
- (II) 分割化によって、既存の先行関係に与える影響が、最小である。

* 原稿受付 平成 7 年 6 月 15 日

** 正 会 員 岡山県立大学 (総社市窪木 111)

*** 正 会 員 岡山大学 (岡山市津島中 1-1-1)

以下に(I), (II)の条件に基づく N_i の設定法を示す。

条件(I)に基づく設定

本研究においては、各部分問題でとり得る最大作業数 H_{max} が、最小作業数 H_{min} の2倍以下になるという条件を設ける。そのため次の記号を設ける。

H : 1つの部分問題の平均作業数 ($=n/g$)

M_i : 各部分問題の作業数を等しくした場合のグループ G_i の

末端の作業番号 ($=i \times n/g$)

($n=100, g=3$ の場合, $M_1=33, M_2=67, M_3=100$ となる)

ただし, H, M_i とも小数点以下は四捨五入する。

ここで, N_i のとり得る範囲を未知の整数 Y より,

$$M_i - Y \leq N_i \leq M_i + Y \quad (1)$$

とすると, 図1から

$$H_{max} = H + 2Y, \quad H_{min} = H - 2Y$$

となり, 最大作業数は最小作業数の2倍以下より,

$$H_{max} \leq 2H_{min}$$

H_{max} と H_{min} を代入して

$$Y \leq H/6 \quad (2)$$

を求める。

式(2)を満たす最大の整数を求め, Y^* で表し, この Y^* を式(1)の Y に代入して, N_i のとりうる範囲を

$$M_i - Y^* \leq N_i \leq M_i + Y^*$$

で表す。

条件(II)に基づく設定

任意の作業番号 j を部分問題の末端作業とし, その作業で分割を行うと, $(j+1)$ 以上の作業番号をもつ作業は, それ以前, つまり j 以下の番号をもつ作業より前には割り当てることができない。すなわち, 本来の先行関係に新たな制約条件を加え, 先行関係を強めていることになり, 最適解を得る可能性を小さくしていることになる。したがって, 作業 j はなるべく既存の先行関係に影響を与えないものが望ましい。そこで, 条件(I)で求めた N_i の取り得る範囲内から, 最も先行関係に影響を与えない作業を選び, その作業を N_i^* として分割を行う。

4.2 分割化の具体例

図2のような先行関係図をもつ問題を, 2つの部分問題に分割すると仮定する。

$$n=12, \quad g=2 \text{ より}$$

$$H=6, \quad M_1=6, \quad M_2=12 \text{ となり}$$

$$Y \leq H/6 = 6/6 \text{ より}$$

$$Y^*=1$$

が求められ

$$5 \leq N_1 \leq 7$$

となる。

Fig.1 Relation between H_{max} and H_{min}

$$R=0.758$$

Fig.2 Given precedence relation diagram

$$R=0.879$$

Fig.3 Precedence relation diagram where work element ⑤ is cutting-off element

以上より, 図2の問題は⑤~⑦のいずれかの作業で分割を行うことになる。⑤の作業で分割を行った場合の先行関係図は図3のようになり, その場合の R の値は0.879である。同様に⑥と⑦の作業で分割を行うと, R の値はそれぞれ0.879, 0.833となる。そこで, R の値は⑦の作業で分割を行うときに最小値を取るのので, $N_i^*=7$ が選択される。

5. 部分問題へのラインバランシングの適用

5.1 実行可能解の生成

実行可能解の生成については各部分問題に属する作業に限定して行うことによって, 分岐数を少なくする。なお, 最終の部分問題以外では, 部分問題内の最後のワークステーションでの割り当て作業の総作業時間が, ステーションタイムの下限值(6.1節参照)に達しない場合は, それらの作業を次の部分問題に送り, 部分問題の調整を行う。

5.2 下界の計算

解法として用いる分岐限界法での下界 LB の計算式は, 文献1)より

$$LB = \sqrt{F_1 + F_2}$$

を用いる。ただし,

$$F_1 = \sum_{k=1}^i (L - p_k)^2$$

$$F_2 = \{L - (u+1)\}^2 \times q_1 + (L - u)^2 \times (q - q_1)$$

$$q = \left\lceil \frac{(T - \sum_{k=1}^i p_k)}{L} \right\rceil$$

$$u = \left\lfloor \frac{(T - \sum_{k=1}^i p_k)}{q} \right\rfloor$$

$$q_1 = \text{mod} \left\{ \left(T - \sum_{k=1}^i p_k \right) / q \right\}$$

T : 総作業時間

R : 作業が割り当てられたワークステーション数

$\lceil X \rceil$: X 以上の最小の整数

$\lfloor Y \rfloor$: Y を越えない最大の整数

$\text{mod}\{A/B\}$: A を B で割った余り

5. 3 ステーションタイムによる分岐選択

従来の解法では、下界の値だけを次の分岐選択の対象にしているが、本研究では、下界の値が同じ場合にはステーションタイムを考慮に入れて分岐を行う。つまり、最終の部分問題以前では作業時間の大きいものを、最終部分問題では作業時間の小さいものを優先して分岐を行う。この理由は、最終部分問題以前では、分割化によってワークステーション数最小化を得る可能性が大きくなるように、最終部分問題までになるべく多くの作業をワークステーションに割り当てるためである。最終部分問題では6章に述べる境界部分のラインバランシングを行うにあたり、なるべくバランスの悪い（ステーションタイムが小さい）部分を、境界部分のラインバランシングの対象になる可能性が高くなるよう、前方のワークステーションに割り当てるためである。このように、各部分問題でラインバランシングの最適解を求める。

6. 境界部分のラインバランシング

6. 1 ラインバランシングの範囲

試行解を求めた後に、問題の分割化により強まった先行関係の影響が大きいと予想される部分問題の境界部分の両端において、以下に述べる範囲内のワークステーション間で、再び分岐限界法でラインバランシングを行い、試行解を改善することによって満足解を求める。その場合、対象範囲を大きくする程、より最適解に近づく可能性は大きくなるが、それと同時に計算時間も大きくなる。本研究の目的は、満足解をより早く求めることなので、その範囲を効率的に設定する必要がある。

対象範囲の決定手順

【手順1】 対象範囲の境界線を中心に、異なる部分問題に属する要素作業間で、本来なら互いに先行関係がなく、分割化によって強制的に先行関係が与えられた作業を探す。

【手順2】 手順1で探した作業が割付けられたワークステーション中から、境界線上から最も離れたものを、前後から1つずつ選ぶ。

【手順3】 手順2で求めたワークステーション間を、境界部分のラインバランシングの対象範囲とする。（図4では、AとBのワークステーションが、境界線から最も離れていると仮定している。）

【手順4】 手順3で求めた範囲は、先行関係が弱いときには大きくなり、先行関係が強いときは逆に小さくなる。そこで、対象範囲の上限値と下限値を設定し、上限値を超える場合は上限値までを、逆に下限値まで達しない場合は下限値までを対象範囲とする。

Fig. 4 Scope of line balancing

Fig. 5 Scope of upper limit

上限値の設定

図5のように、境界部分のラインバランシングを行う部分問題の境界線から、その前後の部分問題の境界線までを範囲の上限とする。

下限値の設定

先行関係が強いほど、結合部分のラインバランシングに必要な範囲は小さくなる。また、かなり先行関係が強い時は、隣接する部分問題間では作業の入れ替えが行えず、試行解が改善されるのは図6(a), (b)のような作業の移動によってのみである。

ここで、図7(a)のように境界線を中心に、前後S個のワークステーションの総作業時間を t_a , t_b とする（ただし、 $t_a \geq t_b$ とする）。もし、要素作業（作業時間 t ）の移動があれば総作業時間はそれぞれ、図7(b)のように $t_a - t$, $t_b + t$ となり、試行解の評価値がよくなるには、前後S個のワークステーションの総作業時間の差が縮まる必要があり、 $t_a - t_b \geq |(t_a - t) - (t_b + t)|$ (3)

$$= |t_a - t_b - 2t|$$

が成立する必要がある。

$t_a - t_b > 2t$ のとき

式(3)は成立

$t_a - t_b \leq 2t$ のとき

$t_a - t_b \geq t$ の条件下で式(3)は成立

$$\therefore t_a - t_b \geq t \quad (4)$$

のとき

式(3)は満たされる。

よって、式(4)から、Sが最小値をとるのは、 t の値が最小かつステーションタイムの並びが図8のようなときであり、

Fig. 6 An example of moving work element

Fig.7 Total working time of S work stations

Fig.8 Allotment station time in case of minimum S

$S(L - p_k \text{ の下限値}) \geq \text{最小要素作業時間}$ (5)
を満たす時に S は最小値をとる。しかし、式(5)は理想的な場合なので S ではなく、

$S_1 \mid T_{i+1} - T_i \mid \geq \text{平均要素作業時間}$ (6)
を満たす S_1 の最小値を下限値の 1 つとする。

ただし、

T_i : 部分問題 G_i の平均ステーションタイム

また、各部分問題の平均ワークステーション数を E とすれば、
 $S_2 = E / 2$ (ただし、 S_2 は整数)

も下限値の一つと考え、下限値 $S_{\min}$ の設定は、

$S_{\min} = \min \{S_1, S_2\}$

とする。

$S_{\min}$ の値が決まると、下限値の範囲は図 9 のように決まる。

Fig.9 Scope of lower limit

6. 2 再ラインバランシングの下界の計算式

試行解を改善するにあたり、その時点でのワークステーション数はほとんど変化しないと考えられる。そこで、前述の下界 LB にワークステーション数の設定を加えた下界 LB'

$$LB' = \sqrt{F_1 + F_3}$$

を境界部分のラインバランシングに用いる。

ただし、

$$F_3 = (L - a)^2 \times (W - r - b) + (L - a - 1)^2 \times b$$

$$a = \left\lceil (T - \sum_{k=1}^r p_k) / (W - r) \right\rceil$$

$$b = \text{mod} \left\{ (T - \sum_{k=1}^r p_k) / (W - r) \right\}$$

W: 境界部分のラインバランシングを行う範囲内のワークステーション数

7. コンピュータによる計算と考察

7. 1 問題設定

コンピュータによる計算で、計算時間と評価値の比較を従来法と提案法について行い、7.2 節で述べる 5 段階評価で示す。そのため、サイクルタイムを 30 単位時間とし、要素作業時間のばらつきは 5 ~ 8 単位時間とする。総要素作業数 n、グループ数 g については、それぞれ値を変化させ、それらの問題を (n, g) で表すこととし、表 2 に示す 10 パターンの問題の計算を行う。さらに、先行関係の強さ R を決めるパラメータ P (0.1 ~ 1.0) を 10 種類、乱数の初期値を 10 種類与え、合計 1000 問題をコンピュータによって自動作成し、その問題を従来法と提案法で解いた。また、従来法では、(総要素作業数 × 2) 秒ですべての計算を打ち切ることとし、提案法では、(平均部分問題数 × 2) 秒で各部分問題での計算を打ち切ることとする。なお、使用計算機は SUN ワークステーション・S s 2 である。

7. 2 計算結果の評価

計算結果の評価は以下の通りである。なお、本研究の目的は、満足解をより早く求めることであるので、A1 ~ A3 の評価が比較的良好な結果だとみなされる。

A1: 提案法によって得られた解の評価値が、従来法で得られた最適解の評価値と等しく、提案法の計算時間が従来法に比べ短縮されている場合。

A2: 提案法によって得られた解の評価値が、途中計算が打ち切られ従来法で得られた試行解の評価値と比較して、同等もしくは優れている場合。

A3: 提案法によって得られた解の評価値が、従来法で得られた解の評価値より 1 割以下の増加で済んでおり、提案法の計算時間が従来法に比べ増加されていない場合。

B1: 提案法によって得られた解の評価値が、従来法で得られた解の評価値よりも良好、もしくは 1 割以下の増加で済んでいるが、提案法での計算時間が従来法に比べ増加した場合。

B2: 提案法によって得られた解の評価値が、従来法で得られた解の評価値より 1 割以上大きな場合。

7. 3 計算結果

問題 (100, 3) の計算結果の一部 (10 問) と、その結果に基づく計算結果の評価を表 1 に示す。

また、すべての問題における計算結果の評価の個数を表 2 に、

Table 1 Results of calculation and grade of problem (100, 3)

P	R	Conventional		Proposed		Grade
		B I	T	B I	T	
0.10	0.6063	3.9	3.75	3.9	1.45	A1
0.13	0.7109	3.9	1.63	3.9	0.82	A1
0.15	0.7834	3.9	0.85	4.1	0.33	A3
0.18	0.8323	3.9	0.53	3.9	10.23	B1
0.20	0.8444	4.1	0.55	4.3	6.83	B1
0.25	0.8909	4.4	*	4.6	100.12	A3
0.30	0.9188	11.6	*	10.9	125.18	A2
0.50	0.9634	11.2	*	11.2	43.01	A2
0.70	0.9865	11.6	8.08	11.6	0.65	A1
1.00	1.0000	18.1	0.33	20.3	0.05	B2

T: CPU time (*: Break off)

Table 2 Number of each grade for various problems

Problem	A1	A2	A3	B1	B2	Total
(100, 2)	35	25	14	21	5	100
(100, 3)	29	22	20	18	11	100
(100, 4)	27	24	15	23	11	100
(200, 2)	29	35	14	15	7	100
(200, 3)	30	36	10	15	9	100
(200, 4)	25	40	11	2	22	100
(300, 2)	23	37	26	5	9	100
(300, 3)	20	38	17	4	21	100
(300, 4)	12	42	22	0	24	100
(300, 6)	11	42	16	1	30	100
Total	241	341	165	104	149	1000
	747			253		1000

Table 3 Percentage of grade for each number of elements (%)

Number of Elements	A1	A2	A3	B1	B2
100	30.3	23.7	16.3	20.7	9.0
200	28.0	37.0	11.7	10.7	12.7
300	16.5	39.8	20.3	2.5	21.0

Thick line shows conventional method
Thin line shows proposed method (g is dividing number)

Fig.10 Average CPU time in case of 300 work elements

各要素作業数ごとの評価の個数の割合は表3に示す。

さらに、要素作業数300の問題の平均計算時間について、従来法と提案法の比較を図10に示す。

7. 4 考 察

表2より全体の75%程度は良好な結果が得られた。また、全体的な傾向としては、要素作業数と分割数が増えると、B1の評価は少なくなりB2の評価は増えているが、この理由は明らかである。すなわち、分岐限界法では、大規模な問題になるほど計算時間は指数関数的に増加する。そこで、要素作業数と分割数を増やすほど、従来法に対する提案法の計算時間の短縮の効果は大きくなる。図10では、分割数の増加が計算時間の短縮を示している。しかし、分割数を多くすることは、本来の先行関係に与える影響を大きくするので、求める満足解の評価値を悪くする可能性が大きくなる。従って、本研究の解法を用いる場合、要素作業数に応じて分割数を設定することになるが、計算時間の短縮を目指すなら分割数を多くし、解の評価値を重視するなら分割数は少なくすればよい。

8. お わ り に

本研究では、ラインバランシング問題に対する計算時間短縮のために以下のことを行った。

- (1) ラインバランシング問題を、いくつかの部分問題に分割して、求めた解の結合部分に対して、再度ラインバランシングを行う方法を提案した。
- (2) 実際に、コンピュータによって1000問の計算を行うことで、従来法との比較を行い、全体の75%程度の問題については、提案法の有効性を示すことができた。

本研究は、作業数に対して計算時間が指数関数的に増加するアルゴリズム（分岐限界法や総当たり法など）を用いる場合に、計算時間短縮の効果が大きいと予想される。ワークステーションの設備に制約のある場合や混合品種ラインについては、今後の課題にしたい。

参 考 文 献

- 1) S. Miyazaki and H. Ohta: The Assembly Line Balancing Problem with the Balancing Index as an Objective Function, Lecture Notes in Control and Information Science (M. Iri and K. Yajima. eds), 113, Springer-Verlag, Berlin, (1988) 681-690.
- 2) E. P. C. Kao and M. Queyranne: Dynamic Programming Methods for Assemble Line Balancing, Operations Res, 30, 2, (1982) 375.
- 3) L. Salvesson and K. R. Baker: Dynamic Programming Solution of Sequencing Problems with Procedure Constraints, Operations Res, 26, 3, (1978) 444.
- 4) R. V. Johnson: Efficient Modular Implementation of Branch-and-Bound Algorithms, Decision Sci, 19, (1988) 17.
- 5) 人見勝人：生産管理工学，コロナ社，(1978) 129-132.