

# クラウドソースドマニュファクチャリング環境下における 企業間のリソース配分手法の一提案

A proposal of resource allocation method among companies under Crowdsourced Manufacturing

原田佳明・神戸大学

Yoshiaki Harada, Kobe University

國領 大介・神戸大学

Daisuke Kokuryo, Kobe University

貝原俊也・神戸大学

Toshiya Kaihara, Kobe University

藤井 信忠・神戸大学

Nobutada Fujii, Kobe University

Key Words: Crowdsourced manufacturing system, Cloud-based manufacturing system, Resource allocation, Combinatorial Auction

## 論文要旨

近年のIoTの発展により、企業間でリソース情報を共有しリソースの融通を可能とするクラウドソースドマニュファクチャリングと呼ばれる生産形態に注目が集まっている。クラウドソースドマニュファクチャリングとは、複数の企業がリソースの情報を共有・融通することにより生産性の向上を目指すものであり、そこで重要となるのが効果的なリソース配分である。本稿ではリソースの配分と取引価格を決定する組合せオークション方式を用いたリソース配分手法を提案する。組合せオークションにおける勝者決定問題を2種類の定式化を行い、計算機実験により有効性の確認を行う。

### 1. はじめに

近年IoT (Internet of Things) の発展に伴い、工場や製造機器をインターネット上に繋ぐことにより生産性を向上させること動きが活発化している [1]。また、企業や工場内だけでなく、企業の壁を超えたつながりを利用した生産に注目が集まっており [2]、その中でIoT環境下において複数の工場や複数の企業をつなぎ設備・材料・労働力・工法を融通し合う生産形態であるクラウドソースドマニュファクチャリングと呼ばれる生産形態が提案された [3]。クラウドソースドマニュファクチャリングを形成することで、リソースが不足した処理や、自社にはないリソースを使用する処理をつなぐ他企業に委託することで、顧客ニーズに合わせたカスタム生産を低コストで実現可能となる [4]。クラウドソースドマニュファクチャリングにおいて重要となるのがどの企業のリソースの要求に対して、どの企業の提供リソースを割り当てるかを決定するリソース配分手法である [5]。そこで本稿では、発生したリソース要求に対し、リソースの配分と取引価格を決めるため、組合せオークション方式を用いたリソース配分手法を提案する。

### 2. 対象モデル

本稿の対象とするクラウドソースドマニュファクチャリングモデルを以下のように定義する。

- I. 顧客からのオーダーは各企業に対して発生する。
- II. 各企業は出来ない処理がある場合にリソースを要求する。
- III. 各企業は空いているリソースがあればリソースを提供する。
- IV. リソース要求に対して、提供リソースを割り当てる。
- V. リソース要求企業はリソースが調達できれば、製品の生産を行う。
- VI. 対象とする製品は、工程毎に分割可能とする。

### 3. 組合せオークション方式を用いたリソース配分手法

本研究では2-IVのリソース要求に対して、組合せオークション方式を用いてリソースの配分と、その取引価格を決める。

#### 3-1 提案手法の概要

組合せオークション方式を用いた提案手法の流れを以下に示す。

STEP1. リソース提供企業とリソース要求企業は入札を作成し、オークション主催社に提出する。(入札作成)

STEP2. オークション主催者は、評価値の合計が最大(最少)となる入札を決定する。(勝者決定)

STEP3. 各企業はオークションの結果に応じ、定められた取引価格において取引を行う。(取引価格決定)

オークション主催者は利益を求めないとする。またオークションにかけられる財はリソース $r(r = 1, \dots, R)$ とする。

#### 3-2 入札作成

以下に従ってリソース提供側とリソース要求側は以下に従って、入札を作成する。

- リソース提供企業 $i(i = 1, \dots, I)$ について
  - 提供するリソース $r$ のコスト $c_{ir}$ と提供時間 $TP_{ir}$ からなる入札を作成する。
  - 提供時間の一部のみを提供することが可能とする。
  - 各企業は $R$ 個の入札を作成する。ただし提供できないリソースに関しては、コスト、提供時間ともに0とする。
- リソース要求企業 $j(j = 1, \dots, J)$ について
  - 予算 $v_j$ 要求するリソース $r$ の要求時間 $TP_{jr}$ を提示する。
  - 必要なリソースの組合せに対してリソースの要求を作成する。これは全てのリソースが揃わないと製品の生産が出来ないからである。
  - 各企業が作成する入札は1つとする。

Fig.1に作成した入札の例を示す。要求企業1はリソース1を125[Ts]、コスト0.1で提供可能、リソース2を100[Ts]、コスト0.2で提供可能であること表示。要求企業2は、予算100で、リソース1を100[Ts]を要求していることを表示。

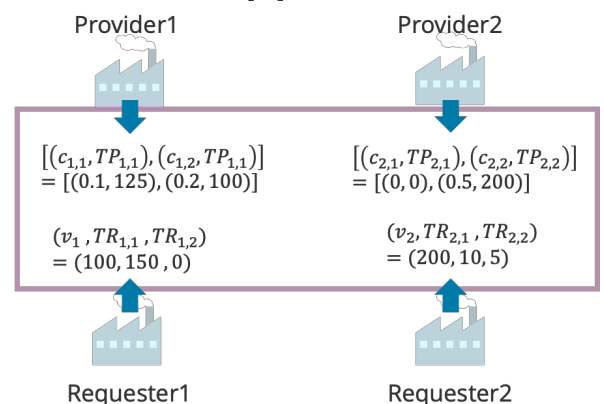


Fig. 1 Example of bids

#### 3-3 勝者決定問題

前節で作成された入札に対し、2種類の目的関数を用いた勝者決定問題の定式化を述べる。

(i) コスト最小化問題としての定式化を(1)~(6)に述べる.

$$\begin{aligned} \min \quad & \sum_{i=1}^I \sum_{r=1}^R \sum_{j=1}^J c_{ir} \times TR_{jr} \times x_{ir,j} \quad (1) \\ \text{s.t.} \quad & \sum_{j=1}^J TR_{jr} \times x_{ir,j} = TR_{ir} \quad (\forall i, \forall r) \quad (2) \\ & \sum_{i=1}^I TR_{jr} \times x_{ir,j} \leq TP_{jr} \quad (\forall j, \forall r) \quad (3) \\ & \sum_{i=1}^I x_{ir,j} \leq 1 \quad (\forall j, \forall r) \quad (4) \\ & \sum_{i=1}^I \sum_{r=1}^R PAY_{ji,r} \leq v_i \quad (\forall i) \quad (5) \\ & x_{ir,j} \in \{0,1\} \quad (6) \end{aligned}$$

(1)は目的関数であり、総コストの最小化を表す。式(2)は発生したリソース要求を満たす提供企業が1つ存在するたすための制約である。式(3)は各提供企業の提供リソースの容量制約。式(4)が要求企業*i*の要求リソース*r*をそれぞれ提供するのはただか1企業とする制約である。式(5)は、リソース要求企業*j*がリソース提供企業*i*からリソース*r*を借りるときに支払う価格PAY<sub>ji,r</sub>はリソース要求企業*j*の予算以内とする制約である。式(6)の*x*<sub>ir,j</sub>はリソース提供企業*i*がリソース*r*をリソース要求企業*j*に提供するとき1、しないとき0となる決定変数である。この問題は、全てのリソース要求を満たす上で、コストが最小となる提供リソース側の勝者を決定する。よって前提として、全てのリソース要求を満たす時間リソース提供されていないと解を求めることが出来ない。

(ii) 利益最大化問題としての定式化を(8)~(12)に述べる.

$$\begin{aligned} \max \quad & \sum_{j=1}^J v_j \times y_j - \sum_{i=1}^I \sum_{r=1}^R \sum_{j=1}^J c_{ir} \times TR_{jr} \times x_{ir,j} \quad (7) \\ \text{s.t.} \quad & \sum_{j=1}^J TR_{jr} \times x_{ir,j} \leq TP_{jr} \quad (\forall i, \forall j) \quad (8) \\ & \sum_{i=1}^I x_{ir,j} \leq 1 \quad (\forall j, \forall r) \quad (9) \\ & \sum_{i=1}^I \sum_{r=1}^R PAY_{ji,r} \leq v_i \quad (\forall i) \quad (10) \\ & \begin{cases} x_{ir,j} = 0 \quad (\forall i, \forall r) \quad (\text{if } y_j = 0) \\ \sum_{i=1}^I TR_{jr} \times x_{ir,j} = TR_{jr} \quad (\forall i, \forall r) \quad (\text{if } y_j = 1) \end{cases} \quad (11) \\ & y_j, x_{ir,j} \in \{0,1\} \quad (12) \end{aligned}$$

式(7)は目的関数であり、各企業の受け取る利益の合計の最大化を表す。式(8)は、式(3)と同じ提供企業のリソースの容量制約、式(9)は式(4)と同じ勝者となるリソース提供企業をただか1企業とする制約である。式(10)は式(5)と同じ予算制約、式(11)は*y*<sub>j</sub>と*x*<sub>ir,j</sub>の関係を表す制約式である。式(12)の*y*<sub>j</sub>リソース要求企業*j*の入札が勝者となった場合1、ならなかった場合0となる決定変数であり、*x*<sub>ir,j</sub>は式(6)と同様の決定変数である。この問題は提供リソースが不足する場合に、リソース要求側の勝者も同時に決めることが出来る。

### 3-4 取引価格決定

本稿では取引価格PAY<sub>ji,r</sub>を、参考文献 [6]を参考に、以下のように定める [4]。

$$PAY_{ji,r} = \frac{c_{jr} + v_i \times (\frac{TR_{jr}}{\sum_{j=1}^J TR_{jr}})}{2} \times TR_{jr} \quad (13)$$

ここでsumTR<sub>j</sub>はリソース要求企業*j*がリソースを要求する時間の合計であり、以下のように定まる。

$$\text{sumTR}_j = \sum_{r=1}^R TR_{jr} \quad (14)$$

## 4. 計算機実験

リソースの配分を行う提案手法において、勝者決定問題の目的関数の違いによる影響を検証する。実験1は提供リソースが全てのリソース要求を満たせる時間提供されるとき、実験2は提供リソースが全ての要求を満たせない状況とする。勝者決定問題はCPLEXを用いて求解する。

### 4-1 実験条件

(i) 実験1では、リソース提供企業数*I* = 4、リソース要求企業*J* = 8、オークションにかけられるリソースの種類を*R* = 2とする。提供企業は、それぞれ250[Ts]リソースを提供する。

リソース1を提供する企業が2企業、リソース2を提供する企業が2企業存在する。各提供企業でコストが異なるとする。要求企業はそれぞれリソース1とリソース2を合計で100[Ts]要求する。各要求企業で予算が異なるとする。提供リソースは全ての要求を満たせる時間提供されるときとする。

(ii) 実験2では、実験1の提供企業数*I* = 2とし、提供リソースが不足した場合を想定する。他の条件は実験1と同様である。

## 4-2 実験結果

### (i) 実験1

実験1の各目的関数に対する総コストと総利益をTable1に示す。2つの目的関数に対してTotal CostとTotal Profitは同じ値となった。これは、本実験がリソースの要求は全て満たされる条件下で行なったため、利益最大化問題の定式化の決定変数*y*<sub>j</sub>は任意の*j*で1となり、コスト最小化問題と利益最大化問題の定式化が一致したためである。以下詳しく説明する。式(11)は、*y*<sub>j</sub> = 1の場合の制約が有効となることで、コスト最小化問題の定式化の制約式(2)と一致する。また目的関数(7)は、第1項は定数となり、目的関数(1)と同じ式である第2項が最小となるとき目的関数値は最大となる。従って、利益最大化問題の定式化はリソースの要求が全て満たされる条件下のとき、コスト最小化問題の定式化と一致する。

Table 1 Experimental result1

	Total Cost	Total Profit
Cost Min	110	1090
Profit Max	110	1090

### (ii) 実験2

実験2の各目的関数に対する総コストと総利益をTable2に示す。コスト最小化問題では求解できない場合でも、利益最大化問題では求解できることが確認できた。

Table 2 Experimental result2

	Total Cost	Total Profit
Cost Min	-	-
Profit Max	50	850

## 5. まとめ

本研究では、クラウドソースドマニファクチャリングのリソース配分に組合せオークション方式を用いてその配分を決定した。勝者決定問題において、2種類の目的関数を用いた勝者決定問題の定式化を行なった。計算機実験の結果、提供リソースが全てのリソース要求を満たせない場合でも求解できる利益最大化の定式化において、提供リソースが全てのリソース要求を満たせる場合でしか求解できないコスト最小化の定式化と同じ結果を得ることが確認できた。この結果より、どちらの場合においても、勝者決定問題を利益最大化の定式化で行うことが有効であると考えられる。今後の展望としては、より大規模な条件においても同様の結果を導くかを検証する。

## 参考文献

- [1] Germany, 9 8 2018. Available: Federal Ministry for Economic Affairs and Energy, Germany.
- [2] IVI, 9 12 2017.[Online].Available: <https://www.iv-i.org/faq.html>.
- [3] IEC,2102017.Available: <http://www.iec.ch/whitepaper/pdf/iecWPfuturefactory-LR-en.pdf>.
- [4] 勝村 義輝, 他,日本機械学会論文集 vol.83,No.848, 2017.
- [5] D. Wu, Journal of Manufacturing Systems 32, 2013.
- [6] P. Samimi, Information Sciences 357 (2016) 201-216, 2016.