Deep Learning for Optimal Resource Allocation in IoT-enabled Additive Manufacturing

Farshad Mashhadi, *Student Member*, and Sergio A. Salinas Monroy, *Member*
Department of Electrical Engineering and Computer Science
Wichita State University
Wichita, Kansas, USA
{fxmashhadi@shockers, sergio.salinasmonroy@}.wichita.edu

***Abstract*—Additive manufacturing is revolutionizing the way that we produce, deliver, and consume objects in many industries. Compared to traditional manufacturing, where large factories mass produce objects far away from consumers, additive manufacturing can build customized objects in many small factories closer to where they are needed. Building these simplified supply chains requires manufacturers to employ IoT technologies that can provide real-time monitoring of the 3D-printers, and automatically adjust their operation to meet a highly-dynamic object demand. To this end, researchers have proposed the Additive Manufacturing (AM) cloud that can automatically manage the manufacturing resources based on real-time data collected from IoT-enabled equipment. However, most existing works on the AM Cloud only focus on finding production decisions that minimize the operating costs of the AM Cloud, and disregard setting prices for their services, which limits the profits of the manufacturers. To bridge this gap, we design a deep-learning auction that maximizes the utility of the AM Cloud by assigning 3D-printers to the buyers who are willing to pay the highest prices for their objects, and by assigning production orders to the manufacturers that can build the objects at the lowest cost. The auction prevents buyers from unfairly affecting the results, and its assignments satisfy production capacity and raw material constraints. We conduct extensive simulations, and see that our proposed auction mechanism can both find production control decisions and improve the utility of the AM Cloud by up to 50% compared to existing approaches.**

I. Introduction

Additive manufacturing (AM) is revolutionizing the way that we produce, deliver, and consume objects in many industries including healthcare, transportation, and consumer products [1]. Compared to traditional manufacturing, additive manufacturing offers significantly lower marginal costs that can lead to simplified supply chains. In particular, since 3D-printers can build different object designs without retrofitting its tools, their per-object cost is mostly independent of the production volume [2]. For this reason, small additive manufacturing facilities with only a few machines and small production volumes can be placed closer to where their objects are needed while remaining profitable.

Building these simplified supply chains requires manufacturers to employ IoT technologies that can provide real-time monitoring of the 3D-printers, and automatically adjust their operation to meet a highly-dynamic object demand. To this end, researchers have proposed the AM Cloud, which centrally manages the production resources of a set of small additive manufacturing enterprises, called micro-manufacturers, based on real-time data obtained by IoT-enabled manufacturing equipment. The AM Cloud maintains short shipping distances by assigning production orders to the micro-manufacturer that is nearest to the object's destination, and can help micro-manufacturers fulfill large orders that they would have been unable to fulfill on their own. Furthermore, buyers can access the resources from multiple micro-manufacturers without having to deal with each of them individually.

However, managing the AM Cloud poses two significant challenges. First, the AM Cloud needs to allocate production orders to its micro-manufacturers in such a way that its operating costs are minimized while satisfying the production capacity and raw material availability constraints of the manufacturers. Second, the AM Cloud needs to set prices for accessing its manufacturing resources in such a way that its profit is maximized, and object buyers are incentivized to participate in the AM Cloud.

There have been some research efforts on resource allocation in a single additive manufacturing facility [3]–[6]. Researchers have also investigated additive manufacturing clouds where multiple manufacturers collaborate to fulfill large production orders [7]–[9]. However, these works only aim to minimize the production costs, and thus lack a mechanism to set prices for access to their manufacturing resources, which is crucial to maximize the profits of the AM Cloud.

To set prices, researchers have successfully employed auctions in other cyber-physical systems, including edge and cloud computing [10], cognitive radio networks [11], power systems [12], and supply-chains [13]. In an auction, bidders compete over a set of resources by submitting their bids to an auctioneer who allocates the resources to the bidders who value them the most. However, these works either assume there are always enough resources to satisfy all the demand, e.g., cloud-computing auctions, or assume that bidders have the same valuation for resources provided by different sellers, e.g., power systems. Thus, they are inadequate for the AM Cloud where demand can exceed the production capacity, and bidders prefer to receive objects from nearby manufacturers due to the shorter shipping times.

In our recent work [14], we considered the problem of optimally allocating 3D-printing area from a set of micro-manufacturers to a set of buyers in the AM Cloud. We used a randomized auction to allocate the (possibly limited)

978-1-7281-5503-6/20/$31.00 ©2020 IEEE

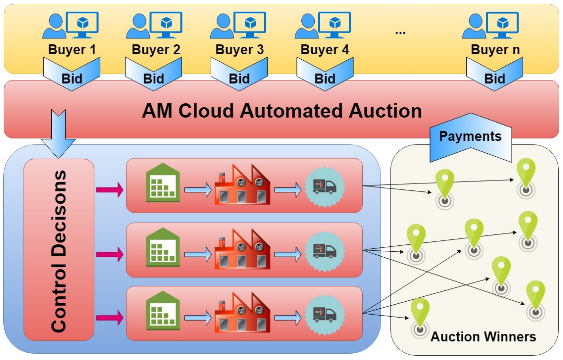


Fig. 1: The AM Cloud Architecture.

manufacturing resources to a set of buyers who have different valuations for objects that are built by different micro-manufacturers. In this work, we instead use a *deep learning auction mechanism* that also considers heterogeneous object valuations, but can achieve a significantly higher utility for the AM Cloud compared to randomized auctions. To the best of our knowledge, this is the first mechanism that uses deep learning to simultaneously manage the production and set prices in additive manufacturing.

Specifically, we design a deep neural network that learns how to match the 3D-printers from the micro-manufacturers that can build the object at the lowest cost to the buyers who are willing to pay the highest prices to access the 3D-printers. The production order allocation decisions satisfy the capacity and raw material availability constraints of the micro-manufacturers. To calculate the price for accessing the 3D-printers, we design a second deep-neural network that learns how to calculate a payment for each buyer that maximizes the utility of the AM Cloud, and prevents the buyers from gaming the auction. We train the proposed deep learning networks with a large data set of sample bids. Our extensive testing shows that our proposed scheme can results in utility gains for the AM Cloud that are up to 26% higher than that of a posted price auction, and more than 50% compared to multi-unit auctions. Note that neither the posted price or multi-unit auctions can satisfy the physical constraints of the system, i.e., the production capacity and raw material constraints. We also see that the 3D-printer allocation decisions and payments can be computed by the trained neural networks within a few seconds, which is practical.

II. Problem Formulation

A. System Model

We consider an AM Cloud where an operator centrally manages the production resources of a set of micro-manufacturers $\mathcal{M} = \{1, 2, \ldots, M\}$ to build objects for a set of buyers $\mathcal{B} = \{1, 2, \ldots, B\}$. Micro-manufacturers build objects on-demand and according to the buyers' object designs. A micro-manufacturer can fabricate objects for several buyers, and the objects from a single buyer can be produced by multiple micro-manufacturers. Moreover, buyers are usually located near each other inside consuming centers, e.g., cities, industrial parks, etc., and micro-manufacturers can be located close to buyer clusters or in remote areas.

The operator aims to maximize the revenue of the micro-manufacturers by selling as many objects as possible subject to the production capacity of micro-manufacturers 3D-printers, and the amount of raw material available at the micro-manufacturers.

The total amount of objects that a micro-manufacturer can produce is limited by the number of 3D-printers and available raw material that it owns. Let $\mathcal{Q}_m = \{1, 2, \ldots Q_m\}$ be the set of 3D-printers at micro-manufacturer m, where $Q_m = |\mathcal{Q}_m|$ is the total number of printers, and let $\mathbf{1}_m^{j,b}$ be an indicator function that is equal to one if the jth 3D-printer at micro-manufacturer m has been assigned to buyer b, and zero otherwise. Then, the total object production at micro-manufacturer m is bounded by

$$\sum_{b \in \mathcal{B}} \mathbf{1}_m^{j,b} \leq \min \left\{ Q_m, \left\lfloor \frac{R_m}{\beta_b} \right\rfloor \right\} \tag{1}$$

for all $m \in \mathcal{M}$, where R_m is the total amount of raw material at micro-manufacturer m, $\beta_b < \beta^{max}$ is the amount of raw material needed to fabricate one object for buyer b, and β^{max} is determined by the printing area of m's 3D-printers. When a 3D-printer is assigned to a buyer, the buyer may choose to build multiple objects per printer instead of only one as long as its objects remain within the dimensions of the printer and there is enough raw material. In this case, β_b represents the raw material needed for all of b's objects inside a single printer.

To calculate the total cost of operating the AM Cloud, we consider the cost of raw material, object fabrication, and shipping. Let r_j^m, f_j^m, and s_b^m be the per unit cost of raw material, fabrication, and shipping between micro-manufacturer m and buyer b. Then, the total cost of building and delivering one object for b with the jth printer at micro-manufacturer m is given by

$$c_{j,b}^m = \beta_b (r_j^m + f_j^m + s_b^m), \tag{2}$$

for all $m \in \mathcal{M}, j \in \mathcal{Q}_m$, and $b \in \mathcal{B}$. The fabrication cost f_j^m accounts for energy consumption, object handling, and administrative costs. The raw material cost r_j^m can differ between micro-manufacturers due to differences in their suppliers and machines. The shipping cost s_b^m is given by a monotonically increasing function of the distance between m and b.

B. An Auction and Control Mechanism for the AM Cloud

The operator conducts an auction that allocates a certain amount of 3D-printers to each buyer considering the physical constraints of the micro-manufacturers, i.e., production capacity and raw material availability. The auction also calculates the buyers' payments.

Specifically, each buyer $b \in \mathcal{B}$ has a private bid defined by the tuple $P_b = \{v_{b,j}, D_b\}$ that specifies its valuation $v_{b,j}$ of receiving printer j, and its budget D_b. Buyer b draws its bid P_b from its bid space $\mathcal{P}_b$ according to a probability distribution F_b that is known to the operator, but not to the other buyers. We denote the set of private bids by $\mathbf{P} = \{P_1, P_2, \ldots, P_B\}$, and the space of buyer bid sets by $\mathcal{P}$. Moreover, we denote the set of bids without the bid from buyer b by $\mathbf{P}_{-b} = \{P_1, ..., P_{b-1}, P_{b+1}, ...P_B\}$.

The auction starts with buyers reporting their bids to the operator. Buyers may submit bids that are different to their private bids to attempt to increase their utility. We denote the reported bid from buyer b by P'_b, and the set of reported bids by $\mathbf{P}' = \{P'_1, P'_2, ..., P'_B\}$. The auctioneer then finds a randomized 3D-printer allocation rule $a : \mathcal{P} \to [0,1]^{B\times Q}$, where $Q = \sum_{m\in\mathcal{M}} Q_m$ is the total number of 3D-printers in the AM Cloud. The allocation rule a maps the vector of reported buyer bids $\mathbf{P}' \in \mathcal{P}$ to a matrix of allocation probabilities $\mathbf{A}(\mathbf{P}') \in [0,1]^{B\times Q}$. The matrix $\mathbf{A}(\mathbf{P}')$ has a row for each buyer, and a column for each printer in the system. Hence, the element $a_{b,j'}(\mathbf{P}')$ is the probability that the jth printer at micro-manufacturer m builds an object for buyer b, and it is located in the bth row and j'th column, where $j' = (\sum_{n=1}^{m-1} Q_n + j)$ (for all $b \in \mathcal{B}$, $j \in \mathcal{Q}_m$ and $m \in \mathcal{M}$). Since a single 3D-printer can only be allocated to one buyer, we require that $\sum_{b=1}^{B} a_{bj}(\mathbf{P}') \leq 1$, $\forall j$. A single buyer b can be allocated multiple 3D-printers, and some 3D-printers may remain unallocated.

The operator also finds a payment rule $p : \mathcal{P} \to \mathbb{R}^{B\times 1}_{\geq 0}$ that maps the reported buyer bids $\mathbf{P}' \in \mathcal{P}$ to the buyers' expected payments $p_b(\mathbf{P}')$ (for all buyers $b \in \mathcal{B}$).

Moreover, based on their allocated printers and required payments, the buyers receive a utility. We consider buyers that equally value all of their allocated 3D-printers and prefer to receive zero objects rather than to exceed their budgets. Let $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2 \cup \cdots \cup \mathcal{Q}_M$ be the set of all 3D-printers in the system. Then, the utility of buyer b is given by

$$u_b(P_b, \mathbf{P}') = \begin{cases} \sum_{i\in\mathcal{Q}} v_{b,i} a_{b,i}(\mathbf{P}') - p_b(\mathbf{P}') & \text{if } p_b(\mathbf{P}') \leq D_b \\ -\infty & \text{if } p_b(\mathbf{P}') > D_b. \end{cases}$$

Similarly, the AM Cloud receives a utility based on the difference between the cost of building objects and the buyer payments, i.e.,

$$U = \sum_{b\in\mathcal{B}} p_b(\mathbf{P}') - \sum_{b\in\mathcal{B}} \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{Q}} a_{b,i}(\mathbf{P}') c^m_{i,b} \quad (3)$$

where $c^m_{i,b}$ is defined in (2).

C. Auction Requirements

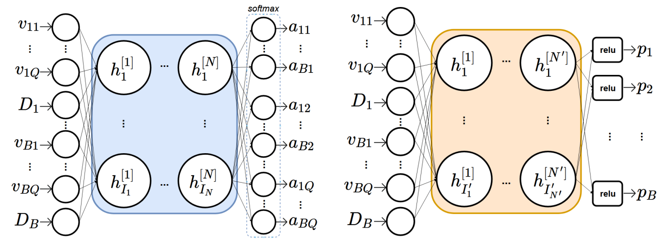
For the auction to be attractive to both the buyers and the AM cloud, the allocation and payment rules need to satisfy the following constraints.

Definition 1. *Budget Constraint (BC). An auction (a, p) satisfies the budget constraints of buyers if it requires payments that are less than their budgets, i.e.,*

$$p_b(\mathbf{P}') \leq D_b, \forall b \in \mathcal{B}, \forall \mathbf{P}' \in \mathcal{P}. \quad (4)$$

Definition 2. *Dominant Strategy Incentive Compatible (DSIC) Constraint. Let $\mathbf{P}'_b = \{P_1, ..., P'_b, ..., P^B\}$ be the set of reported bids, and P'_b be the buyer b's misreported bid, i.e., $P'_b \neq P_b$. Then, an auction (a, p) is said to be DSIC if it satisfies the BC constraint, and guarantees that no buyer can improve its own utility by bidding untruthfully, i.e.,*

$$u_b(P_b, \mathbf{P}') \geq u_b(P_b, \mathbf{P}'_b), \forall b \in \mathcal{B}, \forall \mathbf{P}' \in \mathcal{P}, \forall P'_b \in \mathcal{P}_b. \quad (5)$$



(a) The allocation network. (b) The payment network.

Fig. 2: Neural Networks for the AM Cloud Auction.

Definition 3. *Individual Rationality (IR) Constraint. An auction (a, p) maintains individual rationality if no buyer obtains a negative utility for participating in the auction, that is,*

$$u_b(P_b, \mathbf{P}') \geq 0, \forall b \in \mathcal{B}, \forall \mathbf{P}' \in \mathcal{P}. \quad (6)$$

Definition 4. *Raw Material Availability Constraint. An auction (a, p) satisfies the raw material availability constraint if the amount of raw material needed to produce the objects for the buyers that have 3D-printers allocated at micro-manufacturer m does not exceed the amount of available raw material R_m, i.e.,*

$$\sum_{b\in\mathcal{B}} \sum_{i\in\mathcal{Q}_m} \beta_b a_{b,i}(\mathbf{P}') \leq R_m, \ \forall m \in \mathcal{M}, \forall \mathbf{P}' \in \mathcal{P}. \quad (7)$$

We note that the Raw Material Availability Constraint enforces the second term of the minimum operator in (1), and that the allocation matrix directly enforces the first term through its number of columns and constraints on allocated 3D-printers.

III. A Deep Neural Network for AM Auctions

In this section, we describe the architecture of the proposed auction neural networks for the AM Cloud, and a method to train them.

A. Network Architecture

To find the 3D-printer allocation and payment rules (a, p) that maximize the AM Cloud's utility and satisfy the constraints in Section II-C, we design two feed-forward neural networks that take the bids from buyers as input and output 3D-printer allocations and payments, respectively.

1) Allocation Network: The object allocation neural network (ANN) takes the reported bid profile $\mathbf{P}'$ as input and outputs the values $a_{b,j'}(\mathbf{P}')$'s of the allocation matrix $A(\mathbf{P}')$ as shown in Fig. 2a. The ANN is formed by N fully-connected hidden layers with hyperbolic tangent activation functions, and a fully connected output layer with `softmax` activation.

We denote the parameters of the allocation network by $\mathbf{w}$.

The output layer takes the output of the Nth hidden layer as input, and uses a `softmax` function to convert its values into the 3D-printer allocation probabilities $a_{b,j'}$'s in matrix $\mathbf{A}$. To account for unallocated objects, we have an additional element in the Nth hidden layer for each printer in the AM Cloud.

2) Payment Network: The payment rule neural network (PNN) takes the reported bids $\mathbf{P}'$ as input and outputs the payments $p_b(\mathbf{P}')$ for all the buyers as shown in Fig. 2b. The PNN is formed by N' fully connected hidden layers where the first $N'-1$ layers apply hyperbolic tangent activation functions. In the N'th layer, the network uses a sigmoid activation function to find a fractional payment $\tilde{p_i} \in [0,1]$. Then, the PNN computes the payments for buyer b based on the output of its N' hidden layer and the ANN output as follows: $p_b = \tilde{p}_b \sum_{i=1}^{Q} a_{b,i} v_{b,i}, \forall b \in B$. Note that this formulation of p_b enforces the individual rationality constraint in (6) because buyer b's payment is greater than zero only if it receives printers from the auction.

B. An Optimization Problem for Training the Auction Networks

Before formulating the optimization problem for training the allocation and payment neural networks, we define a set of penalty functions to measure the constraint violations given an auction (a,p).

To measure the degree by which the auction (a,p) deviates from the DSIC constraint in Section II-C, we define regret penalty which shows the maximum gain in *ex post* utility that buyer b could obtain by misreporting its bid, i.e.,

$$rgt_b(a,p) \\ = E_{\mathbf{P}'\sim F}\left[\max_{P_b'\in\mathcal{P}_b}\left\{\mathbf{1}_{p_b(\mathbf{P}_b')\le D_b}\cdot\left(u_b(P_b,\mathbf{P}_b')-u_b(P_b,\mathbf{P}')\right)\right\}\right] \tag{8}$$

where $\mathbf{1}_{\left(p_b(\mathbf{P}_b'))\le D_b\right)}$ is a binary variable that is equal to one if $p_b(\mathbf{P}_b') \le D_b$ holds, and zero otherwise, and $\mathbf{P}_b'$ is the set of bids with buyer b's bid misreported.

We measure the deviation from the BC constraint which we call budget penalty, as follows:

$$bcp_b = E_{\mathbf{P}'\sim F}[\max\{0, p_b(P_b,\mathbf{P}') - D_b\}] \tag{9}$$

To ensure that the auction results satisfy the raw material availability constraint in (7), we measure the difference between the available and required amount of raw material by the auction at each micro-manufacturer m:

$$rmp_m = E_{\mathbf{P}'\sim F}[\max\{0, -e_m(\mathbf{P}')\}], \forall m \in \mathcal{M} \tag{10}$$

where $e_m = R_m - \sum_{b\in\mathcal{B}}\sum_{i\in\mathcal{Q}_m} a_{b,i}(\mathbf{P}')\beta_b$.

The objective of the auction (a,b) is to maximize the utility of the AM Cloud. To this end, we set the objective of the allocation and payment networks as the minimization of the negated expected utility of the AM Cloud as follows:

$$\mathcal{L}(a,p) = -E_{\mathbf{P}'\sim F}[U(\mathbf{P}')] \tag{11}$$

Based on the above constraint violation metrics and objective, we can train the neural networks with the following optimization problem. Let $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{w}' \in \mathbb{R}^{d'}$ denote the vector of parameters of the allocation and payment networks, respectively, where d and d' are the number of tunable parameters in the neural networks. Let a^w and $p^{w'}$ be the induced allocation and payment rules due to the neural network parameters $\mathbf{w}$, and $\mathbf{w}'$, respectively.

We define the loss function for the allocation and payment rules $(a^w, p^{w'})$, as the negated expected utility $\mathcal{L}_(a^w, p^{w'}) = -E_{P\sim F}[\mathbb{U}(P)]$, where $\mathbb{U}(P)$ is the utility of the AM Cloud for the bid profile P. Then, the training problem is given by

$$\begin{aligned} &\min_{\mathrm{w}\in\mathbb{R}^d,\mathrm{w}'\in\mathbb{R}^d} \mathcal{L}(a^w, p^{w'}) \\ &\text{s.t.}\;\; rgt_b(a^w, p^{w'}) = 0, \forall b \in B \\ &\qquad bcp_b(a^w, p^{w'}) = 0, \forall b \in B \\ &\qquad rmp_m(a^w, p^{w'}) = 0, \forall m \in M. \end{aligned} \tag{\textbf{P1}}$$

C. Data Set Generation and Deep Network Training

Directly solving **P1** is challenging because the constraint violation metrics and objective are in terms of expected values. Instead, we reformulate **P1** in terms of sample bid profiles. Specifically, we sample L bid profiles i.i.d. from the distribution of profiles F and group them in set $\mathcal{S} = \{\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \ldots, \mathbf{P}^{(L)}\}$. To calculate the sample mean of the regret penalty, we generate an additional set of misreported bids $\mathcal{S}_l' = \{\bar{\mathbf{P}}^{(1)}, \ldots, \bar{\mathbf{P}}^{(K')}\}$ for each $\mathbf{P}^{(l)}$ in $\mathcal{S}$, where $\bar{\mathbf{P}}^k \in \mathcal{S}_l'$ (for all $k \in [1, K']$) are sampled from the space $\mathcal{P}$ according to a distribution that is not necessarily equal to F. By feeding the misreported bid profiles to the neural networks, we calculate the sample mean of the regret. Similarly, we can use the output of the neural networks with the sample profiles as input to calculate the sample mean of the budget constraint, and the raw material availability constraint as well as for the AM Cloud Utility.

Using the sample bids, we form a sample mean optimization problem that can be solved with the augmented Lagrangian method [15] and the ADAM solver [16]. Specifically, we first replace the expected values in **P1** with their sample mean expressions, and form the following augmented Lagrangian function:

$$\begin{aligned} &\mathcal{F}_\rho(w, w'; \lambda_{rgt}, \lambda_{irp}, \lambda_{bcp}, \lambda_{rmp}) \\ &\quad = \hat{\mathcal{L}}(a^w, p^{w'}) \\ &\quad + \sum_{b\in\mathcal{B}} \lambda_{rgt,b}\widehat{rgt}_b(a^w, p^{w'}) + \frac{\rho}{2}\sum_{b\in\mathcal{B}} \widehat{rgt_b}^2(a^w, p^{w'}) \\ &\quad + \sum_{b\in\mathcal{B}} \lambda_{bcp,b}\widehat{bcp}_b(a^w, p^{w'}) + \frac{\rho}{2}\sum_{b\in\mathcal{B}} \widehat{bcp_b}^2(a^w, p^{w'}) \\ &\quad + \sum_{m\in\mathcal{M}} \lambda_{rmp,m}\widehat{rmp}_b(a^w, p^{w'}) + \frac{\rho}{2}\sum_{m\in\mathcal{M}} \widehat{rmp_m}^2(a^w, p^{w'}) \end{aligned}$$

where $\lambda_{rgt} \in \mathbb{R}^B$, $\lambda_{bcp} \in \mathbb{R}^B$, and $\lambda_{rmp} \in \mathbb{R}^M$ are vectors of Langrangian multipliers corresponding to the equality constraints in **P1**, and $\rho > 0$ is a fixed parameter used to control the weight on the augmented quadratic terms. The terms $\hat{\mathcal{L}}(a^w, p^{w'}), \widehat{rgt}_b(a^w, p^{w'}), \widehat{bcp}_b(a^w, p^{w'})$, and $\widehat{rmp}_b(a^w, p^{w'})$ denote the sample mean of the AM Cloud utility, the regret penalty constraint, the budget penalty constraint, and the raw material availability constraint, respectively.

Then, we use the ADAM solver to update the parameters w and w' and only update the Lagrangian multipliers after a constant number of iterations of the ADAM optimizer to avoid large swings in the parameter updates. Every iteration of the ADAM solver, feeds the $\mathbf{P}^{(l)}$'s to the neural networks defined

TABLE I: Simulation Scenario Settings.

Scenario	I	II	III	IV	V
Number of buyers B	4	8	8	15	20
Number of manufacturers M	2	3	6	10	10
Total number of printers Q	4	10	20	30	60
Maximum number of objects R	3	6	12	20	40

by w and w' to calculate the updated utility and the constraint violations sample means.

IV. Simulation Results

To evaluate the performance of our proposed deep learning auction mechanism for the AM Cloud, we consider several scenarios as described in Table I. In this table, the 1st row indicates the scenario, the 2nd row indicates the total number of buyers B, the 3rd row is the total number of micro-manufacturers in the AM Cloud M, the 4th row is the number of 3D-printers in the AM Cloud Q, and the 5th row is the total number of objects that the AM Cloud can produce with the available raw material at all micro-manufacturers, i.e., $R = \sum_{m \in \mathcal{M}} R_m$. Both the 3D-printers and the raw material are randomly distributed among the micro-manufacturers. The amount of raw material needed for each object β_b is set to 1 for all buyers. We choose the per unit cost of raw material, fabrication, and shipping for each of the micro-manufacturers uniformly at random from the intervals $[\$0.06, \$0.08]$, $[\$0.07, \$0.1]$, and $[\$0.05, \$0.1]$, respectively. In Scenarios I-III, the budgets D_b's are drawn randomly from the interval $[\$0.01, \$2]$, in Scenario IV from $[\$0.01, \$4]$, and in Scenario V from $[\$0.01, \$10]$. Buyers' valuations v_b's are chosen at random from the interval $[\$0.01, \$1]$ in all scenarios. We allocate 3D-printers to the buyer that receives the highest allocation probability in $\mathbf{A}(\mathbf{P}')$.

To train the neural networks, we generate a sample set $\mathcal{S}$ with $L = 640,000$ sample bid profiles, along with their corresponding misreported bid sets $\mathcal{S}_l'$'s, and divide it into 5,000 mini-batches. The network is then trained over 80 epochs of $5,000$ iterations, an overall of 400,000 iterations.

To find the optimal values of the neural network hyper-parameters, we trained the networks with varying number of layers and layer sizes. The networks performed best with 6 hidden layers of 100 neurons each. The parameter ρ in the Langangian function was initialized to 0.01. We test the networks' auction results with a set of 10,000 bid profiles that generated in the same way as $\mathcal{S}$. We implement the neural networks in software using TensorFlow, and use the Glorot uniform method to initialize them. We train both networks on a general purpose PC with an Nvidia GTX 1070 Ti GPU, 16GB RAM, and 1TB SSD external memory.

We first investigate the utility of the AM Cloud under the proposed auction mechanism and compare it to results obtained under a posted price auction [17], and a multi-unit auction with budget constraints [18]. Since these auctions are unable to handle the raw material availability constraint, and considering the shipping distances would significantly increase their computational complexity, we compute their utility by first finding the buyers that win the auction, and then matching the winners to specific 3D-printers based on an optimization problem that minimizes the production cost and satisfies the physical constrains. Fig. 3a shows the expected utility of the AM Cloud as the number of training iterations increases in Scenario II. We observe that the utility stabilizes around \$1.7 after 300,000 iterations, and that it is higher than the utility found by the posted-price and multi-unit auctions.

TABLE II: The AM Cloud's utility and constraint violations for all scenarios.

Scenario	I	II	III	IV	V
Utility	0.7574	1.74	3.18	5.82	16.73
Regret penalty	0.0011	0.0013	0.0016	0.0013	0.0014
Budget constraint penalty	0.0011	0.0012	0.0014	0.0014	0.0014
Raw material availability	0.0011	0.0010	0.0011	0.0011	0.0015

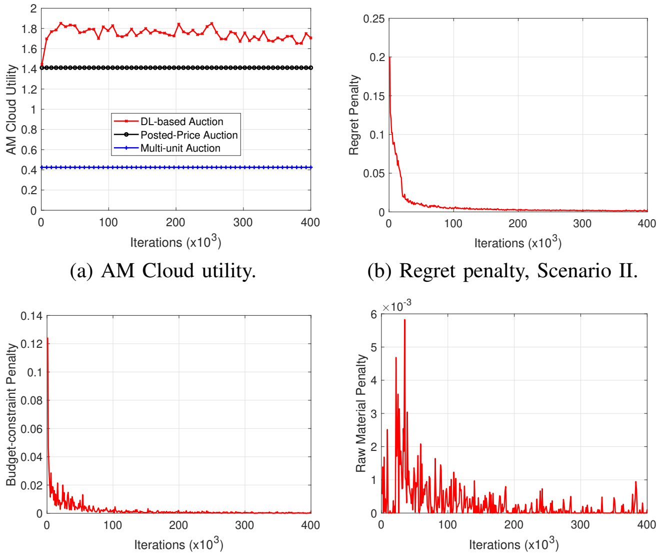
In Fig. 4, we show the AM Cloud utility for all the scenarios, and compare it to [17] and [18]. We observe that our proposed AM Cloud auction mechanism achieves a utility of up to 26% higher than that of the posted price auction, and at least 50% higher than that of the multi-unit auction in all scenarios. This shows the main contribution of our approach: to achieve a higher utility for the AM Cloud it is crucial to simultaneously compute the auction winners, their payments, and the allocation of 3D-printers to specific buyers.

Next, we evaluate how well the auction and allocation payment (a, p) found by our proposed auction mechanism satisfies the auction constraints of dominant strategy incentive compatible (DSIC), buyer budgets, and raw material availability. In Fig. 3b, we measure how much the proposed auctions deviate from the DSIC property, i.e., the total utility buyers could obtain by submitting bids with valuations and budgets different from their true valuations and surveys. We call this regret. We see that as the regret approaches zero as the number of iterations increases. Hence, buyers have no incentive to deviate from the true budgets and valuations, thus the DSIC constraint is satisfied. Fig. 3c shows the total amount of money that buyers need to pay above their budgets according to the payment p found by the proposed auctions. We observe that the extra payment buyers need to pay above their budgets is close to zero nearly after 200,000 iterations. In Fig. 3d, we show the amount of total raw material that micro-manufacturers need beyond their current inventory, and see that it approaches zero after 200,000 iterations as well. Note that the individual rationality and the production capacity constraints are directly enforced by the structure of the payment neural network and by the structure of the allocation matrix, respectively. We observe similar results in all five scenarios.

Table II summarizes the violation from the regret, budget, and raw material availability constraints for all the scenarios after 400,000 iterations.

We report the training and computing times of the proposed neural networks in Table III. The computing time measures how long it takes the neural networks to find the allocation a and payment p given a bid profile $\mathbf{P}'$ on a virtual machine with 5 vCPUs and 8GB RAM. We see that it takes only a few seconds to find the allocation and payments for all scenarios, which is practical for the AM Cloud.

We see that the training time increases as the number of buyers and 3D-printers increases. We can expect to have larger neural networks for larger systems, where there are a higher number of parameters that need to be trained. However, the



(a) AM Cloud utility.

(b) Regret penalty, Scenario II.

(c) Budget penalty, Scenario II.

(d) Raw material availability, Scenario II.

Fig. 3: Performance of the proposed scheme.

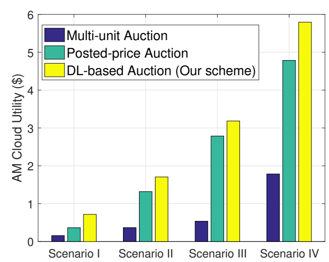


Fig. 4: The utility of the AM Cloud under different auction approaches.

TABLE III: Training and computing time.

Scenarios	**I**	**II**	**III**	**IV**	**V**
Training Time (hours)	4.1	5.5	7.1	12.9	28.7
Computing Time (seconds)	0.03	0.29	1.57	2.5	4.1

proposed neural networks only need to be trained once for a given set of buyers and micro-manufacturers. To enable the cloud to seamlessly expand in the future, we can train the neural networks with additional buyers and micro-manufacturers, and set their bids and capacity to zero, respectively, until new buyers and 3D-printer join the system. Moreover, if buyers leave the AM Cloud, we can also set their bids to zero without having to retrain the neural networks. Similarly, if a 3D-printer becomes unavailable we can set its available raw material to zero so it receives no production orders.

V. Conclusions

In this paper, we studied the problem of optimally setting prices to access the production resources of micro-manufacturers in the AM Cloud and matching the object buyers to 3D-printers. Since additive manufacturing facilities can be profitable even at small scales, we can place many of them close to the consumers. However, managing their production decisions and setting fair prices for the object buyers is a challenging problem. To address this challenge, we proposed a deep learning based auction that maximizes the utility of the AM Cloud by finding optimal allocation decisions and payments for the buyers, The auction mechanism prevent the buyers from unfairly manipulating the results of the auction, and satisfies the production capacity, and raw material constraints of the AM Cloud. We conducted extensive simulations and observe that the proposed neural network auctions can find a utility for the AM Cloud that is higher than the utility found by auction schemes.

References

[1] D. Thomas, "Economics of the us additive manufacturing industry," *NIST Special Publication*, vol. 1163, 2013.

[2] M. Baumers, L. Beltrametti, A. Gasparre, and R. Hague, "Informing additive manufacturing technology adoption: total cost and the impact of capacity utilisation," *International Journal of Production Research*, vol. 55, no. 23, pp. 6957–6970, 2017.

[3] I. Kucukkoc, Q. Li, N. He, and D. Zhang, "Scheduling of multiple additive manufacturing and 3d printing machines to minimise maximum lateness," *Twent Int Work Semin Prod Econ*, vol. 1, pp. 237–247, 2018.

[4] A. Chergui, K. Hadj-Hamou, and F. Vignat, "Production scheduling and nesting in additive manufacturing," *Computers & Industrial Engineering*, vol. 126, pp. 292–301, 2018.

[5] F. Dvorak, M. Micali, and M. Mathieug, "Planning and scheduling in additive manufacturing," *Inteligencia Artificial*, vol. 21, no. 62, pp. 40–52, 2018.

[6] Y. Oh, C. Zhou, and S. Behdad, "Production planning for mass customization in additive manufacturing: Build orientation determination, 2d packing and scheduling," in *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2018, pp. V02AT03A033–V02AT03A033.

[7] J. Mai, L. Zhang, F. Tao, and L. Ren, "Customized production based on distributed 3d printing services in cloud manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1-4, pp. 71–83, 2016.

[8] L. Zhou, L. Zhang, Y. Laili, C. Zhao, and Y. Xiao, "Multi-task scheduling of distributed 3d printing services in cloud manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 96, no. 9-12, pp. 3003–3017, 2018.

[9] Q. Li, D. Zhang, S. Wang, and I. Kucukkoc, "A dynamic order acceptance and scheduling approach for additive manufacturing on-demand production," *The International Journal of Advanced Manufacturing Technology*, pp. 1–19, 2019.

[10] N. C. Luong, Z. Xiong, P. Wang, and D. Niyato, "Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[11] L. Gao, L. Duan, and J. Huang, "Two-sided matching based cooperative spectrum sharing," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 538–551, 2017.

[12] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, 2017.

[13] Q.-h. Zhang and J.-w. Luo, "Study on the supply chain coordination of perishable items based on double auction," *Industrial Engineering and Management*, vol. 14, no. 3, pp. 33–44, 2009, double sided.

[14] F. Mashhadi and S. S. Monroy, "Economically-robust dynamic control of the additive manufacturing cloud," *IEEE Transactions on Services Computing*, 2019.

[15] Z. Feng, H. Narasimhan, and D. C. Parkes, "Deep learning for revenue-optimal auctions with budgets," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 354–362.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] P. Khezr, "Auction versus posted-price and the informed seller problem," 2012.

[18] C. Borgs, J. Chayes, N. Immorlica, M. Mahdian, and A. Saberi, "Multi-unit auctions with budget-constrained bidders," in *Proceedings of the 6th ACM conference on Electronic commerce*. ACM, 2005, pp. 44–51.