EAI Endorsed Transactions
on Smart Cities

Research Article EAI.EU

Aspects of Mechanism Design for Industry 4.0 Multi-Robot Task Auctioning

Ajay Kattepur[1], Sourav Khemkha[2,*]

[1]Ericsson Artificial Intelligence Research, Bangalore India.
[2]Dept. of Mathematics, Indian Institute of Technology, Kharagpur, India.

Abstract

Collaborative multi-robotic tasks are essential in complex Industry 4.0 deployments, involving autonomous robots. As such autonomous robots work with minimal centralized control, analysis of strategies for effective cooperation in task completion are needed. This is specially crucial when automation tasks are outsourced to specialized vendors without centralized control. *Mechanism Design* techniques have been proposed to create scenarios among multiple autonomous (possibly selfish) entities to result in desired outcomes. In this paper, we study the effect of varying bidding auction mechanism design protocols to result in multi-agent task auctioning. We demonstrate this approach over a realistic use case of task allocation involving multiple pick and place robots in Industry 4.0 warehouses. Multiple realistic auction/bidding scenarios considered including selfish agents, erroneous estimates of temporal features, heterogeneous capacities and composite bids. The results demonstrate that effective mechanisms can lead to fair outcomes despite erroneous or biased bids for tasks from agents.

Received on 07 August 2020; accepted on 05 August 2021; published on 12 August 2021

Keywords: Smart Things and Smart Objects, Software Defined Networks and Internet of Things, Smart Homes, Buildings and Malls

doi:10.4108/eai.12-8-2021.170670

1. Introduction

Industry 4.0 integrates the use of robotics, cyber-physical systems and intelligent automation [1]. This has led to increasing proliferation into manufacturing, supply chain, retailing and warehousing sectors. The key enablers for Industry 4.0 include:

1. The ability to *interoperate* between computing, Internet of Things (IoT) [2], robotics and human participants.
2. The ability to add *information* to physical systems, such as those provided by sensor data.
3. Replacing human participants in *technical tasks*, specially those requiring repetitive automation or robotic precision.
4. Robotic systems performing tasks in an *autonomous* fashion, with minimal human intervention.

A fundamental characteristic required in Industry 4.0 deployments is the ability of autonomous robotic devices to self-configure in dynamic goal and deployment conditions. This requires autonomous goal evaluation, reasoning and task completion capabilities in agents.

An Industry 4.0 use case of interest is Amazon's warehouse [1] [6], wherein multiple Kiva robots are used for pick and delivery tasks. Tasks arrive at varying rates and may be be completed by robotic agents available in the warehouse. Warehouse systems have been typically controlled using a centralized monitors, which track inventory and locations of agents on the shop floor. However, this is antagonistic to the principles of Industry 4.0, requiring distributed, autonomous and decentralized participants. This also requires specific locations, states,

[*]Corresponding author. Email: ajay.kattepur@gmail.com

[1]https://www.amazonrobotics.com/

task capacities of individual agents to be monitored and maintained in a centralized fashion, which is not scalable.

The design of systems that have distributed, autonomous and (possibly) selfish agent interactions requires a careful analysis of desired goals. Systematic techniques are needed to model the interactions, constraints and the resultant outcome of such system constraints. *Mechanism Design* [3][4] is concerned with settings where a policy maker faces the problem of aggregating the announced preferences into a system-wide decision. Mechanism design solves a decision or optimization problem with incomplete information on agent capabilities. The most widely used mechanism is the *Vickery-Clarkes-Groves* (VCG) mechanism [5][4], that can guarantee *Pareto Optimality* and promote truthful bidding as a dominant strategy.

In this paper, we apply various mechanism designs to study coordination problems among multiple intelligent robotic agents [7]. We consider the Industry 4.0 warehouse scenario where multiple *picking*, *placing* and *inventory management* tasks are to be completed via an auction mechanism. The agents are autonomous, may have incomplete information about the bids and heterogeneous capacities. The combinatorial auction is formulated with agents vying for tasks, in order to be rewarded with point scores. Point scores gained due to completion of tasks are traded off with battery charging times, that may be used by agents (analogous to cash payments received to human participants). Such a federation of agents is required for large scale deployments, with vendors and heterogeneous robots competing for common tasks.

We integrate the VCG mechanism to enable task auctioning and coordination among the robotic agents in Industry 4.0. Through exhaustive simulations, nuances are studied in scenarios including heterogeneous agent capacities, individual task bids, combinatorial bids and collusion with erroneous estimates. We demonstrate how the mechanisms may be made robust enough to ensure fair and truthful allocation to auction participants. Such a model ensures fair, scalable and efficient deployments of autonomous agents in Industry 4.0 deployments.

The **principal contributions** of this paper are:

1. Porting the Industry 4.0 autonomous agent task allocation problem into a game theoretic mechanism formulation.

2. Evaluating the use of VCG auctioning mechanism designs for combinatorial auctions in cooperative tasks.

3. Consider auctions under multiple scenarios with homogeneous/heterogeneous agents, single/combinatorial tasks and truthful/erroneous bids.

4. Extensive simulation of the above mechanisms, to demonstrate efficacy of solutions in Industry 4.0 context.

The rest of this paper is organized as follows: A survey of the state of the art is presented in Section 2. Section 3 provides an overview of the automated task procurement processes in Industry 4.0 warehouses. An overview of mechanism designs, combinatorial auctions and bidding languages are provided in Section 4. Mapping problems from Industry 4.0 task allocation into mechanism design formalisms, with bidding, allocation and payment is provided in Section 5. Comprehensive simulations are examined in Section 6, including use of heterogeneous agents, composite bids and erroneous estimates. This is followed by conclusions of the paper.

2. Related Work

We provide an overview of the state of the art in combinatorial auctions, mechanism design and Industry 4.0 warehouse automation.

2.1. Mechanism Designs and Auctions

Auctions and bidding techniques have been proposed to allocate resources to parties in a fair manner. This has been extended in *combinatorial auctions* [8], where bidders propose bids on combinations of available items/tasks. As bidders are autonomous and may behave in individualistic manner, it is important to set up games that can result in desirable outcomes both to the auctioneers as well as the bidders.

Mechanism design [3][4] has been proposed to generate social interactions among agents, in order to meet certain goal objectives. It is the assumption of the mechanism design process, that the participants in the social interaction will hold private information and behave in a self-centered manner to maximize private goals. The most widely used mechanism is the *Vickery-Clarkes-Groves* (VCG) mechanism [5][4]. A simple used case of the VCG mechanism is the Vickery Auction or the Second Price Sealed Bid Auction, wherein each buyer submits a sealed bid, the buyer with the highest bid is declared the winner.

The use of combinatorial auctions and mechanisms have been proposed in the logistics, autonomous robotic and vehicular transport segments. In [9], the application of single-round combinatorial auctions have been applied to Home Depot's transporter handling network. The results indicate that the choice of auction mechanisms not only provided better rates, but many of the carriers also expressed increased satisfaction in the awarded tasks. In [10], the use of VCG auctions in multi-tenant autonomous vehicle scheduling is proposed, that would help improve the utilization of resources. In [11], the use of software actors interacting over mechanisms for improved agility and scalability is proposed. In our work, we apply efficient mechanism design to autonomous robots deployed in industrial settings.

2.2. Industry 4.0 Automation

Industry 4.0 deployments [1] propose the use of autonomous robotic entities to complete complex tasks. Commercial deployments have been used in warehouses [12] to improve throughput of automated tasks. Amazon[2] has deployed hundreds of autonomous robots to aid in reducing costs of warehouse logistics [6]. Inspiration is drawn from the use of autonomic computing technologies [13], that allow robotic runtime reconfiguration and adaptation. Architectures with self-aware, self-configuring and self-optimizing capabilities have also been proposed [14], that may be applied to such automation frameworks. For smaller scale deployments, coordinating robotic entities via a centralized cloud [15], could prove useful.

In [16], a smart factory framework is proposed that incorporates industrial network, cloud, and supervisory control terminals with smart shop-floor objects such as machines, conveyors, and products. As the smart factory is characterised by a self-organized multi-agent system, an intelligent negotiation mechanism is proposed for agents to cooperate with each other. Analysis done in [17] demonstrates that the system between the picking and storage area represents the most critical subsystem in automated warehouses. These requirements suggests the development of decentralized control solutions, involving coordination among multi-agent systems. In [18], a distributed optimization framework is proposed to handle task allocations in Industry 4.0 warehouses. In [21], a decentralised multi-agent variant of robotic coordination in an open factory setting with multiple owners of robots as well as different owners of the items to be produced, both considered self-interested and individually rational are considered. This is solved using a multi-agent decentralised optimisation approach that is computationally efficient.

In this paper, we apply mechanism design to the combinatorial auctioning of tasks in Industry 4.0 warehouses. Table 1 provides a detailed comparison with respect to the categories of papers. We have contrasted work using multi-agent optimization, game theoretic models and reinforcement learning. To the best of our knowledge, there is limited work in the intersection of combinatorial auctioning and Industry 4.0 robotic task allocation. We consider auctions under multiple scenarios with homogeneous/heterogeneous agents, single/combinatorial tasks and truthful/erroneous bids. Such an in depth analysis of mechanisms would bring us closer to practical deployments of autonomous entities in industrial settings.

3. Automated Task Procurement Process

In this section, we provide an overview of Industry 4.0 warehouse automation, that makes use of multiple intelligent robotic agents for task completion.

3.1. Intelligent Agents

Traditional techniques to coordinate machines and robots in large warehouses involve centralized architectures. A typical use case could include data analytics and coordination performed over a centralized cloud repository [6]. However, there are multiple drawbacks of centralized coordination, including: (i) Inefficient latency overheads to transmit large datasets to the cloud (ii) Inability to reconfigure in real-time to changes, that is a requirement of multiple manufacturing and transportation scenarios (iii) Lack of scale, dependent on a single computational node to optimize operations.

An alternative to such centralized systems, is to make use of multi-agent systems [7]. Multi-agent systems consist of multiple coordinating *intelligent agents*, that can perform task computations in an autonomous fashion. The intelligent agents posses perception/actuation capabilities to sense/act on the environment – however, this information may be restricted to a limited viewpoint. In order to perform more complex tasks, it is necessary for the agents to coordinate with each other. The data and knowledge captured by agents may be shared amongst the agents via hierarchical or peer-to-peer mechanisms. This information may be used to perform more complex sets of tasks, than would have been possible by individual agents.

To model the robotic components in warehouses, we make use of the *Autonomous Robot* abstraction, inspired by intelligent agents [19]. Typical activities, for instance with a pick & place robot in a smart warehouse, include:

1. *Goals*: Understanding goals of each task and subtask, such as, placing correct parts into correct bins within the given time constraints.

2. *Perception*: Object identification and obstacle detection using camera and odometry sensors that sense the environment. This aids the robot in object detection and identification.

3. *Actions*: Identifying granular actionable sub-tasks, such as, moving to particular location, picking up parts of orders or sorting objects. Constraints may be placed on the robot capabilities, motion plans and accuracy in performing such actions.

4. *Knowledge Base*: Using domain models of the world for goal completion, such as warehouse environment maps, rack type and product features.

We further elaborate on task allocation in the Industry 4.0 automation setting, next.

[2] https://www.amazonrobotics.com/

Table 1. State of the Art Comparison on Industry 4.0 Multi-Robot Task Auctions.

Papers	Approach	Contrast with our approach.
[13] [15]	Cloud/Edge Robotics for Industry 4.0	Techniques in this area largely rely on a centralized coordinator to orchestrate the tasks. The cloud, edge or master controller is responsible for allocating tasks to the agents and monitoring progress. In contrast, our approach makes use of an auctioning mechanism where the agents may bid for jobs and available tasks. Any deviations in bidding or errors re handled by the system. The advantage is that multiple vendors / robot types may participate in the auctioning process.
[16] [18] [21]	Distributed Optimization, Multi-Agent Systems for Industry 4.0	These category of papers consider an open and flexible deployment of agents within the factory floor. There may be variations in demands, heterogenous agents and agents incoming/leaving the system. Unlike centralized optimization approaches, these rely on decentralized optimization approaches. This is in line with the area of this work. However, we make use of combinatorial auctions that are an alternative to optimization based approaches. The optimization happens at the bidding level rather than the task level.
[11] [9] [10]	Game Theory and Auctions for Autonomous Agents	These set of approaches make use of game theory, mechanism design and combinatorial auctions to solve problems in autonomous agents. However, the settings are in the case of autonomous electric vehicles and software, which have their own set of constraints. There is not much analysis of selfish agents and overbidding that has been done in this paper. Moreover, Industry 4.0 robotic coordination comes with its own set of challenges and bidding specifications, that are to be included.
[22] [23]	Reinforcement Learning Approaches for Multi-Robot Coordination	These approaches make use of machine learning and reinforcement learning to coordinate multiple robots. However, an extended training period is needed, which cannot be guaranteed in all cases. Our solution is a higher level alternative to this, wherein coordination of autonomous agents is carried out via VCG auctioning.

3.2. Industry 4.0 Warehouse Tasks

Industry 4.0 warehouse tasks require coordination between multiple autonomous agents. This is specifically needed in large warehouse set-ups where a "parts-to-picker" model involving items on a conveyor belt has to be replaced by a "picker-to-parts" model, such as making use of autonomous mobile robots. To further elaborate we present the following realistic scenario:

A large warehouse is presented, such as those managed by Amazon or Dell. Orders arrive periodically and are retrieved by a fleet of autonomous robots. A set of server robots receive the orders and efficiently allocate them to the delivery robots. The delivery robots move along the warehouse floor and approach appropriate product locations. The delivery robots recognize the correct items and make use of robotic arms to pick the objects. There are constraints on the task completion times that must be met. The robots have limited carrying capacities and drain batteries as they perform tasks. There are also variations in order arrival rates.

The scenario described above requires multiple aspects to be taken into account. First, there is a lack of a centralized control mechanism, requiring a protocol for coordination among the agents. Second, rather than accumulating tasks in batches to be procured at a later stage, the processing must be done in a first come first serve basis. Third, the number of agents and the tasks allocated should be appropriately scaled up in accordance with order rates.

Fig. 1 provides an overview of the warehouse automation model. Specialized agents are used to schedule and procure products with minimal external control. Server agents broadcast goal tasks to delivery agents. Mobile delivery agents acquire a subset of tasks and execute them within the time constraints. They are provided with points for completion of tasks, that may be used to recuperate battery power at charging stations. The agents may be penalized for unfulfilled orders. The server agents keeps track of the increase/decrease in item inventory in the warehouse.

Do note that the tasks are typically complex, requiring coordination among multiple agents. A typical case would be products to be re-tried from disparate locations, that would be efficiently re-tried by agents

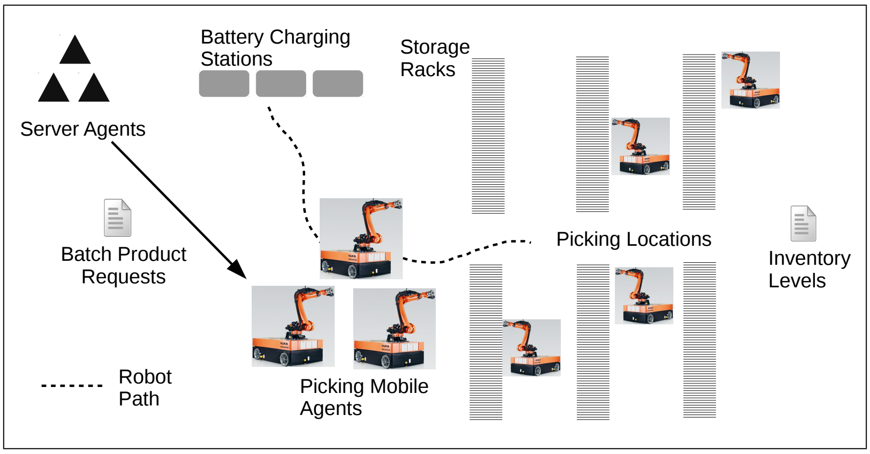


Figure 1. Multiple Agents deployed in Warehouses.

located in close proximity. The auctioning and allocation of tasks is typically limited by the following constraints:

- *Utilization* – as every agents has limited load carrying capacity, the agents may choose to bid or abstain from tasks that would not maximize utilization.
- *Latency* – every task has a time range that must be met, which might restrict the agents that may participate.
- *Battery Limitation* – the agents also must consider battery capacities, that must last throughput the task duration.

Further elaboration of how auction mechanism formulations may be ported to distributed settings are elaborated in the proceeding sections. Do note that as there is autonomy involved in the process, there are relative estimates of latency and battery depletion entailed with completion of a task – these should be incorporated into the allocation model.

4. Mechanism Design and Auctions

In this section, we introduce combinatorial task auctions that may be applied towards multi-agent task allocation. Mechanism designs that are responsible for generating scenarios where autonomous agents bid for tasks are also studied.

4.1. Combinatorial Task Auctions

Auctions and bidding techniques have been proposed to allocate resources to parties in a fair manner. This has been extended in *combinatorial auctions* [8], where auctions of multiple non-identical items is performed and the bidding price may depend on compositions of other item bids. In general, as agents intend to bid for combinations of items, the combinatorial auctions may lead to superior allocations. However, due to the exponential number of combinations, typically a subset of combinations are allowed to make the complexity tractable.

The four aspects to be specified in combinatorial auctions are:

1. *Bidding*: As each bidder provides bids for a combination of items, the protocol to specify the bids are to be efficiently specified.
2. *Allocation*: The allocation of items to various bidders will choose to maximize an utility function based on the bids.
3. *Payment*: The rules of payment will be such that the auctioneers' revenues are maximized, while ensuring fair allocation. Fairness here refers to the ability of the mechanism to identify collusion or over-optimistic bidding among agents that could lead to tasks not being fulfilled.
4. *Strategy*: As every bidding agent is autonomous, it is assumed that the strategy employed would be motivated by individual gains. The auction should be formulated such that despite individual motivation, the allocation would maximize overall utility.

This paper considers only sealed bid auctions with a private value model for each bidding agent. The modelling is as follows:

1. A single auctioneer presents m items for sale. This is bid on my n bidders, having id i and individual valuation functions v_i.
2. $v_i(S)$ is the valuation provided by bidder i for a subset of items S.
3. The auctioneer determines the winning bidders based on an allocation algorithm: find a pairwise

disjoint set $S_1, \ldots, S_n$ to maximize the overall utility $\sum_i v_i(S_i)$.

As the combinatorial auctions must posses capabilities to express combinations of items [5], the following bidding combinations are typically employed: (i) *Atomic Bids*: The bidder submits a price p for a subset of items S. (ii) *OR bids*: Bidders submit a number of atomic bids (S_i, p_i), with the subset of items S_i being valued at p_i. The bidder may be awarded more than one subset. (iii) *XOR bids*: This is similar to the OR bids; however, only one of the subsets is awarded to each bidder. Such bidding techniques are integral to designing suitable mechanisms for multi-agent task auctioning, described next.

4.2. Mechanism Design

The main focus of mechanism design [3][4] is on on the design of social institutions that satisfy certain objectives, despite the fact that participating individuals hold private information. An instance would be in an auction setting, where the auctioneer would act in favour of increasing the price of items; on the contrary, the bidder would attempt to acquire the goods at the lowest possible value.

Formally speaking, for a finite set of individuals $N = \{1, 2, \ldots, n\}$ represented by i, the set of possible decisions are represented as $d \in D$. We define a few terms that are used in the mechanism design context [3].

Definition 1. Individual Preferences The private information held by individual i is denoted by $\theta_i \in \Theta_i$. The utility function representing preferences over decisions is denoted by $v_i : D \times \theta_i \to \mathbb{R}$. The preference of an individual $v_i(d, \theta_i)$ denotes the advantageous benefit from decision $d \in D$. If an individual prefers decision d over d', it is denoted by $v_i(d, \theta_i) > v_i(d', \theta_i)$.

Definition 2. Allocating a Private Good: In an auction, an atomic good is allocated to a bidder. The allocation is represented as $D = \{d \in \{0,1\}^n : \sum_i d_i = 1\}$, where $d_i = 1$ denotes successful bidder. The successful bidder benefits by θ_i thus producing the valuation $v_i(d, \theta_i) = d_i \theta_i$.

Definition 3. Efficient Decision A decision rule $d(\theta)$ is efficient if:

$$\sum_i v_i(d(\theta), \theta_i) \geq \sum_i v_i(d', \theta_i) \tag{1}$$

for all θ and $d' \in D$. This presents the Pareto Optimal front of allocating the good.

Definition 4. Mechanisms A mechanism is defined as a pair (M, G), where M is the message space and G is an outcome function dependent on the decision D. So, for a set of messages $(m_1, \ldots, m_n)$, the resulting outcomes of decisions are represented by $(g_d(m), g_{t,1}(m), \ldots, g_{t,n}(m))$.

A good mechanism is one wherein the participants individually choose messages dependent on their private information, yet leading to socially desired overall outcomes. Dominant strategies are ones wherein the individuals have the best possible messages with respect to other participants in the mechanism. In other words, the dominant strategy is optimal irrespective of the behaviour of other participants.

Definition 5. Dominant Strategies For an agent i with private information θ_i, a strategy $m_i \in M_i$ is said to be dominant if: $\begin{array}{l} v_i(g_d(m_{-i}, m_i), \theta_i) + g_{t,i}(m_{-i}, m_i) \geq \\ v_i(g_d(m_{-i}, \overline{m}_i), \theta_i) + g_{t,i}(m_{-i}, \overline{m}_i) \end{array}$
for all m_{-i}, $\overline{m}_i$; $\overline{m}$ are the messages made public.

As each agent holds private information θ_i, the mechanism must provide incentives to reveal this information truthfully. Individuals are taxed or subsidized based on the revealed θ_i. This incentive is provided using a transfer function $t : \theta \to \mathbb{R}^n$. For every decision d, the *social choice function* is provided as $(d(\theta), t(\theta))$. Some typical desired properties of social choice functions, include:

1. *Pareto optimality*: Implementing an outcome that is not Pareto-dominated by any other outcome, so no other outcomes make one agent better-off while making other agents worse-off.

2. *Maximized social welfare*: Implementing an outcome that maximizes the total utility across agents. This is often called the efficient outcome. Agent i with type θ_i has utility $v_i(\theta_i, o)$ for outcome $o \in \mathcal{O}$, where $\mathcal{O}$ is the possible set of outcomes. we might wish to achieve efficiency in the system by maximizing the total utility gained across all agents, in which case:

$$f(\theta) = \max_{o \in \mathcal{O}} \sum_{i \in N} v_i(\theta_i, o) \tag{2}$$

3. *Budget balance*: The total payment that agents make equals exactly zero (a strict budget balance), so money is not injected into or removed from a system. Or, the total payment is non-negative (a weak budget balance), so the mechanism does not run at a loss.

4. *Individual rationality*: We can consider individual rationality, in which an agent has non-negative utility in expectation to a given mechanism.

The most widely used mechanism is the *Vickery-Clarkes-Groves* (VCG) mechanism [5][4]. A simple used case of the VCG mechanism is the Vickery Auction or the Second Price Sealed Bid Auction, wherein each buyer submits a sealed bid, the buyer with the highest bid is declared the winner. The winning bidder pays an amount equal to the second highest bid. It can be shown that the mechanism is truthful – a mechanism where bidding the true valuation is a dominant strategy [3].

VCG Mechanism

1. Each agent reports a valuation $\widehat{v}_i$.
2. The mechanism chooses the allocation $(m_1, \dots, m_n)$ that maximizes $\sum_j \widehat{v}_j(m_j)$ and outputs it.
3. For each agent i:
   (a) The mechanism finds the allocation $(m'_1, \dots, m'_n)$ that maximizes $\sum_{j \neq i} \widehat{v}_j(m'_j)$.
   (b) Agent i pays $p_i = \sum_{j \neq i} \widehat{v}_j(m'_j) - \sum_{j \neq i} \widehat{v}_j(m_j)$.

We apply the VCG mechanism designs to evaluate auctions in Industry 4.0 settings in the next section.

5. Industry 4.0 Automation Task Auctions

We re-visit the scenario on Industry 4.0 warehouses from Section 3.2, wherein multiple mobile picker robots are deployed in order to complete a task. The pickup/delivery tasks can arrive at varying rates and may also differ in the number of agents/time-lines expected for completion. *We formulate this problem as via decentralized bidding – the factory/warehouse can only publish tasks; agents that may be managed by multiple vendors bid on tasks.* Each agent is a potential bidder – bidders will place bids on any subset of tasks they want to complete. For each bidding horizon, we assume there are n agents and m tasks. Agents charge a certain amount to complete a task – these points may be traded for charging times at stations (akin to remuneration provided to human agents for task completion). A central system receives all the bids, processes them and allocates the tasks in an optimal way. The charging mechanism ensures that bidders bid truthfully.

Table 2 provides the notations for the agents and tasks in the Industry 4.0 warehouse scenario. Agents are provided additional labels to specify their location coordinates, battery life, carrying capacities and bidding information. An important point to note here is that the agents will bid based on estimation algorithms, that determine the cost (in terms of latency, battery deletion rates) used for valuing the bids. There may be agents who provide gross under/over erroneous estimates of the valuations (akin to human agents lying). It is the ability of the mechanism to handle such cases that determines efficacy in practical deployments.

On receipt of orders to be procured from the warehouse, the agents coordinate to ensure timely completion of the procurement process. This can be delayed by order congestion or unavailability of sufficient agent resources. The process of auctioning tasks among autonomous agents can be broken into three parts: (i) **Bidding process**: The input given to the auctioning problem would be the vector of bidder's valuation and the number of combinations permitted per bidder. (ii) **Allocation of the Winners**: An optimization problem that can be posed as a knapsack problem – relaxations in conditions may be needed to compute the allocation. (iii) **Paying the Winners**: The amount that a winner pays that is socially optimal must be determined by the mechanism.

5.1. Task Auctions

The process for task auctioning starts by the server agents displaying the task, time constraints (if any) and baseline scores for task completion (Fig. 1). If the delivery agents bid for a subset S of tasks $(s_1, s_2, \dots, s_k)$ then his true valuation of this subset is the sum of the distances between agent's current location and each task's location. The actual bid might be greater than the true valuation, when the agent is greedy to gain more point scores. Depletion in the battery is a function of the task valuation to be performed.

Thus, each bid is of the type (S_i, bids_i) where S_i is the subset of the tasks considered for the i^{th} bid and $bids_i$ the value of the bid. As already mentioned we have a total of L bids. The *Python*[3] code snapshot of bid allocation to tasks is provided below:

```python
def assign_bids(bids,erroneous,n,m):
    cnt = 0
    for i in range (erroneous):
        for j in range(m):
            if(i<erroneous):
                bids[cnt] = (int)(bids[cnt]*(1+abs(
                    np.random.normal(0,1))))
                cnt = cnt+1
    return bids

def solve(agents,task,discharge_rate,n,m,erroneous)
    :
    bids = [0 for x in range(m*n)]
    dist = [0 for x in range(m*n)]
    cnt = 0
    for j in range(n):
            for k in range(m):
                dist[cnt] = (agents[j].pos).
                    comp_dis(task[k])
                bids[cnt] = agents[j].bidding_func(
                    dist[cnt] * discharge_rate)
                cnt = cnt+1
    bids = assign_bids(bids,erroneous,n,m)
```

We notice that the `bids` are a function of the `dist` distance to perform the task and `discharge_rate` the battery discharge rate. Bids that are provided by agents

[3] https://www.python.org/

Table 2. Notations for Task Auctioning.

Agent Attributes	n	Number of agents
	agent id	Agent identity
	bids	A set consisting of all the bids
	v	Represent the true valuation to corresponding bid
	L	Total number of bids per task
	n'_{error}	Number of agents who provide erroneous estimates
	n''_{light}	Number of agents who can pick up only light objects
	score	Total amount charged/gained for a completed task
	life	Battery capacity remaining on the agent
	pos	Agent Position
Task Attributes	m	Number of tasks
	task id	Representing the task
	pos	Location of the tasks
	type	0 if task if light weight, 1 if task if heavy weight

with `erroneous` estimates are incremented by a normally distributed random value.

5.2. Allocation of the winners

Given a set of bids in a combinatorial auction, the objective is to find an allocation of items to bidders that maximizes the auctioneer's revenue. The bids are expressions in a bidding language (Section 4.1), by which bidders report valuations for subsets of items The auctioneer's revenue is maximized by choosing an allocation that maximizes the sum of the bidders' valuations for the subset of items that they receive.

Definition 6. Winner Determination Problem: Given bids $bids_i, i = 1, \ldots, n$, the winner determination problem is the problem to compute:

$$x \in arg\max \sum_{i \in N} bids_i(S)x_i(S)|x \text{ is a feasible solution}\Bigg) \tag{3}$$

We have a total of L bids of the form $(S_i, Bids_i)$, and the allocating the best bids could be thought of as a multidimensional knapsack problem [20]. We denote x_i as binary variable which is equal to 1 if the i^{th} bid has won else 0, and we also denote k as the maximum no of bids any agent can win. This is formulated as:

$$\begin{array}{lll} \text{min:} & \sum_{i=1}^{L} x_i \cdot bids_i & \\ \text{s.t.:} & \sum_{i|j \in S_i} x_i \leq 1 & \forall j = 0, 1, 2, \ldots, m-1 \\ & \sum_{i|j^{th}\text{agent bids}} x_i \leq k & \forall j = 0, 1, 2, \ldots, n-1 \\ & x_i \in \{0, 1\} & \forall i = 1, 2, \ldots, L \end{array} \tag{4}$$

The first constraint ensures that every task is allocated at most once as the tasks are indivisible, while the second constraint denotes that every agent can win at most k bids. Since the multi-dimensional knapsack problem is NP-complete, we use branch and bound heuristics to generate feasible solutions [20].

5.3. Winner Scores

We use the VCG mechanism to pay the winners (price scores), which ensures truthful bidding. We will later see that if bidders lie they would suffer relative losses, given this mechanism. The total amount paid to the $agent_i$:

$$\begin{array}{l} \text{(Social Welfare of others if agent } i \text{ was absent) -} \\ \text{(Social Welfare of others if agent } i \text{ was present)} \end{array} \tag{5}$$

where, Social Welfare of all players $= \sum_{i=1}^{L} x_i \cdot bids_i$.

The code snipped provided below with the winning agents provided a pricing score (`agents[winner].score`), updating the battery discharge rate (`agents[winner].life`) and current position (`agents[winner].pos`).

```
winning_bids,s=knapsack(bids,weights,n,m)
print(s)
for i in winning_bids:
        winner = i//m
        new_bids=[0 for x in range(m*(n-1))]
        c1=0; c2=0;
        for j in range(n):
            if j==winner:
                c2=c2+m
                continue
            for k in range(m):
                new_bids[c1]=bids[c2]
                c1=c1+1
                c2=c2+1
        new_weights=assign_weights(len(new_bids),n
          -1,m)
        _,s1=knapsack(new_bids,new_weights,n-1,m)

        agents[winner].score+=s1+bids[i]-s
        agents[winner].life-=dist[i]*discharge_rate
        agents[winner].pos=task[i%m]
```

The pricing score attained after multiple rounds may be used by agents to re-charge batteries at the battery charging stations (Fig. 1). It would be advantageous for individual agents to secure maximum pricing scores for minimal work done; it is the responsibility of a good mechanism to identify these scenarios and award scores fairly. Multiple scenarios involving such multi-agent task auctions are analyzed via simulations in the next section.

In the case of our mechanism design, the agents can bid for tasks using their estimates of bid valuations. An awarded bid results in payment in the form of point scores – this may be redeemed by agents to charge batteries (readers may notice similarity with prices paid to human agents for tasks).

Our approach is decentralized, wherein all the bidders compute their bids in parallel and then pass it to the central system. This is different when compared to a centralized task allocation process, where it would be the central system's responsibility to compute these allocations in a sequential manner.

If $T_1(m)$ denotes the time taken to calculate the bids (optimization problem for a particular agent) and $T_2(L, n, m)$ be the time taken to solve the winner determination problem, then the overall time complexity for the two approaches are :

1. Centralized : $n \times T_1(m) + T_2(L, n, m)$
2. Decentralized : $T_1(m) + T_2(L, n, m)$

Thus, the computational time complexity reduces significantly for higher number of agents.

6. Simulation Results

We study the effects of varying mechanism designs on the Industry 4.0 warehouse demand auctioning process (Fig. 1). We have considered a grid size of 100×100 m. In all our simulations we consider 50 iterations(unless specified). In each iteration there are m task locations generated from a uniform distribution. The initial battery capacity is 100% and the discharge rate of the battery is $0.1/m$. If there is a particular task which a agent will not be able to complete (due to its low battery), it does not participate in the auctioning process. Only after exchanging obtained processing scores for re-charging battery station times can the discharged agent participate in the bidding process. Thus, it is imperative for agents to have sufficient battery capacities to complete tasks.

Every simulated case has two sub-parts: (i) Each agent bids the truthful valuation that is estimated – this is an estimate of the line of sight distance needed to move to the task location and corresponding battery usage (ii) Some agents provide erroneous bids – under/overestimating the tasks so as to gain a foothold on the auction. It is the goal of the mechanism design to study and analyse varied situations that can occur in Industry 4.0 autonomous task auctions.

6.1. Case 1: Homogeneous Agent Capacities – Individual Task Bids.

The first case considered is with homogeneous agent task capacities. The agents bid individually for all tasks, resulting in a total of $m \times n$ bids. Fig. 2 (a)(b) show the final scores of 10 agents after 50 iterations of task bids. When all agents provide accurate bids, the final scores of all agents are uniformly distributed, and the difference between min-median-max agent's cumulative scores are also uniformly distributed. On the other hand, in Fig. 2 (c)(d) when there are 5 agents providing erroneous estimates. We notice that there is a clear separation in agent scores – the agents who provide erroneous estimates have a relatively lower score after 50 iterations. This clearly demonstrates the efficacy of the VCG mechanism – agents have an incentive to provide correct (truthful) estimates in order to obtain better scores.

6.2. Case 2: Heterogeneous Agent Capacities – Individual Task Bids.

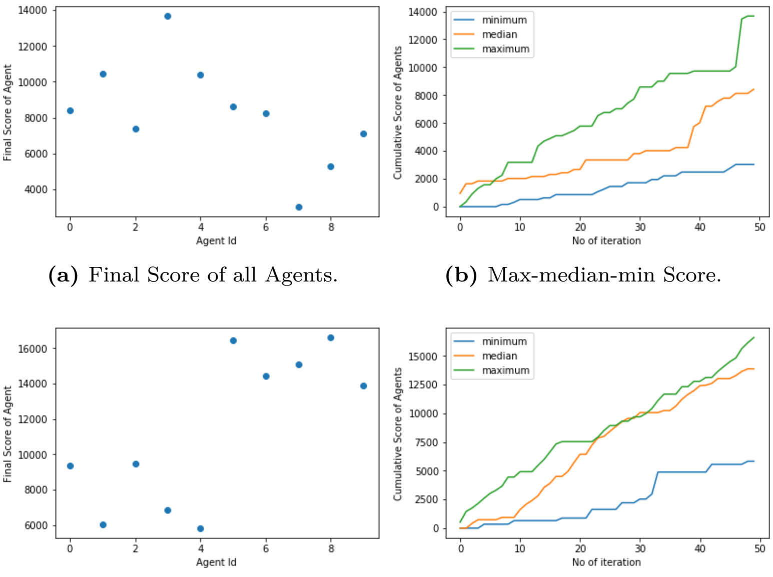
In this case, we introduce heterogeneous tasks – there are two types of tasks, light-weight and heavy-weight. The agents are also divided into types one who can perform only light-weight tasks and the others who can do both. We consider that the first five agents can do only light-weight tasks, and others that can do both. We can see in Fig. 3, the mechanism ensures that the agents who can do both tasks receive superior scores compared to agents who do less tasks.

6.3. Case 3: Homogeneous Agents Bids for Combinatorial Subsets of Items.

We now introduce bidding for a combinatorial subset of tasks. There are at total of m available tasks resulting in 2^{m-1} non empty subsets of tasks. For each agent we randomly choose 5 possible subsets, and make the agent bid for it. The VCG mechanism described in Section 5 determines the winner. As seen in Fig. 4 (a)(b), the total score of all agents provided due to the composite nature of bids rises, when compared to Fig. 2. We also notice similar results when agents provide erroneous estimates in Fig. 4 (c)(d), with overbidding agents provided lower scores.

6.4. Case 4: Erroneous Estimates with Collusion.

There are scenarios where agents collude together to provide erroneous estimates [5]. This is akin to human agents inflating the market with higher rates for lower level of services. We see this in Fig. 5, wherein differences between cases where there is accurate bidding vs. collusion – this implies that the the auctioneer has to pay more scores for similar tasks. By colluding, the agents



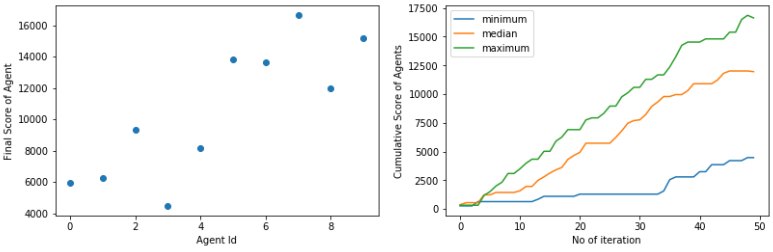
(a) Final Score of all Agents.

(b) Max-median-min Score.

(c) Final score of all agents.

(d) Max-median-min Score.

Figure 2. Homogeneous Agent Bidders with Accurate Estimates (a)(b) and Erroneous Estimates of 5 Agents (c)(d).



(a) Final Score of all Agents.

(b) Max-median-min Score.

Figure 3. Heterogeneous Agents with Accurate Estimates.

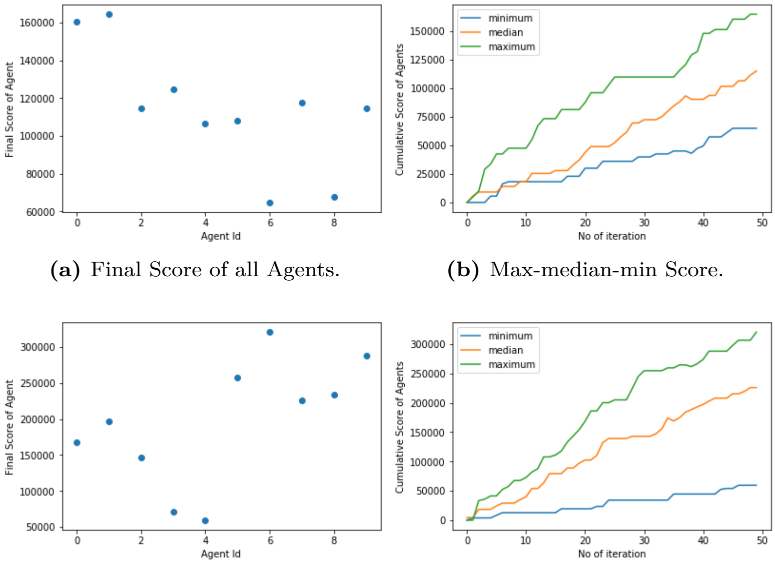
receive higher scores for task, thus enabling them to recharge their batteries

In order to prevent such a situation, we can set an upper-bound on the bidding value, and dismiss bids having a value more than the upper bound. An upper-bound for a particular task could be calculated by taking the maximum distance from all the four corners of the map grid – any accurate estimate of the task should not cross this bound UB. While UB might not be a tight upper-bound, it should be reasonable enough to detect if all of them are colluding. We simulate the effect of three upper bounds in Fig. 6 – UB, $UB \times (1 + x)$, $UB \times (1 - x)$ where x is a random variable taken from a uniform distribution with mean 0 and standard deviation 1. Fig. 6 simulates the failure rate (number of tasks that were not allocated) when 30% agent lie (provide erroneous estimates), 70% agent lie and all agents lie. Fig. 6 demonstrates that an upper-bound of UB or $UB \times (1 + x)$ (optimistic values) may be preferable, as the pessimistic upper bound of $UB \times (1 - x)$, that produces a high failure rate irrespective of the number of colluding agents.

Fig. ?? summarises the findings of the simulated cases under various mechanism design schemes. While the scores may vary based on the task distributions and rewards, the general trend may be applied to multiple use cases involving combinations of multiple agents in an auction setting.

In **summary**, revisiting the principal contributions of the paper:

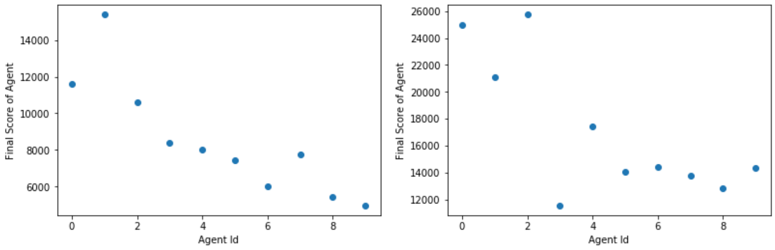
1. Porting the Industry 4.0 autonomous agent task allocation problem into a game theoretic mechanism formulation. The general structure of mechanism auctions have been described in Section 4 with specific instances of Industry 4.0 multi-robot task allocation covered in Section 5.



(a) Final Score of all Agents. **(b)** Max-median-min Score.

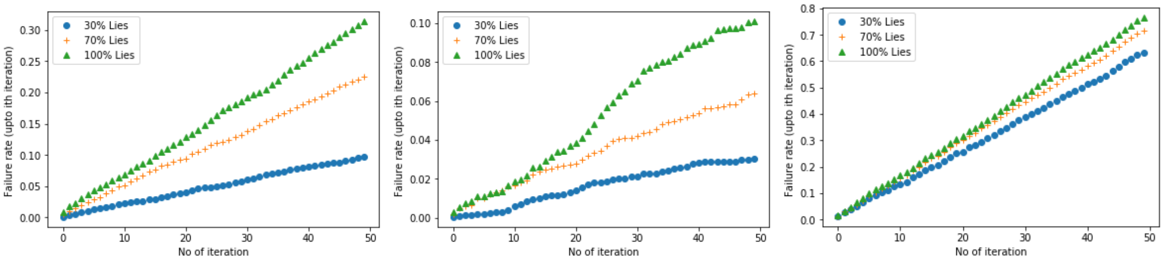
(c) Final Score of all Agents. **(d)** Max-median-min Score.

Figure 4. Composite Bids with Accurate Estimates (a)(b) and Erroneous Estimates of 5 Agents (c)(d).



(a) Accurate Estimates. **(b)** Collusion of Erroneous Estimates.

Figure 5. Final Scores of Agents.



(a) Upper-bound is UB. **(b)** Upper-bound is $UB \times (1 + x)$. **(c)** Upper-bound is $UB \times (1 - x)$.

Figure 6. Cumulative Failure Rates with varied Upper-bounds of Bids.

2. Evaluating the use of VCG auctioning mechanism designs for combinatorial auctions in cooperative tasks. This has been evaluated in the simulations in Section 6.

3. Consider auctions under multiple scenarios with homogeneous/heterogeneous agents, single/combinatorial tasks and truthful/erroneous bids. Figures 2 to 6 demonstrate the efficacy of the technique under a variety of scenarios.

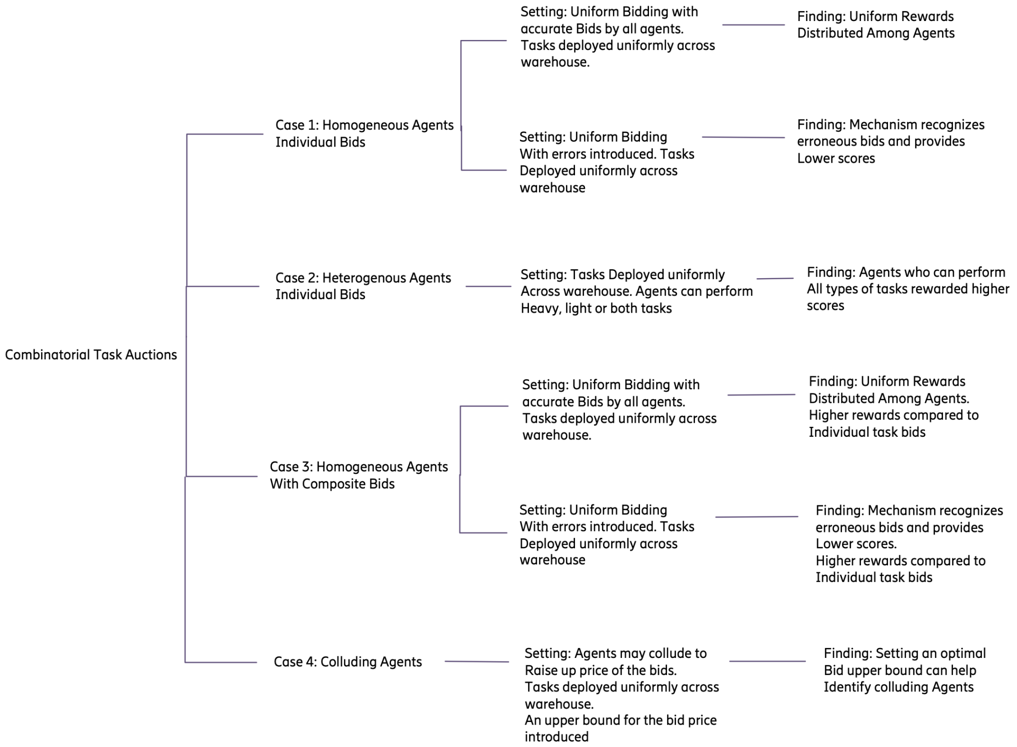

Combinatorial Task Auctions
Case 1: Homogeneous Agents Individual Bids
Setting: Uniform Bidding with accurate Bids by all agents. Tasks deployed uniformly across warehouse.
Finding: Uniform Rewards Distributed Among Agents
Setting: Uniform Bidding With errors introduced. Tasks Deployed uniformly across warehouse
Finding: Mechanism recognizes erroneous bids and provides Lower scores
Case 2: Heterogenous Agents Individual Bids
Setting: Tasks Deployed uniformly Across warehouse. Agents can perform Heavy, light or both tasks
Finding: Agents who can perform All types of tasks rewarded higher scores
Case 3: Homogeneous Agents With Composite Bids
Setting: Uniform Bidding with accurate Bids by all agents. Tasks deployed uniformly across warehouse.
Finding: Uniform Rewards Distributed Among Agents. Higher rewards compared to Individual task bids
Setting: Uniform Bidding With errors introduced. Tasks Deployed uniformly across warehouse
Finding: Mechanism recognizes erroneous bids and provides Lower scores. Higher rewards compared to Individual task bids
Case 4: Colluding Agents
Setting: Agents may collude to Raise up price of the bids. Tasks deployed uniformly across warehouse. An upper bound for the bid price introduced
Finding: Setting an optimal Bid upper bound can help Identify colluding Agents

Figure 7. Findings for various Simulations on Combinatorial Task Auctions.

Such a systematic evaluation of mechanism design theory would prove useful across multiple Industry 4.0 robotic deployments.

7. Conclusions

The advent of Industry 4.0 automation necessitates intelligent, autonomous and collaborative robotic agents. Specially in task allocation amongst collaborative agents, there is a need to move away from centralized task allocation to decentralized multi-agent coordination. This is specifically needed when vendors are outsourced to manage specialized robotic agents. In this paper, we have made use of combinatorial auctioning mechanisms to allocate tasks to autonomous robotic agents. To ensure fairness in the auctioning mechanism despite heterogeneity, erroneous bidding estimates, agent collusion or combinatorial bids, we utilize the *Vickery-Clarkes-Groves* (VCG) mechanism. These aspects are demonstrated over a realistic case study in Industry 4.0 warehouses, where the use of appropriate mechanisms ensures appropriate allocation of warehouse pickup–delivery tasks. Such a model for mechanism design with combinatorial auctions would prove useful across a host of deployments involving multiple autonomous robots.

References

[1] M. Hermann, T. Pentek & B. Otto, "Design Principles for Industrie 4.0 Scenarios", *49th Hawaii Intl. Conf. on System Sciences*, 2016.

[2] S. Greengard, "The Internet of Things", *MIT*, 2015.

[3] M. Jackson, "Mechanism Theory", *Optimization and Operations Research*, Encyclopedia of Life Support Systems, Oxford UK, 2003.

[4] N. Nisan, "Algorithmic Mechanism Design: Through the lens of Multiunit auctions", *Handbook of Game Theory with Economic Applications*, vol. 4, pp. 477–515, 2015.

[5] Y. Narahari, "Game Theory and Mechanism Design", *IISc Press and the World Scientific Publishing*, 2014.

[6] P. Wurman, R. D'Andrea & M. Mountz, "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses", *AAAI Artificial Intelligence Mag.*, vol. 29, no. 1, pp. 9–19, 2008.

[7] Y. Shoham & K. Leyton-Brown, "Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations", *Cambridge University Press*, 2009.

[8] N. Nisan, "Bidding and Allocation in Combinatorial Auctions", *Hebrew University*, 2000.

[9] W. Elmaghraby & P. Keskinocak, "Combinatorial Auctions in Procurement", *Georgia Tech*, 2002.

[10] A. Lam, "Combinatorial Auction-Based Pricing for Multi-tenant Autonomous Vehicle Public Transportation System", *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, 2016.

[11] R. Dash, N. Jennings & D. Parkes, "Computational-Mechanism Design: A Call to Arms", *IEEE Intelligent*

Systems, vol. 3, 2003.

[12] J. Bartholdi & S. Hackman, “Warehouse and Distribution Science”, *The Supply Chain and Logistics Institute*, Georgia Institute of Technology, 2016.

[13] M. Huebscher & J. McCann, “A survey of autonomic computing – degrees, models, and applications”, *ACM Computing Surveys*, vol. 40, no. 3, 2008.

[14] F. Faniyi, P. R. Lewis, R. Bahsoon & X. Yao, “Architecting Self-Aware Software Systems”, *IEEE/IFIP Conf. on Software Architecture*, 2014.

[15] G. Hu, W. Tay & Y. Wen, “Cloud robotics: architecture, challenges and applications”, *IEEE Network*, vol. 26, no. 3, pp. 21–28, 2012.

[16] S. Wang, J. Wan, D. Zhang, D. Li & C. Zhang, “Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination”, *Computer Networks*, vol. 101, 2016.

[17] F. Basile, P. Chiacchio and E. Di Marino, “An auction-based approach for the coordination of vehicles in automated warehouse systems”, *IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, 2017.

[18] A. Kattepur, H. Kumar Rath, A. Simha and A. Mukherjee, “Distributed optimization in multi-agent robotics for industry 4.0 warehouses”, *Proceedings of the 33rd Annual ACM Symposium on Applied Computing* , 2018.

[19] S. Russell & P. Norvig, “Artificial Intelligence: A Modern Approach”, *Pearson*, 2015.

[20] M. Hifi, M. Michrafy & A. Sbihi, “Heuristic algorithms for the multiple-choice multidimensional knapsack problem”, *J. of the Operational Research Society*, vol. 55, no. 12, pp. 1323-1332, 2004.

[21] Marin Lujak, Alberto Fernandez & Eva Onaindia, “Spillover Algorithm: A decentralised coordination approach for multi-robot production planning in open shared factories”, *Robotics and Computer-Integrated Manufacturing*, vol. 70, 2021.

[22] C. Amato, “Decision-Making Under Uncertainty in Multi-Agent and Multi-Robot Systems: Planning and Learning”, *Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018.

[23] H. Hu, X. Jia, K. Liu and B. Sun, “Self-adaptive traffic control model with Behavior Trees and Reinforcement Learning for AGV in Industry 4.0”, *IEEE Transactions on Industrial Informatics*, 2021.