Distributed Optimization in Multi-Agent Robotics for Industry 4.0 Warehouses

Ajay Kattepur, Hemant Kumar Rath, Anantha Simha & Arijit Mukherjee
Embedded Systems and Robotics, TCS Research & Innovation, India
ajay.kattepur,hemant.rath,mukherjee.arijit@tcs.com;ananthasimha@googlemail.com

ABSTRACT

Robotic automation is being increasingly proselytized in the industrial and manufacturing sectors to increase production efficiency. Typically, complex industrial tasks cannot be satisfied by individual robots, rather coordination and information sharing is required. Centralized robotic control and coordination is ill-advised in such settings, due to high failure probabilities, inefficient overheads and lack of scalability. In this paper, we model the interactions among robotic units using *intelligent agent* based interactions. The autonomous behavior of these agents requires task/resource allocation to be performed via distributed algorithms. We use the motivating example of warehouse inventory automation to optimally allocate and distribute delivery tasks among multiple robotic agents. The optimization is decomposed using *primal* and *dual* decomposition techniques to operate in minimal latency, minimal battery usage or maximal utilization scenarios. These techniques may be applied to multiple deployments involving coordination and task allocation between autonomous agents.

CCS CONCEPTS

• **Mathematics of computing** → *Continuous optimization*; • **Applied computing** → *Supply chain management*; • **Computer systems organization** → *Robotics*;

KEYWORDS

Industry 4.0, Multi-agent Systems, Warehouse Automation, Distributed Optimization, Robotics.

ACM Reference Format:
Ajay Kattepur, Hemant Kumar Rath, Anantha Simha & Arijit Mukherjee. 2018. Distributed Optimization in Multi-Agent Robotics for Industry 4.0 Warehouses. In *SAC 2018: SAC 2018: Symposium on Applied Computing , April 9–13, 2018, Pau, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3167132.3167221

1 INTRODUCTION

Integration of robotics, cyber-physical systems and the cloud has come to the forefront with Industry 4.0 requirements [1]. Industry 4.0 proposes the "smart factory" solution, wherein, modular cyber-physical systems coordinate to make decentralized decisions. Some of the key requirements for this include [1]:

1. *Interoperability*: Machines, Internet of Things (IoT) [2] and humans connected and coordinating with each other.
2. *Information Transparency*: Physical systems enhanced with sensor data to create added value.
3. *Technical Assistance*: This involves the use of intelligent devices to aid in informed decision making. Robotic automation may be identified to perform repetitive, unsafe or precise tasks.
4. *Decentralized Decisions*: The ability of such systems to make autonomous decisions; only critical cases will involve human intervention.

Warehouse and factory floor automation [3][4] have been principal areas of interest with respect to these requirements. Automated Guided Vehicles (AGVs) are employed in warehouses to move products from one place to another [5][6]. AGVs follow fixed routes (using wires or markers) that are pre-programmed on them. As they have limited on board computational intelligence, *Networked Robotics* [7] have been proposed, wherein robotic AGVs may link to an Internet based infrastructure to seamlessly exchange data. This data may be autonomously exchanged or co-ordinated via a central control station. This has been extended to the *Cloud Robotics* [8] framework, where robots make use of the cloud to coordinate or offload computational tasks.

In warehouses that may have hundreds of robots on the floor, complex problem domains (scheduling, optimization, planning) require modular and scalable solutions [4]. A number of functional, modular components (agents) may be deployed to solve complex problems. Decomposing large problems allows agents to autonomously solve problem subsets, without relying on a failure prone central controller. When there are inter-dependencies among sub-problems, agent coordination should ensure solutions to the global problems. Multi-agent systems [9][10] are typically employed to model interactions between heterogeneous, distributed and mobile communicating devices. This serves as the right modeling approach when dealing with networked robotic devices that autonomously cooperate to complete missions. Multi-agent deployments are suitable for real-time distributed control in dynamic and distribution prone environments. Important requirements of Industry 4.0 including information transparency and decentralized decision making are satisfied by multi-agent systems.

Due to the decomposition of tasks to agents, typical AGV dependent warehouses can be upgraded with robotic Autonomous Mobile Robot (AMR) deployments [5][6]. AMRs have more powerful on-board computational power that can

SAC 2018, April 9–13, 2018, Pau, France
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5191-1/18/04. . . $15.00
https://doi.org/10.1145/3167132.3167221

understand dynamically changing environments, produce a real-time map of surroundings and efficiently plan paths to destinations. In warehouses where there are dynamic variations in demand and supply of inventory, assigning tasks to robotic AMR agents can be modeled as optimization problems [4]. Typically, such optimizations are solved at a central entity such as Warehouse Management Systems [11], which are neither scalable nor can handle autonomous entities. In this paper, we exploit the agent based architecture to pose the centralized optimization problem to distributed entities. We make use of *primal* and *dual* decomposition techniques to distribute the resource optimization problem among agents. As a realistic use case, we employ the "picker-to-parts" autonomous robots in a warehouse setting, similar to the *Kiva* robot delivery system employed by Amazon[1]. We demonstrate that, through application of this distributed optimization framework, problems such as minimizing overall procurement latency and maximizing utilization of robotic agents may be solved. We further demonstrate that such a distributed optimization is fault tolerant in nature, striving for completion of tasks, despite robotic failures. To the best of our knowledge, this is one of the few papers addressing the problem of warehouse automation with intelligent multi-agent systems, decentralized optimization.

The main contributions of this paper are:

(1) Propose an agent based model for warehouse automation.
(2) Identify the optimization parameters that constrain resources of agents.
(3) Formulate primal and dual decomposition techniques that can be distributed among agents.
(4) Demonstrate optimization of task allocation to maximize resource utilization or minimize latency/energy usage.
(5) Demonstrate fault-tolerance in the task allocation problem due to distributed implementation.

The rest of this paper is organized as follows: State of the art is summarized in Section 2. Section 3 provides a brief overview of multi-agent system modeling. Description of the warehouse scenario with agent based interactions are provided in Section 4. Section 5 provides analysis of distributed optimization involving primal and dual decomposition. Porting such distributed optimization to the warehouse automation scenario is done in Section 6. In Section 7, the results of applying such optimizations to the warehouse automation scenario are presented. Conclusions follow in Section 8.

2 RELATED WORK

The emergence of Industry 4.0 requirements [1] aims to introduce smart, reconfigurable and modular devices in the warehouse and factory environments. This has led to a flurry of research in the warehouse space with the inclusion of autonomous robots [5][6][12] for improving efficient management. A concurrent research area concerns the use of networked and cloud enabled robots [7][8], for coordinating multiple robots deployed on the shop floor. In such deployments, typical path planning algorithms go hand in hand with resource optimization and coordinated goal completion among deployed robots [13]. Warehouse Management Systems (WMS) [11] typically are used to coordinate machines and human interactions in enterprise warehouses.

Handling intelligent, autonomous robots requires an appropriate modeling and analysis framework. Multi-agent systems [9] have been used to encapsulate modular autonomous systems in multiple fields. In the case of robotic systems, multi-agent systems may be used to encapsulate autonomous robots that can coordinate without central control [10]. Coordinating such devices in a decentralized setting is an active area of research [14]. One such widely used protocol is the contract-net protocol [15], that makes use of a bidding mechanism to allocate tasks. In this work, we further append such a protocol with optimization models, which are able to allocate tasks among autonomous agents.

The use of resource optimal protocols in the robotic world has been studied in centralized deployments [16]. Decomposition of optimization algorithms across multiple nodes has been used extensively in the networking literature to design cross-layer resource allocation mechanisms [17]. In [18][19], theoretical models for decomposition techniques in multi-agent systems are provided. An early industrial use case for intelligent agents in manufacturing was provided by the Holonic Manufacturing Systems [20]. The use of genetic algorithms for coordinating multiple agents has also been proposed in literature, as seen in [21].

We apply the distributed optimization among a set of agents to warehouse automation [4]. Deploying autonomous robotic agents in the warehouse setting has received considerable industrial interest, ever since Amazon acquired Kiva robotics in 2012 [22]. Robot automation employed in Amazon's warehouses has drastically reduced procurement times from 60 minutes taken by human participants to around 15 minutes. Automation has increased the inventory capacity per square foot by 50% and reduced Amazon's operating cost by 20% ($225 million/warehouse) [22].

In this work, we take the realistic case of autonomous mobile robots deployed in Industry 4.0 warehouses. The agents coordinate among themselves using distributed optimization techniques to allocate tasks within warehouses. We model division of tasks based on agent utilization, procurement latency, energy depletion and fault tolerance capabilities.

3 MULTI-AGENT SYSTEMS

Classical architectures for control of distributed systems involve a central coordinating node. An example of this might be on the warehouse floor, wherein all robots submit data to a central (cloud) repository for coordination. While a centralized solution is easy to manage (global picture) and might lead to full efficient deployments, there are cases where central deployments are not possible. Typical drawbacks of centralized decision and control include (i) *Inefficiency*: As all the decisions are processed through a central node, latency overheads are substantial (ii) *Constraints*: Multiple safety-critical scenarios (manufacturing, transportation) require real-time decisions that may be repeatedly violated by the remote central node (iii) *Reconfiguration*: Complex

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

deployments typically involve reconfiguration and dynamic changes, which needs autonomous responses.

Intelligent *agents* [10] are autonomous, problem-solving computational entities that can coordinate over a distributed, dynamic environment. Multi-agent systems are a group of agents deployed with specific architecture, coordination and messaging protocols. Each agent senses/actuates based on data computed from its own field of view. Typically, the autonomous agents have incomplete information and limited viewpoints of a global problem. As there is no global control or central daemon available, coordination among agents are needed. Consensus on the global task may be provided by sharing knowledge among agents in a peer-to-peer or hierarchical fashion [15]. Data is also decentralized as agents may sense, store and perform decision updates periodically.

When considering such agent based deployments for robotics, partitioning distributed device clusters as agents may be performed:

- *Physical Distinction*: Each robot is an agent.
- *Encapsulation*: Every agent should be encapsulated with enough information to complete a sub-task.
- *Ownership*: Only a few agents have permissions to modify data.
- *Resources*: Allocating limited resources (computation time, carrying capacity, network bandwidth) to a few robots can correspond to agents.

We further enunciate on dividing complex tasks to agents in warehouses in the next section.

4 CASE STUDY: WAREHOUSE AUTOMATION

In order to study a realistic scenario, wherein, multiple agents autonomously interact with each other, we use the warehouse procurement set-up motivated by Industry 4.0 requirements. The typical “parts-to-picker” model (items on a conveyor belt) is replaced by the “picker-to-parts”(Autonomous Mobile Robot) model. The set-up is described below:

Consider a large warehouse (e.g. Amazon, Dell) where inventory is processed and delivered to customers. Human interaction is kept minimal with a fleet of hundreds of robots handling the order processing. A set of server robots receive the goods/tasks that need to be delivered. The order-processing information is assigned to a set of mobile delivery robots that move along the warehouse floor. Robotic picking arms are located at each shelf, that pick and drop goods on the delivery robots. Inventory management is performed by periodically polling the racks for RFID tagged products. As it is a large warehouse, no centralized control is possible and the robots must handle task sets in an autonomous, collaborative manner [22]. There is a significant variation in the demand of procured items for collaborative task completion.

Such a scenario has motivated by [22][4], where hundreds of autonomous robots are deployed on the warehouse floor. The objective is to move from (time-bound) batch processing, which leads to increasing procurement latency, to a real-time procurement system. The use of multiple decentralized robots also prevents a single point of failure. This can also be used to scale up the number of robots with varying demand.

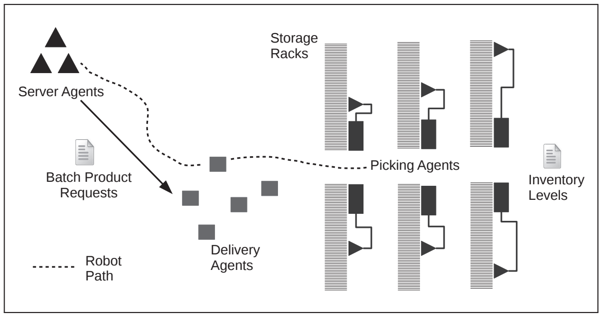

Figure 1: Multiple Agents deployed in Warehouses.

Inventory management is done in conjunction with Radio-Frequency IDentification (RFID) tagging [23]. Procurement and dispatch of items from the warehouse can be monitored with a common database with RFID tag readers located on procurement robots and entry/exit points.

As seen in Fig. 1, we decompose the problem space using specialized robotic agents spawned within a warehouse to automate delivery of products. A set of server agents A_S tabulate the task requirements (product list); a set of mobile delivery agents A_D are employed to subdivide/allocate delivery tasks and a set of picking agents A_P are located close to the shelves to pick and drop required items. Inventory is monitored using RFID tags through periodically polling, such that replacement of inventory is done periodically (A_S, A_P crosschecks). Properties of physical distinction, encapsulation, ownership and resource allocation amongst these agents are set as described in Section 3.

Once a set of products are to be procured by the server agent A_S, the products are subdivided based on dimensions (length, breadth, weight) and location coordinates (rack, row, column). These are then assigned to the delivery agents A_D that have a limited capacity of battery lifetime, load bearing capacity and load bearing dimensions. The delivery robots A_D then approach the picker robots A_P to procure specific items in the provided list. The picker agent A_P detects the proximity of delivery robots and picks/drops the required products. After procuring the items, the server robot A_S updates the inventory list about decrease in stocks of a particular inventory item. Note that this delivery process requires handling of dynamic variations in demand and location of delivery robots. While in the computational setting, resources may include CPU cycles, memory, network and battery capacities, in case of such cyber-physical interactions, allocating tasks with physical and environmental constraints is also needed. Rather than processing each order individually, the server agent may queue up a batch of tasks that need to be completed. These tasks may be sub-divided among multiple delivery and picking agents to optimize procurement of products across multiple warehouse racks. The process of division/allocation of delivery tasks can occur in multiple modes:

- *Utilization of Delivery Agents* – wherein, the tasks as subdivided such that utilization of individual agents are maximized. Each agent makes a choice as to participate or abstain from the procurement process (based on constraints).

- ○ *Procurement Latency* – wherein, the autonomous delivery robots coordinate to procure items within a given latency constraint.
- ○ *Energy Limitation* – wherein, delivery robots accept tasks as long as overall battery degradation of each agent is minimized.

This formulation allows us to move from AGV → AMR, distributing intelligence to the robotic entities. Further elaboration of how optimization formulations may be ported to distributed settings are elaborated in the proceeding sections.

5 OPTIMIZATION TECHNIQUES

In this section, we formulate optimization problems when applied to multi-agent systems for resource allocation. We further provide an overview of primal and dual decomposition techniques [24], that are used for distributed optimization among agents.

5.1 Multi-agent Optimization

For the multi-agent optimization formulation, we consider m agents, where each agent $i \in [1, m]$ has a local convex objective function $f_i(x)$, with $f_i : \mathbb{R}^n \to \mathbb{R}$, and a local convex constraint set X_i (known only by the agent). $x \in \mathbb{R}^n$ represents a global decision vector that the agents are collectively trying to decide on. The decision vector x can either be a resource allocation vector or a global parameter vector to be estimated via agent coordination. The goal of the agents is to cooperatively optimize a global objective function $f(x)$, which is a function of local objectives:

$$f(x) = T\Big(f_1(x), \ldots, f_m(x)\Big) \tag{1}$$

where $T : \mathbb{R}^m \to \mathbb{R}$ is a monotonically increasing convex function. The decision vector x is constrained to lie in a set $x \in \mathcal{C}$, which is a combination of local and global constraints:

$$\mathcal{C} = \Big(\cap_{i=1}^{m} X_i\Big) \cap \mathcal{C}_g \tag{2}$$

where $\mathcal{C}_g$ represents the global constraints. This model leads to the following optimization problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{C} \end{aligned} \tag{3}$$

We further develop this formulation for the warehouse scenario in Section 6.

5.2 Optimization Decomposition

Resource allocation among multiple entities can be formulated as a Knapsack optimization problem [24]. However, multiple agents coordinating in real time require alternatives to centralized coordination [19]. Moreover, in industrial warehouses, where demand of products varies dynamically, a distributed optimization approach is needed. As seen in Fig. 2, the centralized control/optimization model which needs a central authority is moved to a decentralized model, with agents coordinating with each other to solve a global problem. Note that the coordinating agent in Fig. 2 does not represent a central entity, but an additional coordination function that may be performed by an elected agent node. In addition, data transferred between the coordinator and other agents is limited to an update function that controls the global constraints $\mathcal{C}_g$. As this decouples the coordination function from a particular agent (unlike the centralized version), migration or duplication of the role may be performed in case of agent failures. Such a decomposition approach has been employed in [18], which we will now describe in brief.

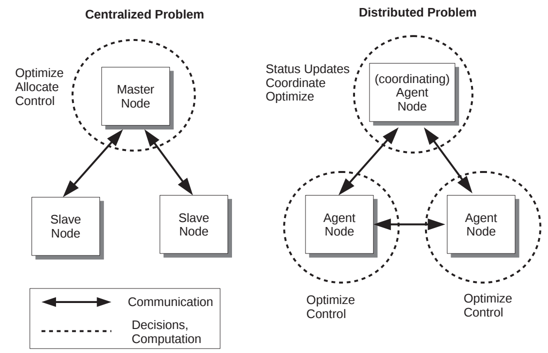

Figure 2: Centralized vs. Distributed Coordination.

Building from eq. 3, we consider a set of m agents allocated a subset of the tasks $a_i,\ i \in [1, m]$, that coordinate in order to optimize resource/task allocation. The centralized version of this optimization problem is:

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} f_i \cdot a_i \\ \text{s.t.} \quad & A_i \cdot a_i \leq x_i, \qquad \forall i \in [1, m] \\ & \sum_{i=1}^{m} F_i \cdot a_i \leq h \end{aligned} \tag{4}$$

where f_i represents the set of resource constraints to be minimized and A_i the constraints inherent in every agent. The individual sub-problems are coupled by the constraints $\sum_{i=1}^{m} F_i \cdot a_i \leq h$, which might represent limits on shared resources. While this is traditionally solved using linear programming/Knapsack formulations, decomposition presents the solution with a master algorithm and linear programming sub-problems solved (possibly in parallel for scalability) at each iteration. The assumption is that the constraints are separable and the local objective functions/constraints may be distributed over the agents. Two possible techniques exist: *Primal* and *Dual* decomposition [24].

5.2.1 Primal Decomposition (Allocation).

In this decomposition technique, the coupling constraint $\sum_{i=1}^{m} F_i \cdot a_i \leq h$ is subdivided into two sets of individually solvable constraints dependent on a shared variable z. The individual optimizations that are solved at each agent are given below (agent m is the co-ordinator):

$$\begin{array}{ll|ll} \min & \mathcal{P}_i = f_i \cdot a_i,\ \forall i \neq m & \min & \mathcal{P}_m = f_m \cdot a_m,\ i = m \\ \text{s.t.} & A_i \cdot a_i \leq x_i & \text{s.t.} & A_m \cdot a_m \leq x_m \\ & F_i \cdot a_i \leq z & & F_m \cdot a_m \leq h - (m-1) \cdot z \end{array} \tag{5}$$

The master optimization that is updated and solved at the coordinating agent is $\min \sum_{i=1}^{m} \mathcal{P}_i$. In order to solve this update equation, the sub-gradient is given as:

$$z = z - \alpha_k \Big(- \sum_{i=1}^{m-1} S_i + S_m \Big) \tag{6}$$

where S_i and S_m are the solutions to the decomposed problems in eq. 5, with step size α_k.

5.2.2 Dual Decomposition (Pricing). The dual decomposition problem makes use of Lagrange multipliers to eliminate the coupling constraint [17], using a decomposition variable λ:

$$\begin{aligned} \min \quad & \big(F_i \lambda + f_i\big) \cdot a_i, \qquad \forall i \in [1, m] \\ \text{s.t.} \quad & A_i \cdot a_i \le x_i \end{aligned} \tag{7}$$

The master update performed at the coordinating agent is given by:

$$\lambda = \Big(\lambda - \alpha_k \Big(- \sum_{i=1}^{m} F_i \cdot S_i + h \Big) \Big)_+ \tag{8}$$

where S_i are the solutions to decomposed problems in eq. 7, $(\cdot)_+$ denotes the non-negative part of a vector and α_k is step size. At each step, the master algorithm sets the prices for the resources. The individual agents each optimize independently, taking into account the expense and income generated by using the resources. The master algorithm adjusts the prices such that the price for an over-used resource is increased and the price for an under-used resource is decreased (non-negative).

Both the primal and dual decomposition techniques make use of the sub-gradient method for distributed optimization. The sub-gradient method is a simple algorithm for minimizing a non-differentiable convex function [24]. The sub-gradient method uses the iteration:

$$z^{k+1} = z^k - \alpha_k \cdot g(k) \tag{9}$$

Here z^k is the k^{th} iterate, $g(k)$ is any sub-gradient of function f at z^k, and $\alpha_k > 0$ is the k^{th} step size. Several step size rules may be applied including, constant step size $\alpha_k = b$ and non-summable diminishing $\alpha_k = a/\sqrt{k}$, where $a > 0$. At each step, the best point found since the first iteration is tracked:

$$f^k_{best} = \min \Big(f^{k-1}_{best}, f(z^k) \Big) \tag{10}$$

The sub-gradient algorithm is guaranteed to converge to within some range of the optimal value [24].

6 OPTIMIZATION IN WAREHOUSE AUTOMATION

In this section, we formulate the optimization problem in a warehouse floor, where (varying) number of products need to be procured and delivered via robotic agents. We demonstrate that this can be formulated as maximizing utilization, minimizing procurement latency or minimizing energy consumption of resources. The optimization problems are further decomposed using primal/dual techniques to be solved by individual agents. The real deployments may be ported onto Robot Operating System (ROS)[2].

Parameter	Description
A_D, A_S, A_P	Delivery, Server, Picking Agent types
i	ID of the Agent with $i \in [1, m]$
a_i	Task allocated to the i^{th} node $\in A_D$
U_i	Device Utilization on node i
D_i	Dimension of products (length × width) on node i
w_i	Weight of products on node i
l_i	Procurement latency for node i
d_i	Distance to be traveled by node i
v_i	Average velocity of node i
μ	Picking agent ($\in A_P$) efficiency
$\mathfrak{D}_i$	Maximum Dimension Capacity of node i
$\mathfrak{W}_i$	Maximum Weight Capacity of node i
ΔC_i	Battery Depletion associated with node i
C_i	Total Battery Capacity of node i
I_i	Current drawn from battery for node i
T_i	Operating Battery Temperature for node i
$\mathfrak{D}$	Total Dimensions of products to be delivered
$\mathfrak{W}$	Total Weight of products to be delivered

Table 1: Optimization Formulation Parameters.

Algorithm 1: Distributed Optimal Task Allocation among Warehouse Delivery Agents.

```
Input: Coordinating Agent with Global Task; Participating Agent
       Status (Number, Location, Capacity, Energy Levels); Task
       Constraints; Initial Update Value;
Profile Global Task and Related Constraints;
if $Allocate \leftarrow Task$ then
    Perform Primal Decomposition (eq. 5);
else
    Perform Dual Decomposition (eq. 7);
Allocate Sub Tasks to Agents with Distributed Optimization;
while Update Non Convergent do
    Agents perform Optimization with Local Constraints;
    Master Algorithm Update (eq. 6 for primal, eq. 8 for dual)
Agents Complete Tasks Optimally;
Agents Update Status to Coordinator;
Output: Global Task completed within Constraints;
```

When an order with a given number of products are to be delivered from the warehouse, the agents (A_D, A_S, A_P) coordinate such that the procurement is satisfied. The coordination may be restricted based on parameters relating to device utilization or procurement latency. Table 1 describes some of the important parameters to be used in our formulation. This can be viewed as extension to the *contract net* protocol [15] consisting of the following steps:

(1) Recognition of goals and whether it can be completed within constraints by an individual or group of agents.
(2) Broadcasting of the task to other agents including objectives and constraints.
(3) Agents bid for the task taking into account tradeoffs for completing tasks and resources to be allocated.
(4) Contracts are awarded to successful bidding agents with communication on job allocation.
(5) A contracted agent may further sub-contract parts of the job to complete the task.

The algorithmic overview of our methodology is presented in Algorithm 1, wherein the traditional tasks $2 - 4$ in the contract net protocol are set into the distributed optimization formulation. Note that the agents we consider typically

[2] http://www.ros.org/

operate as delivery agents A_D as specified in Section 4. Problems such as path planning, stocking and pick–place are out of scope of this formulation. We elaborate on specific situations where the optimization may be formulated based on utilization, latency or energy constraints.

6.1 Maximize Utilization

When there are multiple robots on the warehouse floor ready for delivery, one strategy is to maximize the per-trip utilization of resources. By utilization, we refer to load bearing capacity of the robots rather than utilization of hardware (CPU, memory, network). The optimization may be formulated to maximize the utilization of each node, subject to maximum dimensions/weight that the node can bear. In order to deploy this optimization over agents, we perform primal decomposition as given below:

$$\begin{array}{ll|ll} \max & PB_i = U_i \cdot a_i,\ \forall i \neq m & \max & PA = U_m \cdot a_m,\ i = m \\ \text{s.t.} & D_i \cdot a_i = \mathcal{X} & \text{s.t.} & D_m \cdot a_m = \mathfrak{D} - (m-1) \cdot \mathcal{X} \\ & w_i \cdot a_i = \mathcal{Y} & & w_m \cdot a_m = \mathfrak{W} - (m-1) \cdot \mathcal{Y} \\ & D_i \cdot a_i \leq \mathfrak{D}_i & & D_m \cdot a_m \leq \mathfrak{D}_m \\ & w_i \cdot a_i \leq \mathfrak{W}_i & & w_m \cdot a_m \leq \mathfrak{W}_m \\ & a_i \geq 0 & & a_m \geq 0 \\ & 0 \leq U_i \leq 1 & & 0 \leq U_m \leq 1 \end{array} \tag{11}$$

where, $\mathfrak{D}_i$ and $\mathfrak{W}_i$ are the maximum carrying dimensions of node i having normalized utilization U_i. In order to satisfy the primal decomposition constraints, one of the agents (agent with index $i = m$) performs the pivot optimization satisfying the global constraint. A master update equation is performed at the coordinating agent node (Fig. 2) to update the variables $(\mathcal{X}, \mathcal{Y})$ that are used in the primal problems:

$$\begin{aligned} \mathcal{X} &= \mathcal{X} - \alpha_k \Big(-PA + \sum_i PB_i \Big) \\ \mathcal{Y} &= \mathcal{Y} - \alpha_k \Big(-PA + \sum_i PB_i \Big) \end{aligned} \tag{12}$$

6.2 Minimize Procurement Latency

As collecting products from picking stations and delivery to appropriate stations would involve overheads, formulating cases where latency must be minimized is of interest. The following optimization is proposed where the latency parameter l_i is dependent on a number of factors including distance between picking and delivery robot d_i, average robot speed v_i and picking efficiency μ (average time to pick items dependent on the speed of the picker robot). As this reflects a pricing problem, we perform dual decomposition as follows on all agents:

$$\begin{array}{lll} \min & D_i\lambda_1 + w_i\lambda_2 + l_i \cdot a_i & \forall i \in [1, m] \\ \text{s.t.} & D_i \cdot a_i \leq \mathfrak{D}_i & \\ & w_i \cdot a_i \leq \mathfrak{W}_i & \\ & l_i = \dfrac{d_i}{v_i} + \mu & \\ & a_i \geq 0 & \end{array} \tag{13}$$

where l_i is the procurement latency associated with agent i. The update equations for each of the variables λ_1 and λ_2 is based on the gradient descent method:

$$\begin{aligned} \lambda_1 &= \Big(\lambda_1 - \alpha_k \Big(-\sum_i D_i a_i' + \mathfrak{D} \Big) \Big)_+ \\ \lambda_2 &= \Big(\lambda_2 - \alpha_k \Big(-\sum_i w_i a_i' + \mathfrak{W} \Big) \Big)_+ \end{aligned} \tag{14}$$

where a_i' refers to task-allocation solutions provided from the optimization in eq. 13.

6.3 Minimize Energy Utilization

Minimizing the battery depletion rates of robotic agents may also be a priority. Battery discharge lifetimes have been studied using circuit, electrochemical, stochastic or analytical models [25]. For constant loads, the battery lifetime L can be derived from the battery capacity C and the discharge current I as $L = C/I$. However, modeling the discharge capacity of lithium-ion or lead acid batteries is complicated by the non-linear discharge profiles. Voltage can vary based on the discharge currents, rendering the actual battery capacity lower for high discharge currents (rate capacity effect). Additionally, during periods of low discharge, the battery can recover some of its capacity (recovery effect). We make use of an improved version of Peukert's law [26] that takes into account current drawn and temperature variation during battery discharge:

$$\Delta C = \gamma \cdot \Big(\frac{I}{I_{\text{ref}}} \Big)^{\delta} \cdot \Big(\frac{T_{\text{ref}}}{T} \Big)^{\eta} \tag{15}$$

where ΔC is the battery capacity consumed, I_{ref} and T_{ref} are the reference current and temperature rating of the battery, I and T are the current and temperature observed during battery usage, parameters δ, η are exponents relating non-linear battery depletion rates and constant γ on A-h relates the capacity removed to the non-dimensional discharge current and temperature components.

This is formulated as an optimization problem with the objective to reduce total battery consumption of participating agents using dual-decomposition:

$$\begin{array}{lll} \min & D_i\lambda_1 + w_i\lambda_2 + \Delta C_i \cdot a_i & \forall i \in [1, m] \\ \text{s.t.} & D_i \cdot a_i \leq \mathfrak{D}_i & \\ & w_i \cdot a_i \leq \mathfrak{W}_i & \\ & \Delta C_i < C_i & \\ & a_i \geq 0 & \end{array} \tag{16}$$

The update equations for each of the variables λ_1 and λ_2:

$$\begin{aligned} \lambda_1 &= (\lambda_1 - \alpha_k(-\sum_i D_i a_i' + \mathfrak{D}))_+ \\ \lambda_2 &= (\lambda_2 - \alpha_k(-\sum_i w_i a_i' + \mathfrak{W}))_+ \end{aligned} \tag{17}$$

where a_i' refers to solutions provided from the optimization in eq. 16.

We demonstrate the efficacy of these optimization formulations on deployment simulations in the next section.

Metric	Value
Model	
Physical Dimensions	2 × 0.45 × 0.8 m, 750 kg
Payload	1500 kg
Max. Velocity (straight)	3.6 km/h
Battery Specs.	Lithium-ion, 160 Ah, 48 V
Runtime	8 hours
Sensors	Laser scanners, safety/warning sensors for 360° monitoring

Table 2: KUKA Mobile Platform (KMP) 1500.

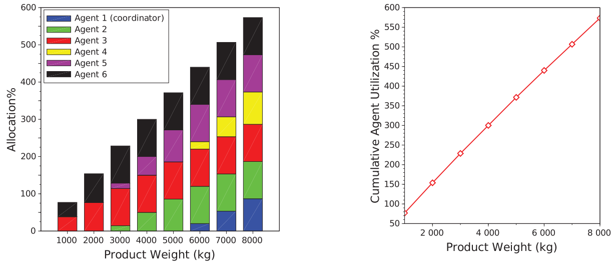

Figure 3: Allocation Ratio to Agents and Utilization Levels.

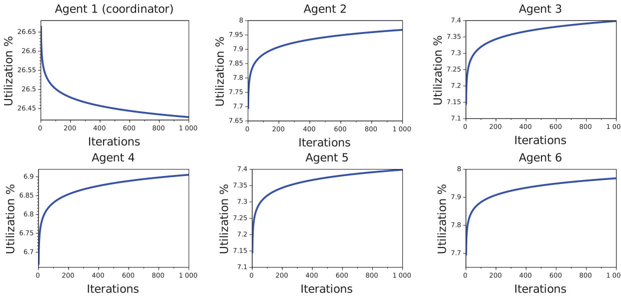

Figure 4: Convergence Analysis with Primal Decomposition.

7 RESULTS

In order to demonstrate the optimization output on real-world robotic agents, we make use of the KUKA mobile platform (KMP) 1500[3] with specifications in Table 2. The KUKA KMP 1500 robot is an autonomous platform that may be deployed on warehouse floors. By the use of safety laser scanners and wheel sensors, the robots are able to perform real time Simultaneous Localization and Mapping.

We now apply the optimization formulations developed in Section 6 to a set of KUKA robots simulated in *Gazebo* on the warehouse floor to collect and deliver products. For our simulations, we simulate *six* delivery agents of homogeneous capacities, that coordinate over complex tasks. A further assumption is that the static locations of the server A_S and picking agents A_P are preset with the optimization performed over picking allocations to delivery agents A_D. The optimizations are solved in Scilab[4] using the *Karmarkar* optimization solver, setting the sub-gradient update from Section 5 as $\alpha_k = 0.1/\sqrt{k}$. The following cases are studied:

- *Utilization*: In order to maximize utilization of the robotic agents, the primal-decomposition techniques provided in eqs. 11, 12 are deployed. Fig. 3 provides an output of this optimization when applied to increasing demand in products (assumption is that a stock keeping unit weighs 1 kg). As seen from Fig. 3, allocation to the delivery agents are gradually increased, such that utilization of each agent is proportionately maximized. The convergence analysis of primal decomposition (eqs. 11, 12) for one instance is seen in Fig. 4. One of the agents serves the coordination function and performs the master update.
- *Latency*: In order to minimize procurement latency of the robotic agents, the dual-decomposition techniques provided in eqs. 13, 14 are deployed. Fig. 5 provides an output of this optimization when applied to increasing demand in products when the delivery agents are randomly placed on the warehouse floor, with velocity as specified in Table 2. As seen from Fig. 5, allocation provided to the distant agents 3 and 5 is minimized so as to maintain acceptable latency of around 3 minutes (differing from allocation in Fig. 3). The convergence analysis of the dual decomposition variable λ (eqs. 14) is seen in Fig. 6, where convergence to a stable neighbourhood is observed within a few iterations.
- *Energy*: In order to minimize battery degradation of the robotic agents, the dual-decomposition techniques provided in eqs. 16, 17 are deployed. Fig. 7 provides an output of this optimization when applied to KUKA robots with battery specifications in Table 2. We notice variations in allocation ratios such that the battery degradations of each agent is minimized.
- *Fault-tolerance*: One of the key advantages of moving to an agent based system is fault tolerance. As there is no central entity replicating/check-pointing, the coordinating agent in Fig. 2 is responsible for varying the primal/dual update factors to take care of faulty agents. As seen in Fig. 8, despite the failure of Agent 3, the optimization reassigns tasks so that other agents participate in higher ratios. Note that we do not perform any check-pointing (partially completed tasks are not reported); rather, the optimization ensures task re-distribution to available agents in the next iteration.

In our analysis, though we have simulated only a limited number of agents to aid in clarity, the procedure may be easily extended to tens or hundreds of agents. The scalable approach provided due to the optimization decomposition in Section 5.2 inherently allows distributed, parallel execution of the optimization problems. Processing capacities available on the autonomous agents may be exploited in industrial settings to aid in distributed cooperative scheduling among the agents. The message passing among agents has overheads, whose analysis has been earmarked for future work.

[3] https://www.kuka.com/en-us/products/mobility/mobile-platforms/kmp-1500

[4] https://www.scilab.org/

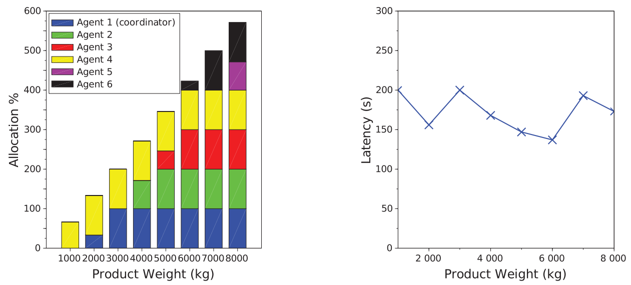

Figure 5: Allocation Ratio to Agents and Latency.

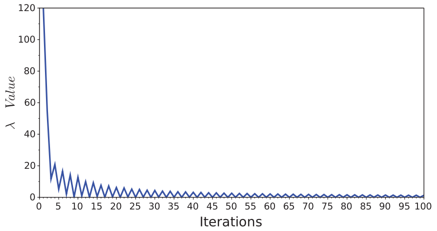

Figure 6: Convergence Analysis with Dual Decomposition.

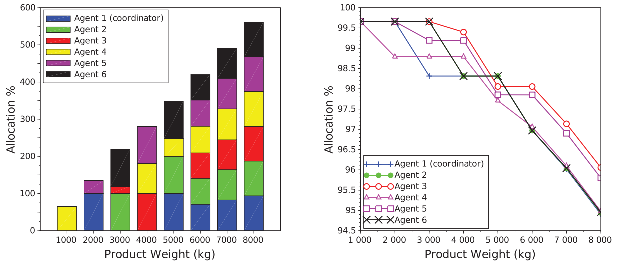

Figure 7: Allocation Ratio to Agents, Battery Degradation.

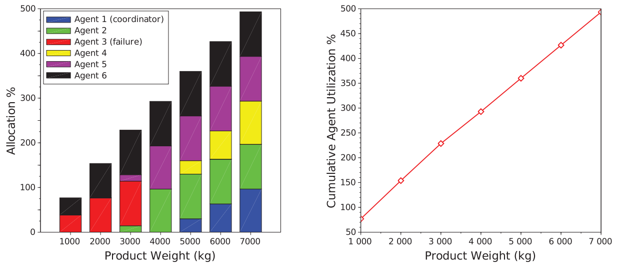

Figure 8: Allocation Ratio with Agent Failure.

8 CONCLUSIONS

Multi-agent intelligent systems are ripe for application into real-world industrial problems. In this paper, we exploit multi-agent system interactions to solve some of the Industry 4.0 automation requirements in warehouses. Through the use of decomposed optimization techniques, the intelligent agents are efficiently enabled utilization, latency and energy resources when traded off with task allocation. Through distributed optimization solvers, these situations are analyzed based on commercially available warehouse robots to demonstrate efficacy in delivery tasks. Such a distributed optimization framework may be extended to other deployments involving intelligent agents.

REFERENCES

[1] M. Hermann, T. Pentek and 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. Hompel and T. Schmidt, "Warehouse Management: Automation and Organization of Warehouse and Order Picking Systems", *Springer-Verlag*, 2012.

[4] J. Bartholdi and S. Hackman, "Warehouse and Distribution Science", *The Supply Chain and Logistics Institute*, Georgia Institute of Technology, 2016.

[5] V. Marik & D. McFarlane, "Industrial Adoption of Agent-Based Technologies", *IEEE Intelligent Manufacturing Control*, 2005.

[6] W. Shen, "Distributed Manufacturing Scheduling using Intelligent Agents", *IEEE Intelligent Systems*, 2002.

[7] R. Luo, K. Su, S. Shen and K. Tsai, "Networked intelligent robots through the Internet: issues and opportunities", *IEEE Magazine*, vol. 91, no. 3, pp. 371–382, 2003.

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

[9] K. Sycara, "Multiagent Systems", *AI Magazine: Intelligent Agents*, vol. 19, No.2, 1998.

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

[11] JDA Warehouse Management System (WMS), https://jda.com/solutions/profitable-omni-channel-retail-solutions/intelligent-fulfillment/warehouse-management, 2017.

[12] W. Shen, Q. Hao, H. Yoon and D. Norrie, "Applications of agent-based systems in intelligent manufacturing: An updated review", *Advanced Engineering Informatics* vol. 20, pp. 415–431, 2006.

[13] B. Brummit, "A Mission Planning System for Multiple Mobile Robotics in Unknown, Unstructured, and Changing Environments", *tech. report CMU-RI-TR-98-42*, Carnegie Mellon University, 1998.

[14] L. Panait and S. Luke, "Cooperative Multi-Agent Learning: The State of the Art", *Autonomous Agents and Multi-Agent Systems*, 11, pp. 387–434, 2005.

[15] T. Sandholm, "An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations," *University of Massachusetts*, 1993.

[16] A. Kattepur, H. Dohare, V. Mushunuri, H. Rath and A. Simha, "Resource Constrained Offloading in Fog Computing", *ACM Middleware Workshops*, 2016.

[17] D. Palomar and M. Chiang, "A Tutorial on Decomposition Methods for Network Utility Maximization", *IEEE J. on Selected Areas in Communications*, vol. 24. no. 8, 2006.

[18] H. Terelius, U. Topcu and R. Murray, "Decentralized Multi-Agent Optimization via Dual Decomposition", *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11245–11251, 2011.

[19] A. Nedic and A. Ozdaglar, "Cooperative distributed multi-agent optimization", *Convex Optimization in Signal Processing and Communications*, Cambridge University Press, 2009.

[20] M. Fletcher, R. Brennan and D. Norrie, "Modeling and reconfiguring intelligent holonic manufacturing systems with Internet-based mobile agents," *J. of Intelligent Manufacturing*, no. 14, vol. 1, pp. 7–23, 2003.

[21] A. Cardon, T. Galinho and J. Vacher "Genetic algorithms using multi-objectives in a multi-agent system," *Robotics and Autonomous Systems*, no. 33, pp. 179–190, 2000.

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

[23] B. Hardgrave, J. Aloysius and S. Goyal, "Does RFID improve inventory accuracy? A preliminary analysis", *Intl. J. of RF Technologies*, vol. 1, no. 1, pp. 44–56, 2009.

[24] S. Boyd and L. Vandenberghe, "Convex Optimization", *Cambridge University Press*, 2004.

[25] R. Rao, S. Vrudhula and D. Rakhmatov,"Battery Modeling for Energy-Aware System Design", *IEEE Computer*, vol. 36, no. 12, 2003.

[26] A. Hausmann and C. Depcik, "Expanding the Peukert equation for battery capacity modeling through inclusion of a temperature dependency", *J. of Power Sources*, vol. 235, pp 148–158, 2013.