Distributed Auction Algorithms for the Assignment Problem with Partial Information[1]

15th ICCRTS
The Evolution of C2
C2 Approaches and Organization

Chulwoo Park[3], Woosun An[3], Krishna R. Pattipati[2,3], and David L. Kleinman[4]

***Abstract*—Task-asset assignment is a fundamental problem paradigm in a wide variety of applications. Typical problem setting involves a single decision maker (DM) who has complete knowledge of the weight (reward, benefit, accuracy) matrix and who can control any of the assets to execute the tasks. Motivated by planning problems arising in distributed organizations, this paper introduces a novel variation of the assignment problem, wherein there are multiple DMs and each DM knows *only a part* of the weight matrix and/or controls a subset of the assets. We extend the auction algorithm to such realistic settings with various partial information structures using a blackboard coordination structure. We show that by communicating the bid, the best and the second best profits among DMs and with a coordinator, the DMs can reconstruct the centralized assignment solution. The auction setup provides a nice analytical framework for formalizing how team members build internal models of other DMs and achieve team cohesiveness over time.**

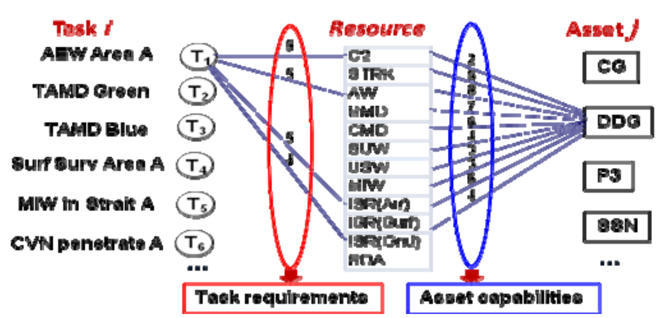
***Index Terms*—Assignment problem, blackboard coordination structure, distributed auction algorithm, maritime operations centers (MOC), partial information structure.**

I. Introduction

A. Motivation

In large-scale organizations, information-processing and decision making functions are distributed among decision makers (DMs) who perform the lower-level tasks themselves and coordinate their information and actions in order to achieve the overall organizational goals. This is because the information processing and decision making capabilities of a (human) DM are limited; distributed decision making facilitates the workload of each DM to remain below their processing capacity thresholds.

This research is motivated by the mission planning and monitoring activities associated with the Navy's maritime operations centers (MOC), in which multiple DMs with partial information and partial control over assets are involved in the development of operational level plans. The MOC emphasizes standardized processes and methods, centralized assessment and guidance, networked distributed planning capabilities, and decentralized execution for assessing, planning and executing missions across a range of military operations [1]. In this vein, we are developing analytical and computational models for multi-level coordinated mission planning and monitoring processes associated with MOCs, so that they can function effectively in highly dynamic, asymmetric, and unpredictable mission environments. Two key problem areas are: 1) realistic modeling of multi-level coordination structures that link tactical, operational and strategic levels of decision making; and 2) collaborative planning with partial information and partial control of assets. In the collaborating planning problem, each DM "owns" a set of assets and is responsible for planning certain tasks. Each task is characterized by a vector of resource requirements, while each asset is characterized by a vector of resource capabilities (see Fig. 1). Multiple assets (from the same DM or multiple DMs) may be required to process a task. The degree of match between the task-resource requirement vector and asset-resource capability vector determines the accuracy of task execution. In addition, the elements of



Legend	Description	Legend	Description
AEW	Airborne early warning	USW	Undersea warfare
TAMD	Theater air/missile defense	BDA	Battle damage assessment
MIW	Mine warfare	ISR	Intelligence, surveillance and reconaissance
C2	Command and control		
STRK	Strike	CVN	Nuclear aircarft carrier
AW	Air warfare	CG	Guided-missile cruiser
BMD	Ballistic missile defense	DDG	Guided-missile destroyer
CMD	Command	P3	Anti-submarine aircraft
SUW	Surface warfare	SSN	Nuclear submarine

Fig. 1. Illustration of task-asset matching problem.

1 This work was supported by the U.S. Office of Naval Research under Grant N00014-09-1-0062.
2 To whom correspondence should be addressed: krishna@engr.uconn.edu.
3 C. Park [STUDENT], W. An [STUDENT], and Prof. K. R. Pattipati are with the Electrical Engineering Department, University of Connecticut, Storrs, CT 06029 USA, (phone: 860-486-2890; fax: 860-486-5585; e-mail: {chp06004; woa05001; krishna}@ engr.uconn.edu).
4 Prof. D. L. Kleinman is with the Department of Information Sciences, Naval Postgraduate School, Monterey, CA 93943, USA (phone: 831-656-7627; e-mail: dlkleinm@nps.edu).

task-resource requirement and asset-resource capability vectors may be affected by the mission environment (e.g., weather), and there may be precedence constraints on tasks. This leads to a stochastic allocation problem of matching the task requirements with the asset capabilities to maximize the task execution accuracy. The distributed assignment problem with *partial information* considered in this paper is a simplified and abstracted version of the collaborative planning problem.

The objective of assignment problem is to match each row (corresponding to a task i) of an $n \times n$ benefit matrix, A

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} = \left[a_{ij} \right]_{n\times n} \quad (1)$$

to a distinct column (corresponding to an asset j) in such a way that the sum of the corresponding matched entries is maximized. In other words, we want to select n elements of the benefit matrix, so that there is exactly one element in each row i and one in each column j and the sum of the corresponding benefits is a maximum. In this paper, we assume that each DM knows *only a part* of the benefit matrix and has control over certain rows (tasks) and certain columns (assets). We consider a blackboard coordination structure[1], which is a coordination mechanism used by team members to 'share their bidding information'. Here, we extend the auction algorithm to the following various information structures: horizontal, vertical, block diagonal and checkerboard. We show that by posting the bid, the best and the second best profits to the blackboard, the DMs can reconstruct the centralized assignment solution. However, different information structures exhibit different computation and coordination delays to arrive at the centralized solution. The auction setup provides a nice analytical framework to study and quantify the impact of information, coordination and organizational structures on speed of collaborative planning, and for formalizing how team members build internal models of other DMs and achieve team cohesiveness over time.

B. *Related Research*

Since Kuhn's pioneering formulation in 1955, the assignment problem has been one of the most popular problems in linear programming and in combinatorial optimization [2]−[19]. The Hungarian method [2] solved the assignment problem in polynomial time and which anticipated the later primal-dual methods [3], [4]. This method was extended to general transportation problems [5]. Jonker and Volgenant [6] developed a primal-dual algorithm for the Hungarian method, which contains new initialization routines and a special implementation of Dijkstra's shortest path method. A scaling technique was employed to obtain a cost-scaling Hungarian algorithm [7], [8].

The auction algorithm is also a primal-dual algorithm having pseudo-polynomial time complexity, but high average efficiency in practice [9], [10]. It is an iterative method for finding the optimal prices and an assignment that maximizes the net benefit, and is therefore the maximum weighted matching assignment [12]−[14]. Forward (i.e., rows bidding for columns) and reverse (i.e., columns bidding for rows), and forward/reverse (i.e., alternate application of forward and reverse) auction algorithms were developed [18]. The average computational complexity of an efficient implementation of the auction algorithm is considerably better than the one for the Hungarian method [10]. Auction algorithms were extended to solve the transportation problem [11], the minimum cost flow problem [12]−[14] and the shortest path problem [15].

In the 1990s, many sequential methods for the assignment problem (especially auction, shortest path-based Hungarian method, and primal simplex algorithms) have been parallelized and computationally tested on parallel machines. In the original distributed implementation, each node is a processor adjusting its own dual prices on the basis of local information communicated by adjacent nodes [12]. It shows finite convergence of a totally asynchronous, distributed version of the algorithm, wherein some processors compute faster than others, some processors communicate faster than others, and there can be arbitrarily large communication delays [12]. These methods, however, do not provide large speedups [12]. A later parallel auction algorithm, where multiple bids are carried out in parallel, and the calculation of each bid is shared by several processors, provided better speedups [17]. As an extension of [17] to the case where only local information (corresponding to vertical or single column information structure considered here) is available due to limited communication capabilities of agents, a distributed auction algorithm with local communication is proposed in the context of networked systems [18], showing that it maximizes the total assignment benefit via a proper selection of the value for ε-complementary slackness (CS) [16], $\varepsilon < 1 / n$. However, the point-to-point communication can result in significant time delays in propagating the global information due to many communication iterations among DMs, namely, multi-hop information propagation [18], [19].

Different from the previous approaches, we propose a distributed auction algorithm assuming that each DM only knows the partial elements of the benefit matrix of its own assets and tasks with various information structures, viz., horizontal, vertical, block diagonal and checkerboard patterns. We seek to address two key questions: 1) what is the information that each DM needs to coordinate with other DMs in order to provide a solution that is the best for the overall team, viz., a centralized solution; and 2) how to create incentive mechanisms among DMs so that the team achieves this solution. A solution to the first problem is that each DM solves his own assignment problem, and transmits local information, viz., bid, best profit and second best profit, to the blackboard (information sharing space). A solution to the second problem, although not discussed here, is to reward the team members on the basis of both team reward and individual rewards.

[1] These algorithms have been extended to point-to-point communication structures as well.

The contributions of this paper are three fold. We consider distributed assignment problems, wherein there are multiple DMs and each DM knows *only a part* of the weight matrix and controls a subset of the assets (columns). The second contribution is that the paper formalizes distributed auction algorithms with a number of information structures: horizontal, vertical, block diagonal and checkerboard (block matrix) structures. The blackboard communication structure is used to coordinate the bids and assignments. The third contribution is that our approach enables us to quantity the impact of various information structures on speed of collaborative planning.

C. Scope and Organization of the Paper

The paper is organized as follows. The information structures considered in this paper are described in section II. In section III, the distributed auction algorithms for the various information structures based on blackboard communication structure are developed. Performance results for the algorithms are given in section IV. The paper concludes with a summary and future research directions in section V.

II. Information and Coordination Structures

A. Information Structures

Here, we consider four (of many) information structures: horizontal, vertical, block diagonal and checkerboard (block matrix) structures. The information structure for the normal assignment problem is termed the centralized information structure.

In the horizontal information structure, each DM knows certain rows of the benefit matrix corresponding to a set of tasks (see Fig. 2 (a)). Here, the number of DMs is m. In the vertical information structure, each DM knows certain columns of the benefit matrix corresponding to a set of assets (see Fig. 2 (b)). Here, the number of DMs is l. In the block diagonal information structure, each DM knows the benefits for his own task-asset pairs while the coordinator knows the benefits for the rest of the task-asset pairs (see Fig. 2 (c)). Note that there is no overlap in rows/columns among DMs (de-confliction among DMs is inherent in the structure) and the number of DMs is m, and the index of the coordinator is (m+1). In the checkerboard information structure, each DM has its own assets and tasks, but with significant overlaps in both rows and columns. In this structure, each DM knows the benefits for his own task-asset pairs, but needs to collaborate horizontally or vertically to share the bidding information (see Fig. 2 (d)). This configuration seeks to approach edge interaction, where each DM also acts as a coordinator, with neighboring DMs. Here, the number of DMs row-wise is m and the number of DMs column-wise is l, so that the total number of DMs in the checkerboard information structure is $m \times l$. Appendix A formalizes the information structures.

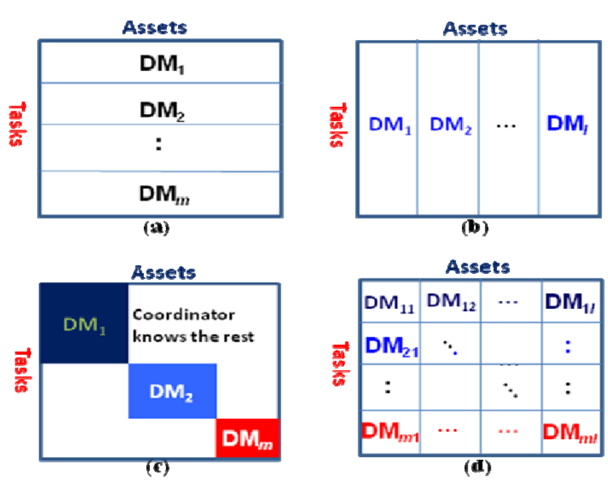


Fig. 2. Information structures. (a) Horizontal. (b) Vertical. (c) Block diagonal. (d) Checkerboard.

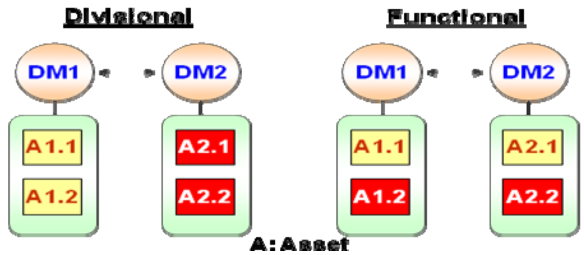


Fig. 3. Information-based organizational models.

The four information structures above correspond to a number of organizational models found in practice, viz., divisional, functional, hybrid and matrix organizations [21]. In a divisional organization (akin to the horizontal information structure), each DM "owns" all the necessary assets (corresponding to columns). Typically, the activities conducted by the members in this organization are restricted to a certain geographic area of responsibility (see Fig. 3). Thus, DMs are responsible for their respective tasks, but this may lead to operational inefficiencies when the task distribution among geographic areas changes. On the other hand, DMs in the functional organization (akin to the vertical information structure) control a single asset type having specialized knowledge of them, and perform a specialized set of tasks (all the tasks in the rows) (see Fig. 3). Thus, the activities of a functional organization may span multiple geographic regions. This structure leads to operational efficiencies within those DMs, but it could also lead to a lack of communication between the functional DMs within an organization, making the organization slow and inflexible. The matrix structure (akin to the checkerboard information) groups DMs by both function and division so that they can take advantage of both structures, i.e., operational responsiveness of a divisional structure and efficiency of a functional structure. DMs must work with each other (both row-wise and column-wise), and collaborate to accomplish their activities; these activities require a great deal of time, communication, effort and skill to collaborate with other DMs. In the hybrid structure (akin to a block diagonal information structure), each DM mainly allocates his own assets to his own tasks, except that the coordinator facilitates supporting-supported relationships among team members. The four information structures provide a range of possible organizational constructs for evaluating the distributed auction algorithm.

B. Parameterization of Information and Organizational Structures

Here, we parameterize the performance of the information structure α by a speedup function $f(\alpha)$

$$f(\alpha) = \frac{t(\alpha)}{t(central)}, \quad \alpha \in \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}, \tag{2}$$

where *t(centralized)* is the centralized auction (termed the normal auction) time with centralized information structure; $t(\alpha)$ is the distributed auction time for each information structure. Here α_1, α_2, α_3 and α_4 denote horizontal, vertical, block diagonal and checkerboard information structures, respectively. The performance (speedup) is the ratio of (sequential, centralized) normal auction time and the (parallel) distributed auction time. Efficiency is the ratio of speedup and the number of DMs. The relative performance measures (computation and coordination delays) are measured via numerical simulations in section IV.

C. Blackboard Coordination Structure

We consider the blackboard coordination structure (how the team of DMs 'shares bidding information'). In the blackboard coordination structure (see Fig. 4), the DMs and a coordinator post their bids, best profit and second best profit information to the blackboard. It is operationalized as follows: 1) each DM bids for the best asset to process each task, as well as the best and the second best profits to the blackboard; 2) if the coordinator can ensure better profit for a task using another asset, it posts a revised bid for the new best asset to the blackboard; 3) each DM may choose to update its bid after observing the bids on the blackboard. Suppose there are M DMs, where $M = m$ ⟨l⟩ for the horizontal ⟨vertical⟩ information structure; $M = (m + 1)$ for the block diagonal information structure, and $M = (m \times l)$ for the checkerboard information structure. Here, m is the number of DMs row-wise and l is the number of DMs column-wise (see Fig. 2). Let $D_{k\{\bullet\bullet\bullet\}}$ = {bid, best profit, second best profit}$_k$ be the bidding data set of DM k, where '•' denotes the transmission status of bidding data attribute. For example, $D_{k\{111\}}$ denotes that the transmitted data set includes the bid, the best profit and the second best profit, respectively. Then, the cumulative bidding data set on the blackboard is the union of the all DM's bidding data given by

$$D_B = \{\bigcup_{k=1}^{M} D_{k\{\bullet\bullet\bullet\}}\}. \tag{3}$$

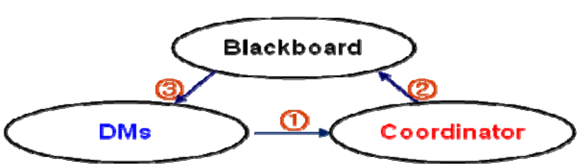


Fig. 4. Blackboard coordination structure.

III. Distributed Auction Algorithms with Various Information Structures

Appendix B formulates the assignment problem and briefly describes the forward and reverse auction algorithms. Appendix C includes the distributed formulations of the assignment problem for the four information structures. Appendix D provides the pseudo code for the distributed auction algorithms. See Table III of Appendix D for variable definitions in this section.

A. Distributed Forward ⟨Reverse⟩ Auction with Horizontal ⟨Vertical⟩ Information Structure

Let ⟨•⟩ denote entities for the reverse auction algorithm. For the blackboard coordination structure, the distributed forward ⟨reverse⟩ auction algorithm has four processing steps: 1) for each assigned task ⟨asset⟩, each DM bids for its current best asset ⟨task⟩; 2) DMs send their bids, viz., $D_{k\{100\}}$ to the blackboard; 3) after scanning the bids on the blackboard, the coordinator invokes the assignment phase of the forward ⟨reverse⟩ auction and posts it to the blackboard; and 4) each DM updates his bid after observing the bids on the blackboard.

The algorithm for the distributed forward ⟨reverse⟩ auction with horizontal ⟨vertical⟩ information structure is listed as 'Algorithm 1' in Appendix D.

As a running illustrative example, we consider a 5 × 5 benefit matrix A (see Fig. 5 (a)) with initial prices p_j, the value for ε-complementary slackness (CS) [16], $\varepsilon = 0.2$, and 5 DMs having a row-wise task $i \in I_k = I_i$ and entire column-wise assets $j \in J_k = J_T$. The forward auction process steps are as follows: 1) Each DM k bids for its tasks $\{i_k\}$ and finds the best asset $j_{i_k} \in J$: bids = {7.2, 2.2, 9.2, 33.2, 14.2}, e.g., bid for task 1 = 92 – 85 + 0.2 = 7.2, bid for task 2 = 97 – 95 + 0.2 = 2.2, etc.; 2) DMs 1, 2, 3 and 4 send their bids to the blackboard to share their bids (see Fig. 5 (b)); 3) Comparing his own bids and his subordinates' bids on the blackboard, the coordinator assigns an asset j to the best task i attaining the maximum bid and posts the bid to the blackboard for each asset j; and 4) DMs 2 and 3, updates their bids after observing the bids on the blackboard (see Fig. 5 (c)).

B. Distributed Forward Auction with Block Diagonal Information Structure

In the block diagonal information structure, the coordinator accesses the blackboard and revises DMs' bids if he can ensure better profits for tasks. The distributed (forward) auction algorithm with block diagonal information structure has five processing steps: 1) the bidding step for each DM k is the same for all the algorithms in this paper, even though they work differently for each information structure; we call this bidding step as 'the common bidding step for each DM' from now on; 2) DMs send their bids, as well as the best and the second best profits, viz., $D_{k\{111\}}$ to the blackboard; 3) if the coordinator can ensure better profit for a task, the coordinator revises the bid and posts it to the blackboard; 4) the coordinator invokes the assignment phase in the same way as in the horizontal information structure case and posts it to the blackboard; and 5) each DM and the coordinator update their bids after observing the bids on the blackboard.

	Asset 1	Asset 2	Asset 3	Asset 4	Asset 5
Task 1	**74**	85	43	29	92
Task 2	95	**59**	57	94	97
Task 3	37	38	**92**	83	58
Task 4	85	52	51	**14**	20
Task 5	38	68	82	38	**8**

(a)

Price, p_j	0	0	0	0	0
Bids	Asset 1	Asset 2	Asset 3	Asset 4	Asset 5
For task 1	~~74.2~~				7.2
For task 2		~~59.2~~			2.2
For task 3			9.2, ~~92.2~~		
For task 4	33.2			~~14.2~~	
For task 5			14.2		~~8.2~~

(b)

Fig. 6. (a) Benefit matrix for block diagonal information structure. (b) Bids on the blackboard updated by the coordinator.

The distributed forward auction algorithm with block diagonal information structure is listed as 'Algorithm 2' in Appendix D. If DMs employ reverse auction algorithm, the coordinator must employ reverse auction algorithm as well. It can alternately use forward and reverse auction steps as well.

As an illustrative example, we consider the same benefit matrix A, initial prices p_j, ε (= 0.2) as in the horizontal information structure case with 5 DMs having a row-wise task $i \in I_k = I_i$ and a column-wise assets $j \in J_k = J_i$, and the coordinator knowing the rest, i.e., $i \in I_6 = I_T \setminus \cup_{k=1}^{5} I_k$, and $j \in J_6 = J_T \setminus \cup_{k=1}^{5} J_k$. Here $\cup$ and $\setminus$ denote set union and set subtraction, respectively. Note that the DM's tasks and assets are diagonal components as highlighted in Fig. 6 (a). The algorithm steps are as follows: 1) The common bidding step for each DM: bids = {74.2, 59.2, 92.2, 14.2, 8.2}, e.g., bid for task 1 = 74 + 0.2 = 74.2, bid for task 2 = 59 + 0.2 = 59.2, etc.; 2) DMs send their bids as well as the best profits of each DM = {74, 59, 92, 14, 8}, to the blackboard. Note that there are no second best profits in this example because all DMs have only one task and one asset; 3) The coordinator finds the best asset for each task $i_k \in T_k(j_k)$, where $T_k(j_k)$ is the set of tasks of a DM k from which an asset j_k receives a bid, the best profits = {92, 97, 83, 85, 82} and the second best profits = {85, 95, 58, 52, 68}. Based on these, the coordinator decides on the best bids = {7.2, 2.2, 9.2, 33.2, 14.2}, e.g., bid for task 1 = 92 – 85 + 0.2 = 7.2, bid for task 2 = 97 – 95 + 0.2 = 2.2, etc., and posts these bids on the blackboard (see as Fig. 6 (b)); 4) The coordinator assigns an asset j to the best task i and posts the bid to the blackboard for each asset j; and 5) DMs 2 and 3, and coordinator update their bids after observing the bids on the blackboard (same as in Fig. 5 (c))

Price, p_j	0	0	0	0	0
	Asset 1	Asset 2	Asset 3	Asset 4	Asset 5
Task 1	74	85	43	29	92
Task 2	95	59	57	94	97
Task 3	37	38	92	83	58
Task 4	85	52	51	14	20
Task 5	38	68	82	38	8

(a)

Price, p_j	0	0	0	0	
Bids	Asset 1	Asset 2	Asset 3	Asset 4	Asset 5
For task 1					7.2
For task 2					2.2
For task 3			9.2		
For task 4	33.2				
For task 5					

(b)

Price, p_j	**33.2**	0	**14.2**	0	**7.2**
Bids	Asset 1	Asset 2	Asset 3	Asset 4	Asset 5
For task 1					7.2
For task 2					~~2.2~~
For task 3			~~9.2~~		
For task 4	33.2				
For task 5			14.2		

(c)

Fig. 5. (a) Benefit matrix. (b) and (c): Bids on the blackboard sent by DMs and posted by the coordinator.

C. Distributed Forward Auction with Checkerboard Information Structure

For the checkerboard information structure, the distributed forward auction algorithm has five processing steps: 1) the common bidding step for each DM (*r.c*); 2) all DMs post their bids, as well as the best and the second best profits, viz., $D_{k\{111\}}$ to the blackboard. Note that multiple DMs may send bids for the same task *i*; 3) after scanning the bids on the blackboard, the coordinator decides on the best bid of a task *i* for assets j_i of all DMs in the same row; 4) the assignment is made by the coordinator after completing the bidding process for every row, and posts it to the blackboard; and 5) the bid update step is performed after observing the bids on the blackboard.

The distributed forward auction algorithm with checkerboard information structure is listed as 'Algorithm 3' in Appendix D.

As an illustrative example, we again consider the same benefit matrix *A*, initial prices p_j, ε (= 0.2) as in the previous examples, except that we have 25 DMs ($DM_{r.c}$) having a row-wise task $i \in I_r = I_i$ and column-wise asset $j \in J_c = J_i$. Note that each cell (highlighted) corresponds to a DM, who is responsible for only one task and one asset (see Fig. 7 (a)). The (forward) auction process with checkerboard information structure has five processing steps: 1) The common bidding step for each DM; 2) All DMs post their bids, as well as the best profits to the blackboard (see Fig. 7 (b)); 3) The coordinator decides on the best bid of task $i_r \in I_r$ for assets $j_{i_k} \in J_T$ in the same row and updates bids on the blackboard (see Fig. 7 (c)); 4) The coordinator assigns an asset *j* to the best task *i* and posts the bid to the blackboard for each asset *j*; 5) DMs 2, 3 and coordinator update their bids after observing the bids on the blackboard (same as in Fig. 5 (c)).

IV. SIMULATION RESULTS

A. Numerical Model Setup

We compare the performance of distributed auction algorithms in terms of computation and coordination delays of the blackboard communication structure for the four information structures. We also provide results for the centralized (normal) auction algorithm that has access to the entire benefit matrix. The computation delay t_{comp} includes bidding time t_{bid}, processing time t_{proc} and assignment time t_{assign}. The coordination delay t_{coord} measures data distribution time t_{dist}, that is, delay in distributing the benefit matrix among DMs, bid-update/posting time t_{post} and communication time t_{comm} between DMs and the coordinator and between DMs and the blackboard communication structure. Two key parameters for analyzing the performance of distributed auction algorithm are the size of benefit matrix for a fixed number of DMs, as well as the number of DMs for a fixed size benefit matrix. This enables us to synthesize the optimal number of DMs for a given benefit matrix size and vice versa.

B. Performance (Delay) Measurement Model

The computation delay in the normal auction is defined as the sum of bidding time, processing time and assignment time, i.e., , i.e., $t_{comp} = t_{bid} + t_{proc} + t_{assign}$. The computation delay of a DM *k* in the distributed auction is defined as the sum of bidding time, processing time and assignment time of a DM *k*, i.e., $(t_{comp})_k = (t_{bid})_k + t_{proc} + t_{assign}$. Then, the overall computation delay for distributed auction is

$$t_{comp} = \max_k\{(t_{bid})_k\} + t_{proc} + t_{assign}. \quad (4)$$

Note that bidding is parallel; thus, the bidding time can be measured by finding the maximum bidding time among DMs and the coordinator (the latter in the case of block diagonal information structure only).

For the coordination delay, the normal auction has posting/bid-update time, but there is no data distribution time and communication delay, i.e., $t_{coord} = t_{post}$. The coordination delay of a DM *k* is defined as the sum of data distribution time, bid-update/posting time, and communication delay, i.e., $(t_{coord})_k = t_{dist} + t_{post} + (t_{comm})_k$. Then, the overall coordination delay for distributed auction is

$$t_{coord} = t_{dist} + t_{post} + \max_k\{(t_{comm})_k\}. \quad (5)$$

Note that the communication between DMs and the blackboard occurs simultaneously; thus, the communication time can be measured by finding the maximum communication time among DMs.

	Asset 1	Asset 2	Asset 3	Asset 4	Asset 5
Task 1	**74**	**85**	**43**	**29**	**92**
Task 2	**95**	**59**	**57**	**94**	**97**
Task 3	**37**	**38**	**92**	**83**	**58**
Task 4	**85**	**52**	**51**	**14**	**20**
Task 5	**38**	**68**	**82**	**38**	**8**

(a)

Price, p_j	0	0	0	0	
Bids/ Best Profits	Asset 1	Asset 2	Asset 3	Asset 4	Asset 5
For task 1	74.2/ 74	85.2/ 85	43.2/ 43	29.2/ 29	92.2/ 92
For task 2	95.2/ 95	59.2/ 59	57.2/ 57	94.2/ 94	97.2/ 97
For task 3	37.2/ 37	38.2/ 38	92.2/ 92	83.2/ 83	58.2/ 58
For task 4	85.2/ 85	52.2/ 52	51.2/ 51	14.2/ 14	20.2/ 20
For task 5	38.2/ 38	68.2/ 68	82.2/ 82	38.2/ 38	8.2/ 8

(b)

Price, p_j	0	0	0	0	
Bids	Asset 1	Asset 2	Asset 3	Asset 4	Asset 5
For task 1	~~74.2~~	~~85.2~~	~~43.2~~	~~29.2~~	7.2
For task 2	~~95.2~~	~~59.2~~	~~57.2~~	~~94.2~~	2.2
For task 3	~~37.2~~	~~38.2~~	9.2	~~83.2~~	~~58.2~~
For task 4	33.3	~~52.2~~	~~51.2~~	~~14.2~~	~~20.2~~
For task 5	~~38.2~~	~~68.2~~	14.2	~~38.2~~	~~8.2~~

(c)

Fig. 7. (a) Benefit matrix for checkerboard information structure. (b) and (c) Bidding data on the blackboard sent by DMs and updated by the coordinator.

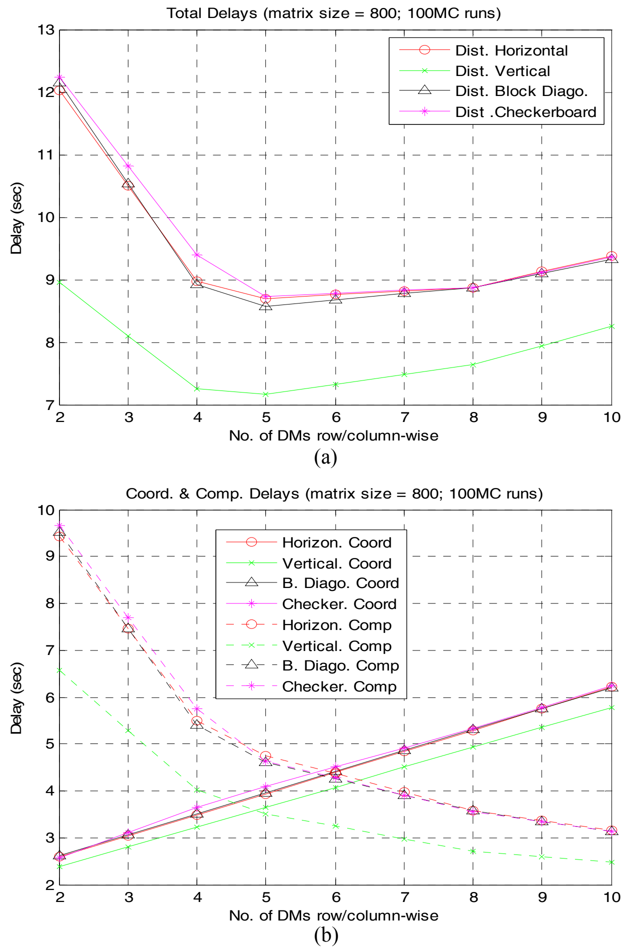


Fig. 8. Performance measures for Case 1. (a) Total delays. (b) Coordination and Computation delays.

Total delay is the sum of the computation delay and coordination delay, i.e., $t_{delay} = t_{comp} + t_{coord}$.

C. Numerical Results

In this section, we discuss the results of applying the distributed auction algorithm using three cases. Our experimental setup is as follows: benefit matrix sizes range from 160 × 160 to 800 × 800, we vary the number of DMs from 2 to 10, and we average delays over 100 Monte Carlo simulation runs on a Quad-Core AMD Opteron™ Processor 2376 (3.29GHz, 15.9GB of RAM) and implemented in MATLAB.

1) Case 1− *benefit matrix size is fixed*: In this case, we fix the benefit matrix size as 800 × 800 and test the algorithms by varying the number of DMs from 2 to 10 (see Fig. 8 and Table I). Fig. 8 (a) shows the total delays for each information structure. Here, all the distributed auction algorithms exhibit less total delay than the normal auction. Note that the total delay for normal auction is 24.71 (sec.). The numbers of DMs having least total delay are 5, 5, 6 and 25 (5 DMs row-wise and 5 DMs column-wise) for the horizontal, vertical, block diagonal and checkerboard information structures, respectively (see Fig. 8 (a)). Fig. 8 (b) displays the coordination and computation delays of distributed auction. The performance of distributed auction becomes worse with increasing number of DMs because the coordination delay increases linearly for more than 2 DMs (4 DMs with checkerboard information structure) and its increase is more than the decrease in computation delay for more than 6 DMs (see Fig. 8 (b)). This implies that the coordination delay including data distribution time, bid-update/posting time and communication time are not significant up to 5, 5, 6 and 25 DMs for the horizontal, vertical, block diagonal and checkerboard information structures, respectively. The speedups with optimal numbers of DMs for distributed auction with horizontal, vertical, block diagonal and checkerboard information structures are 2.7, 3.3, 2.8 and 2.7, respectively (see Table I). The corresponding efficiencies (ratios of speedup and number of DMs) are 0.55, 0.66, 0.46 and 0.11. The distributed forward/reverse auction with horizontal/vertical information structure provides the best performance for 5 DMs, while the distributed auction with block diagonal information structure has the best performance for 6 DMs because it has less coordination delay than other structures.

2) Case 2−*number of DMs is fixed*: Here, we fix the number of DMs as 5 and test the algorithms for various sizes of the benefit matrix up to 800 × 800 (see Fig. 9 and Table II). Fig. 9 (a) shows total delays for each information structure. Beyond 500 × 500 benefit matrix size, all the distributed auction algorithms show gradually less delay than the normal auction because the computation delay of normal auction increases rapidly, while the coordination delay of distributed auction increases slowly (see Fig. 9 (b)). The matrix size having maximum speedup is 800 × 800 for distributed auction algorithms with horizontal, vertical and block diagonal information structures, while it is 640 × 640 for distributed auction algorithm with checkerboard information structure, and the corresponding speedups are 2.7, 2.8, 2.8 and 3.1, respectively (see Table II). The corresponding efficiencies are 0.50, 0.60, 0.24 and 0.05. The horizontal/vertical information structure has better performance than other structures.

TABLE II
THE NUMERICAL RESULTS FOR CASE 1 (800 × 800 MATRIX / UP TO 10 DMS. CHECKER BOARD: UP TO 100).

Auction	# of DMs (Checker-board)	Horizon-tal	Vertical	Block Diagonal	Checker-board
Delay[a] (sec.)	2 (4)	12.03	8.96	12.14	12.24
	4 (16)	8.98	7.25	8.91	9.39
	5 (25)	8.70	7.17	8.58	8.73
	6 (36)	8.75	7.33	8.68	8.78
	8 (64)	8.87	7.63	8.88	8.87
	10 (100)	9.38	8.26	9.32	9.36
Speedup	2 (4)	2.0	2.0	2.0	2.0
	4 (16)	2.7	2.7	2.8	2.6
	5 (25)	2.9	2.9	2.9	2.8
	6 (36)	2.8	2.8	2.9	2.8
	8 (64)	2.8	2.8	2.8	2.8
	10 (100)	2.6	2.6	2.7	2.6

[a] The corresponding delay for normal auction is 24.7 (sec).

TABLE II
THE NUMERICAL RESULTS FOR CASE 2 (5 DMS, CHECKER BOARD: 25 / UP TO 800 × 800 MATRIX).

Auction	Matrix Sizes	Horizon-tal	Vertical	Block Diagonal	Checker board
Delay[a] (sec.)	480	0.04	0.04	0.05	0.06
	640	1.90	1.57	2.04	1.41
	800	8.70	7.17	8.58	8.73
Speedup	480	1.9	1.9	1.5	1.2
	640	2.3	2.8	2.2	3.1
	800	2.7	3.3	2.8	2.4

[a] The corresponding delays for normal auction are {0.09, 0.12, 0.07, 4.55, 24.78}

3) Case 3–both parameters are varied: Here, we vary both *n* and *m* to find optimum value of the matrix size and the number of DMs. Assuming a parallel organizational structure, the benefit matrix sizes were varied from 30 × 30 to 810 × 810, and the optimal number of DMs for each information structure was found. The results are shown in Fig. 10. The optimal number of DMs exhibit staircase behavior with the matrix size. The optimal numbers of DMs with both block diagonal and horizontal structures increase slower than the other two structures.

D. Discussion

Now we point out several practical insights into organizational design with the quantified impacts of our experiments. In our experiments, the vertical information structure (akin to a functional structure) with 5 DMs is the best for an 800 × 800 matrix size showing best efficiency (0.66). The horizontal information structure (akin to a divisional structure) with 5 DMs is efficient for matrix sizes larger than 480 × 480. Experiments suggest that horizontal and vertical structures have better performance than block diagonal and checkerboard information structures; specifically, checkerboard information structure (akin to a matrix structure) shows the worst performance due to significant coordination delays and overlap among DMs. However, this structure may be robust to changes in elements of reward matrix and number of DMs. Block diagonal information structure (akin to a hybrid structure) shows reasonable performance because the coordinator resolves row-wise (divisional) and column wise

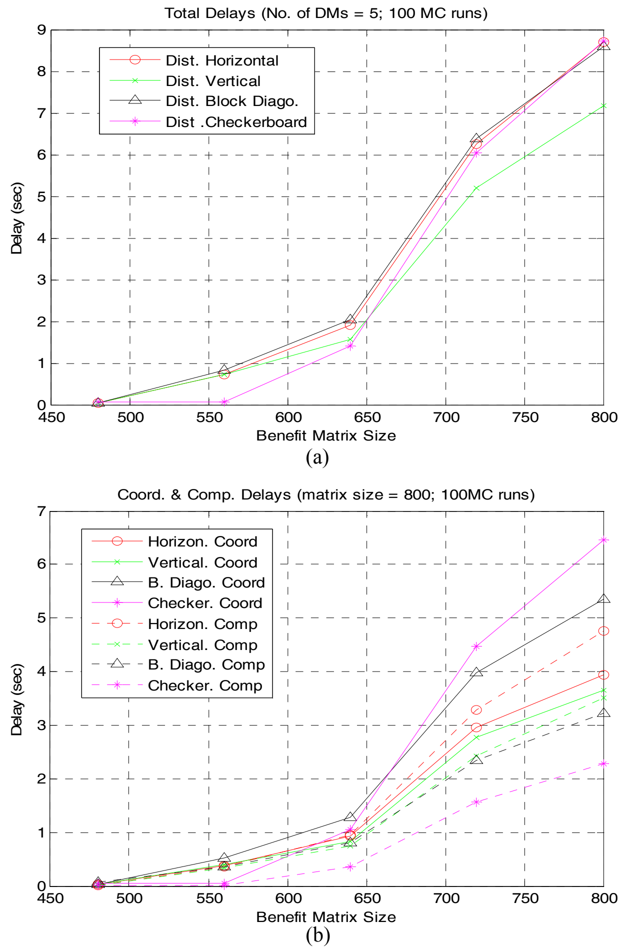


Fig. 9. Performance measures for Case 2. (a) Total delays. (b) Coordination and Computation delays.

(functional) conflicts. Thus, this structure is applicable to either divisional (horizontal) or functional (vertical) structures.

V. CONCLUSION AND FUTURE WORK

In this paper we introduced a novel variation of the assignment problem, wherein there are multiple DMs and each DM knows *only a part* of the weight matrix and controls a subset of the assets. This work was motivated by our ongoing work on analytical and computational models for multi-level coordinated mission planning and monitoring processes associated with MOCs. Here, we extend the auction algorithm to such realistic settings with partial information structures. We show that by posting the bid, the best and the second best profits to the blackboard, the DMs can reconstruct the centralized assignment solution. The performance of various information structures was evaluated by comparing the delays involved in converging to a centralized solution. The distributed auction model in this paper provides a nice analytical framework for formalizing how team members build internal models of other DMs and achieve team cohesiveness over time.

There are numerous extensions of this research. We mention

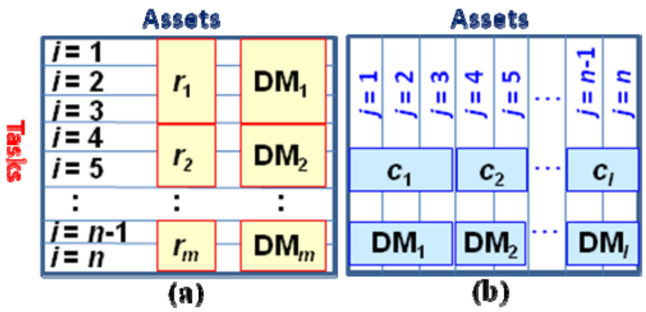


Fig. 11. The set of rows/columns. (a) Horizontal. (b) Vertical.

three here. First, how to develop collaborative planning algorithms with partial information and partial control of assets, when each task is characterized by a vector of resource requirements, and each asset is characterized by a vector of resource capabilities? Second, how to design information and coordination structures to maximize organizational efficiency and be robust to a range of missions? Third, given that DMs in hierarchical organizations operate at multiple time scales, how to synthesize multi-level coordination structures that link tactical, operational and strategic levels of decision making is a major research issue. In addition, our distributed auction model can be applied to network centric enterprises [22] for quantifying the roles of 1) Distributed information structure in generating awareness, 2) Communication structure, e.g., blackboard or point-to-point, for sharing/improving awareness, and 3) Organizational structure for exploiting awareness.

APPENDIX A

SPECIFICATION FOR INFORMATION AND COORDINATION STRUCTURES

For the horizontal information structure, let $I \subseteq \{1, 2, \ldots, n\}$ and $J \subseteq \{1, 2, \ldots, n\}$ be nonempty and non-overlapping subsets of tasks and assets, respectively, that are unassigned, and $\underline{r} = [r_1, r_2, \ldots, r_m]$ denote the vector of rows assigned to the m DMs ($m \leq n$) such that $\sum_{k=1}^{m} r_k = n$. Then, the set of rows and columns of a DM k (see Fig. 11 (a)) is $\{I_k, J_k\}$, where

$$\begin{aligned} I_k &= \left(\sum_{u=1}^{k-1} r_u + 1, \sum_{u=1}^{k} r_u \right), \qquad k = 1, 2, .., m, \\ J_k &= \{1, 2, \cdots, n\}. \end{aligned} \tag{A.1}$$

For the vertical information structure, let $\underline{c} = [c_1, c_2, \ldots, c_l]$ denote the vector of columns assigned to the l DMs ($l \leq n$) such that $\sum_{k=1}^{l} c_k = n$. The set of rows/columns of a DM k (see Fig. 11 (b)), $\{I_k, J_k\}$ are

$$\begin{aligned} I_k &= \{1, 2, \cdots, n\}, \\ J_k &= \left(\sum_{v=1}^{k-1} c_v + 1, \sum_{v=1}^{k} c_v \right), \qquad k = 1, 2, .., l. \end{aligned} \tag{A.2}$$

In the block diagonal information structure, the number of DMs is m, where $m \leq n$. The set of rows and columns of a DM k, $\{I_k, J_k\}$ are

$$\begin{aligned} I_k &= \left(\sum_{u=1}^{k-1} r_u + 1, \sum_{u=1}^{k} r_u \right), \qquad k = 1, 2, .., m, \\ J_k &= \left(\sum_{u=1}^{k-1} c_u + 1, \sum_{u=1}^{k} c_u \right), \qquad k = 1, 2, .., m. \end{aligned} \tag{A.3}$$

The set of rows and columns of the coordinator, denoted as DM (m+1), are

$$\left(I_{(m+1)}, J_{(m+1)} \right) = \left(I_T, J_T \right) \backslash \bigcup_{k=1}^{m} \left(I_k, J_k \right). \tag{A.4}$$

where $I_T = \{1, 2, \ldots, n\}$ and $J_T = \{1, 2, \ldots, n\}$. Here $\cup$ and $\backslash$ denote set union and set subtraction, respectively.

In the checkerboard information structure, the set of rows and columns of a DM ($r.c$), $\{I_r, I_c\}$ are

$$\begin{aligned} I_r &= \left(\sum_{u=1}^{r-1} r_u + 1, \sum_{u=1}^{r} r_u \right), \qquad r = 1, 2, .., m, \\ J_c &= \left(\sum_{v=1}^{c-1} c_v + 1, \sum_{v=1}^{c} c_v \right), \qquad c = 1, 2, .., l. \end{aligned} \tag{A.5}$$

For uniform notation, we number the DMs using a single index as follows: $k = (r - 1)l + c$. This means that a DM ($r.c$), r^{th} row-wise DM and c^{th} column-wise DM, can be represented as a DM k counting his row-wise and column-wise location. For example, when $m = 5$, $l = 5$, $DM_{2.3}$ is numbered as $k = 8$, and the corresponding rows and columns are $I_2 = (r_1 + 1, r_1 + r_2)$ and $J_3 = (c_1 + c_2 + 1, c_1 + c_2 + c_3)$, respectively. For a given DM k,

we can find indices (r, c) as follows: $r = \lceil k / l \rceil$ and $c = k - (r - 1)l$,

where $\lceil \cdot \rceil$ denotes the ceiling function.

APPENDIX B

ASSIGNMENT PROBLEM USING AUCTION ALGORITHM

The objective of the assignment problem is to match n tasks to n' assets to maximize a linear benefit function: there is a benefit matrix, $A = [a_{ij}]$, where a_{ij} denotes the benefit of assigning asset j to task i. When $n = n'$, it is called a symmetric assignment problem; otherwise, it is asymmetric.

B.1. Symmetric Assignment Problem

The symmetric assignment problem is given by

$$\begin{aligned} \text{maxmize} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{n} x_{ij} = 1, \qquad \forall j = 1,2,..,n, \\ & \sum_{j=1}^{n} x_{ij} = 1, \qquad \forall i = 1,2,..,n, \\ & x_{ij} \geq 0, \qquad \forall i,j = 1,2,..,n, \end{aligned} \tag{B.1}$$

where x_{ij} = 1 if the j^{th} asset assigned to the i^{th} task; x_{ij} = 0 otherwise. The first constraint, $\sum_{i=1}^{n} x_{ij} = 1$, requires that every asset is assigned to exactly one task, and the second constraint, $\sum_{j=1}^{n} x_{ij} = 1$, requires that every task is assigned exactly one asset. Therefore, these constraints ensure that the assignment matrix is a permutation such as

$$P = \left[p_{ij}\right]_{n\times n} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.2}$$

Letting π_i and p_j denote the profit of task i and the price of asset j, respectively, the dual of the symmetric assignment problem (B.1) is

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} \pi_i + \sum_{j=1}^{n} p_j \\ \text{subject to} \quad & \pi_i + p_j \geq a_{ij}, \qquad \forall i,j = 1,2,..,n. \end{aligned} \tag{B.3}$$

It is clear from the form of the dual constraints in (B.3) that once the values of $\{p_j\}$ are known, $\sum_{i=1}^{n} \pi_i$ is minimized if we set each π_i to the largest value allowed by the constraints $\pi_i + p_j \geq a_{ij}$, which is

$$\pi_i = \max_j \{a_{ij} - p_j\}. \tag{B.4}$$

This leads to the following equivalent unconstrained dual problem

$$\text{minimize}\left(\sum_{i=1}^{n} \max_j \{a_{ij} - p_j\} + \sum_{j}^{n} p_j\right). \tag{B.5}$$

We consider the complementary slackness (CS) conditions for the assignment problem, viz., if $x_{ij} > 0$, then $\pi_i + p_j = a_{ij}$. Using (B.4) to eliminate π_i, the second CS condition is equivalent to

$$\text{if } x_{ij} > 0, \text{ then } a_{ij} - p_j = \pi_i = \max_h \{a_{ih} - p_h\}. \tag{B.6}$$

Condition (B.6) admits that each asset h carries a price p_h and if it is assigned to a task i, there is a benefit, a_{ih}. The difference $(a_{ih} - p_h)$ is viewed as the profit to task i derived by using asset h. Condition (B.6) then states that each task should be assigned to the asset that would yield the maximal task profit.

B.2. Forward 〈Reverse〉 Auction Algorithm for the Symmetric Assignment Problem

Let 〈•〉 denotes entities for the reverse auction algorithm. In the forward 〈reverse〉 auction algorithm, tasks 〈assets〉 (the bidder) compete for assets 〈tasks〉 (the prize) by bidding for their best assets 〈tasks〉, i.e., each task 〈asset〉 bids for the asset 〈task〉 that provides the best profit 〈benefit〉. The forward 〈reverse〉 auction algorithm for the symmetric assignment problem proceeds iteratively and terminates when a feasible assignment is obtained [16]. At the start of a generic iteration, we have a partial assignment S, which is a set of task-asset pairs $\{(i, j)\}$, and a price 〈profit〉 vector $\underline{p} = [p_1, \ldots, p_n]$ 〈$\underline{\pi} = [\pi_1, \ldots, \pi_n]$〉 satisfying ε-CS:

$$\begin{aligned} & a_{ij} - p_j \geq \max_h \{a_{ih} - p_h\} - \varepsilon \\ & \langle a_{ij} - \pi_i \geq \max_h \{a_{hj} - \pi_h\} - \varepsilon \rangle. \end{aligned} \tag{B.7}$$

As an initial choice, one can use an arbitrary set of prices 〈profits〉 together with the empty assignment, which trivially satisfies ε-CS [16]. The iteration of the forward 〈reverse〉 auction algorithm consists of two phases: bidding and assignment.

1) Bidding Phase: Let $I \subseteq \{1, 2,\ldots, n\}$ and $J \subseteq \{1, 2, \ldots, n\}$ be nonempty subsets of tasks and assets, respectively, that are unassigned. Each task $i \in I$ 〈asset $j \in J$〉 finds an asset $j_i \in J$ 〈task $i_j \in I$〉 which offers maximal profit 〈net benefit〉, that is,

$$j_i = \arg\max_{j\in J}\{a_{ij} - p_j\} \ \langle i_j = \arg\max_{i\in I}\{a_{ij} - \pi_i\}\rangle. \tag{B.8}$$

The corresponding best profit 〈net benefit〉 is

$$v_i = \max_{j\in J}\{a_{ij} - p_j\} \ \langle \beta_j = \max_{i\in I}\{a_{ij} - \pi_i\}\rangle \tag{B.9}$$

and the corresponding second best profit 〈net benefit〉 is

$$w_i = \max_{j\neq j_i, j\in J}\{a_{ij} - p_j\} \ \langle \gamma_j = \max_{i\neq i_j, i\in I}\{a_{ij} - \pi_i\}\rangle. \tag{B.10}$$

Compute the "bid" of task i 〈asset j〉 for asset j_i 〈task i_j〉 given by

$$\begin{aligned} & b_{ij_i} = p_{j_i} + v_i - w_i + \varepsilon = a_{ij_i} - w_i + \varepsilon \\ & \langle b_{i_j j} = \pi_{i_j} + \beta_j - \gamma_j + \varepsilon = a_{i_j j} - \gamma_j + \varepsilon \rangle. \end{aligned} \tag{B.11}$$

This means that a task i 〈asset j〉 bids for an asset j_i 〈task i_j〉 that gives the maximum profit 〈net benefit〉.

2) Assignment Phase: For each asset j ⟨task i⟩, let $T(j)$ ⟨$A(i)$⟩ be the set of tasks ⟨assets⟩ from which an asset j ⟨task i⟩ receives a bid in the bidding phase of the iteration. If $T(j)$ ⟨$A(i)$⟩ is nonempty, increase p_j ⟨π_i⟩ to the highest bid:

$$p_j = \max_{i \in T(j)} b_{ij} \ \langle \pi_i = \max_{j \in A(i)} b_{ij} \rangle. \tag{B.12}$$

Remove from the assignment S any pair (i, j) if an asset j ⟨task i⟩ was assigned to some task i ⟨asset j⟩ under S, and add to S the pair (i_j, j) ⟨(i, j_i)⟩, where i_j ⟨j_i⟩ is a task ⟨asset⟩ in $T(j)$ ⟨$A(i)$⟩ attaining the maximum above. The prize (asset) being auction off goes to the highest bidder (task) and the object is assigned the price of highest bid. This process goes until the assignment matrix become a (non-diagonal) identity matrix shown in (B.2). Note that if two or more tasks find the same asset that is equally maximally beneficial to them, the asset is simply assigned to the lower numbered task.

APPENDIX C

DISTRIBUTED ASSIGNMENT PROBLEM USING DISTRIBUTED AUCTION ALGORITHM

Recall the number of DMs, M is defined as m, l, $(m + 1)$ and $(m \times l)$ for the horizontal, vertical, block diagonal and checkerboard information structures, where m is the number of DMs row-wise and l is the number of DMs column-wise. Let the set of DMs for the rows and columns be defined as $K_i = \{k: i \in I_k\}$ and $K_j = \{k: j \in J_k\}$. Then, the assignment problem is

$$\begin{aligned} &\text{maxmize} && \sum_{k=1}^{M} \sum_{i \in I_k} \sum_{j \in J_k} a_{ij}^{(k)} x_{ij} \\ &\text{subject to} && \sum_{k \in K_j} \sum_{i \in I_k} x_{ij} = 1, \quad \forall j = 1, 2, \ldots, n, \\ &&& \sum_{k \in K_i} \sum_{j \in J_k} x_{ij} = 1, \quad \forall i = 1, 2, \ldots, n, \\ &&& x_{ij} \geq 0, \quad \forall i, j = 1, 2, \ldots, n \end{aligned} \tag{C.1}$$

where $x_{ij} = 1$ if the j^{th} asset assigned to the i^{th} task; $x_{ij} = 0$ otherwise, and $a_{ij}^{(k)}$ denotes that a DM k (k^{th} DM) has knowledge of a_{ij}. The dual function of the linear assignment problem (C.1) [20] is

$$q(\underline{\pi}, \underline{p}) = \max_{x_{ij} \geq 0} \left(\sum_{k=1}^{M} \sum_{i \in I_k} \sum_{j \in J_k} a_{ij}^{(k)} x_{ij} + \sum_{j=1}^{n} p_j (1 - \sum_{k \in K_j} \sum_{i \in I_k} x_{ij}) + \sum_{i=1}^{n} \pi_i (1 - \sum_{k \in K_i} \sum_{j \in J_k} x_{ij}) \right) \tag{C.2a}$$

$$= \max_{x_{ij} \geq 0} \left(\sum_{k=1}^{M} \sum_{i \in I_k} \sum_{j \in J_k} (a_{ij}^{(k)} - \pi_i - p_j) x_{ij} + \sum_{i=1}^{n} \pi_i + \sum_{j=1}^{n} p_j \right). \tag{C.2b}$$

Thus, the dual of the linear assignment problem is

$$\begin{aligned} q^* &= \min_{\underline{\pi}, \underline{p}} \left(\sum_{i=1}^{n} \pi_i + \sum_{j=1}^{n} p_j \right) \\ &\text{subject to} \quad \pi_i + p_j \geq a_{ij}^{(k)}, \\ &\qquad \forall i \in I_k, \ \ \forall j \in J_k, \ \ \forall k = 1, 2, \ldots, M. \end{aligned} \tag{C.3}$$

Again, let ⟨•⟩ denotes entities for the reverse auction algorithm. For the horizontal ⟨or vertical⟩ information structure, the dual of the linear assignment problem for the forward ⟨reverse⟩ auction is

$$\begin{aligned} &q^* = \min_{\underline{p}} \left(\sum_{k=1}^{M} \sum_{i \in I_k} \max_{j} (a_{ij}^{(k)} - p_j) + \sum_{j=1}^{n} p_j \right) \\ &\left\langle q^* = \min_{\underline{\pi}} \left(\sum_{k=1}^{M} \sum_{j \in J_k} \max_{i} (a_{ij}^{(k)} - \pi_i) + \sum_{i=1}^{n} \pi_i \right) \right\rangle. \end{aligned} \tag{C.4}$$

For the block diagonal information structure, the dual of the linear assignment problem for the forward auction formulation is

$$q^* = \min_{\underline{p}} \left(\sum_{k=1}^{M} \left[\sum_{i \in I_k} \max \left\{ \max_{j \in J_k} (a_{ij}^{(k)} - p_j), \max_{j \in J \setminus J_k} (a_{ij}^{(m+1)} - p_j) \right\} \right] + \sum_{j=1}^{n} p_j \right), \tag{C.5}$$

where $(m + 1)$ denotes the coordinator.

For the checkerboard information structure, the dual of the linear assignment problem for the forward auction formulation is

$$q^* = \min_{\underline{p}} \left(\sum_{u=1}^{m} \sum_{i \in I_u} \max_{w} \left[\max_{j \in J_w} (a_{ij}^{(k)} - p_j) \right] + \sum_{w=1}^{l} \sum_{j \in J_w} p_j \right). \tag{C.6}$$

APPENDIX D

DISTRIBUTED AUCTION ALGORITHM

Algorithm 1: Distributed Forward ⟨or Reverse⟩ Auction with Horizontal ⟨or Vertical⟩ Information Structure
(see Table III for variable definitions)

<u>*Step 1:*</u> Each DM k bids for his tasks $\{i_k\}$ ⟨assets $\{j_k\}$⟩: for each task $i_k \in I_k$ ⟨asset $j_k \in J_k$⟩, find the best asset $j_{i_k} \in J_T$ ⟨task $i_{j_k} \in I_T$ ⟩

$j_{i_k} = \arg\max_{j\in A(i_k)}\{a_{i_k j} - p_j\}$ $\langle i_{j_k} = \arg\max_{i\in T(j_k)}\{a_{ij_k} - \pi_i\}\rangle$

Best profit ⟨price⟩:

$v_{i_k} = \max_{j\in A(i_k)}\{a_{i_k j} - p_j\}$ $\langle \beta_{j_k} = \max_{i\in T(j_k)}\{a_{ij_k} - \pi_i\}\rangle$

2nd best profit ⟨price⟩:

$w_{i_k} = \max_{j\in A(i_k);j\neq j_{ik}}\{a_{i_k j} - p_j\}$

$\langle \gamma_{j_k} = \max_{i\in T(j_k);i\neq i_{jk}}\{a_{ij_k} - \pi_i\}\rangle$

Bid: $b_{i_k j_{i_k}} = a_{i_k j_{i_k}} - w_{i_k} + \varepsilon$ $\langle b_{i_{jk} j_k} = a_{i_{jk} j_k} - \gamma_{j_k} + \varepsilon\rangle$

Step 2: DMs send their bids to the blackboard

Set: $\{b_{ij_i}\} = \{\bigcup_{k=1}^{m} b_{i_k j_{i_k}}\}$ $\langle\{b_{i_j j}\} = \{\bigcup_{k=1}^{l} b_{i_{jk} j_k}\}\rangle$

Step 3: The coordinator assigns an asset j ⟨task i⟩ to the best task i ⟨asset j⟩ attaining the maximum below and post the bid to the blackboard

Assign: $p_j = \max_{i\in T(j)} b_{ij}$ $\langle \pi_i = \max_{j\in A(i)} b_{ij}\rangle$

Step 4: Each DM updates his bid after observing the bids on the blackboard.

Algorithm 2: Distributed (Forward) Auction with Block Diagonal Information Structure
(see Table III for variable definitions)

Step 1: Each DM k bids for his tasks $\{i_k\}$: for each task $i_k \in I_k$, find the best asset $(j_k)_{i_k} \in J_k$

$(j_k)_{i_k} = \arg\max_{j_k\in A_k(i_k)}\{a_{i_k j_k} - p_{j_k}\}$

Best profit: $v_{i_k} = \max_{j_k\in A_k(i_k)}\{a_{i_k j_k} - p_{j_k}\}$

2nd best profit: $w_{i_k} = \max_{j_k\in A_k(i_k);j_k\neq(j_k)_{i_k}}\{a_{i_k j_k} - p_{j_k}\}$

Bid: $b_{i_k (j_k)_{i_k}} = a_{i_k (j_k)_{i_k}} - w_{i_k} + \varepsilon$ for asset $(j_k)_{i_k}$

Step 2: DMs send their bids, as well as the best and the second best profits to the blackboard

Set: $\{b_{i_k j_{i_k}}\} = \{\bigcup_{k=1}^{m} b_{i_k (j_k)_{i_k}}\}$;

Set: $\{b_{ij_i}, v_i, w_i\} = \{\bigcup_{k=1}^{m}(b_{i_k j_{i_k}}, v_{i_k}, w_{i_k})\}$

Step 3: The coordinator finds the best asset for each task $i_k \in T_k(j_k)$, updates the best bid, and then posts it to the blackboard

$(j_{(m+1)})_{i_k} = \arg\max_{j_{(m+1)}\in J_T\setminus A_k(i_k)}\{a_{i_k (j_{(m+1)})} - p_{j_{(m+1)}}\}$

Best profit: $v_{i_k}^{(m+1)} = \max_{j_{(m+1)}\in J_T\setminus A_k(i_k)}\{a_{i_k (j_{(m+1)})} - p_{j_{(m+1)}}\}$

2nd best profit: $w_{i_k}^{(m+1)} = \max_{\substack{j_{(m+1)}\in J_T\setminus A_k(i_k);\\ j\neq(j_{(m+1)})_{i_k}}}\{a_{i_k (j_{(m+1)})} - p_{j_{(m+1)}}\}$

If $(v_{i_k} < v_{i_k}^{(m+1)})$

Bid: $b_{i_k (j_{(m+1)})_{i_k}} = a_{i_k (j_{(m+1)})_{i_k}} - \max(v_{i_k}, w_{i_k}^{(m+1)}) + \varepsilon$

Update $b_{i_k j_{i_k}} = b_{i_k (j_{(m+1)})_{i_k}}$

else

If $(w_{i_k} - v_{i_k}^{(m+1)} < 0)$

Bid: $b_{i_k (j_k)_{i_k}} = b_{i_k (j_k)_{i_k}} + w_{i_k} - v_{i_k}^{(m+1)}$

Update $b_{i_k j_{i_k}} = b_{i_k (j_k)_{i_k}}$; *end*

end

Step 4: The coordinator assigns an asset j to the best task i attaining the maximum below and post the bid to the blackboard

Assign: $p_j = \max_{i\in T(j)} b_{ij}$

Step 5: Each DM and the coordinator update their bids after observing the bids on the blackboard.

Algorithm 3: Distributed (Forward) Auction with Checkerboard Information Structure
(see Table III for variable definitions)

Step 1: Each DM $(r.c)$ bids for its tasks $\{i_r\}$ to find the best asset $(j_c)_{i_r} \in J_c$

$(j_c)_{i_r} = \arg\max_{j_c\in A_c(i_r)}\{a_{i_r j_c} - p_{j_c}\}$

Best profit: $v_{i_{(r.c)}} = \max_{j_c\in A_c(i_r)}\{a_{i_r j_c} - p_{j_c}\}$

2nd best profit: $w_{i_{(r.c)}} = \max_{j_c\in A_c(i_r);j_c\neq j_{i_r}}\{a_{i_r j_c} - p_{j_c}\}$

Bid: $b_{i_r (j_c)_{i_r}} = a_{i_r (j_c)_{i_r}} - w_{i_{(r.c)}} + \varepsilon$ for asset $(j_c)_{i_r}$

Step 2: All DMs post their bids, as well as the best and the second best profits to the blackboard

Set: $\{b_{i_r j_{i_r}}, v_{i_r}, w_{i_r}\} = \{\bigcup_{c=1}^{l}(b_{i_r (j_c)_{i_r}}, v_{i_{(r.c)}}, w_{i_{(r.c)}})\}$

Set: $\{b_{ij_i}\} = \{\bigcup_{r=1}^{m}\{b_{i_r j_{i_r}}, v_{i_r}, w_{i_r}\}\}$

Step 3: The coordinator decides the best bid of task $i_r \in I_r$ for assets $j_{i_k} \in J_T$ in the same row

$j_{i_r} = (j_{c*})_{i_r} = \arg\max_{j\in A(i_r)}\{v_{i_{(r.c)}}\}$

Best profit: $v_{i_r} = \max_{j\in A(i_r)}\{v_{i_{(r.c)}}\}$

If $(\max_{j\in A(i_r);j\neq j_{i_r}}\{v_{i_{(r.c)}}\} > w_{i_{(r.c*)}})$

2nd best profit: $w_{i_r} = \max_{j\in A(i_r);j\neq j_{i_r}}\{v_{i_{(r.c)}}\}$

Bid: $b_{i_r (j_{c*})_{i_r}} = b_{i_r (j_{c*})_{i_r}} + w_{i_{(r.c*)}} - w_{i_r}$ for asset j_{i_r}

Update: $b_{i_r j_{i_r}} = b_{i_r (j_{c*})_{i_r}}$

else

2nd best profit: $w_{i_r} = w_{i_{(r.c*)}}$

Update: $b_{i_r j_{i_r}} = b_{i_r (j_{c*})_{i_r}}$

end if

Step 4: The coordinator assigns an asset j to the best task i attaining the maximum below and post the bid to the blackboard

Assign: $p_j = \max_{i\in T(j)} b_{ij}$

TABLE IV
THE CARDINALITY OF PROFITS OF THE DISTRIBUTED AUCTION WITH BLOCK DIAGONAL INFORMATION STRUCTURE.

Case	Best[a]	2nd best[b]	Coordinator
1	v_{i_k}	w_{i_k}	-
2	v_{i_k}	$v_{i_k}^{(m+1)}$	Update bid with new 2nd best and post it to the blackboard
3	$v_{i_k}^{(m+1)}$	v_{i_k}	Update bid with new best/2nd best and post it to the blackboard
4	$v_{i_k}^{(m+1)}$	$w_{i_k}^{(m+1)}$	Update bid with new best/2nd best and post it to the blackboard

[a] v_{i_k} and $v_{i_k}^{(m+1)}$ are the best profits of a DM k and a coordinator
[b] w_{i_k} and $w_{i_k}^{(m+1)}$ are the 2nd best profits of a DM k and a coordinator

Step 5: Each DM updates his bid after observing the bids on the blackboard.

REFERENCES

[1] Commander United State Fleet Forces Command, "Maritime Headquarters with Maritime Operations Center Concept of Operations," Rev. A, Initial draft vol. 2, Jan. 2008.
[2] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics*, vol. 2, pp. 83–97, Mar. 1955.
[3] H. W. Kuhn, "Variants of the Hungarian method for assignment problems," *Operations Research*, vol. 3, pp. 253–258, Dec. 1956.
[4] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society of Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, Mar. 1957.
[5] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.
[6] R. Jonker, and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing* vol. 38, pp. 325–340, 1987.
[7] H. N. Gabow, J. Comp, "Scaling algorithms for network problems," *System Sci.*, vol. 31, no. 2, pp.148–168, 1985.
[8] H. Gabow and R. Tarjan, "Almost-optimum speed-ups of algorithms for bipartite matching and related problems," in *Proc. 20th Annu. ACM Symp. Theory of computing*, Chicago, IL, May 02–04, 1988, pp. 514–527.
[9] D. P. Bertsekas, "A Distributed algorithm for the assignment problem", Lab. for Information and Decision Systems, Working Paper, M.I.T., Cambridge, MA, March 1979.
[10] D. P. Bertsekas, "A New algorithm for the assignment problem," *Math Programming*, vol. 21, pp. 152–171, 1981.
[11] D. P. Bertsekas and D. A. Castanon, "The auction algorithm for transportation problems," *Analysis of Operation Research*, vol. 20, pp. 67–96, 1989.
[12] D. P. Bertsekas and J. Eckstein, "Distributed asynchronous relaxation methods for linear network flows problems," in *Proc. I F AC '87*, Munich, Germany, Jul. 1987.
[13] D. P. Bertsekas and D. A. Castanon, "The auction algorithm for minimum cost network flow problem," Lab. For Information and Decision Systems, Technical Report LIDS-P-1925, M.I.T., Cambridge, MA, 1989.
[14] D. P. Bertsekas and D. A. Castanon, "A generic auction algorithm for the minimum cost network flow problem," Lab. For Information and Decision Systems, M.I.T., Cambridge, MA, 1991.
[15] D. P. Bertsekas, "An Auction algorithm for shortest paths," *SIAM Journal on Optimization*, vol. 1, pp. 425–447, 1991.
[16] D. P. Bertsekas, *Network optimization: continuous and discrete models*. Belmont, MA: Athena Scientific, 1998, 251–336.
[17] D. P. Bertsekas and D. A. Castanon, "Parallel synchronous and asynchronous implementations of the auction algorithm," *Parallel Computing*, vol. 17, pp. 707–732, 1991.
[18] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas, "A distributed auction algorithm for the assignment problem," in *Proc. 47th IEEE Conf. Decision and Control*, Cancun, Mexico, Dec. 9−11, 2008.
[19] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*, Englewood Cliffs, N.J.: Prentice Hall, 1989, pp. 27–88.

TABLE III
VARIABLES USED IN THE DISTRIBUTED AUCTION ALGORITHMS.

Variables	Definitions	Used in[a]
ε	The value for ε-complementary slackness (CS) [16]	1−3
m / l	Number of DMs row-wise / column-wise	1−3
I/T (I_T / J_T)	Nonempty subsets of tasks / assets, that are unassigned (assigned to DMs)	1−3
I_k / J_k	Sets of tasks / assets, that are assigned to a DM k	1, 2
$i_k(i_{j_k})$ / $j_k(j_{i_k})$, and $j_{(m+1)}$	A task / an asset of a DM k (bids for an asset / a task of a DM k), and of a coordinator	1, 2
$(j_k)_{i_k}$ $((j_{(m+1)})_{i_k})$	An asset j of a DM k (a coordinator) bids to a task i of DM k	2
i_r / j_c	A task i / an asset j of a row-wise/column wise DM r / c	3
$j_{i_r}((j_c)_{i_r})$	An asset j (of a column-wise DM c) bids for (the best) task i of a row-wise DM r	3
$\{\pi_i\}$, $\{p_j\}$	The dual prices: the profit of a task i, the price of an asset j	1−3
$p_{j_k}(p_{i_{(m+1)}})$	The price of an asset j of a DM k (a coordinator)	2
p_{j_c}	The price of an asset j of a column-wise DM c	3
v_i / w_i	The best/2nd best profits of a task i (in the blackboard)	2
v_{i_k} / w_{i_k} $(\beta_{j_k}$ / $\gamma_{j_k})$	The best/2nd best profits (prices) of a task i (an asset j) of a DM k	1, 2
$v_{i_k}^{(m+1)}$ / $w_{i_k}^{(m+1)}$	The best/2nd best profits of a task i of a coordinator	2
$v_{i_{(r.c)}}$ / $w_{i_{(r.c)}}$ $(v_{i_r}$ / $w_{i_r})$	The best/2nd best profits of a task i of a DM ($r.c$) (a row-wise DM r)	3
$w_{i_{(r.c^*)}}$	The 2nd best profit of the column-wise DM having the best profit for a task i of a row-wise DM r	3
$T(j)$ / $A(i)$ $(T(j_k)$ / $A(i_k))$	The set of tasks / assets from which an asset j / a task i (of a DM k) receives a bid	1−3 (1)
$T_k(j_k)/A_k(i_k)$ $(A_c(i_r))$	The set of assets/tasks of a DM k (a column-wise DM c) from which a task i/ an asset j of a DM k (a row-wise DM r) receives a bid	2 (3)
$A(i_r)$	The set of assets of from which a task i of a row-wise DM r receives a bid	3
b_{ij_i} / $b_{i_j j}$ $(b_{i_k j_{i_k}}$ / $b_{i_{j_k} j_k})$	The bid from a task i / asset j (of a DM k) to the best asset j / task i	1 (1, 2)
$b_{i_k (j_k)_{i_k}}$ $(b_{i_k (j_{(m+1)})_{i_k}})$	The bid from a task i of a DM k to the best asset j of a DM k (a coordinator)	2
$b_{i_r (j_c)_{i_r}}$ $(b_{i_r (j_{c^*})_{i_r}})$	The bid from a task i of a row-wise DM r to (the best) asset j of a column-wise DM c	3
$b_{i_r j_{i_r}}$	The bid from a task i of a row-wise DM r to the best asset j	3
$a_{i_k j}$ / a_{ij_k} $(a_{i_k j_{i_k}}$ / $a_{i_{j_k} j_k})$	Benefit when a task i / asset j of a DM k bids for (the best) asset j / (the best) task i	1
$a_{i_k j_k}(a_{i_k (j_k)_{i_k}})$, $a_{i_k j_{(m+1)}}$	Benefit when a task i of a DM k bids for (the best) asset j of a DM k, a coordinator	2
$a_{i_k j_{(m+1)}}$	Benefit when task i of a DM k bids for an asset j of a coordinator	2
$a_{i_r j_c}(a_{i_r (j_c)_{i_r}})$	Benefit when task i of a row-wise DM r bids for (the best) asset j of a column-wise DM c	3

[a] 1, 2 and 3 denote Algorithms 1, 2 and 3, respectively

[20] D. P. Bertsekas, *Nonlinear programming*, Belmont, M.A.: Athena Scientific, 2003, pp.365–377.

[21] V. K. Narayanan and R. Nath, *Organization theory: A strategic approach*, Homewood, I.L., IRWIN, 1993, pp.303–349.
[22] D. S. Alberts, J. J. Garstka, and F. P. Stein, *Network centric warfare: developing and leveraging information superiority*, CCRP Publication Series, 1999, pp. 25–51.