

# Effects-Based Design of Robust Organizations

Student Paper Submission  
(Suggested Track: Modeling and Simulation)

Sui Ruan<sup>1</sup>

E-mail: sruan@engr.uconn.edu

Candra Meirina<sup>1</sup>

E-mail: meirina@engr.uconn.edu

Haiying Tu<sup>1</sup>

E-mail: haiying.tu@uconn.edu

Feili Yu<sup>1</sup>

E-mail: yu02001@engr.uconn.edu

Krishna R. Pattipati<sup>1,2</sup>

<sup>1</sup>University of Connecticut, Dept. of Electrical and Computer Engineering

371 Fairfield Road, Unit 1157

Storrs, CT 06269-1157

Fax: 860-486-5585

Phone: 860-486-2890

E-mail: krishna@engr.uconn.edu

\*This work was supported by the Office of Naval Research under contract #N00014-00-1-0101.

<sup>1</sup>Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT 06269-1157, USA.

<sup>2</sup>Correspondence: krishna@engr.uconn.edu

# Effects-Based Design of Robust Organizations

Sui Ruan<sup>a</sup>, Candra Meirina<sup>a</sup>, Haiying Tu<sup>a</sup>, Feili Yu<sup>a</sup>, and Krishna R. Pattipati<sup>a</sup>

**Abstract**—Effects-based design of robust organizations seeks to synthesize an organizational structure and its strategies (resource allocation, task scheduling and coordination), to achieve the desired effect(s) in a dynamically changing mission environment. In this paper, we model the dynamic system associated with the mission environment (e.g., environment faced by a joint task force with a military objective [8] or the competitive environment confronted by a consumer electronic company striving to increase its market share) as a finite-state Markov Decision Process (MDP)[1][2][7]. Using this model, we determine a near-optimal action strategy that specifies which action to take in each state of the MDP model by Monte Carlo control method. The action strategy determines a range of possible missions the organization may face. The range of missions and platform utilization measures, in turn, are used to synthesize a robust organizational structure.

**Keywords:** Organizational Design, Markov Decision Processes, Reinforcement Learning, and Monte Carlo Control Method.

## I. INTRODUCTION

### A. Motivation

Market environments and battlespaces are dynamic and uncertain. Organizations seeking to achieve the desired effects in such environments are confronted with the following: 1) parts of the environment can not be controlled directly; 2) various exogenous events may impact the state of the environment; 3) the interactions between potential organization's actions and the dynamics of the environment may result in consequences that can not be predicted a priori with certainty. Consequently, organizations need to plan for potential contingencies, be flexible and adaptable. In other words, they need to be robust learning organizations. In this paper, we apply concepts from Markov Decision Processes (MDP), reinforcement learning, Monte Carlo control method, and mixed integer optimization, as in

[1]-[5] and [16], to prescribe an optimal decision strategy and the concomitant organization structure to achieve desired effects in an uncertain mission environment.

Over the years, research in organizational decision-making has demonstrated that a strong functional dependency exists between the specific structure of the mission environment and the resulting optimal organizational structure and its decision strategy. That is, the optimality of an organization design depends on the mission environment and the organizational constraints [15]. Such organizations, termed congruent, are expected to perform well. Incongruence to environmental dynamics hinders the organization's ability to achieve the desired effects, results in a higher cost, or likely to encounter adverse effects.

Agility is arguably one of the most important characteristics of successful information age organizations [8]. Agile organizations do not just happen. They are the results of an organizational structure, command and control approach, concepts of operation, supporting systems, and personnel that have a synergistic mix of the right characteristics. Agile organization is a synergistic combination of the following six attributes: Robustness, Resilience, Responsiveness, Flexibility, Innovation and Adaptation [8]. Robustness is the ability to retain a level of effectiveness across a range of missions that span the spectrum of conflicts, operating environments, and/or circumstances. An organization designed to cope with dynamic and stochastic mission environments is said to be robust in the sense that it can cope with a range of contingencies.

### B. Related Research

Over the past several years, mathematical and computational models of organizations have attracted a great deal of interest in various fields of scientific research [15]. Many research efforts have focused on the problem of quantifying the structural

<sup>a</sup>Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT 06269-1157, USA. E-mail: [sruan, meirina, yu02001, krishna]@engr.uconn.edu, haiying.tu@uconn.edu. This work is supported by the Office of Naval Research under contract No. 00014-00-1-0101.

(mis)match between organizations and their tasks. When modeling a complex mission and designing the corresponding organization, the variety of mission dimensions (e.g., functional, geographical, terrain), together with the required depth of model granularity, determine the complexity of the design process [14]. Model-driven synthesis of optimized organizations for a specific (deterministic) mission environment have been widely studied by the operations research community [12][13].

On the other hand, planning and machine learning in uncertain and dynamic environments have been largely studied by the control and artificial intelligence community. In this vein, reinforcement learning, a computational approach for understanding and automating goal-directed learning and decision-making [1]-[4], has become a dominant approach for dealing with stochastic planning problems.

### C. Scope and Organization of Paper

In section II, we provide an overview of our organization methodology based on MDP, Monte Carlo control method, reinforcement learning, and mixed integer optimization techniques. In section III, we formulate the dynamic environment as a finite state MDP, and formalize the objectives of the organizational design problem. In section IV, we apply Monte Carlo control method to prescribe near-optimal action strategies for the MDP model. Section V provides an integer programming formulation for the congruent organization design problem. Simulation results are presented in Section VI. Finally, the paper concludes with a summary and future research directions in section VII.

## II. ORGANIZATIONAL DESIGN METHODOLOGY

The methodology applied in this paper is shown in Fig. 1. In this study, we are considering a dynamic military mission environment, and the design of a robust organization for accomplishing the mission. The mission is represented via tasks and platforms (assets). There are three type of tasks considered in this paper, viz., mission tasks, time critical tasks, and “mosquitoes” (trivial tasks). Mission tasks are those that must be executed, are known in advance, and typically have precedence restrictions in the form of a dependency graph [9][10]. Time critical tasks are those whose

occurrence is uncertain, are time sensitive, and may have substantial impact on the mission. Mosquitoes are the relatively trivial tasks whose occurrence is highly uncertain, but have insignificant impact on the mission. Each task is characterized by a set of resource requirements [12]. A platform (asset) represents a physical entity of an organization that provides resource capabilities used to process the tasks. Each platform has a specific resource capability. Successful task execution requires that the task’s resource requirements are met by the overall resource capabilities of the platforms allocated to that task.

We model the dynamic environment as a finite state Markov Decision Process (MDP) [3][4][7]. The MDP is characterized by its state and action sets and by the the transition probability matrix. Given any state and action, pair  $(s, a)$ , the probability that the next state is  $s'$  is given by

$$\varphi_{ss'}^a = Pr\{s_{t+1} = s' | s_t = s, a_t = a\} \quad (1)$$

We assume that the structure of MDP model of the environment is known, but the environmental parameters ( e.g., the transition probabilities ) are unknown. Since the model parameters are unknown, they need to be estimated (“learned”) [3]; the estimated parameters enables us to find a near-optimal action strategy that optimizes an objective function.

Monte Carlo control methods [3][6] are employed to obtain a near-optimal action strategy. Monte Carlo methods require only experience-based sample sequences of states, actions, and rewards from an on-line or simulated interaction with the environment. Learning from on-line experience is striking because it requires no prior knowledge of the environment’s dynamics, and yet can still attain optimal behavior asymptotically. The near-optimal strategy from Monte Carlo control method enables us to compute the platform (resource) utilization measures of the near-optimal strategy. These measures are used to synthesize an organization that implements the near-optimal action strategy.

## III. MDP FORMULATION AND ORGANIZATION DESIGN OBJECTIVES

The dynamic stochastic mission environment consists of:

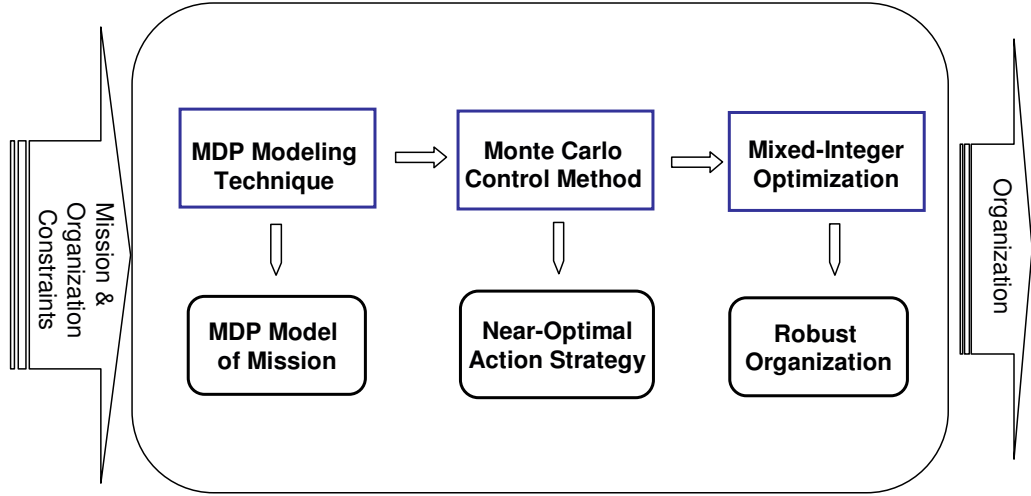


Fig. 1. Robust Organization Design Process

- **Effect set:**  $M = \{m_1, m_2, \dots, m_n\}$ , the desired effects, with some serving as the end goals. There can be dependencies among the effects.
- **Exogenous event set:**  $E = \{e_1, e_2, \dots, e_f\}$ , which represents uncontrollable random events in the environment.
- **Action set:**  $A = \{a_1, a_2, \dots, a_k\}$ , which denotes controllable influences to achieve the desired effects, and minimize the adverse effects of exogenous events. For each action  $a$ , there is a cost  $c(a)$  associated with it.

When applying this modeling approach to a C2 mission environment [9][10], the mission tasks correspond to the effects, and the time critical tasks and “mosquito” tasks correspond to exogenous events. The actions correspond to the asset allocation used to achieve the desired effects and suppress the effects of exogenous events.

Organization is a team of Decision Makers (DM),  $ORG = \{DM_1, DM_2, \dots, DM_D\}$ , i.e., the personnel or automated systems that supervise the actions in the system. DMs coordinate their information, resources, and activities in order to achieve their common goal in the dynamic and uncertain mission environment [12]. The constraint imposed on each DM is in the form of its limited resource handling capability. In this paper, we experiment with the resource handling capability of each DM as a workload threshold. The DM’s workload is

a combination of internal workload and external workload [12][13].

The dynamic mission environment is modeled as a finite state Markov Decision Process (MDP). The main sub-elements of the MDP follow the mission characteristics cited earlier. They are as follows:

- **States:**  $S = \{s_1, s_2, \dots, s_z\}$ 
  - Each state represents the status of effects and exogenous events:  $s_i = (M_i, E_i)$  where  $M_i \subseteq M$  denotes the achieved effects and  $E_i \subseteq E$  denotes the existing, but unmitigated, exogenous events.
  - The initial state,  $s_b = (\emptyset, \emptyset)$ , where no effect has been achieved and no exogenous event has appeared yet; the terminal states,  $S_e \subseteq S$ , represent the attainment of desired end effects.
  - The state has Markov property; that is the next state depends solely on its previous state and action, as in Eq. (1).
- **Actions:**  $A = \{a_1, a_2, \dots, a_k\}$ .
- **Reward mechanism:**
  - Reward : When an end state is reached, a fixed reward  $r(s_e) > 0$  is accrued by the organization.
  - Penalty : When undesirable end effects are reached, a penalty  $r(s_h) < 0$  is imposed on the organization.

- Action cost: Whenever an action  $a_i \in A$  is pursued, a cost  $c(a_i)$  is incurred.

The objectives of the design problem are to:

- 1) Find an optimal closed-loop action strategy, viz., a mapping from state to action, to maximize the expected net reward.
- 2) Design an organization, i.e., the allocation of platforms to decision makers, such that the overall workload of the organization is minimized, subject to constraints on each DM's workload capability.

The MDP problem posed above is a stochastic optimal control problem [1][2]. Dynamic programming (DP) [2] is widely used to characterize the optimal solution. However, it suffers from “the curse of dimensionality”, meaning that its computational requirements grow exponentially with the number of state variables. In addition, DP needs the complete parameters of the model. However, in real world applications, the transition probabilities are rarely known. Instead, we could gain knowledge of the system from sampled data, and train the strategy. We propose Monte Carlo control methods [3][6] to achieve the objective of obtaining a near-optimal strategy for the MDP model without a complete knowledge of the MDP parameters (i.e., when the transition probabilities are not known). Monte Carlo method requires only that the MDP model generate a set of sample state transitions, but not the complete probability distributions of all possible state transitions. The Monte Carlo method learns these quantities from an on-line simulation. This, in turn, enables us to obtain an optimal strategy [3] without prior knowledge of the environment's dynamics. An agent-based simulator, such as the one in [11], can provide a vehicle to operationalize various mission episodes utilized in the Monte Carlo method.

Platform (resource) utilization measures of the near-optimal strategy are employed to design an organizations that is congruent with the mission environment. In this paper, the expected total workload of the organization is minimized. An organization tuned to such a strategy is *robust* in the sense that it covers a range of missions that are most likely to materialize in the dynamic environment.

#### IV. MONTE CARLO METHOD FOR NEAR-OPTIMAL ACTION STRATEGY

Monte Carlo methods are ways of solving the reinforcement learning problem based on averaging sample returns [3][6]. In Monte Carlo methods, we assume that the experience is divided into episodes, and that all episodes eventually terminate regardless of which actions are selected. It is only upon the completion of an episode that value estimates (estimation of the expected reward of state, and state-action pair) and policies (strategy, the mapping from states to actions) are changed. Monte Carlo methods are thus incremental in an episode-by-episode sense [3][6], but not in a step-by-step sense. The term “Monte Carlo” is often used more broadly for any estimation method whose operation involves a significant random component. Here, we use it specifically for methods based on averaging complete returns.

Monte Carlo methods could be applied to evaluate state-value of any given strategy, viz., the mapping of states to actions, by averaging the returns from sample episodes. Starting from Monte Carlo policy evaluation, it is natural to alternate between evaluation and improvement on an episode-by episode basis. After each episode, the observed returns are used for policy evaluation, and then the policy is improved at all the states visited in the episode [3]. In addition, Monte Carlo methods are particularly attractive when one requires the value of only a subset of the states. One can generate many sample episodes starting from these states, averaging returns only from these states, and ignoring all others.

The complete Monte Carlo control algorithm, which combines Exploring Starts (ES) and  $\epsilon$ -greedy, is given in Figure 2. The ES technique starts an episode by randomly choosing the initial state. The concept of  $\epsilon$ -greedy means that most of the time an action that has maximal estimated action value is chosen, but with probability  $\epsilon$  an action is selected at random. That is, all non-greedy actions are given the minimal probability of selection,  $\frac{\epsilon}{|A(s)|}$ , and the remaining bulk of the probability,  $1 - \epsilon + \frac{\epsilon}{|A(s)|}$ , is given to the greedy action [3], where  $|A(s)|$  is the cardinality of action set,  $A(s)$  in state  $s$ . This enables the Monte Carlo control method to get out of local minima.

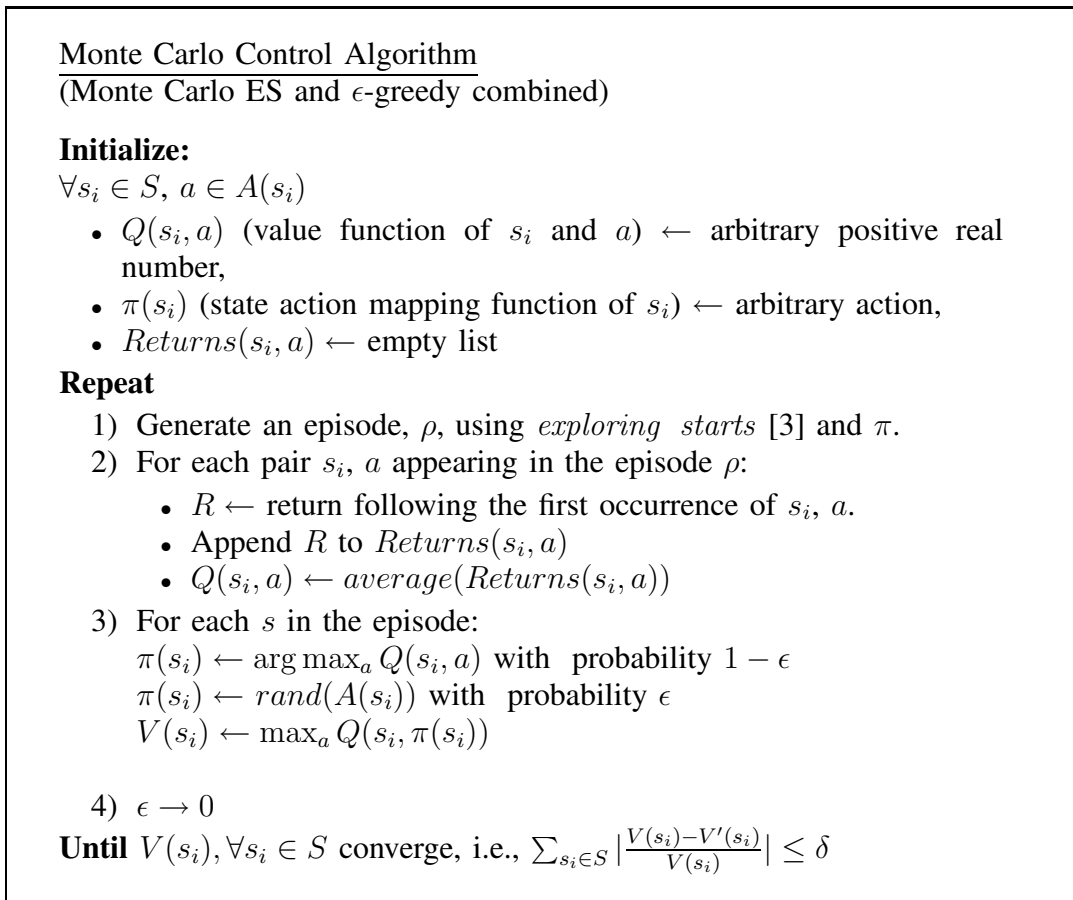


Fig. 2. Monte Carlo control method with Exploring Starts and  $\epsilon$ -greedy

Note that, applying a simple Monte Carlo control method, based on only exploring starts [3], is not satisfactory, i.e. some states are rarely visited, while some other states form cycles. An  $\epsilon$ -greedy [3] method is often combined with the ES method. A fixed  $\epsilon$  is not efficient. A large  $\epsilon$  yields slow convergence and tends to oscillate; with a small  $\epsilon$ , it is difficult to eliminate the possible state cycles. Typically, one chooses  $\epsilon_i \rightarrow 0$  such that  $\sum_i \epsilon_i \rightarrow \infty$ . For example, with  $\epsilon_i = \frac{\epsilon}{N+i}$ ,  $N = (2, 6)$ , the results are satisfactory.

In the Monte Carlo control algorithm, all the returns for each state-action pair are accumulated and averaged, irrespective of which policy was in force and when they were observed. Convergence of the Monte Carlo control method is assured because the changes to the action-value function decrease over time. A formal proof has not yet appeared, and it is one of most fundamental open questions in reinforcement learning [3][4].

## V. ORGANIZATION DESIGN

In a  $C^2$  environment [10], each mission task can be modeled as a desired effect, while time critical tasks and “mosquito” tasks can be viewed as exogenous events. A Task is an activity that entails the use of relevant resources (provided by organization’s platforms) and is carried out by an individual DM or a group of DMs to accomplish the mission objectives [12]. Each task  $T_i (i = 1, \dots, N)$  has resource requirements specified by the row vector  $[R_{i1}, R_{i2}, \dots, R_{iL}]$ . A platform is a physical asset of an organization that provides resource capabilities and is used to process tasks. Each platform belongs to a unique platform class. For each platform class  $P_m (m = 1, \dots, K)$ , we define its resource capability vector  $[r_{m1}, \dots, r_{mL}]$ , where  $r_{ml}$  specifies the number of units of resource type  $l$  available on platform class  $P_m$  [12]. We have assume that the number of platform classes and the available platforms in each platform class is known. In the MDP model, each action  $a_i$  is a set of platforms  $\underline{P}(a_i) = [x_{i1}, x_{i2}, \dots, x_{im}]$ , specifying

the numbers of platforms of each class needed to pursue the action. The workload of each decision maker in the organization embodies the activity of supervising the platforms and coordinating the platforms with other decision makers to pursue the action collaboratively to achieve the desired effect and mitigate the effects of exogenous events.

Starting from the near-optimal strategy, we can design the organization structure which is congruent to this strategy. In this paper, we only study the organization as the ownership of each platform of every platform class, so that the expected total workload of the organization is minimized with each decision maker's workload constraint satisfied.

Let  $[n_1, \dots, n_m]$  be the available numbers of platforms for platform classes  $[P_1, \dots, P_m]$ . Let  $[x_{i1}, \dots, x_{im}]$  be the numbers of platforms of each platform class for action  $a_i$ . Define  $[X_1, \dots, X_m]$  as the numbers of platforms for any action. Since the occurrence of actions in the optimal strategy is probabilistic, so is  $[X_1, \dots, X_m]$ .

Let  $y_{ki}$  be the the number of platforms of class  $P_i$  allocated to  $DM_k$ . For the organization  $ORG = (DM_1, \dots, DM_D)$ , the workload  $WL_k$  of  $DM_k$ ,  $k \in \{1, \dots, D\}$  is approximated as:

$$WL_k = \alpha \sum_{i=1}^m \frac{y_{ki}}{n_i} E(X_i) + \beta \sum_{i=1}^m \sum_{j=1}^m \left[ \frac{y_{ki}(n_j - y_{kj})}{n_i n_j} E(X_i X_j) \right] \quad \forall k = 1, \dots, D \quad (2)$$

where  $\sum_{i=1}^m \frac{y_{ki}}{n_i} E(X_i)$  quantifies the internal workload of  $DM_k$ , which accounts for the supervision of platforms that  $DM_k$  owns. The internal workload of  $DM_k$  incurred by platform class  $P_i$  is proportional to

- expected number of platforms of class  $P_i$ , i.e.,  $E(X_i)$ ;
- the proportion of platforms of class  $P_i$  that  $DM_k$  owns, i.e.,  $\frac{y_{ki}}{n_i}$ ;

That is, the larger is the value of  $E(X_i)$ , the higher is the workload imposed on platform class  $P_i$ ; the larger the proportion of platforms of class  $P_i$  owned by decision maker  $DM_k$ , i.e.  $\frac{y_{ki}}{n_i}$ , the higher is the workload from class  $P_i$  on  $DM_k$ . The

term  $\sum_{i=1}^m \sum_{j=1}^m \left[ \frac{y_{ki}(n_j - y_{kj})}{n_i n_j} E(X_i X_j) \right]$  quantifies the external workload, i.e., the coordination effort of  $DM_k$  cooperating with other DMs in the organization to execute tasks. The external workload on  $DM_k$  incurred by the platform classes  $P_i$  and  $P_j$  is proportional to

- joint expectation of numbers of platforms of classes  $P_i$  and  $P_j$ , i.e.,  $E(X_i X_j)$ ;
- the proportion of platforms of class  $P_i$  that  $DM_k$  owns, i.e.,  $\frac{y_{ki}}{n_i}$ ;
- the proportion of platforms of class  $P_j$  that  $DM_k$  doesn't own, i.e.,  $\frac{(n_j - y_{kj})}{n_j}$ .

Here, the higher order expectations of joint numbers of platform classes, e.g., third order expectation  $E(X_i X_j X_w)$ ,  $i, j, w \in \{1, \dots, D\}$ , are ignored. User specified constants  $\alpha$  and  $\beta$  define the weights on internal and external workloads.

Thus, the total workload of all DMs is:

$$WL_{all} = \alpha \sum_{i=1}^m E(X_i) + \beta \sum_{i=1}^m \sum_{j=1}^m \left[ \left( 1 - \frac{\sum_{k=1}^D y_{ki} y_{kj}}{n_i n_j} \right) E(X_i X_j) \right] \quad (3)$$

The optimization problem associated with organizational design is as follows:

$$\begin{aligned} \text{objective : } \min WL_{all} &= \alpha \sum_{i=1}^m E(X_i) \\ &+ \beta \sum_{i=1}^m \sum_{j=1}^m \left[ \left( 1 - \frac{\sum_{k=1}^D y_{ki} y_{kj}}{n_i n_j} \right) E(X_i X_j) \right] \\ \text{s.t. } WL_k &= \alpha \sum_{i=1}^m \frac{y_{ki}}{n_i} E(X_i) \\ &+ \beta \sum_{i=1}^m \sum_{j=1}^m \left[ \frac{y_{ki}(n_j - y_{kj})}{n_i n_j} E(X_i X_j) \right] \leq C_k \\ &\quad \forall k = 1, \dots, D \\ \sum_{k=1}^D y_{ki} &= n_i \quad \forall i = 1, \dots, m \\ y_{ki} &\in I^+ \quad \forall k = 1, \dots, D \quad \forall i = 1, \dots, m \end{aligned} \quad (4)$$

Note that  $E(X_i)$  is the expected number of platforms of class  $P_i$ , and  $E(X_i X_j)$  is the joint expectation of numbers of platform classes  $P_i$  and  $P_j$ .

These expectations are approximated by their sample means:

$$\begin{aligned} \bar{X}_i &\approx E(X_i) \\ \bar{X}_i X_j &\approx E(X_i X_j), \quad \forall i, j \in \{1, \dots, m\} \end{aligned} \quad (6)$$

This problem, defined in Eq.(5), is typically a nonlinear constrained integer optimization problem, which is NP-hard. We applied MINLP (Mixed Integer Nonlinear Programming) algorithms in *TOMLAB<sup>TM</sup>*[17] optimization software to solve the above integer programming problem. The solver package *minlpBB* for sparse and dense mixed-integer linear, quadratic and nonlinear programming was employed for this problem.

## VI. SIMULATION

### A. Simulation Model

We illustrate our approach to effects-based robust organization design using a simple joint-task-force scenario that can be operationalized in the distributed dynamic decision-making (DDD-III) war-gaming simulator [9]. In the example system, there are four platform classes, i.e.,  $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$ , as defined in Table I, and three mission tasks (desired effects), i.e.,  $M_1$ ,  $M_2$ ,  $M_3$ , and three time critical tasks (exogenous events), i.e.,  $T_1$ ,  $T_2$ ,  $T_3$ , as defined in Table II. The dependencies among mission tasks are shown in Fig. 3.

The following three types of resources are considered:

- ASUW – Anti-SUBmarine Warfare
- STK – STriKe warfare
- SOF – Special/ground Operations

Rules of the simulation model, which are unknown to the learning agent, are as follows:

- All mission tasks should be executed by satisfying the dependency constraints;
- All time critical tasks, if they ever appear, have to be completed before the final mission task is completed;
- Each resource contributes to the task accuracy; the more resources are allocated to a task, the higher is the probability of completing the task.
- If the final mission task is achieved, the game is won and a reward of 5,000 units is earned;
- If there are three time critical tasks existing in the system, the game is lost and a penalty of 3,000 units is incurred.

### B. Monte Carlo Simulations

By using the Monte Carlo control method as shown in Fig. 2, we obtain a near-optimal (stable) action strategy via 85000 runs (episodes). The

TABLE I  
PLATFORMS

Platform	Name	Number	ASUW	STK	SOF	Cost
$P_1$	F18s	3	0	2	0	100
$P_2$	FAB	5	1	0	0	80
$P_3$	FOB	3	1	1	1	160
$P_4$	SOF	2	0	0	1	60
Reward (Win)			5000			
Penalty (Lose)			-3000			

TABLE II  
MISSIONS AND TIME CRITICAL TASKS

Mission	Name	ASUW	STK	SOF
$M_1$	NB - Naval Base	2	0	2
$M_2$	AB - Air Base	0	6	0
$M_3$	PRT - Sea Port (final goal)	2	2	0
$T_1$	SCUD - missile launcher	1	1	0
$T_2$	PH - Hostile ship	1	0	1
$T_3$	TSK - complex ground task	0	1	1

optimal action strategy, i.e., the mapping of states to actions, viz., the combinations of platforms, and values of states, are as listed in Table III. The comparison of net rewards from the near-optimal strategy obtained by Monte Carlo control method and a *randomized* greedy strategy from 2000 runs are shown in Fig. 4. The greedy strategy chooses randomly, at each state, one of the five most economically feasible platform combinations. After 10000 runs (sample episodes), the average net reward of the near-optimal strategy is 2985, while the average net reward of the greedy strategy is only 2127. Thus, the near-optimal strategy provides a 40% better reward than the greedy strategy.

### C. Organization Design Results

The platform utilization statistics, i.e., the sample mean of the numbers of platforms of each class

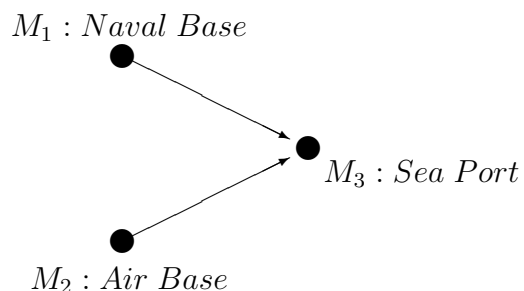


Fig. 3. Effect Dependency



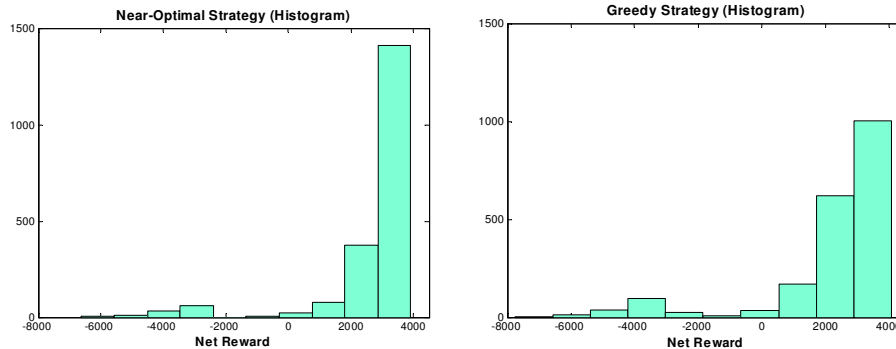


Fig. 4. Comparison of Net Reward Distributions

TABLE III  
NEAR-OPTIMAL STRATEGY

State	Action (Platforms)				State Value
	$P_1$	$P_2$	$P_3$	$P_4$	
$\emptyset$	1	1	1	3	2777.72
$M_1$	3	0	0	0	3008.48
$M_2$	0	2	0	2	3565.36
$T_1$	1	2	0	1	2115.63
$T_2$	0	2	0	2	2174.47
$T_3$	1	0	0	1	2291.07
$T_1, T_2$	1	1	0	0	1796.93
$M_1, T_1$	3	0	0	0	2270.26
$M_1, T_2$	0	2	1	2	2278.97
$M_1, T_3$	3	0	0	0	2100.65
$M_2, T_1$	0	2	0	2	2407.18
...	...	...	...	...	...
$M_1, T_1, T_2$	1	0	1	1	1693.73
$M_1, M_2$	1	2	0	0	3221.11
$M_1, M_2, T_1$	1	2	0	1	2358.82
...	...	...	...	...	...

TABLE IV  
PLATFORM UTILIZATION STATISTICS

	$P_1$	$P_2$	$P_3$	$P_4$
$\bar{X}_i$	1.2	2.3	3.1	2
$X_i X_j$	$P_1$	$P_2$	$P_3$	$P_4$
$P_1$	1.5	0.75	0.25	1.00
$P_2$	0.75	1.916	0.25	1.58
$P_3$	0.25	0.25	0.333	0.667
$P_4$	1.00	1.583	0.667	2.667

TABLE V  
DECISION MAKER PLATFORM OWNERSHIP

DM	$P_1$	$P_2$	$P_3$	$P_4$	Expected Workload ( $\alpha = 1, \beta = 1$ )	Workload Constraint
$DM_1$	1	3	1	1	5.7065	8
$DM_2$	1	1	1	1	4.9265	6
$DM_3$	1	1	1	0	3.3766	6

$\bar{X}_i$ ,  $i \in \{1, \dots, m\}$  and the sample mean of joint numbers of platform class pairs  $\bar{X}_i X_j$ ,  $i, j \in \{1, \dots, m\}$  are listed in Table IV. Using MINLP algorithms in *TOMLAB<sup>TM</sup>*, we can obtain the congruent organization for the near-optimal strategy as shown in Table V.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a methodology for designing a robust organization for stochastic and dynamic environments. The dynamic environment can be modeled as a finite state Markov Decision Process. Using Monte Carlo control methods, a near-optimal action strategy is obtained. An

organization congruent to this strategy is designed by solving an integer optimization problem.

Simulation results support the conclusion that the Monte Carlo control methods are effective in achieving the near-optimal strategies in real world applications, where the system parameters are uncertain. Formulation of organizational design problem and mix-integer optimization algorithms provide nice vehicles to realize the design of organizations that are congruent with their dynamic and uncertain environments.

We are pursuing future research along the follow-

ing directions:

- Incorporate realistic mission environments into the MDP model.
- Include additional organizational structure elements into the design process, e.g., command structure, information flow structure.
- Study the mechanisms of organizational adaptation via agent-based simulations.

- [15] K. M. Carley and Z. Lin, "Organizational design suited to high performance under stress", *IEEE transactions on System, Man and Cybernetics*, Volume 25, 1995, pp.221-231.
- [16] G. L. Nemhauser, L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley-Interscience, New York, 1988.
- [17] K. Holmström, A. O. Göran and M. M. Edvall, *User's Guide for Tomlab /MINLP*, <http://tomlab.biz/download/manuals.php>.

## REFERENCES

- [1] D. P. Bertsekas and S. E. Shreve, *Stochastic Optimal Control: The Discrete-Time Case*, Athena Scientific, 1996.
- [2] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific 2001.
- [3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press.
- [4] L.P. Kaelbling, M. L. Littman, and A. W. Moore, *Reinforcement learning: A survey*, *Journal of Artificial Intelligence Research*, 4:237-285.
- [5] P. R. Kumar and P. Variya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*, Prentice-Hall, Englewood Cliffs, NJ.
- [6] G. Tesauro and G. R. Galperin, *On-line Policy Improvement Using Monte-Carlo Search*, *Advances in Neural Information Processing: Proceedings of the Ninth Conference*.
- [7] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley-Interscience Publication.
- [8] D. S. Alberts and R. E. Hayes. (2003), *Power to the Edge: Command Control in the information Age*, Information Age Transformation Series, CCRP Publications.
- [9] D. L. Kleinman, P.W. Young, and G. S. Higgins, *The DDD-III: A tool for empirical research in adaptive organizations*, Proc 1996 Command Control Res Technol Symp, NPS, Monterey, CA, June 1996.
- [10] D. L. Kleinman, G. M. Levchuk, S. G. Hutchins and W. G. Kemple, *Scenario Design for the Empirical Testing of Organizational Congruence*, 8th International Command and Control Research and Technology Symposium, Monterey, CA, June 2003.
- [11] C. Meirina, G. M. Levchuk, and K. R. Pattipati, *A Multi-Agent Decision Framework for DDD-III Environment*, 8th International Command and Control Research and Technology Symposium, Monterey, CA, June 2003.
- [12] G. M. Levchuk, Y. N. Levchuk, J. Luo, K. R. Pattipati and D.L. Kleinman, *Normative Design of Organizations Part I: Mission Planning*, *IEEE Trans. on SMC: Part A Systems and Humans*, Vol. 32, No. 3, pp. 346-359, May 2002.
- [13] G. M. Levchuk, Y.N. Levchuk, J. Luo, K.R. Pattipati and D. L. Kleinman, *Normative Design of Organizations Part II: Organizational Structures*, *IEEE Trans. on SMC: Part A Systems and Humans*, Vol. 32, No. 3, pp. 360-375, May 2002.
- [14] G. M. Levchuk, Y. Levchuk, J. Luo, F. Tu and K. R. Pattipati, *A Library of optimization Algorithms for Organizational Design*, 2000 Command and Control Symposium, Monterey, CA, June 26-28, 2000.