

Intelligent Operation System for the Autonomous Vehicle Fleet

by

Xingyu Li

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2018

Doctoral Committee:

Professor Bogdan Epureanu, Chair
Professor Mihaela Banu
Professor Xiuli Chao
Professor Ilya Kolmanovsky

Xingyu Li

lixingyu@umich.edu

ORCID iD: 0000-0001-8034-1232

© Xingyu Li September 2018

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge US Army Center of Excellence in Modeling and Simulation of Ground Vehicle Systems, head-quartered at the University of Michigan. for funding my research activities during my Ph.d. student.

I would like to express my sincere gratitude and appreciation to my advisor Prof. Bogdan Epureanu for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me a lot to develop many skills to do my research more effectively and efficiently by challenging and correcting me all the time. There is still so much to learn. I could not have imagined having a better advisor and mentor for my Ph.D study.

I would like to express my special thanks to Prof. Xiuli Chao. He was the one who provides indispensable advice, information and support on different aspects of my project. His endless support was an important factor for me to overcome difficulties in my research. I can never repay him.

I would like to thank my committee members, Prof. Ilya Kolmanovsky and Prof. Mihaela Banu for serving on my committee and providing a lot of valuable suggestion on my research.

Finally, I wish to record my deep gratitude to the support and love from my family - my parents, Li and Wang, and my significant other, Yanchao, Tan. They all kept me going through all the difficulties during my Ph.d. study. This dissertation would not have been possible without them.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
LIST OF FIGURES	vi
LIST OF TABLES	x
ABSTRACT	xi
CHAPTER	
I. Introduction	1
1.1 Problem Definition	1
1.1.1 Benefits from Fleet Modularity	2
1.1.2 Real-time Management of Reconfiguration Manufacturing System	4
1.2 Related Work	5
1.2.1 Design of Fleet Operation System	5
1.2.2 Intelligent Agent Competition	6
1.2.3 Teaming Among Multiple RMSs	7
1.3 Dissertation Contributions	10
1.4 Dissertation Overview	11
II. Literature Review	12
2.1 Modular Vehicle	12
2.2 Model Predictive Control	13
2.3 Agent Based Modeling	16
2.4 Reconfigurable Manufacturing System	19
III. Model Predictive Control	21
3.1 Problem Formation	21
3.2 Model Predictive Control	23

3.2.1	Future State Prediction	27
3.2.2	Cost Function	28
3.3	Case Study 1	30
3.3.1	Operation Management	30
3.3.2	Modular Vehicle Design	35
3.4	Case Study 2	36
3.4.1	System Robustness and Adaptability	36
3.4.2	Stability Margin	45
3.5	Stochastic Model Predictive Control	47
3.6	Case Study 3	54
3.6.1	Fleet Performance with SMPC	56
3.6.2	Fleet Comparison with Supply	58
3.6.3	Fleet Comparison without Supply	59
3.6.4	ADR Action Capacity	62
3.7	Summary	63
IV.	Intelligent Operation System	65
4.1	Problem Formulation	65
4.2	Agent Based Model	67
4.2.1	Camp agent	70
4.2.2	Base agent	71
4.2.3	Supply agent	76
4.3	Case Study	78
4.3.1	JTTS scenario	79
4.3.2	Sensitivity Analysis	84
4.4	Model Implementation	86
4.5	Summary	87
V.	Attack-Defender Game	89
5.1	Game Formation	89
5.2	Attacker-Defender Model	93
5.2.1	Inference Agent	93
5.2.2	Dispatch Agent	97
5.2.3	Base Agent	101
5.3	Numerical Illustrations	109
5.4	Fleet Comparison	111
5.5	Summary	120
VI.	Real-time management of RMS	121
6.1	Model Description	121
6.1.1	Negotiation Among Lines	121
6.1.2	Operation Action Assignment	123

6.1.3	Mining of Potential Configurations	125
6.1.4	Capacity Planning	127
6.1.5	Reconfiguration Action Planning	128
6.1.6	Configuration Path Planning	129
6.1.7	Processing Time Planning	130
6.2	Sample Results	131
6.2.1	Real-time Capacity Management	131
6.2.2	Teamming Among Lines (i.e., among distinct RMS)	133
6.3	Summary	134
VII.	Conclusions	136
7.1	Summary	136
7.2	Summary of Contributions	138
7.3	Limitation of the Study	139
7.4	Future Work	139
BIBLIOGRAPHY	141

LIST OF FIGURES

Figure

1.1	Comparison of a normal tactical vehicle (left) and a modularized tactical vehicle (right).	2
1.2	Operation actions for modularized vehicles: a) assembly, b) disassembly, c) reconfiguration.	2
1.3	Model of fleet operation system through a state space model for both conventional and modular fleets	6
1.4	The operational decisions update at every time point according to the latest demands information and inventory status	7
1.5	Information flow and decision making model used in competition between two adversarial and intelligent fleets.	8
1.6	Teaming between RMS to improve the demand fulfillment through demand and resource sharing	9
3.1	Layout of fleet operation system	22
3.2	The framework of the model predictive control algorithm	27
3.3	ADR actions history during modular fleet operation	32
3.4	Fleet readiness varies with changing capacity	34
3.5	Fitted surface that captures the relationship between vehicle design and fleet readiness	36
3.6	Projection of the fitted surface	37
3.7	An analogy between conventional vehicles and modular vehicles . . .	38

3.8	Reaction to disturbance during conventional fleet operation	40
3.9	Reaction to disturbance during modular fleet	40
3.10	Reaction to disturbance during conventional fleet operation	42
3.11	Reaction to disturbance during modular fleet operation	42
3.12	History of vehicle assembly, disassembly and reconfiguration actions	43
3.13	Reaction to disturbance during modular fleet and conventional fleet operation	44
3.14	Reaction to disturbance during modular fleet and conventional fleet operation	45
3.15	Peak values of convoy attribute changes with the various demand impulses	46
3.16	Stochastically generated demands for 20-day operation	55
3.17	Comparison of fleet performance between MPC and SMPC	57
3.18	Comparison of delayed attributes between different fleets with unlim- ited resupply	59
3.19	Comparison of resupplied resources between different fleets with re- supply	60
3.20	Comparison of suffered insufficiencies between different fleets without resupplies	61
3.21	Comparison of attribute insufficiency and stability with changing ini- tial supplies	62
3.22	Capacity allocation in different available working capacity	63
3.23	Amount of delayed attributes changing with different capacity . . .	64
4.1	Mission scenario in JTTS	66
4.2	Analogy between the military fleet operation and supply chain . . .	67
4.3	Process flow in modular fleet operation	68

4.4	Diagram of the agent-based model for modular fleet operation . . .	69
4.5	Rule-based algorithm for ADR action planning	74
4.6	Comparison of total resupplied resources	80
4.7	Readiness comparison between different fleets	82
4.8	Comparison of ADR actions on different types of vehicles between Empiricist and Optimizer	83
4.9	Modular fleet performances under diverse ADR capacities	85
5.1	Mission scenario created for attacker-defender game	91
5.2	Agent based model and its information flow in attacker-defender game	94
5.3	LSTM architecture used for analyze enemy's historical behaviors and forecast enemy's future actions	96
5.4	Neural network model for feasibility training	99
5.5	Neural network model for success rate training	100
5.6	Comparison of win rate between modular fleet and conventional fleet	112
5.7	Comparison of mismatched convoy attributes	113
5.8	Maximum value of attributes carried by the convoy in different months	115
5.9	Modules damage occurs at each month	116
5.10	Proportion of defense strategy at different stages	117
5.11	Comparison of inference accuracy between modular fleet and conven- tional fleet	118
5.12	Comparison of machine requirements between modular fleet and con- ventional fleet	119
6.1	Structure of optimization models designed for RMS operation . . .	121
6.2	Negotiation among lines base on bidding process	124

6.3	Matching between capacity of RMS for part 1 and its demand . . .	132
6.4	Matching between capacity of RMS for part 2 and its demand . . .	133
6.5	Matching between capacity of RMS for part 1 and its demand . . .	134
6.6	Matching between capacity of RMS for part 2 and its demand . . .	134
6.7	Module level of both lines without negotiation	135
6.8	Module level of both lines with negotiation	135

LIST OF TABLES

Table

3.1	Modular vehicle compositions	31
3.2	System parameter settings	31
3.3	Fleet readiness comparison	31
3.4	Total actions comparison between fleets	32
3.5	Operation cost comparison	33
3.6	The mapping between the modular vehicles and its modules	39
3.7	The mapping between the modular vehicles and its attributes	39
4.1	Comparison of total operational costs between fleets	79
5.1	Dispatch strategies for attacker and defender	91
5.2	Mapping between vehicles and attributes	110
6.1	Machines' composition and their average processing rate	131

ABSTRACT

Modular vehicles are vehicles with interchangeable substantial components also known as modules. Fleet modularity provides extra operational flexibility through on-field actions, in terms of vehicle assembly, disassembly, and reconfiguration (ADR). The ease of assembly and disassembly of modular vehicles enables them to achieve real-time fleet reconfiguration, which is proven as beneficial in promoting fleet adaptability and in saving ownership costs. The objective of military fleet operation is to satisfy uncertain demands on time while providing vehicle maintenance. To quantify the benefits and burdens from modularity in military operation, a decision support system is required to yield autonomously operation strategies for comparing the (near) optimal fleet performance for different vehicle architectures under diverse scenarios.

The problem is challenging because: 1) fleet operation strategies are numerous, especially when modularity is considered; 2) operation actions are time-delayed and time-varying; 3) vehicle damages and demands are highly uncertain; 4) available capacity for ADR actions and vehicle repair is constrained. Finally, to explore advanced tactics enabled by fleet modularity, the competition between human-like and adversarial forces is required, where each force is capable to autonomously perceive and analyze field information, learn enemy's behavior, forecast enemy's actions, and prepare an operation plan accordingly. Currently, methodologies developed specifically for fleet competition are only valid for single type of resources and simple operation rules, which are impossible to implement in modular fleet operation.

This dissertation focuses on a new general methodology to yield decisions in op-

erating a fleet of autonomous military vehicles/robots in both conventional and modular architectures. First, a stochastic state space model is created to represent the changes in fleet dynamics caused by operation actions. Then, a stochastic model predictive control is customized to manage the system dynamics, which is capable of real-time decision making. Including modularity increases the complexity of fleet operation problem, a novel intelligent agent based model is proposed to ensure the computational efficiency and also imitate the collaborative decisions making process of human-like commanders. Operation decisions are distributed to several agents with distinct responsibility. Agents are designed in a specific way to collaboratively make and adjust decisions through selectively sharing information, reasoning the causality between events, and learning the other's behavior, which are achieved by real-time optimization and artificial intelligence techniques.

To evaluate the impacts from fleet modularity, three operation problems are formulated: (i) simplified logistic mission scenario: operate a fleet to guarantee the readiness of vehicles at battlefields considering the stochasticity in inventory stocks and mission requirements; (ii) tactical mission scenario: deliver resources to battlefields with stochastic requirements of vehicle repairs and maintenance; (iii) attacker-defender game: satisfy the mission requirements with minimized losses caused by uncertain assaults from an enemy.

The model is also implemented for a civilian application, namely the real-time management of teams of reconfigurable manufacturing systems (RMSs). As the number of RMS configurations increases exponentially with the size of the line and demand changes frequently, two challenges emerge: how to efficiently select the optimal configuration given limited resources, and how to allocate resources among lines. According to the ideas in modular fleet operation, a new mathematical approach is presented for distributing the stochastic demands and exchanging machines or modules among lines (which are groups of machines) as a bidding process, and for adaptively configuring

these lines and machines for the resulting shared demand under a limited inventory of configurable components.

The main original contributions of this dissertation are: (i) proposed a new stochastic MPC framework for managing the dynamics of operation/manufacturing systems considering the uncertainty; (ii) created a novel agent based model to simulate the human-like decision making process; (iii) formulated an attacker-defender game to highlight the tactical advantages brought by modularity; (iv) designed a negotiation algorithm among teams of vehicles/reconfigurable manufacturing machines for real-time task allocation and resource sharing.

CHAPTER I

Introduction

1.1 Problem Definition

Military vehicles operate in a large variety of environments and scenarios resulting in a diverse set of requirements from fleet mix. The special functionalities of military vehicles and incessantly updated technologies make them hardly to be reused after military operation (*Shinkman*, 2014). In order to reduce the wastage, the US Army requires that fleets of vehicles can be reutilized across a large array of military mission scenarios. Modular vehicles are then introduced by constructing from special components, which are named as modules (*Dasch and Gorsich*, 2016). Modules are assumed to be a special type of components which can be easily coupled/decoupled through simple plug-in/pull-out actions on battlefields. As an example, Fig. 1.1 compares the vehicles structures of a tactical vehicle and a highly modularized tactical vehicle. For a modularized vehicle, five types of modules are specified: a) cabin, b) chassis and power train, c) tires and suspension, d) payload and e) armor.

Vehicle assembly and disassembly can be realized through module connection and disconnection. Vehicle reconfiguration and repair can also be quickly realized by module replacement. On-field assembly, disassembly and reconfiguration (ADR) actions enable modular fleet to vary its configuration in real-time (as shown in Fig. 1.2), and also distinguish the modular fleet operation from conventional fleet operation.

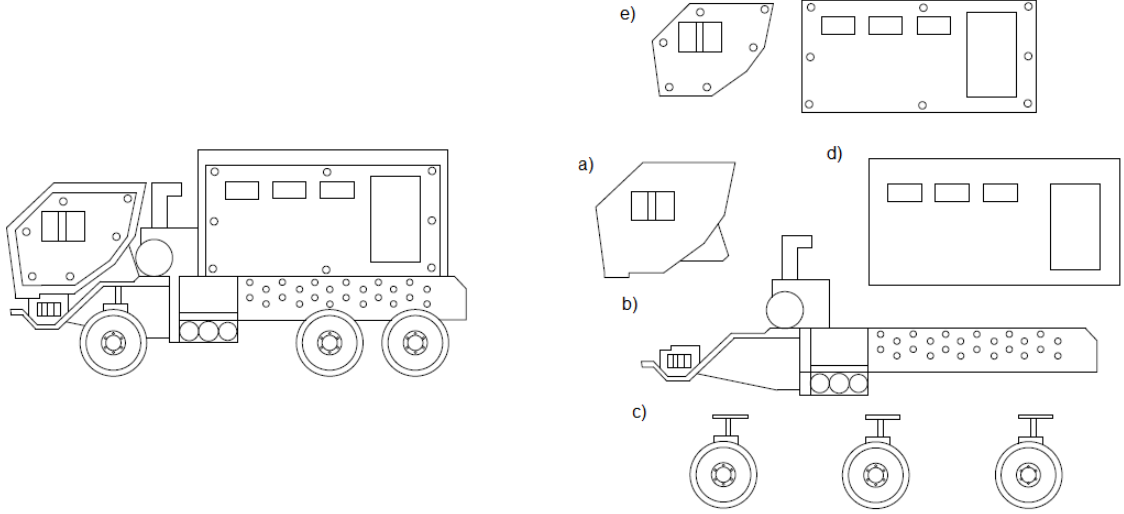


Figure 1.1: Comparison of a normal tactical vehicle (left) and a modularized tactical vehicle (right).

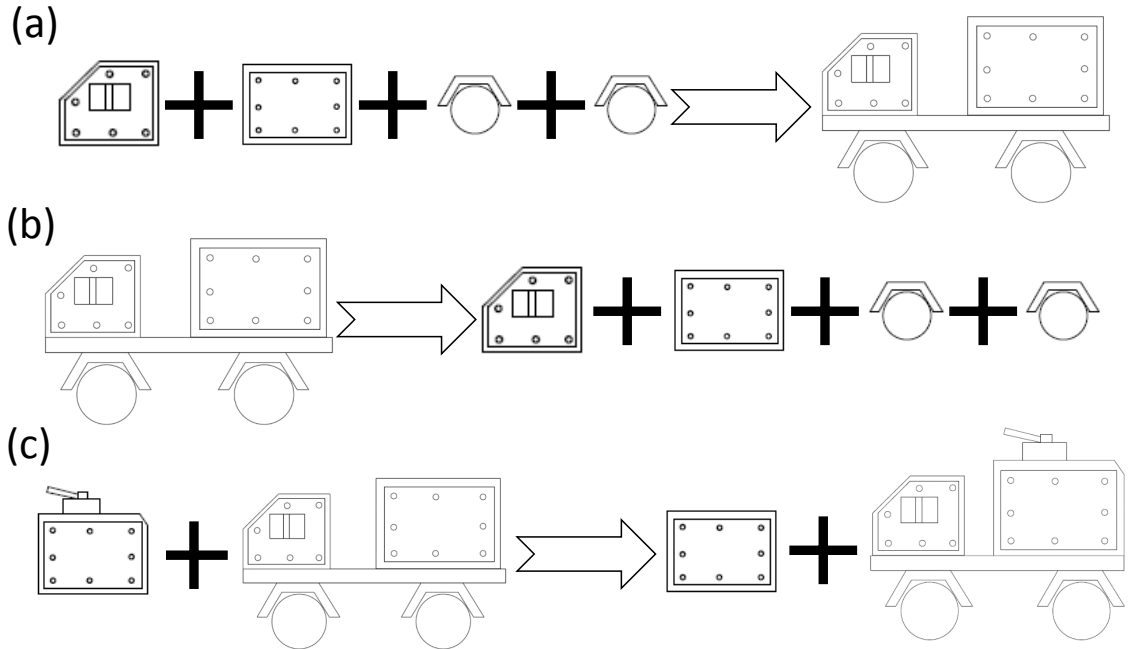


Figure 1.2: Operation actions for modularized vehicles: a) assembly, b) disassembly, c) reconfiguration.

1.1.1 Benefits from Fleet Modularity

In this study, several modular vehicle architectures are firstly generated according to the previous research on modular vehicle design and suggestion from military ex-

perts. Assuming that the modular fleet can be designed and manufactured to enable ADR actions achieved in hours, the first priority of this research is to investigate the benefits from fleet modularity in terms of fleet operational cost, adaptability, resource utility and tactical advantages. Compared to the existing operation problems, the military operation has three special characteristics: (i) the highly uncertain and transient demands which require the decision maker's rapid reaction (ii) the evolutionary demands as commanders from both fleets keep updating their combat and operation decisions based on the others behaviors (iii) high probability of damage during operation, which requires the timely repair and maintenance to balance resource utility rate and demand fulfillment. These conditions make the operation of modular fleet challenging.

The first existing challenge is the management of an inventory with high diversity. We denote a vehicle with one or more damaged components as a damaged vehicle. For example, a single vehicle with 5 different components has 2^5 types of vehicle damage, each requiring a distinct recovery strategy. Next, numerous operation actions need to be determined in real-time, i.e., vehicle reconfiguration, vehicle relocation, convoy formation, vehicle maintenance/repair and etc. Taking vehicle repair as an example, vehicles with different damaged components require different procedure to recover. Given modularity, the damaged vehicles can also be disassembled into modules or reconfigured to other type of vehicles by swapping the damaged modules by modules of other type. Furthermore, all the operational actions require a certain amount of time to be accomplished, and hence the scheduling must be schedule has to be performed ahead of time to guarantee the desired actions can be achieved on time. Given the limited working capacity and limited stocks with diverse conditions, how to effectively and efficiently schedule the operations and distribute the limited capacity in reacting to the stochastically arrived demands while maintaining reasonable healthy stock levels represents another challenge.

Finally, in order to explore the advanced tactics which exploit fleet modularity, the way a human commander makes decisions needs to be considered and modeled. With some simplifications, the main procedure involved in human-like decision making (*Chen and Barnes, 2013*) involves the following steps: 1) perceive field information 2) analyze enemy's behavior based on the received information 3) optimally schedule the on-field actions to beat enemy. Through these procedures, decision makers could adaptively adjust their dispatch strategies and operation plans based on the learned enemy's behaviors, i.e., depending on the strategy the enemy might adopt in a specific situation. Combining with fleet modularity, how to design a decision making model to imitate the human-like thinking process with guaranteed decision optimality remains to be addressed.

1.1.2 Real-time Management of Reconfiguration Manufacturing System

Fleet operation model is also extended into several civilian applications. Unpredictable and frequent market changes including demand fluctuations and rapid introduction of new products make the development of cost effective and responsive manufacturing systems a necessity for companies to survive in the new competitive environment. Reconfigurable manufacturing systems (RMS), which combine high throughput from dedicated manufacturing lines and flexibility from flexible manufacturing systems, have been introduced to address this challenge (*Koren et al., 1999*). With the similar capacity as modularized fleet, each reconfigurable machine can be regarded as a modularized vehicle, thus, the management of RMS, which is a manufacturing line with certain machine layout, can be also regarded as operation of a highly modularized vehicle fleet.

Concretely, a firm that processes multiple types of parts by several autonomous reconfigurable lines is considered. Each part contains diverse features and requires sequential operations. Each line consists of a multi-stage RMS and the associated

configurable components, i.e., modules. Each machine can be reconfigured by swapping its modules. The machines in different configurations are referred as variants. To satisfy the demand, the number and variants of the machines for each stage need to be determined, as well as the assigned operations. As the number of RMS configurations increases exponentially with the size of the line (*Koren and Shpitalni, 2010*) and demand changes frequently, two challenges emerge: how to efficiently select the optimal configuration given limited resources, and how to allocate resources among lines.

Multiple lines are considered for increased throughput. Each line is a system (an RMS), and hence the company is a system of systems. Because of the model complexity, finding near-optimal resource and demand allocation plans for these lines by a centralized management strategy is inefficient. In addition, the number of relocation decisions rapidly grows with increasing number of lines and types of resources. Combined with the complexity in RMS planning, the resource and demand allocation becomes infeasible to be solved by known optimization tools and models; a new method is needed.

1.2 Related Work

1.2.1 Design of Fleet Operation System

Because the number of operational actions is significant and the time delays are considered to complete actions, state space models have been considered to clarify the relationship between resources (vehicles and modules) levels (states) and operational actions (inputs) for both fleets. The outputs of the system describe the performance of vehicle convoys at different battlefields, which is measured by attributes. This expression is shown in Fig. 1.3.

The stochastic damage during military operation results in the vehicles with di-

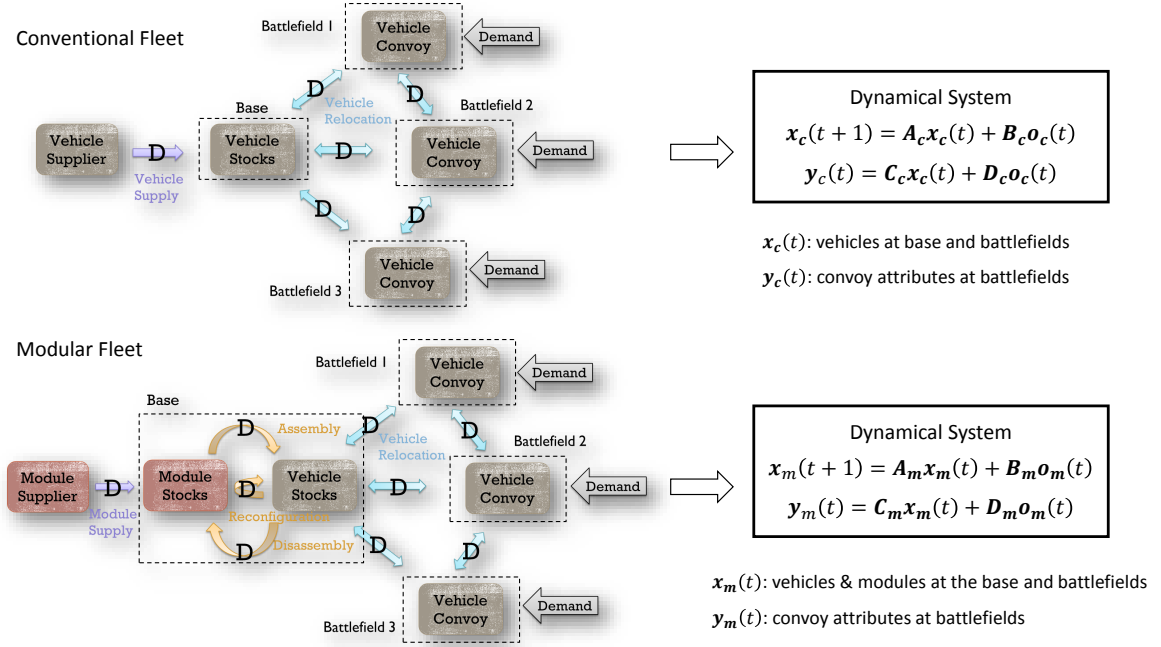


Figure 1.3: Model of fleet operation system through a state space model for both conventional and modular fleets

verse conditions, which require distinct operational actions to recover. As the amount of vehicle damage varies given newly occurred damage and vehicle recovery, the operation system requires to be updated in real-time to reflect the latest inventory status. To ensure decisions are made according to the latest field information and inventory status, the idea from model predictive control is implemented to manage the system dynamics. At each time instant, the operational actions over the short prediction horizon are planned. After each optimization, only the first element of the input sequence (decision for the first day) is implemented. The decision making process for military fleet operation is summarized in Fig. 1.4.

1.2.2 Intelligent Agent Competition

Military demands are time-varying and highly stochastic because commanders keep reacting to enemy's actions. To capture these characteristics, an intelligent agent based model is formulated to imitate decision making process in operating a

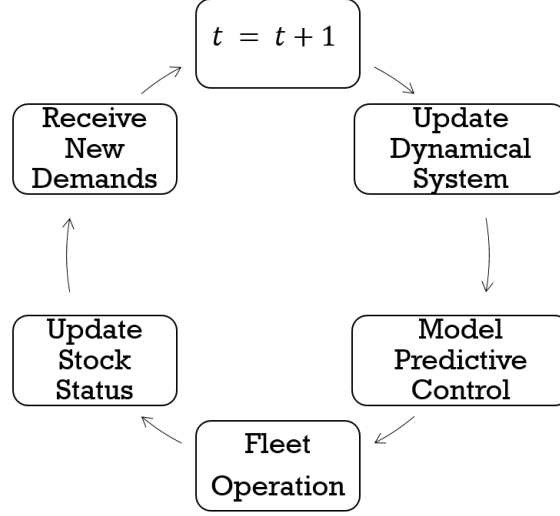


Figure 1.4: The operational decisions update at every time point according to the latest demands information and inventory status

military fleet by applying real-time optimization and artificial intelligence techniques. The agents are capable to infer enemy's future move based on historical data and optimize dispatch/operation decisions accordingly.

The model is implemented in an attacker-defender game between two adversarial and intelligent players to provide a platform for evaluating the quality of decisions. Given the same level of combat resources and intelligence, the tactical advantages from fleet modularity are highlighted, in terms of win rate, unpredictability and suffered damage. Fig. 1.5 provides the overview of agent based model, including the information flow among agents and decision making models.

1.2.3 Teaming Among Multiple RMSs

Personalized production poses new challenges to reconfigurable manufacturing systems due to a dramatic increase in the variety and stochasticity of the manufacturing demand. To better satisfy the demands and guarantee the resource utility rate, autonomously reconfigurable manufacturing systems need to be connected with a decentralized management of individual autonomous lines that can be reconfigured for

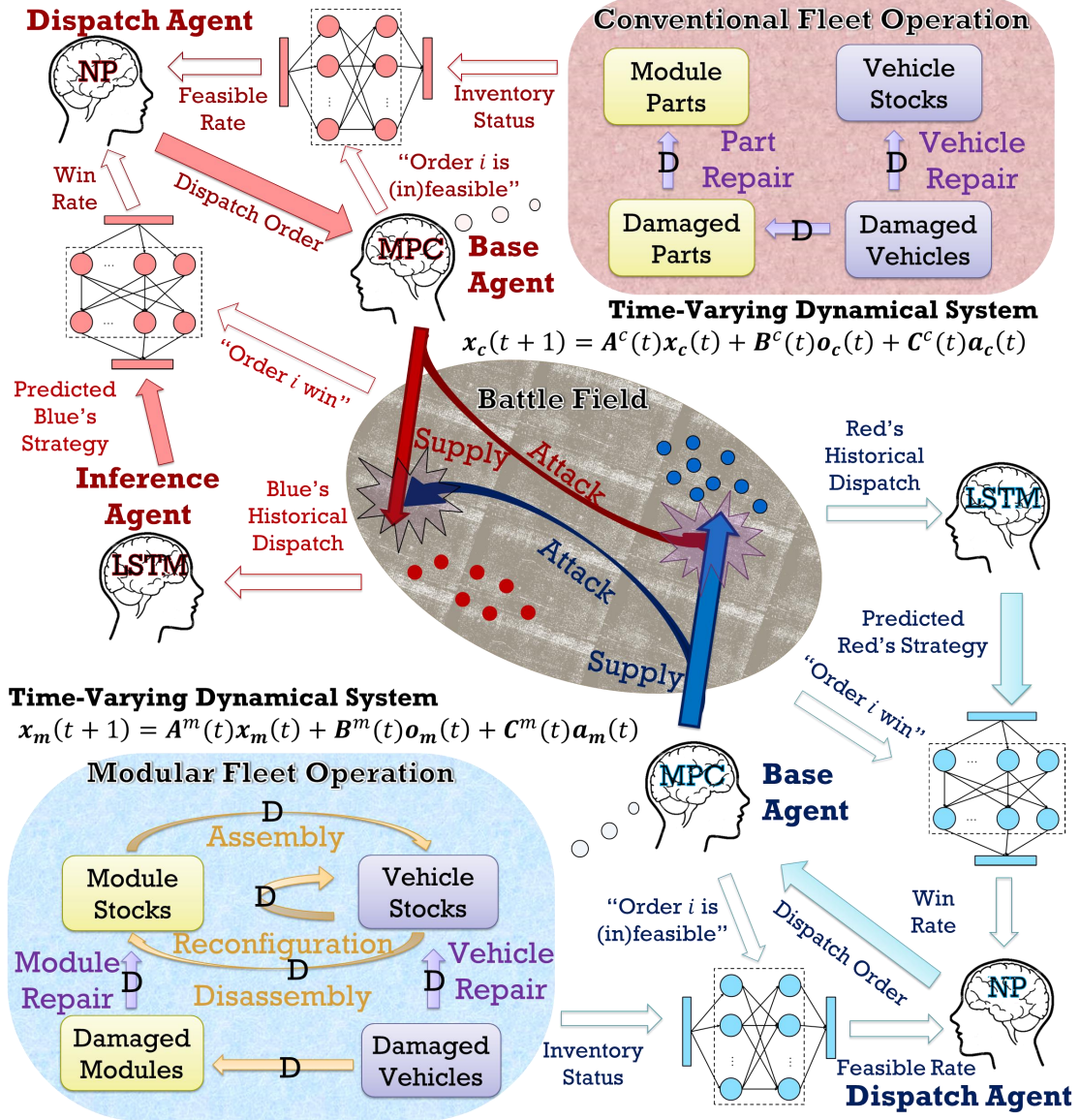


Figure 1.5: Information flow and decision making model used in competition between two adversarial and intelligent fleets.

diverse manufacturing tasks via modular manufacturing modules. A mathematical approach is presented to achieve the teaming of RMSs, i.e., distributing the demands and exchanging machines or modules among lines, through a bidding process, and for adaptively configuring these lines and machines for the resulting shared demand under a limited inventory of configurable components. Fig. 1.6 provides an example in which the operational decisions are improved through RMS reconfiguration, and

through resource and demand sharing to eliminate the remaining demands.

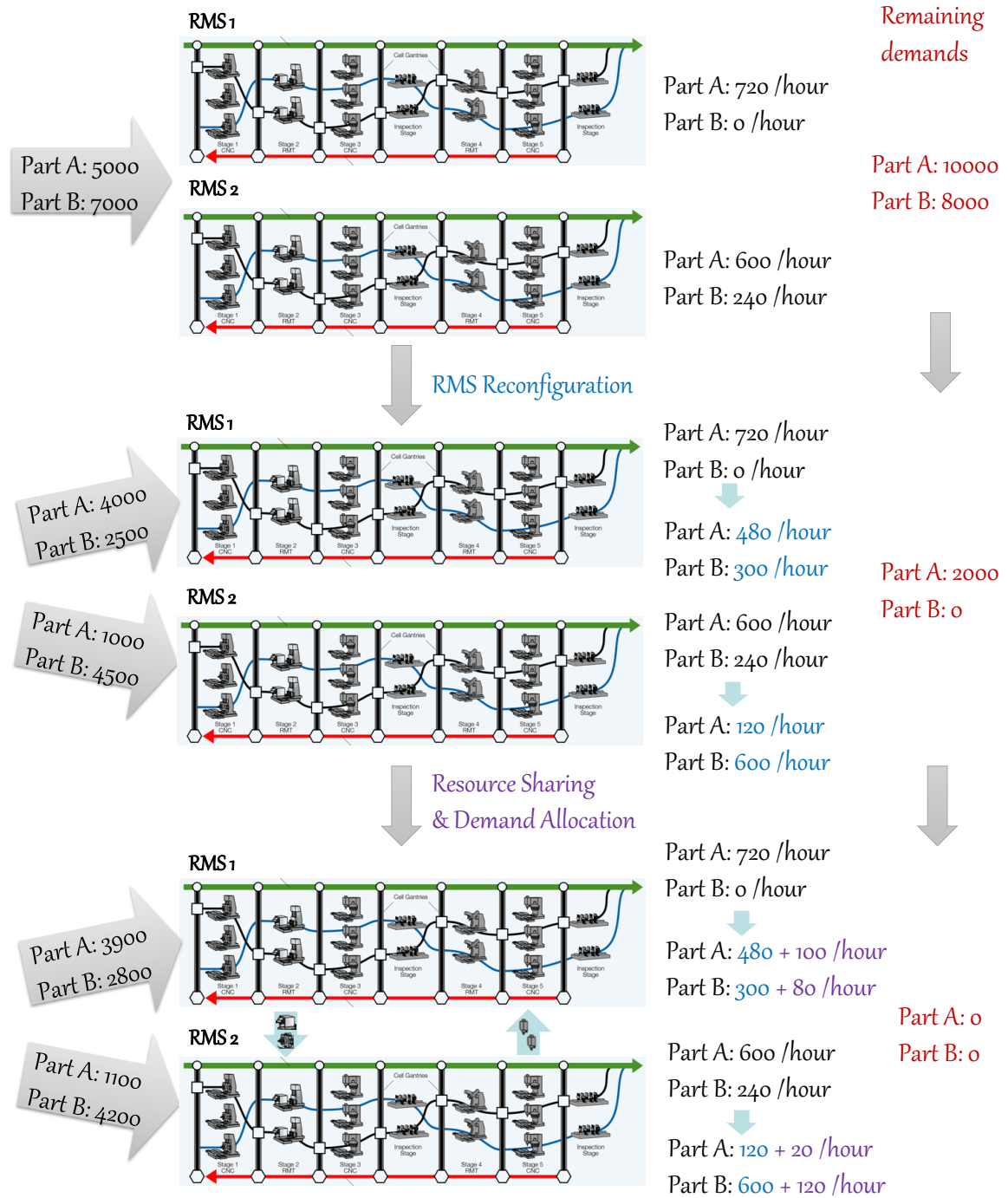


Figure 1.6: Teaming between RMS to improve the demand fulfillment through demand and resource sharing

1.3 Dissertation Contributions

The research contributions of the dissertation are summarized as:

1. Modeled fleet operation as a time-delayed dynamical system, and implemented model predictive control to manage system dynamics in real-time.
2. Analyzed the increased adaptability and robustness resulting from fleet modularity in various mission scenarios.
3. Created an intelligent agent based model to simulate the human-like decision making process in military fleet operation.
4. Investigated the tactical advantages from fleet modularity by performing an attacker-defender game between two intelligent and adversarial players.
5. Designed a group of optimization models to enable each RMS to autonomously reconfigure itself based on the received demands and available resources.
6. Created a negotiation algorithm to efficiently and effectively reallocate demands and resources among the RMSs to obtain a better team performance.

1.4 Dissertation Overview

An introduction to the problem is presented in Chapter 1. Chapter 2 contains a review of the relevant literature. In Chapter 3, the modeling of operation system and implementation of MPC/SMPC are described, as well as the corresponding results from different case studies. In Chapter 4, the model is extended to an agent based model for solving a realistic military logistic problem, i.e., joint tactical transportation system. Chapter 5 outlines the attacker-defender game created to model the competition between two intelligent and adversarial forces to reveal the tactical advantages from fleet modularity. In Chapter 6, the model is implemented for a civilian application to achieve the real-time teaming among different RMSs. Finally, Chapter 7 summarizes main conclusions and identifies both limitations of the current research and recommendations for further research.

CHAPTER II

Literature Review

2.1 Modular Vehicle

Modular architecture refers to using interchangeable components to create products in different variants (*Ulrich, 1994*). Architectures of modular military vehicle have been developed for many years. In the 1980s, Family of Vehicles (FOVs) were already designed exploiting the ideas of modular architectures. Later, two examples of an entire FOV have been proposed, named as Armored Family of Vehicle (AFV) in the 1980s and Future Combat Systems (FCS) in the 2000s (*Dasch and Gorsich, 2016*). In general, previous research in military vehicle modularity proposed a range of modular vehicle architectures based on the manufacturing cost and commonality of components. To evaluate the actual performance of modular fleet, it is important to capture the dynamic perspective of fleet level performance, i.e., fleet readiness, operation cost, etc.

Research on operation management strategies for modular fleet has been conducted for many scenarios and methods. In 2017 for instance, D’Souza et al., (*D’Souza et al., 2016*) built an integrated fleet operation model to evaluate the effectiveness and cost for operating a modular fleet, in terms of functionality model, fleet operation model, manufacturing model and transportation model. In 2018, Bayrak et al. (*Bayrak et al., 2018*) formulated a mathematical model to compare the fleet

performance under equivalent operation strategies designed for conventional fleet and modular fleet respectively. Their results show that fleet modularity is beneficial in life cycle cost saving. The existing studies compare fleet performance with considerable assumptions, i.e., no vehicle reconfiguration, heuristic and time-invariant operation strategy. There is still lack of in-depth efforts in seeking an adaptive operation strategy that could be real-timely updated according to the inventory fluctuation and stochastically arrived demands.

2.2 Model Predictive Control

Decisions require timely update to reduce the negative effects from uncertainty, research has been done in the areas of dynamic scheduling and predictive control. Dynamic scheduling has been defined under three categories, which are completely reactive scheduling, robust pro-active scheduling and predictive-reactive scheduling (*Herroelen and Leus, 2005*). In completely reactive scheduling, the decisions are made based on the short-term events, some heuristic dispatching rules are usually used with it. This type of scheduling is known as myopic, and always applied in a case with high difficulty in forecasting (*Ouelhadj and Petrovic, 2009*). On the contrary, robust pro-active scheduling approach predict the demands based on received information in a dynamic environment (*Vieira et al., 2003*). Predictive-reactive scheduling is the most common approach in the manufacturing systems. Predictive scheduling is initialized the first place, then it is revised by the rescheduling triggered in response to the real-time events. Once rescheduling is triggered, two strategies are commonly used: schedule repair and complete rescheduling (*Sabuncuoglu and Bayiz, 2000*). Schedule repair refers to some local adjustment of the current schedule for reducing the computational load. In contrast, complete rescheduling regenerates a new schedule from scratch, which is better in maintaining the optimality of the solutions.

There are connections between predictive-reactive scheduling with complete reschedul-

ing strategy and model predictive control (MPC). For one, they both optimize the decisions based on the current system status and future demand forecasts. In MPC, the current control action is obtained by solving on-line an open-loop optimal control problem for each sampling instant (*Mayne et al.*, 2000; *Scattolini*, 2009), but usually only one future step of the decisions is selected, which make the MPC have more responsiveness than the dynamic scheduling methods in reacting to disturbances and noise. However, it substantially increase the computational time in problem solving,

Several researchers have successfully implemented the MPC approach to control networks and supply chains. In 2007, Negenborn et al. (*Negenborn et al.*, 2008) considered multi-agent control schemes in which each agent is designed in a MPC approach. They applied the multi-agent MPC for the control of large-scale transportation networks. Dunbar and Desa (*Dunbar and Desa*, 2007) also implemented a distributed MPC for the dynamic management of supply chain networks. Their results showed that MPC can improve system performance over other approaches.

Military fleet control problem involves systems which are inherently uncertain, have non-linear dynamics and are subjected to some form of constraints. Uncertainties to a system include random disturbances and noise, which results additive or multiplicative stochastic variables in the dynamical system. A fundamental question about MPC is its robustness to model uncertainty and noise, to prevent violation of the inequality constraints, which presents additional complexity in terms of computation and analysis. As MPC is very sensitive to even the slightest of disturbance, new model is required to handle the uncertainty.

Robust MPC and stochastic MPC are selected as two alternatives to taking account of model uncertainty. For robust MPC method, the optimal control problem at each time step is typically described as a min-max optimization problem such that all given constraints are satisfied for all possible uncertainties (*Fukushima and Bitmead*, 2003). However, one of limitations of this control design is its conservativeness which

easily leads to infeasibility. Another one is the heavy computation to find the optimal solutions, especially for a large-dimension optimization problem.

The shortcomings of robust MPC motivated the development of stochastic MPC (SMPC). Compared to robust MPC, SMPC takes uncertainty distribution (combination of mean and variance of the future states and outputs) in to consideration, and ensure the outputs and states to be restricted to a specified confidence interval. Research on SMPC has been focused on the stochastic formulation of constraints and objective function. Stability of SMPC are also of interest. Couchman et al. proposed an SMPC algorithm with guaranteed closed loop stability without consideration of constraints. The idea of SMPC has been widely used in solving diverse practical problems, in terms of process control (*Van Hessem and Bosgra, 2006; Camacho and Bordons, 2012*), power management (*Kumar et al., 2018; Ripaccioli et al., 2010*), vehicle path planning (*Bichi et al., 2010; Di Cairano et al., 2014; Qu et al., 2015*), etc. In 2018, Tsao et al. presented a stochastic model predictive control algorithm for solving autonomous mobility on demand problem (*Tsao et al., 2018*). They proposed the model for short-term probabilistic forecasts on dispatching and rebalancing demands and formulated stochastic optimization to real-timely distributes the vehicles to satisfy the demands timely.

However, very limited research has been done regarding the implementation of SMPC on the operation management of a fleet, especially for a highly-modularized vehicle fleet. In such case, decisions need to be made concerning resupply, inventory management, ADR actions scheduling and dispatch. In this dissertation, fleet operation model is formulated as well as probabilistic objective function and constraints. A closed loop control architecture is created to yield the operational decisions in real-time. The aim of this model is to analyze the system-level performance in terms of the adaptability and responsiveness in demands disturbance and system uncertainty.

2.3 Agent Based Modeling

ADR actions enabled by fleet modularity also introduces an additional layer in fleet operation, which dramatically raise the complexity and difficulty in fleet operation, especially when diverse range of vehicle types are involved. It is hard to formulate the problem with all details by using a centralized management strategy. Even if the model is established, it is often intractable to solve because of computation complexity. In order to address this issue, increasing attention have been focused on the multi-agent systems (MAS), where the decisions are made by multiple autonomous or semi autonomous problem-solving agents (*Adhau et al.*, 2013; *Böhnlein et al.*, 2011). Compared to other simulation modeling techniques, i.e., system dynamic (*Linnéusson et al.*, 2018) and discrete event (*Sharda and Akiya*, 2012), ABM is more active in reacting to the changes in the environment (*Maidstone*, 2012), which can truly reflect the situation in military mission environment. Furthermore, the interaction between agents can also represent the information flows among commanders. In this study, we customize multiple types of agents for modular fleet to autonomously yield operation decisions with consideration of main characters in military operations.

Although literature on managing a modular fleet operation through MAS is insignificant, attention has been received in other areas such as manufacturing operations (*Reaidy et al.*, 2006; *Anosike and Zhang*, 2009; *He et al.*, 2014) and supply chain management (*Julka et al.*, 2002; *Giannakis and Louis*, 2011; *Meng et al.*, 2017). According to the relationship among agents, the frameworks of MAS can be broadly classified into three categories: hierarchical master/slave relationships (*Jones and McLean*, 1986), heterarchical cooperation (*Duffie and Prabhu*, 1994; *Maione and Naso*, 2001) and hybrid framework (*Ryu and Jung*, 2003). In 2009, an agent-based model (*Anosike and Zhang*, 2009) was applied to the dynamically integrated manufacturing systems (DIMS), which consists of a modeling and planning layer, a process flow layer and a simulation layer, to make planning and machine control decisions to-

gether with system reconfiguration and restructure. In 1998, Swaminathan (*Swaminathan et al.*, 1998) presented a high fidelity model representing supply chain dynamics with implementation of ABM. The agents are classified into structural elements and control elements. Structural elements contain production agent to manage the inventory; transportation agent to relocate the product from one agent to another. The control elements are used to manage inventory stocks, forecast demands, control material and information flow. Previous studies show the capacity of ABM in decision making, especially in the complex and time-varying system.

However, most of previous research focused on a single decision maker who operates (play a game) against environment. However, unpredictability from the enemy's reaction is essential and non-negligible in every form of warfare, which leads to an inability to forecast the outcome of actions or weakly perceived causal links between the events (*Lynch*, 2015). Further more, smart system and artificial intelligence are playing an ever-increasing role in our daily lives. This trend does not spare military operations. Autonomous vehicles, especially Unmanned Aerial Vehicles, has been widely used to assist military operation (*Landa*, 1991; *Jose and Zhuang*, 2013; *Evers et al.*, 2014). With no surprise, artificial intelligence will play a significant role in management of a large-scale fleet of autonomous vehicles in the near future. Given autonomous decision making system, the goal is to emphasize the synergy between modularity and autonomy by performing an attacker-defender game between conventional fleet and modular fleet.

The use of games in modeling the relationship between an attacker and a defender has a long history starting with the work of Dresher (*Dresher*, 1961). The variety of applications and research relate to issues in military OR and defense studies is rich (*Kardes and Hall*, 2005; *Hausken and Levitin*, 2009; *Zhuang et al.*, 2010; *Paulson et al.*, 2016). There are also several studies of attacker-defender game considering resource dependent strategies. Power (*Powell*, 2007) used game theoretical approach

for analyzing the defender’s allocation decisions against a strategic adversary, i.e., terrorist group, with stochastic target. Given the distribution of enemy’s behavior, they derived a pure strategy for defender that leads to a Bayesian Nash equilibria. Hausken and Zhuang (*Hausken and Zhuang*, 2011) considered a multi-period game where the defender can allocate resources to defend its own resources or attack enemy’s resources. Similarly, the attacker can also determine the use of their resources for attacking defender or protecting itself. They proved a strategy pair that leads to a subgame perfect Nash equilibrium and revealed a correlation between adopted strategy and enemy’s resources.

Game theory related techniques are popular in the existing literatures to seek the equilibrium in resource dependent attacker-defender game. However, these applications mainly focus on single period games or repeated games where the significant information from previous periods are ignored. Further more, strong assumptions on previous approach, i.e., single resource type, sequential move, perfect information on enemy, etc., also make the previous research cannot reflect the operation in real-world military mission, where the demands and environment are unpredictable (*Shinkman*, 2014; *Lynch*, 2015; *Xu et al.*, 2016). Furthermore, the recent study also prove that the performance of modular fleet is heavily influenced by the optimality of operation decisions (*Li and Epureanu*, 2018, 2017a). It is hard to summarize all the strategy pairs according to the optimization results, which makes the proof of equilibrium intractable. A new method is required to investigate the tactical advantages brought from fleet modularity.

In this study, an intelligent agent based model (*Adhitya et al.*, 2007; *Yu et al.*, 2009; *Onggo and Karatas*, 2016) is created to imitate the human-like decision making process. Optimization techniques and artificial intelligence is combined to enable each player (fleet) real-timely make decisions based on experience and optimize their decisions accordingly. By selecting one player as modular fleet and another as con-

ventional fleet, tactical advantages from fleet modularity are revealed by simulating an attacker-defender game in diverse mission scenarios.

2.4 Reconfigurable Manufacturing System

Flexibility in the RMS structure not only enhances the RMS responsiveness through upgradable capacity and modifiable functionality, but also provides the opportunity to boost the utility of machines by using the exact functionality and capacity once needed (*Koren et al.*, 2017). To dynamically match RMS configurations to demands, efficient and effective real-time management strategies are indispensable for RMS to guide system reconfiguration.

There is a rich literature on RMS planning, including process (*Bensmaine et al.*, 2014; *Azab and ElMaraghy*, 2007), capacity (*Asl and Ulsoy*, 2003; *Ceryan and Koren*, 2009), and configuration planning (*Makssoud et al.*, 2013; *Spicer and Carlo*, 2007). RMS configuration planning requires a detailed consideration of operational actions. Wang and Koren (*Wang and Koren*, 2012) investigated capacity planning by adjusting the RMS configuration to optimize the system throughput. Youssef et al. (*Youssef and ElMaraghy*, 2007) proposed an optimal configuration selection strategy to customize RMS configurations to different demand scenarios by optimizing the similarity between the adjacent configurations to increase the system utilization.

Previous studies that consider the stochastic nature of market demands assume that the distribution of demand over time is deterministic and given before investment and planning. However, emerging demands in personalized manufacturing and remanufacturing increase the difficulty of demand forecasting and the diversity of products, which limit the application of existing RMS solutions. Such diverse and rapidly fluctuating demands require new smart solutions in RMS with real-time decision-making that is rarely discussed in literature. In the present study, an autonomous RMS is proposed that synergistically integrates modularity and autonomy interpreting manufac-

turing lines as groups of manufacturing machines which share resources and perform self-reconfiguration. The autonomy of individual RMS enables the decentralization of their management. Thus, in addition to the characteristics of existing RMS solutions (*Koren et al.*, 1999), the proposed approach possesses two additional characteristics, namely autonomy and teaming (decentralization). Hence, a real-time decentralized management framework is introduced, where reconfigurable machines are grouped to form reconfigurable manufacturing lines. An agent-based decision framework is also proposed to manage the teaming of lines that can share a limited number of machines or modules, as well as performing autonomous reconfiguration as often as needed.

CHAPTER III

Model Predictive Control

3.1 Problem Formation

Modular vehicles are vehicles with interchangeable substantial components also known as modules. The objective of a military operation is to satisfy requirements from each battlefield by forming vehicle convoys at designated places in a timely fashion. Considering a main base and several spatially distributed battlefields, each mission requires a group of vehicles of different types, i.e., convoys, to operate starting at a certain time until finished. The requirements for each convoy depend on the demand type and size. We quantify these requirements into material capacity, personnel capacity and firepower. The decision maker needs to determine the convoys for each battlefield, and adjust their decisions in real-time in order to react to the changes in field demands. The base provides the place and infrastructures for modular vehicle assembly, disassembly and reconfiguration (ADR), as well as module and vehicle storage. A layout of the fleet operation system is illustrated in Fig. 3.1.

For conventional fleet, assembled vehicles are the only resources to be supplied and the only decisions made are the number of vehicles to be supplied and dispatched. For modular fleet, the on-base ADR actions enable the transition between vehicles and modules, which make the modules act as the only necessary resources that need to be resupplied. Such characteristics also render the management of the fleet opera-

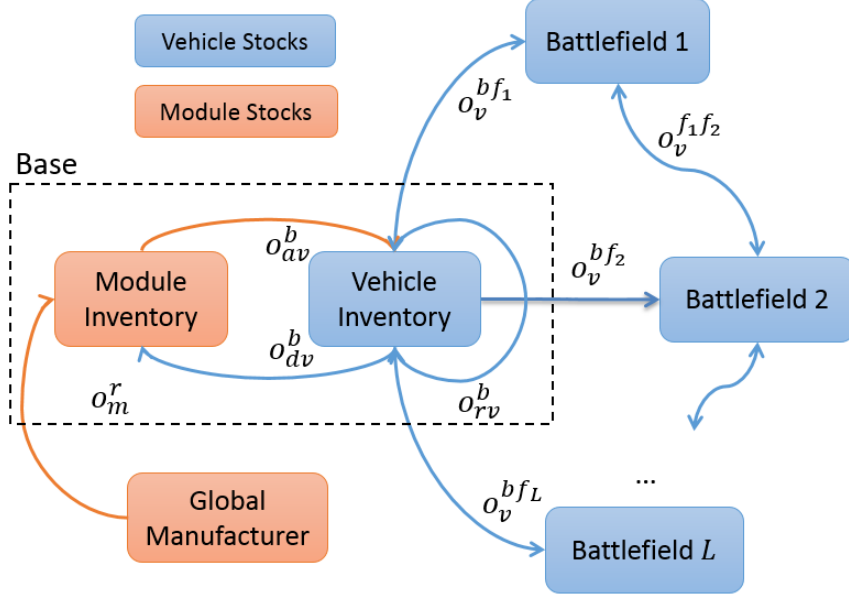


Figure 3.1: Layout of fleet operation system

tions more complex because modules must be assembled into vehicles before fulfilling the attribute demands from battlefields. Time delays from ADR actions, resource resupply, and vehicle dispatches that are considered in this study also increase the difficulty in operating the fleet between base and battlefields.

Researchers have proposed several ways to operate a modular fleet under deterministic field demands (*Bayrak et al.*, 2018; *Li and Epureanu*, 2018), but the management of a modular fleet under a highly uncertain environment is still incomplete. To perform an analysis of the robustness and adaptability to demands disturbances and noise, we formulated the operation management as an MPC problem to determine the real-time system responses through repeated system status forecasting.

3.1.0.1 Dynamics of Demand

The high uncertainty of the combat environment leads to disturbances and noise in the demand attributes. Demand disturbances might come from sudden events, i.e., unexpected assault from enemy. In that case, these disturbances can be modeled as

an impulse signal or ramp signal according to the strike type. In addition, we account for additional demand attributes that can be triggered by delayed demand attributes. The triggered demand attributes from field i $\mathbf{r}_t^{f_i}$ are calculated by

$$\mathbf{r}_t^{f_i}(t+1) = \mathbf{T} \mathbf{e}^{f_i}(t), \quad (3.1)$$

$$\mathbf{T} = \begin{bmatrix} T_{11} & T_{21} & \dots & T_{Q1} \\ T_{12} & T_{22} & \dots & T_{Q2} \\ \dots & \dots & \dots & \dots \\ T_{1Q} & T_{2Q} & \dots & T_{QQ} \end{bmatrix}, \quad (3.2)$$

where, \mathbf{T} is a demand trigger matrix. Because of the high unpredictability during fleet operation, the demand trigger matrix \mathbf{T} is assumed to be unknown to the decision makers. Once the extra attributes are triggered, the optimizer only assumes that these attributes will remain constant and last for a certain time. Thus, we can define the overall future demands as

$$\begin{aligned} \underline{\mathbf{r}}_{t+1} = & [\mathbf{r}_s(t+1), \mathbf{r}_s(t+2), \mathbf{r}_s(t+3), \dots, \mathbf{r}_s(t+t_p)]^T \\ & + \mathbf{d}(t+1) + \mathbf{n}(t+1) + \mathbf{r}_t(t+1), \end{aligned} \quad (3.3)$$

where $\mathbf{d}(t)$ is the demand disturbance received at time t , $\mathbf{n}(t)$ is the noise from the prediction at time t , which originates from the stochasticity in operation actions. Note that, given enough pairs of delayed attributes and triggered attributes, we can also extract the trigger matrix through a Kalman observer. Machine learning techniques can also be implemented when the trigger mechanism cannot be modeled as linear.

3.2 Model Predictive Control

In this section, we present the state space model that represents the modular fleet operation system, and a closed loop to monitor and control the fleet performance.

First, the discrete time state space model is introduced with consideration of time delays from certain actions. Next, we present a method to forecast the system states and a cost function to find the optimal future operation decisions that can minimize the total operational costs. Finally, we present MPC method to manage the fleet operation in real-time under an uncertain environment.

Assume there are L different battlefields, where each battlefield receives demand attributes of Q types at each time. We consider M types of modular vehicles constructed by N types of modules. At time t , the status of the system is described by the number of vehicle stocks $\mathbf{I}_v^b(t)$ and module stocks $\mathbf{I}_m^b(t)$ at the base, and the number of vehicles at field i , $\mathbf{I}_v^{f_i}(t)$, namely

$$\mathbf{I}_v^{b/f_i}(t) = [I_{v_1}^{b/f_i}(t), I_{v_2}^{b/f_i}(t), I_{v_3}^{b/f_i}(t), \dots, I_{v_M}^{b/f_i}(t)]^T, \quad (3.4)$$

$$\mathbf{I}_m^b(t) = [I_{m_1}^b(t), I_{m_2}^b(t), I_{m_3}^b(t), \dots, I_{m_N}^b(t)], \quad (3.5)$$

$$\mathbf{I}(t) = [\mathbf{I}_v^b(t), \mathbf{I}_m^b(t), \mathbf{I}_v^{f_1}(t), \mathbf{I}_v^{f_2}(t), \dots, \mathbf{I}_v^{f_L}(t)]. \quad (3.6)$$

As time delays exist in the system, the decisions placed at the current time t only take effect after a certain time delay τ . Thus, current decisions influence system states in the future. In order to track all system information, the current states also need to cover the future inventory status, in which all the current decisions will be finalized. Thus,

$$\mathbf{s}(t) = [\mathbf{I}_t(t), \mathbf{I}_{t+1}(t), \mathbf{I}_{t+2}(t), \mathbf{I}_{t+3}(t) \dots \mathbf{I}_{t+\tau_{max}}(t)]^T, \quad (3.7)$$

where, $\mathbf{I}_{t+\tau}(t)$ represents the expected system status at time $t + \tau$ that is received at time t . During the fleet operation, the actions to be determined are: a) the number of vehicles to be transported between battlefields and base, $\mathbf{o}_v^{f_i f_j}(t)$; b) the number of modules to be supplied from global manufacturer to base, $\mathbf{o}_m^b(t)$; c) ADR action orders to be placed, $\mathbf{o}_a^b(t), \mathbf{o}_d^b(t), \mathbf{o}_r^b(t)$. We stack all the operation actions for a single

time instant into a vector, $\mathbf{o}(t)$.

According to the dynamic model created for the centralized control of modular fleet operations [8], we also create a dynamic model for this mission scenario and implement this model for system forecasting.

For the number of vehicles of type k at the base

$$\begin{aligned} I_{v_k}^b(t+1) = & I_{v_k}^b(t) - o_{dv_k}^b(t) + o_{av_k}^b(t - \tau_{av_k}) - \sum_{k' \neq k} o_{v_k v_{k'}}^b(t) \\ & + \sum_{k' \neq k} o_{v_{k'} v_k}^{b_i}(t - \tau_{v_{k'} v_k}) - \sum_i o_{v_k}^{bf_i}(t) + \sum_i o_{v_k}^{f_i b}(t - \tau_{f_i b}). \end{aligned} \quad (3.8)$$

For the number of modules of type p at the base

$$\begin{aligned} I_{m_p}^b(t+1) = & I_{m_p}^b(t) - \sum_k m_{v_k m_p} o_{av_k}^b(t) + \sum_k m_{v_k m_p} o_{dv_k}^b(t - \tau_{dv_k}) \\ & + o_{m_p}^r(t - \tau_r) - \sum_k \sum_{k' \neq k} (m_{v_k m_p} - m_{v_{k'} m_p})^+ o_{v_k v_{k'}}^b(t) \\ & + \sum_k \sum_{k' \neq k} (m_{v_k m_p} - m_{v_{k'} m_p})^+ o_{v_k v_{k'}}^b(t - \tau_{v_k v_{k'}}). \end{aligned} \quad (3.9)$$

For the number of vehicles of type k at battlefield i

$$I_{v_k}^{f_i}(t+1) = I_{v_k}^{f_i}(t) - \sum_{x \neq f_i} o_{v_k}^{f_i x}(t) + \sum_{x \neq f_i} o_{v_k}^{x f_i}(t - \tau_{x f_i}). \quad (3.10)$$

According to the dynamic Eqns. (3.8), (3.9), and (3.10), the current states are only influenced by previously-determined action. This indicates that the current actions change future states. We create input matrices $\mathbf{B}(\tau)$, which connect the current actions at time t to states at a later time $t + \tau$. The outputs of system are the attributes at battlefield i , which are composed of the attributes $a^{f_i}(t)$ at battlefield i , and the outputs of system as $\mathbf{y}(t)$, namely

$$\mathbf{a}^{f_i}(t) = [a_1^{f_i}(t), a_2^{f_i}(t), a_3^{f_i}(t), \dots, a_Q^{f_i}(t)]^T, \quad (3.11)$$

$$\mathbf{y}(t) = [\mathbf{a}^{f_1}(t), \mathbf{a}^{f_2}(t), \mathbf{a}^{f_3}(t), \dots, \mathbf{a}^{f_L}(t)]^T. \quad (3.12)$$

We can then write the state space model as

$$\mathbf{s}(t+1) = \mathbf{A}\mathbf{s}(t) + \mathbf{B}\mathbf{o}(t), \quad (3.13)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{s}(t) + \mathbf{D}\mathbf{o}(t), \quad (3.14)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{n_s \times n_s} & \mathbf{I}_{n_s \times n_s} & \mathbf{0}_{n_s \times n_s} & \dots & \mathbf{0}_{n_s \times n_s} \\ \mathbf{0}_{n_s \times n_s} & \mathbf{0}_{n_s \times n_s} & \mathbf{I}_{n_s \times n_s} & \dots & \mathbf{0}_{n_s \times n_s} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{0}_{n_s \times n_s} & \mathbf{0}_{n_s \times n_s} & \mathbf{0}_{n_s \times n_s} & \dots & \mathbf{I}_{n_s \times n_s} \\ \mathbf{0}_{n_s \times n_s} & \mathbf{0}_{n_s \times n_s} & \mathbf{0}_{n_s \times n_s} & \dots & \mathbf{I}_{n_s \times n_s} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}(1) \\ \mathbf{B}(2) \\ \mathbf{B}(3) \\ \dots \\ \mathbf{B}(\tau_{max}) \end{bmatrix}, \quad (3.15)$$

$$\mathbf{C} = [\mathbf{0}_{LQ \times (M+N)}, \begin{bmatrix} \mathbf{M}_{va} & \mathbf{0}_{M \times Q} & \dots & \mathbf{0}_{M \times Q} \\ \mathbf{0}_{M \times Q} & \mathbf{M}_{va} & \dots & \mathbf{0}_{M \times Q} \\ \dots & \dots & \dots & \dots \\ \mathbf{0}_{M \times Q} & \mathbf{0}_{M \times Q} & \dots & \mathbf{M}_{va} \end{bmatrix}, \mathbf{0}_{LQ \times (\tau_{max}-1)n_i}], \quad (3.16)$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{0}_{LQ \times n_i} \end{bmatrix}. \quad (3.17)$$

In the decision making process, the predictions of future system states and demands are always involved. For example, given the estimated future target demands, one may want to know what are the consequences of dispatching 20 vehicles to a certain field. Compared to the classical control methodologies, e.g., PID control, MPC makes better use of future target information and historical system states. Fig. 3.2 demonstrates the designed framework of the MPC approach to manage the fleet operation.

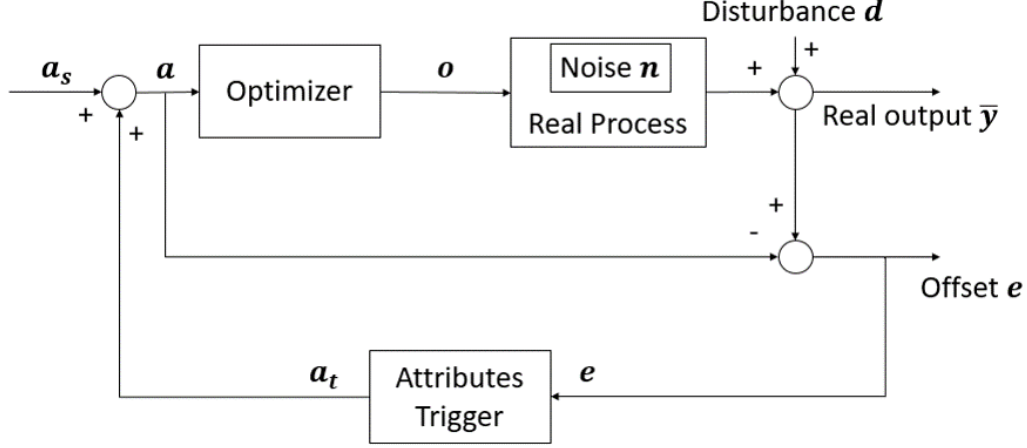


Figure 3.2: The framework of the model predictive control algorithm

3.2.1 Future State Prediction

In this study, we assume that the majority of the field demands are deterministic except for the unpredictable disturbances and noise. The received demands specify the number of attributes required at each battlefield. Thus, the goal is to control the fleet operation to match the attributes of the vehicles at all fields to the received demands. Because of time delays in the operation actions, the operation decisions made at the current time have to guarantee the matching between the resulting future system outputs and the corresponding attribute requirements. Inputs to the future state prediction are the current system states $\mathbf{s}(t)$ and the operation actions in the future $\underline{\mathbf{q}}_t$. The future system states $\underline{\mathbf{y}}_{t+1}$ are predictable by iterations, where

$$\underline{\mathbf{y}}_{t+1} = [\mathbf{y}(t+1), \mathbf{y}(t+2), \mathbf{y}(t+3), \dots, \mathbf{y}(t+t_p)]^T, \quad (3.18)$$

$$\underline{\mathbf{q}}_t = [\mathbf{o}(t), \mathbf{o}(t+1), \mathbf{o}(t+2), \dots, \mathbf{o}(t+t_p-1)]^T. \quad (3.19)$$

By iteratively substituting Eqn. (3.13) into Eqn. (3.13), we can express the $\underline{\mathbf{y}}_{t+1}$

as a function of $\underline{\mathbf{q}}_t$ and $\mathbf{s}(t)$ as

$$\underline{\mathbf{y}}_{t+1} = \mathbf{P}\mathbf{s}(t) + \mathbf{H}\underline{\mathbf{q}}_t, \quad (3.20)$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \dots \\ \mathbf{CA}^n \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{CB} & \mathbf{0}_{LQ \times n_i} & \dots & \mathbf{0}_{LQ \times n_i} \\ \mathbf{CAB} & \mathbf{CB} & \dots & \mathbf{0}_{LQ \times n_i} \\ \dots & \dots & \dots & \dots \\ \mathbf{CA}^{n-1}\mathbf{B} & \mathbf{CA}^{n-2}\mathbf{B} & \dots & \mathbf{CB} \end{bmatrix}. \quad (3.21)$$

with \mathbf{P} being the matrix that connects the future system outputs with the current system states, and \mathbf{Q} being the matrix that connects the system outputs with the future operation actions.

Operation actions are not always accomplished on time, which involves stochasticity in the matrix \mathbf{B} , and that increases the difficulty in forecasting. In this study, we focus on comparing the performance between modular fleet and conventional fleet. Thus, all operation actions are assumed to be accomplished exactly on time. In addition, the mapping between vehicles and modules remains fixed over time, which makes \mathbf{P} , \mathbf{Q} be constant matrices. In the next section, we perform a study to investigate the model performance in reaction to the noise from system state prediction and demand attributes.

3.2.2 Cost Function

We modeled this MPC problem as a trajectory tracking problem, where we minimize the difference between system outputs and attribute requirements. The difference between this problem and the traditional trajectory tracking problem is in the different importance of matching errors. In our study, a negative matching error corresponds to insufficient attributes at the fields and a positive matching error represents redundant attributes. In operation, insufficient attributes lead to much worse outcomes than the redundant supplies, which requires a distinct weighting factor.

Therefore, we design an optimizer to find the actions for the current time and future time that will minimize the total operating costs during the horizon of prediction t_p . The costs of particular interest are from module supplies, vehicle transportation, ADR actions and attribute insufficiency. The cost function is designed as

$$J = \underline{\mathbf{c}}_b(\underline{\mathbf{r}}_{t+1} - \underline{\mathbf{y}}_{t+1}) + \underline{\mathbf{c}}_o \underline{\mathbf{o}}_t, \quad (3.22)$$

where $\underline{\mathbf{c}}_b$ represents insufficiency cost, $\underline{\mathbf{c}}_o$ stores all the costs for actions, c_{av_k} , c_{dv_k} , $c_{v_kv'_k}$, c_{m_p} . By substitute Eqn. (20) into Eqn. (22), we can formulate the fleet operation problem as

$$\text{minimize} \quad \underline{\mathbf{c}}_b(\underline{\mathbf{r}}_{t+1} - \mathbf{P}\mathbf{s}(t)) + (\underline{\mathbf{c}}_o - \underline{\mathbf{c}}_b\mathbf{H}) \underline{\mathbf{o}}_t \quad (3.23)$$

$$\text{subject to} \quad \underline{\mathbf{s}}_{t+1} \geq 0 \quad (3.24)$$

$$\underline{\mathbf{o}}_t \geq 0 \quad (3.25)$$

$$\begin{bmatrix} \mathbf{I}_{M \times (M+1)} \\ \mathbf{0} \end{bmatrix} \mathbf{o}(t) \leq \bar{\mathbf{P}}, \quad \forall t \quad (3.26)$$

$$\underline{\mathbf{y}}_{t+1} \leq \underline{\mathbf{r}}_{k+1} \quad (3.27)$$

Our model is designed to solve modular fleet operations management and design problems, as well as various operation problems. In the following section, we will apply our model to optimally schedule the actions in modular fleet operations for randomly generated military mission scenarios. We then also apply Monte Carlo analysis to solve the module design problem with constrained design costs.

3.3 Case Study 1

We start from a simplified operation scenario with three pieces of infrastructures on the field: one base and two camps. Each piece of infrastructure is connected. In other words, the vehicles can be transported between any two of them. We will also consider two types of demand attributes, e.g., firepower and material capacity. The arrival demand for each camp is modeled as a discrete event, with the rate per hour as $\mu^{c_1} = \frac{1}{6}$, $\mu^{c_2} = \frac{1}{4}$. The demands are randomly generated from two sources, e.g., combat and transportation. Combat demands only require the firepower attributes from the dispatched convoy, $a_{s1}^c \sim N(10, 4)$. Meanwhile, transportation demands require both personnel capacity and firepower (for scout), $a_{s2}^c \sim N(10, 4)$, $a_{s1}^c = a_{s2}^c/2$.

Correspondingly, two types of modular vehicles are created, where type 1 is a combat vehicle with high firepower, and type 2 is a truck with large material capacity. There are four types of modules that are considered, where modules in types 1 and 2 have main attributes in power and mobility, type 3 is a weapon module, and a type 4 is a payload module. The composition and attributes of modular vehicles are shown in Tab. 3.1. We have also created two types of conventional vehicles with same attributes on them, and compare their performance in a randomly generated 30-day mission scenario in terms of fleet readiness and operation costs. The system parameter settings are shown in Tab. 3.2. We have supplied a certain amount of resources to the bases before the mission begins. In particular, 10 vehicles of both types are supplied for the conventional fleet.

3.3.1 Operation Management

As for the modular fleet, 20 modules of types 1, 2, 3, and 10 modules in type 4 are supplied. We use the lateness of demand attributes during the mission as the metric to represent fleet readiness. All the decisions are computed by a *MATLAB* linear programming solver, which takes less than 2 seconds. A fleet performance

Table 3.1: Modular vehicle compositions

Modular Vehicle	Fire Power	Capacity	Module Type			
			1	2	3	4
Type 1	4	1	1	1	1	0
Type 2	0	6	1	1	0	1

Table 3.2: System parameter settings

$c_{av_1} = 4, c_{av_2} = 3$	$\tau_{av_1} = 4, \tau_{av_2} = 3$
$c_{dv_1} = 2, c_{dv_2} = 1$	$\tau_{dv_1} = 2, \tau_{dv_2} = 1$
$c_{v_1v_2} = 2, c_{v_2v_1} = 1$	$\tau_{v_1v_2} = 2, \tau_{v_2v_1} = 1$
$c_{b_1c_1} = c_{c_1b_1} = 2$	$\tau_{b_1c_1} = \tau_{c_1b_1} = 5$
$c_{b_1c_2} = c_{c_2b_1} = 2$	$\tau_{b_1c_2} = \tau_{c_2b_1} = 5$
$c_{c_1c_2} = c_{c_2c_1} = 2$	$\tau_{c_1c_2} = \tau_{c_2c_1} = 5$
$c_h^{b_1} = 0.1$	$\tau_m = 6$
$c_h^{c_1} = c_h^{c_2} = 5$	$t_p = 24$
$c_b^{c_1} = c_b^{c_2} = 10^4$	$t_o = 24$
$p_{max} = 10$	$t_{max} = 5$

comparison is presented in Tab. 3.3.

Table 3.3: Fleet readiness comparison

Fleet	Camp 1		Camp 2	
	Fire Power	Capacity	Fire Power	Capacity
Mod.	47.6	24.4	5.2	0
Conv.	113.2	0	96.2	0

As the conventional vehicles have identical attributes as modular vehicles, we can regard the conventional vehicles as a combination of conventional modules with the same attributes as modules in modular vehicles. Based on the results, a modular fleet only has 10 more initial weapon module supplies, but the total number of delayed attributes is reduced by more than a half. In order to explain these results, we compare the total fleet operational actions in Tab. 3.4. We also present the ADR action history of the modular fleet in Fig. 3.3.

Table 3.4: Total actions comparison between fleets

Fleet	$b_1 \leftrightarrow c_1$	$b_1 \leftrightarrow c_2$	$c_1 \leftrightarrow c_2$	o_{av}	o_{dv}	o_{rv}
Mod.	366	539	95	63	49	93
Conv.	306	461	124	N/A	N/A	N/A

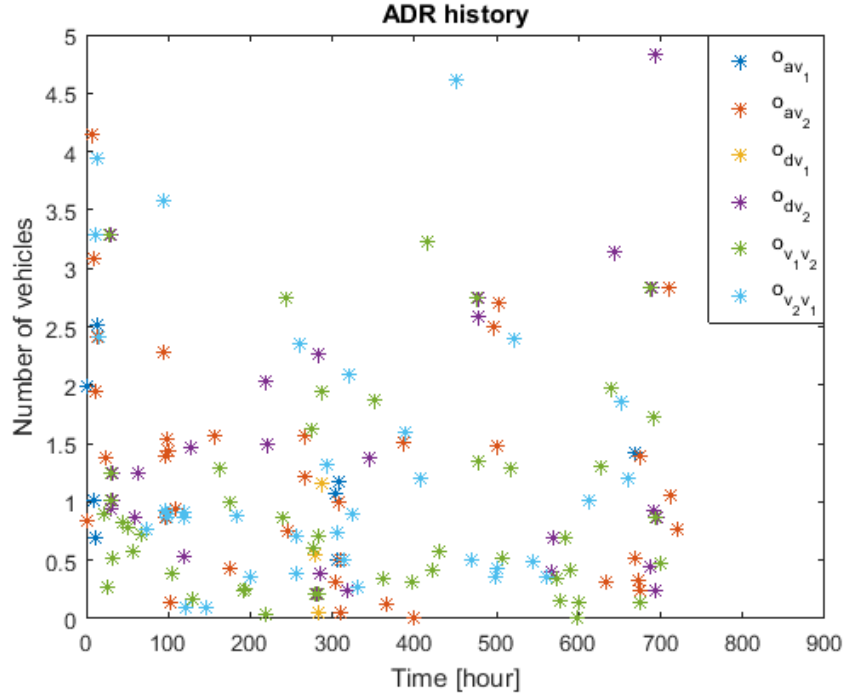


Figure 3.3: ADR actions history during modular fleet operation

Compared with a conventional fleet, a modular fleet facilitates more transportation between a base and one of the camps, but less between camps. In addition to the concern of reducing inventory holding costs, more callback actions originate from the requirements of fleet reconfiguration. According to the performance of conventional fleet, the vehicles supplied are insufficient to satisfy all the field demands on time. Therefore, the extra operation flexibility gained from fleet modularity enables the modular fleet to process fleet reconfiguration to increase the utility rate of supplied resources, and match vehicle composition to the demand attributes.

We also compare the total fleet operation costs in Tab. 3.5. The costs we are particularly interested in are inventory holding costs, ADR action costs, transportation costs, and delay costs. The results show that for a modular fleet, transportation and ADR action costs are higher. However, these actions lead to noteworthy reductions in total holding and delayed demand attribute costs. These results provide a typical example of how the flexibility gained from modularity brings about a significant readiness and utility rate boost in fleet operations.

Table 3.5: Operation cost comparison

Fleet	Holding	ADR	Transportation	Delay
Mod.	1064	313	5352	77254
Conv.	1900	0	4372	219820

Minimum ADR Action Capacity

All the ADR actions are assumed to be processed in a mixed-model production line, which has proven to be beneficial in increasing the production efficiency and reducing the product changeover time. We assume that production capacity is proportional to the number of production lines. Thus, the optimal ADR action capacity

is an important metric to determine the spaces of bases and the amount of production facilities. We present a parametric study in ADR action capacity through 20 randomly generated mission scenarios, and show its influence on fleet readiness in Fig. 3.4.

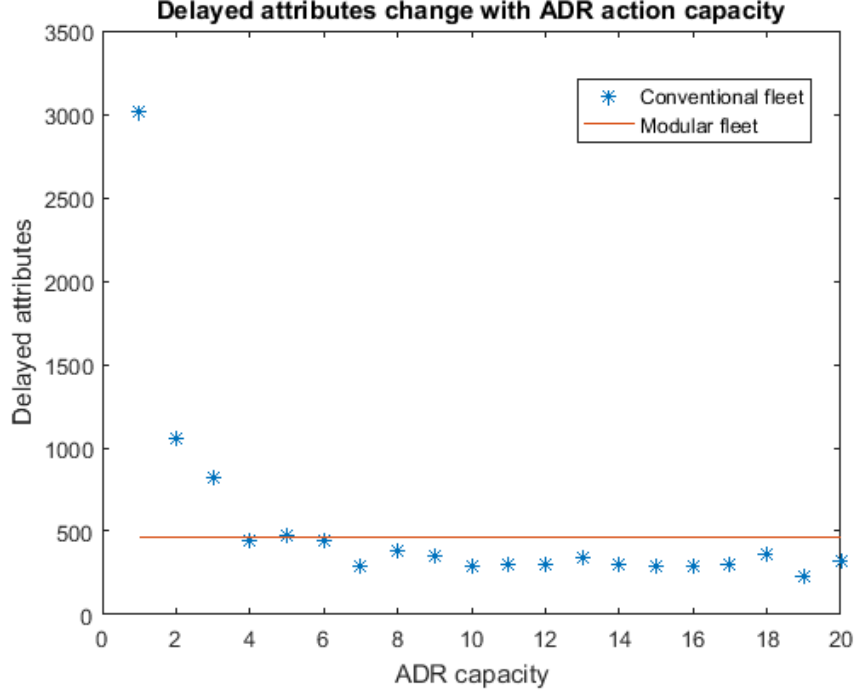


Figure 3.4: Fleet readiness varies with changing capacity

Unsurprisingly, an improvement in fleet readiness occurs with increasing ADR action capacity. As shown in Fig. 3.4, $p_{max} = 5$ is the threshold of action capacity that a modular fleet performance surpasses that of a conventional fleet. $p_{max} = 10$ is a saturation point, beyond which, further capacity enlargement only brings an inconspicuous reduction in delayed attributes. The remaining delayed attributes come from the stochasticity in demand trigger and time delays in the system, i.e., with pure module supplies. It is impossible to have vehicles ready in camps at the beginning of the mission. Creating a preparation period before the mission or supplying completed modular vehicles to bases can further reduce the delayed attributes.

3.3.2 Modular Vehicle Design

We apply Monte Carlo analysis to find the optimal vehicle attribute allocations. We assume the design costs are proportional to the allocated attributes. Taking modular vehicles consisting of three modules as an example again, given that modules in type 1 and type 2 are power and mobility modules, it will cost more to assign the attributes of capacity and firepower to them. Similarly, it will also be expensive to add firepower attributes on a capacity module (type 4) and vice versa. Based on this fact, we assigned $c_{m_1a_{fp}} = 6, c_{m_2a_{fp}} = 6, c_{m_3a_{fp}} = 5, c_{m_4a_{fp}} = 1$ and $c_{m_1a_{cp}} = 6, c_{m_2a_{cp}} = 6, c_{m_3a_{cp}} = 1, c_{m_4a_{cp}} = 5$.

Given the limited design budget, 1500 groups of modular vehicles with diverse attributes are randomly generated subject to the constraint

$$\sum_n \sum_l c_{m_l a_l} m_{m_n a_l} \leq 15 \quad (3.28)$$

In order to proceed with the Monte-Carlo analysis, we use these 1,500 groups of modular vehicles to form a fleet of 1,500 modular vehicles. We operate them using an identical mission scenario, and record their design costs and fleet performance. In addition, a 3D surface is generated to fit the simulation results. Fig. 3.5 shows the relationship between module design and fleet readiness.

Fig. 3.6 shows the projection of the fitted surface on an x-y plane (investment of modules types 3 and 4) to intuitively show the optimal investment strategy. Based on the plot, the optimal investment strategy for this mission scenario is 3.7 for designing module in types 1 and 2, 5 for modules in type 3, and 6.3 for modules 4. It is worth noting that the design cost per attribute is low when it comes to fire power and capacity in type 3 and 4 modules respectively. Despite the costs, the budget is not completely spent on these two types of modules. These results can be explained by the module sharing effect where all the vehicles require modules in types 1 and 2.

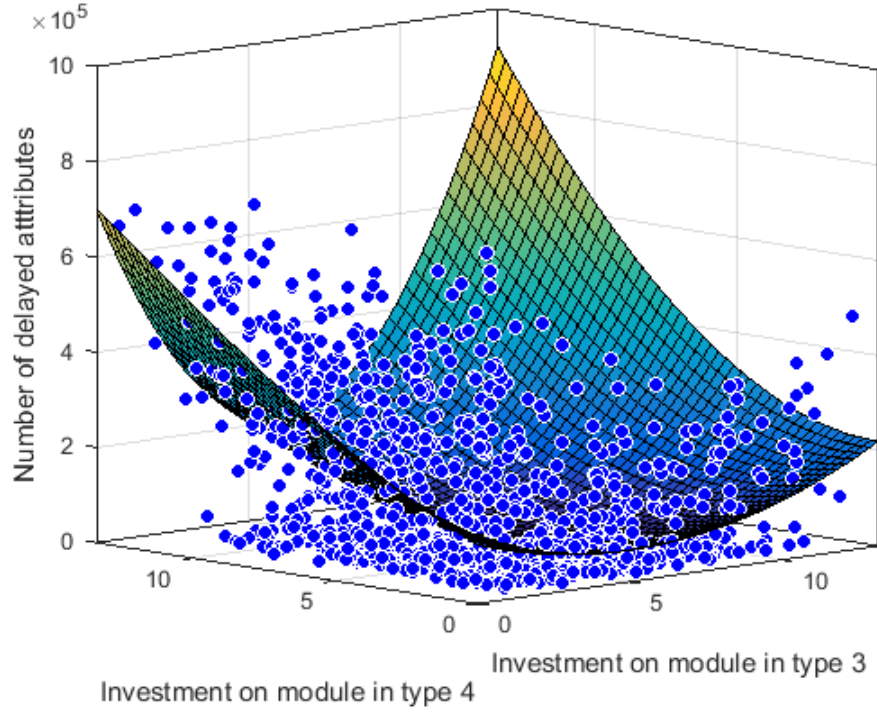


Figure 3.5: Fitted surface that captures the relationship between vehicle design and fleet readiness

In other words, the attributes from these two types of modules are shared across all types of vehicles used in this study. This benefit can be significant in an example involving a small amount of material transportation. The firepower gained from the modules in types 1 and 2 might be enough for scort, which reduces the number of combat vehicles that satisfy transportation demands.

3.4 Case Study 2

3.4.1 System Robustness and Adaptability

In this study, we discuss five types of modular vehicles which will be compared with five military vehicles. These military vehicles are M1008, HMMWV, M985, FMTV, and JERRV. A comparison between these conventional vehicles and their

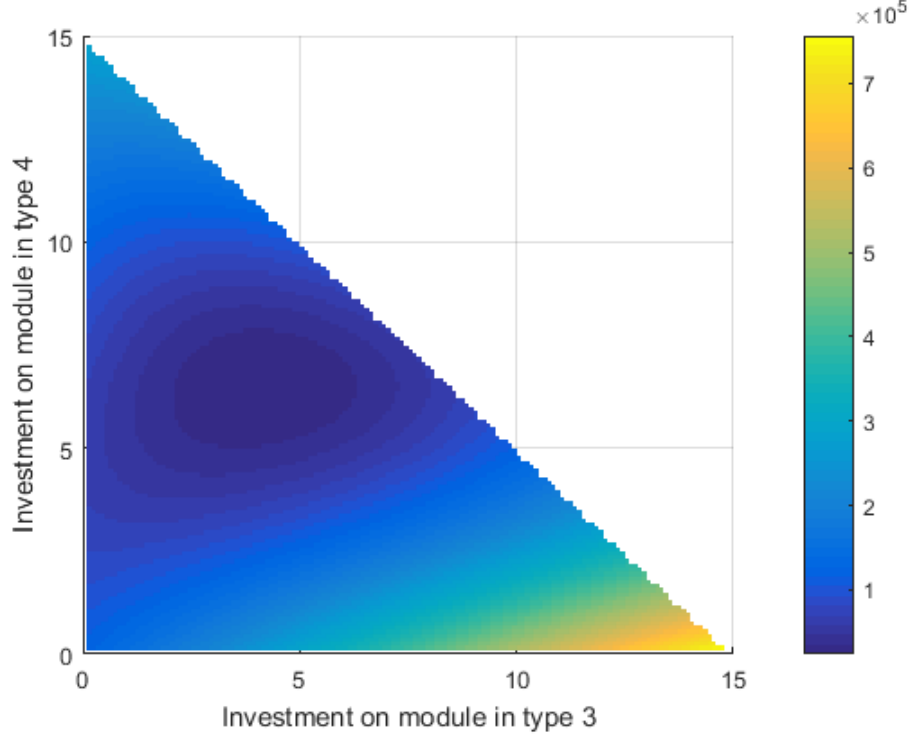


Figure 3.6: Projection of the fitted surface

corresponding modular vehicles is shown in Fig. 3.7. We quantify the vehicle capacities in terms of several attributes such as material capacity, personnel capacity, and firepower. We also assume that the modular vehicles possess identical attributes as their conventional vehicle counterparts.

The structure of the modular vehicles are summarized in Tab. 3.6, and the mapping between vehicles and their attributes is shown in Tab. 3.7. It is worth noting that the modules in the different types of modular vehicles may exhibit different types of attributes. For example, the type 4 modules in an M1008 vehicle have personnel capacity; however, this type of module also provides a significant material capacity once both of them are assembled into an FMTV-like modular vehicle.

We implemented the model mentioned in the previous section to solve the modular fleet management problem in another specified mission scenario. In the following subsections, we first find the steady state of the dynamic system for given constant

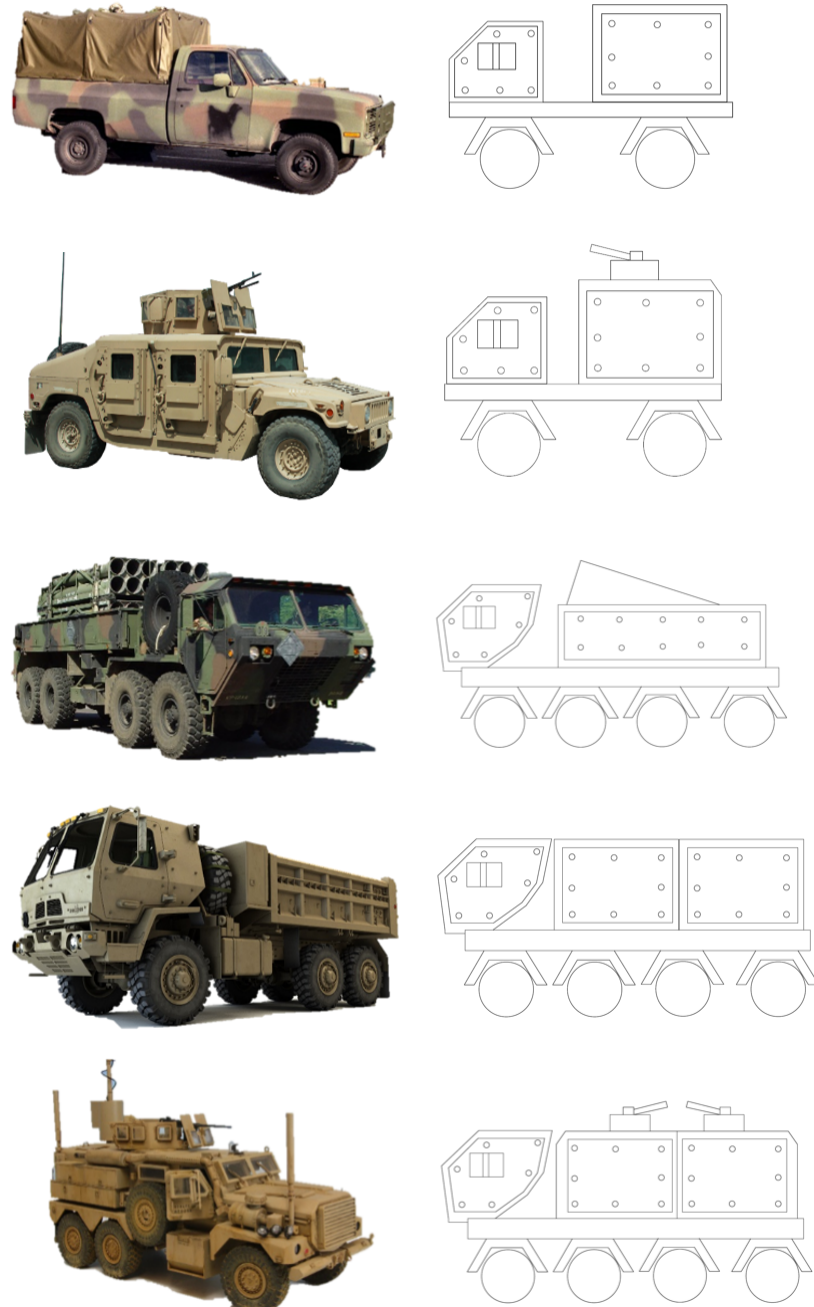


Figure 3.7: An analogy between conventional vehicles and modular vehicles

demands through MPC control. Next, we inject disturbance modeled by step function to the system after the system reaches the steady state, and then compare the system response between a conventional and modular fleet.

Table 3.6: The mapping between the modular vehicles and its modules

	Veh 1	Veh 2	Veh 3	Veh 4	Veh 5
Mod 1	1	1	0	0	1
Mod 2	0	0	1	1	0
Mod 3	1	1	2	2	2
Mod 4	1	0	0	2	0
Mod 5	0	1	0	0	2
Mod 6	0	0	1	0	0

Table 3.7: The mapping between the modular vehicles and its attributes

	Veh 1	Veh 2	Veh 3	Veh 4	Veh 5
Personnel Capacity	2	6	0	1	2
Material Capacity	4	1	0	10	5
Firepower	1	3	8	0	6

3.4.1.1 System Steady State

We implement five types of modular and conventional vehicles in our model to form modular and conventional convoys. We start from a simplified scenario, where there are three types of attributes consistently required from two battlefields, e.g., $r_{si}(t) = 600, \forall i, t$. These attribute demands are identical for both fleets, as well as the zero initial resource stocks, e.g., $\mathbf{s}(0) = \mathbf{0}$. The maximum ADR action capacity is set to 5. We also assume the time to finish the assembly of one module is 1 hour, and disassembly is 0.5 hours. The lead times for on-field transportation and resupply are 5 hours and 12 hours respectively. Based on these settings, the changes in vehicle attributes at the two fields are shown in Fig. 3.8, 3.9.

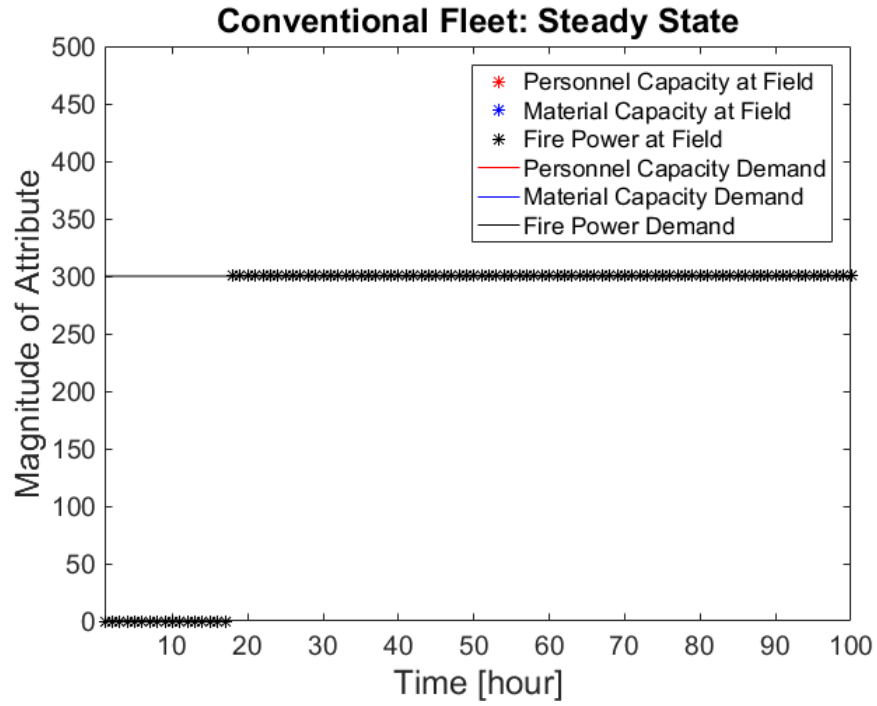


Figure 3.8: Reaction to disturbance during conventional fleet operation

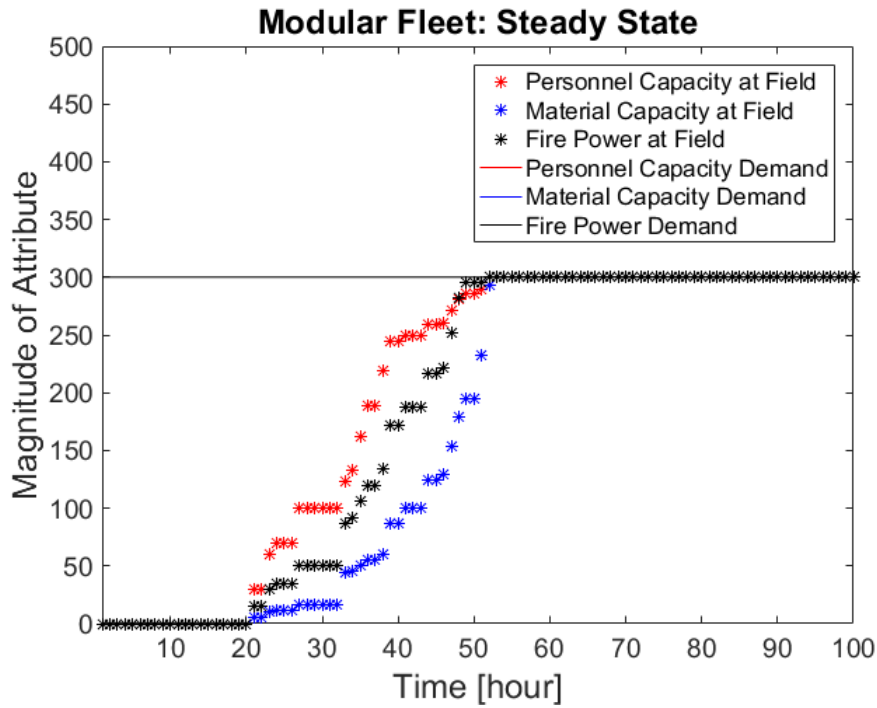


Figure 3.9: Reaction to disturbance during modular fleet

The resupply lead times for vehicles and modules are identical, which is the reason using the modular fleet reaches the steady state later than the conventional fleet. The conventional fleet only needs to wait for the newly-arrived vehicles, and dispatch them to the battlefields. However for the modular fleet, extra time is needed for ADR actions, and the constraint arising from ADR capacity makes this process even longer.

Disturbance Response

After both the conventional and modular fleets enter a steady state, we create a disturbance to the system, which is 10 unexpected fire power attribute requirement for battlefield 1, namely, $d_{a3}^{f_1} = 10$. The system responses are shown in Fig. 3.10, 3.11. These results show that a modular fleet has a better responsiveness to the impulse disturbance than conventional fleet, where the maximum attribute overshoot is around 100, which is only half of what is experienced in a conventional fleet. The resettling time for the modular fleet is also dramatically less than the conventional fleet. In addition, because vehicles always have multiple types of attributes, the sudden change in a single type of target demands induce also fluctuations of other types of demands. For example, the demand attributes in personnel and material capacity also fluctuate significantly once the disturbance in fire power is injected. Because conventional vehicles can only be transported between bases and battlefields, this in turn provides fewer choices for convoy formation.

Compared to conventional fleets, these influences are much quieter for the modular fleet because the time to accomplish on-base ADR actions is significantly shorter than the resupply lead times. Thus, modularity provides the fleet with the ability to reconfigure itself to satisfy the mission requirements. Fig. 3.12 demonstrates the records of assembly, disassembly and reconfiguration actions once the systems detect disturbances.

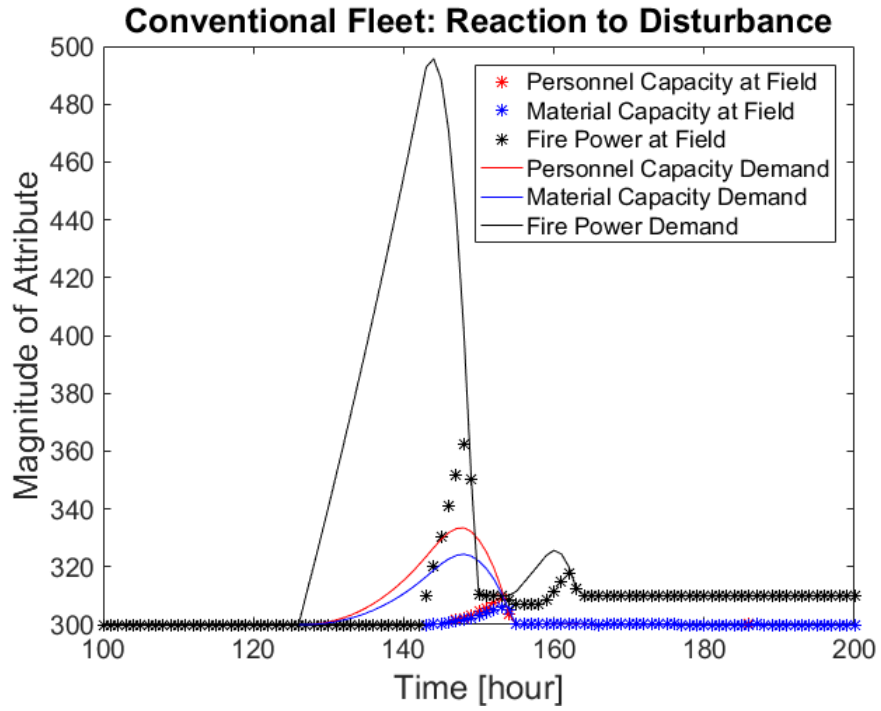


Figure 3.10: Reaction to disturbance during conventional fleet operation

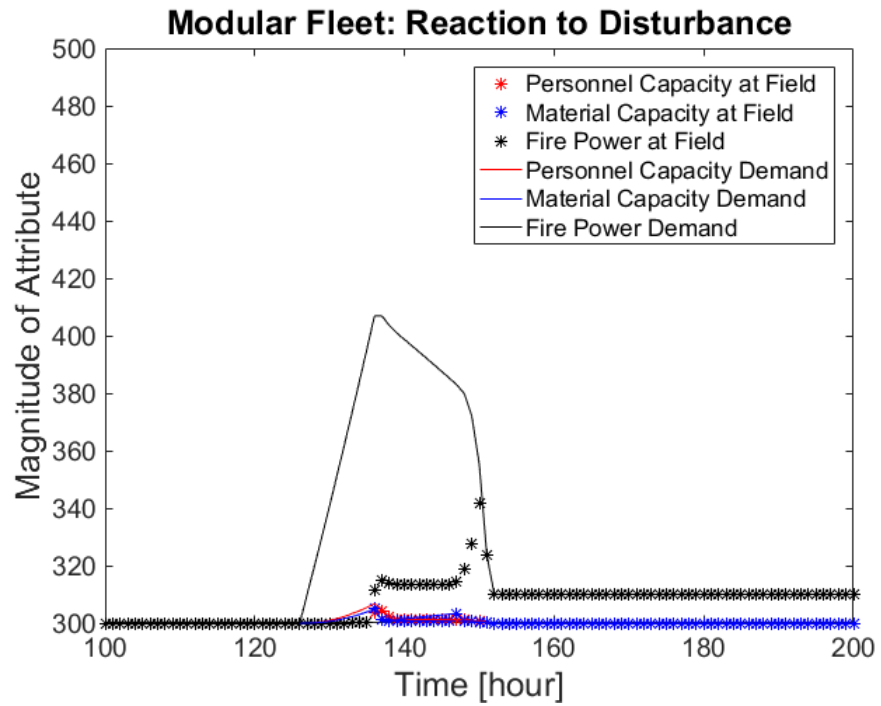


Figure 3.11: Reaction to disturbance during modular fleet operation

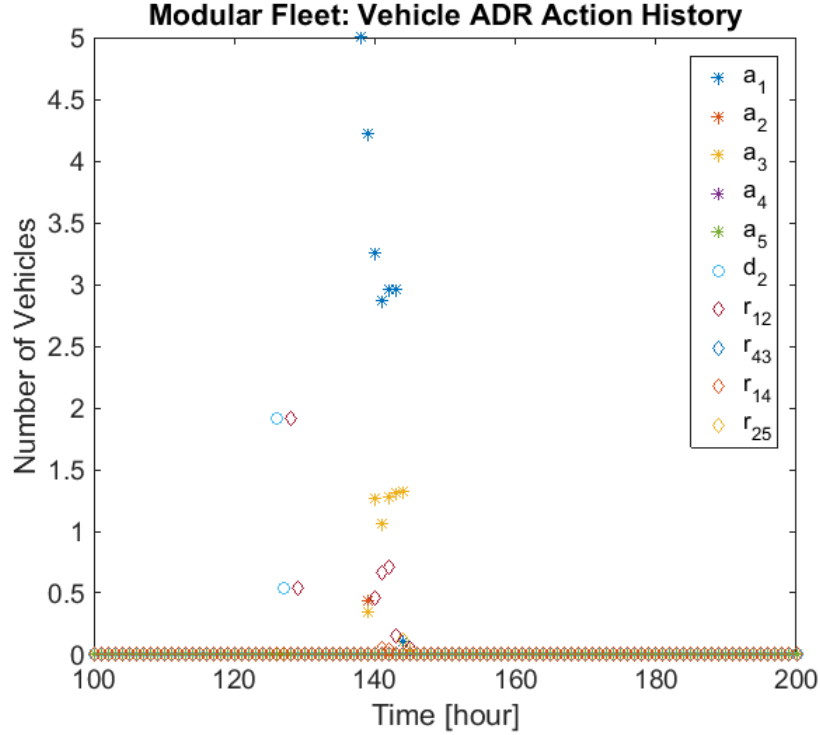


Figure 3.12: History of vehicle assembly, disassembly and reconfiguration actions

The results from Fig. 3.12 show that the modular fleet starts disassembly part of vehicles of type 4 that do not have firepower. These actions release the spare power modules and mobility modules. At the same time, the resupply orders for modules of types 1, 3, 4, and 6 are placed, which are the necessary modules for vehicles with strong fire power. These amounts of harvested modules become essential source for the reconfiguration work. Reconfiguration work is initiated after the accomplishment of disassembly jobs, which mainly covers the transition from type 2 to type 5 vehicles, and from type 4 to 3. Based on Tab. 3.7, both of these reconfiguration actions will lead to a boost in fleet fire power to react to the disturbance. On the contrary, the conventional fleet can only wait for the vehicles from resupply to satisfy the unexpected demand, and order more vehicles to compensate the triggered battlefield demands.

Noise Response

The uncertainty during the fleet operation creates errors in forecasting the behavior of the system and demand attributes. For example, the increased transportation time between the base and battlefields in bad weather conditions can lead to an additional attribute, which leads to the demand underestimations. In this study, we combine all the uncertainty factors as a noise signal which is modeled as uniformly distributed random numbers, e.g., $n_{aq}^{f_i} \sim U(0, 5), \forall q, i$. In Fig. 3.13, 3.14, the performance of two fleets are compared for an environment with noise.

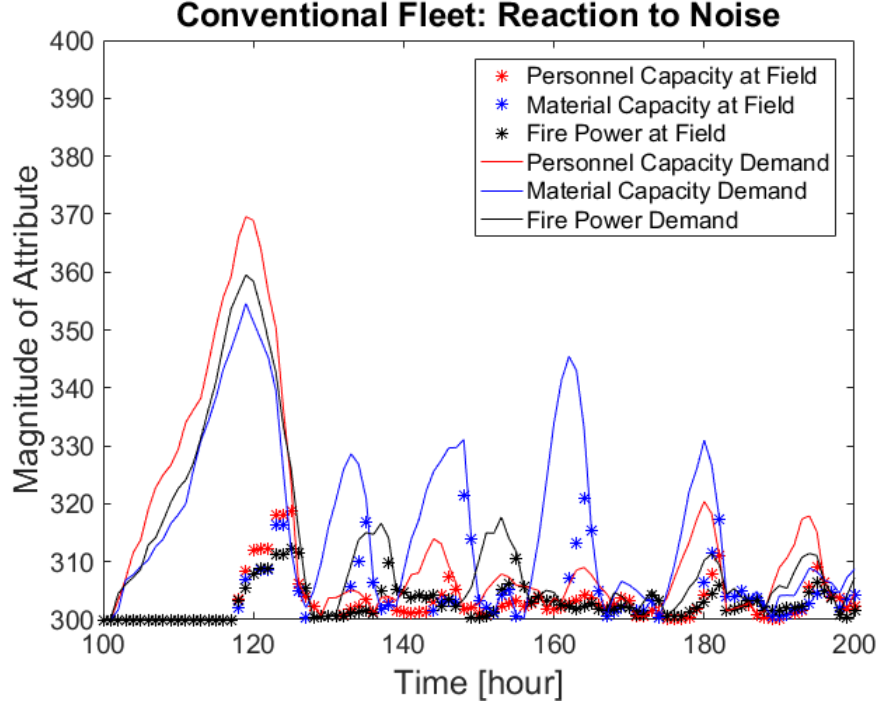


Figure 3.13: Reaction to disturbance during modular fleet and conventional fleet operation

Based on the results shown in Fig. 3.13, 3.14, the modular fleet again has a higher reaction speed than the conventional fleet, where the flexibility gained from on-base ADR actions leads to more reduction in the backlogged demand attributes than the conventional fleet. Once the noise is injected, we noticed that a lot of modular vehicles with high material capacities are reconfigured to other types of

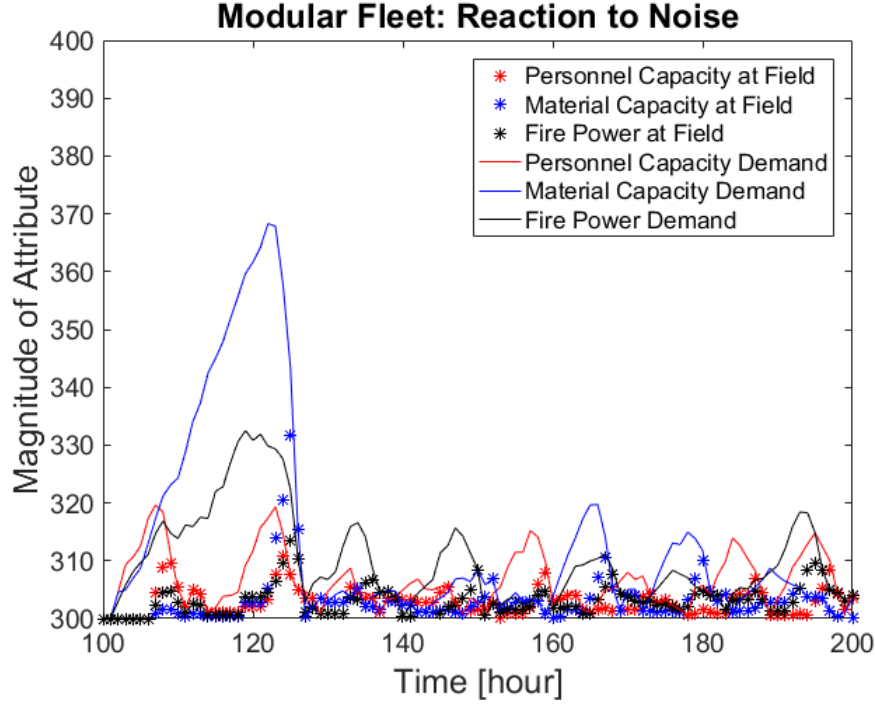


Figure 3.14: Reaction to disturbance during modular fleet and conventional fleet operation

vehicles. This reconfiguration work leads to a significant reduction in the backlogged attributes in personnel capacity and firepower without a markedly loss in material capacity attributes. After the first resupply arrives, the maximum delayed attributes in the modular fleet is less than 20, which is half compared to the conventional fleet. These observations show that ADR actions enable the modular fleet to achieve real-time fleet reconfiguration to readily react to noise from the fields rather than waiting for vehicles from returning convoys or resupplies as in the conventional fleet.

3.4.2 Stability Margin

Improper operation decisions lead to stacks of demand attributes. The system becomes unstable once its capacity cannot satisfy the stacked demands as well as its triggered demands. In this sense, the stability of the fleet operation depends on the trigger matrix, the ADR action capacity and the magnitude of the disturbance.

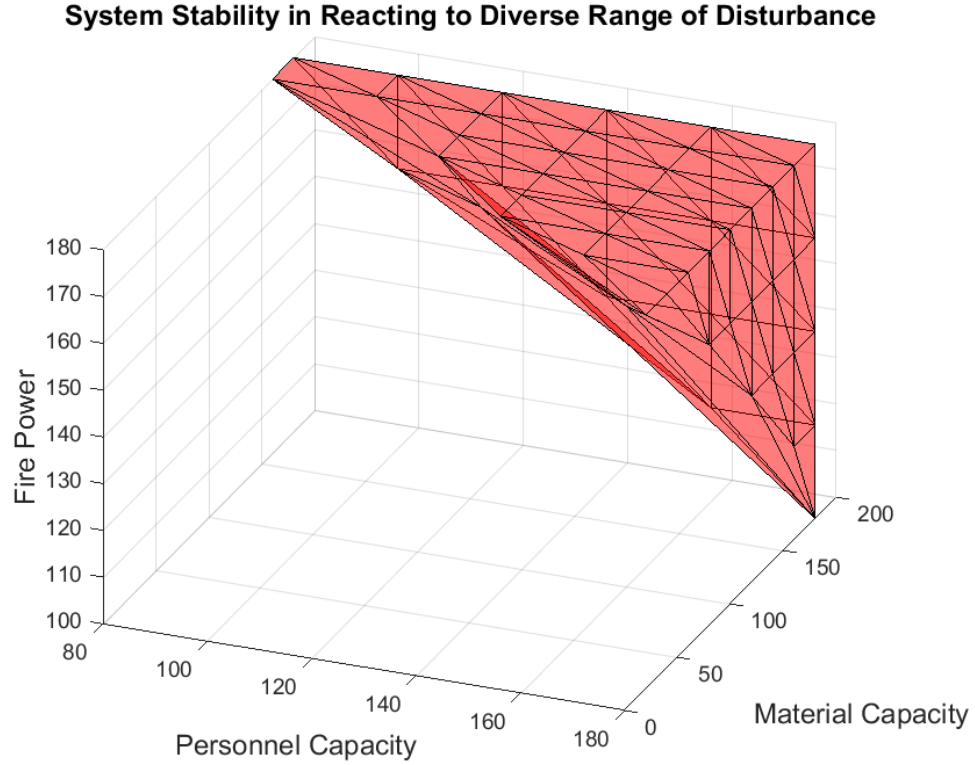


Figure 3.15: Peak values of convoy attribute changes with the various demand impulses

Research on stability and optimality of constrained MPC with a quadratic cost function has been going for many years. For example, a direct method is commonly used to analyze the stability of linear constrained systems (*Mayne et al.*, 2000). In 2010, Vesely et al. designed a robust MPC for a linear system with input constraints, and also verified the quadratic stability by using linear matrix inequalities (*Vesely et al.*, 2010). However, the distinct meanings in the matching errors of this problem disable the quadratic formulation, and also theoretical proof of system stability. In this section, we present the implementation of our model to directly compute the stability margin of the system.

The peak attributes of the total field demands increase with augmenting demand impulses, because a higher amount of impulse requires more resupplies and a longer

ADR action time, which also triggers additional demand attributes. We generated diverse combinations of impulses in the range between 0 and 200 with an interval of 20, and the computed stable and unstable regions are shown in Fig. 3.15. For this specific mission scenario, it can be observed that the system is more sensitive to the firepower disturbance than other disturbances. For example, the system is stable once the magnitude of injected fire power disturbance is below 100, where the stability limits for material capacity and personnel capacity are 20 and 80 respectively. This observation can also verify the the results in noise response, where the material capacity is the latest delayed attributes to be fulfilled. Slight convexity is also noticed in the surface, which indicates that less effort is required to overcome a combination of different types of disturbances than a single type. As for a single type demand disturbance, it is more likely that the total attribute capacity is exceeded, which incurs the resupplies and long waiting time. In contrast, for a combination of multiple types of demands attributes, we can increase the proportion of multi-purpose vehicles to satisfy the different types of disturbance simultaneously.

3.5 Stochastic Model Predictive Control

As commanders of fleet always need to react to enemy's decisions, uncertainty commonly exists in the military fleet operation, which may origin from

1. unexpected delays vehicle dispatch and return
2. damage during operation
3. enemy's unexpected strike

In the previous section, all the uncertainties are treated as disturbance, which can only be sensed a short time ahead of the mission. However, it is also possible to model the distribution of uncertainties, i.e., trigger matrix, fluctuation of inventory stocks,

vehicle damage and etc, as additive or multiplicative stochastic variables. There is no doubt that these additional information can lead to more considerable decisions. Thus, a stochastic dynamical system is firstly created to capture the estimable uncertainty during fleet operation. Next, a suitable controller is required to manage the dynamics of the stochastic system. A fundamental question about MPC is its robustness in reacting to model uncertainty and noise (*Bemporad and Morari, 1999*). To seek a less conservative solution with desired computational speed, we implement ideas of SMPC and create a chance-constrained model to generate operation decisions in real-time with considering uncertainty. Stability constraints are also created to avoid system instability.

With consideration of fluctuations of vehicle stocks \mathbf{w} and received demands \mathbf{v} at battlefields, we follow the previous work and create a stochastic dynamical system to capture the uncertainty of the system (*Ripaccioli et al., 2010; Tsao et al., 2018*). The states \mathbf{s} of system include vehicle stocks at battlefields, vehicle stocks and module stocks at the base. The outputs of system \mathbf{y} is the remaining demands measured by attributes of convoy at battlefields. The system dynamics are dominated by the following equations.

For the number of vehicles of type k at the base

$$\begin{aligned}
I_{v_k}^b(t+1) = & I_{v_k}^b(t) - o_{dv_k}^b(t) + o_{av_k}^b(t - \tau_{av_k}) - \sum_{k' \neq k} o_{v_k v_{k'}}^b(t) \\
& + \sum_{k' \neq k} o_{v_{k'} v_k}^{b_i}(t - \tau_{v_{k'} v_k}) - \sum_i o_{v_k}^{bf_i}(t) + \sum_i o_{v_k}^{f_i b}(t - \tau_{f_i b}).
\end{aligned} \tag{3.29}$$

For the number of modules of type p at the base

$$\begin{aligned}
I_{m_p}^b(t+1) &= I_{m_p}^b(t) + o_{m_p}^r(t - \tau_r) + \sum_k m_{v_k m_p} o_{dv_k}^b(t - \tau_{dv_k}) \\
&\quad - \sum_k m_{v_k m_p} o_{av_k}^b(t) - \sum_k \sum_{k' \neq k} (m_{v_k m_p} - m_{v_{k'} m_p})^+ o_{v_{k'} v_k}^b(t) \\
&\quad + \sum_k \sum_{k' \neq k} (m_{v_k m_p} - m_{v_{k'} m_p})^+ o_{v_k v_{k'}}^b(t - \tau_{v_k v_{k'}}).
\end{aligned} \tag{3.30}$$

For the number of vehicles of type k at field i

$$I_{v_k}^{f_i}(t+1) = I_{v_k}^{f_i}(t) - \sum_{x \neq f_i} o_{v_k}^{f_i x}(t) + \sum_{x \neq f_i} o_{v_k}^{x f_i}(t - \tau_{x f_i}) + w_k^{f_i}(t). \tag{3.31}$$

For the amount of remaining attributes of type q at field i

$$I_{a_q}^{f_i}(t+1) = \sum_{q'} T_{q' q}(t) I_{a_{q'}}^{f_i+}(t) - \sum_k m_{v_k a_q} I_{v_k}^{f_i}(t) + r_{a_q}^{f_i}(t) + v_q^{f_i}(t), \tag{3.32}$$

where, $I_{a_q}^{f_i+}(t)$ is the positive part of the remaining demands at time t . By denoting the negative part as $I_{a_q}^{f_i-}(t)$, we have

$$I_{a_q}^{f_i}(t) = I_{a_q}^{f_i+}(t) - I_{a_q}^{f_i-}(t) \tag{3.33}$$

By denoting system inputs as $\mathbf{u}(t) = [\mathbf{o}(t), \mathbf{y}^+(t-1), \mathbf{y}^-(t-1)]$, the state space model can be expressed by

$$\mathbf{s}(t+1) = \mathbf{A}\mathbf{s}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{E}\mathbf{w}(t), \tag{3.34}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{s}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{F}[\mathbf{v}(t) + \mathbf{r}(t)], \tag{3.35}$$

As a first step, prediction of future state is conducted for making decisions in this time-delayed system. Given the current system states $\mathbf{s}(t)$ and operation actions in

the future $\underline{\mathbf{u}}_t$, future system states $\underline{\mathbf{y}}_{t+1}$ are predictable by iterations, where

$$\underline{\mathbf{y}}_t = [\mathbf{y}(t), \mathbf{y}(t+1), \mathbf{y}(t+2), \dots, \mathbf{y}(t+t_p)]^T, \quad (3.36)$$

$$\underline{\mathbf{u}}_t = [\mathbf{u}(t), \mathbf{u}(t+1), \mathbf{u}(t+2), \dots, \mathbf{u}(t+t_p)]^T, \quad (3.37)$$

$$\underline{\mathbf{w}}_t = [\mathbf{w}(t), \mathbf{w}(t+1), \mathbf{w}(t+2), \dots, \mathbf{w}(t+t_p-1)]^T, \quad (3.38)$$

$$\underline{\mathbf{v}}_t = [\mathbf{v}(t), \mathbf{v}(t+1), \mathbf{v}(t+2), \dots, \mathbf{v}(t+t_p)]^T. \quad (3.39)$$

By iteratively substituting Eqn. (3.34) into Eqn. (3.35), we can express the $\underline{\mathbf{y}}_{t+1}$ as a function of $\underline{\mathbf{u}}_t$ and $\mathbf{s}(t)$, namely

$$\underline{\mathbf{y}}_t = \mathbf{P}\mathbf{s}(t) + \mathbf{H}\underline{\mathbf{u}}_t + \mathbf{G}\underline{\mathbf{w}}_t + \mathbf{L}(\underline{\mathbf{v}}_t + \underline{\mathbf{r}}_t), \quad (3.40)$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \dots \\ \mathbf{CA}^{t_p} \end{bmatrix}, \mathbf{H} = \begin{bmatrix} \mathbf{D} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{CB} & \mathbf{D} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{CA}^{t_p-1}\mathbf{B} & \mathbf{CA}^{t_p-2}\mathbf{B} & \dots & \mathbf{CB} & \mathbf{D} \end{bmatrix}, \quad (3.41)$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{CE} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{CAE} & \mathbf{CE} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{CA}^{t_p-1}\mathbf{E} & \mathbf{CA}^{t_p-2}\mathbf{E} & \dots & \mathbf{CE} \end{bmatrix}, \mathbf{L} = \begin{bmatrix} \mathbf{F} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{F} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{F} \end{bmatrix}. \quad (3.42)$$

with \mathbf{P} being the matrix that connects the future system outputs with the current states, and \mathbf{Q} being the matrix that connects the system outputs with operation actions. \mathbf{G}, \mathbf{L} are matrices connecting system outputs to uncertainties from stocks and demands respectively.

By involving auxiliary variables, we consider both positive part and negative part of output, which are corresponding to the overused attributes and insufficient attributes delivered by dispatched convoy. Again, we assume the attributes insufficiency leads to a much worse outcome than attribute redundancy, thus the assigned cost for insufficient attributes is much higher than that of overused attributes. Similar as previous approach, the objective is to minimize unfulfilled demands with considering the costs from resource supply, overuse and operation actions. The problem can be formulated by a linear programming model.

$$\min_{\mathbf{y}^{+/-}(\tau), \mathbf{o}(t)} \sum_{\tau=t+1}^{\tau=t+t_p} [\mathbf{c}_b^T \mathbf{y}^+(\tau) + \mathbf{c}_o^T \mathbf{y}^-(\tau)] + \sum_{\tau=t}^{\tau=t+t_p} \mathbf{c}_o^T \mathbf{o}(\tau) \quad (3.43)$$

$$\text{s.t.} \quad \mathbf{y}(\tau) = \mathbf{y}^+(\tau) - \mathbf{y}^-(\tau), \quad \forall \tau \quad (3.44)$$

$$I_{v_k}^b(\tau), I_{m_p}^b(\tau), I_{v_k}^{f_i}(\tau) \geq 0, \quad \forall \tau \quad (3.45)$$

$$\sum_k [o_{av_k}^b(\tau) + o_{dv_k}^b(\tau) + \sum_{k' \neq k} o_{v_k' v_k}^b(\tau)] \leq \bar{P}, \quad \forall \tau \quad (3.46)$$

$$\underline{\mathbf{q}}_t, \mathbf{y}^+(\tau), \mathbf{y}^-(\tau) \geq 0, \quad \forall \tau \quad (3.47)$$

where $\mathbf{c}_b/\mathbf{c}_o$ represents attribute insufficiency/redundancy costs, \mathbf{c}_o stores all the costs for actions. Constraint (3.44) ensures the balance between auxiliary variables and outputs; (3.45) ensures that all vehicle stocks and module stocks are non-negative; (3.46) ensures that the on-base ADR actions are always scheduled according to the limited capacity; (3.47) ensures that all the operation actions and auxiliary variables are non-negative.

However, the model above only targets the minimization of total operational cost,

where system instability is not included. As insufficient attributes may trigger additional attributes requirements, the system will never meet the demands if it cannot even fulfill the triggered demands. In that case, the demands keep exploding and make system unstable, i.e.

$$\mathbf{T}(\mathbf{I}_a^{fi+}(\tau) + \mathbf{d}_a^{fi}(\tau)) \geq \mathbf{M}_{va}\mathbf{I}_v^{fi}(\tau), \quad \forall \tau \in [t, t + t_p] \quad (3.48)$$

where $\mathbf{d}_a^{fi}(\tau)$ is the demands received at time τ . However, this condition is shown to be too strict to determine the system stability. For example, a battlefield without any stationed vehicle receives some unexpected demands with significant magnitude, Eqn. (3.48) will be violated with no doubt at that time. As time delays commonly exist in the operation system, the triggered demands cannot be satisfied before the vehicle relocation is done. But the system may still remain stable because of the sufficient stocks available at the base. Thus, this condition cannot be a sufficient condition to evaluate the system stability. To consider that, we relax the constraint: the triggered attributes need only to be bounded at the end of planning horizon, namely

$$\mathbf{T}\mathbf{y}^{fi}(t + t_p) \leq \mathbf{M}_{va}\mathbf{I}_v^{fi}(t + t_p), \quad (3.49)$$

To handle the stochasticity in dynamical system, stochastic MPC takes uncertainty into consideration and in particular uses information on a combination of mean and variance of the future output. By taking expectation of Eqn. (3.34) (3.35), the mean of states and outputs can be described by

$$\bar{\mathbf{s}}(t + 1) = \mathbf{A}\bar{\mathbf{s}}(t) + \mathbf{B}\mathbf{u}(t) \quad (3.50)$$

$$\bar{\mathbf{y}}(t) = \mathbf{C}\bar{\mathbf{s}}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{F}\mathbf{r}(t) \quad (3.51)$$

Next, the propagation of variance during planning horizon is also necessary for implementing SMPC. According to $\Sigma_x = E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T]$. The propagation of variance can be expressed by

$$\Sigma_s(t+1) = \mathbf{A}\Sigma_s(t)\mathbf{A}^T + \mathbf{E}\Sigma_w\mathbf{E}^T \quad (3.52)$$

$$\Sigma_y(t) = \mathbf{C}\Sigma_s(t)\mathbf{C}^T + \mathbf{F}\Sigma_v\mathbf{F}^T \quad (3.53)$$

Based on Eqn. (3.32), variance from the positive part of outputs also propagate to next time instant. In this study, we assume these amounts of variance are negligible, because the delayed demands (positive part of outputs) are minimized during fleet operation, and hence are insignificant compared to the uncertainty from demands and on-field stocks. By substitution, the expectation and variance of states and outputs over the planning horizon can be derived, which are important information in constraint relaxation. Model expressed by (3.43) can be reformulated as

$$\min_{\bar{\mathbf{y}}(\tau), \mathbf{o}(t)} \sum_{\tau=t+1}^{\tau=t+t_p} [\mathbf{c}_b^T \bar{\mathbf{y}}^+(\tau) + \mathbf{c}_o^T \bar{\mathbf{y}}^-(\tau)] + \sum_{\tau=t}^{\tau=t+t_p} \mathbf{c}_o^T \mathbf{o}(t) \quad (3.54)$$

$$\begin{aligned} \text{s.t.} \quad & p\left(\sum_{q'} T_{qq'} y_{q'}(t+t_p) \leq \sum_q M_{v_k a_q} I_{v_k}(t+t_p)\right) \geq 1 - \alpha, \\ & \Rightarrow \sum_{q'} T_{qq'} \bar{y}_{q'}(t+t_p) \leq \sum_{q'} M_{v_k a_{q'}} \bar{I}_{v_k}(t+t_p) - \\ & \quad \left(2 \sum_{q'} T_{qq'} \Sigma_{y_{q'}}\right)^{\frac{1}{2}} \text{erf}^{-1}(1 - 2\alpha), \quad \forall q \end{aligned} \quad (3.55)$$

$$\begin{aligned} & p(I_{v_k}^b(\tau), I_{m_p}^b(\tau), I_{v_k}^{f_i}(\tau) \geq 0) \leq \alpha \\ & \Rightarrow \bar{I}_x(\tau) \geq (2\Sigma_{I_x}(\tau))^{\frac{1}{2}} \text{erf}^{-1}(1 - 2\alpha), \quad \forall \tau \end{aligned} \quad (3.56)$$

$$\sum_k [o_{av_k}^b(\tau) + o_{dv_k}^b(\tau) + \sum_{k' \neq k} o_{v_k' v_k}^b(\tau)] \leq \bar{P}, \quad \forall \tau \quad (3.57)$$

$$\bar{y}_q(\tau) + (2\Sigma_{y_q})^{\frac{1}{2}} \text{erf}^{-1}(1 - 2\alpha) = \bar{y}_q^+(\tau) - \bar{y}_q^-(\tau), \quad \forall \tau \quad (3.58)$$

$$\mathbf{o}_t, \bar{\mathbf{y}}^+(\tau), \bar{\mathbf{y}}^-(\tau) \geq 0. \quad \forall \tau \quad (3.59)$$

All the state-related and output-related constraints are now chance constraints, which allow them to be violated by a certain probability α . Constraint (3.44) which specifies the balance between auxiliary variables and outputs is replaced by constraint SMPC-constraint-4. The idea is to reduce the insufficiency to a certain confidence interval once fluctuation of attributes are considered. This modeling approach further suppresses the variance propagated from the positive part of previous outputs to make the assumptions reliable. Moreover, slack variables are created at constraint (3.55), (3.56) to guarantee feasibility of the optimization. The corresponding costs for auxiliary variable are also created at objective function to ensure the auxiliary variables are minimized. For simplicity, these terms are not expressed in the optimization model.

3.6 Case Study 3

The model is implemented in a simplified mission scenario: only 1 base and 1 camp are considered. Vehicles used in this case study are same as Fig. 3.7. In this case study, robustness boost in fleet operation SMPC is emphasized through comparison firstly. Then, modular fleet and conventional fleet performances are compared in different conditions: resupplies are allowed and not allowed. Finally, we also vary the available ADR action capacity to investigate its impact on modular fleet operation.

We assume the demands stochastically arrived to the fields in random amount and will last over a period, which implies the length of operation required to satisfy the demands. Each demand is modeled as a combination of two step functions, where each of them generates the required attributes in a certain type. The magnitude of the demand is created following Gaussian distribution, i.e.

$$r_f \sim \mathcal{N}(200, 150^2) \quad (3.60)$$

$$r_p \sim \mathcal{N}(250, 125^2) \quad (3.61)$$

$$r_m \sim \mathcal{N}(280, 140^2) \quad (3.62)$$

In previous scenario, demands will last until new demands arrive. It represents the scenario that troops serve certain area. In contrast, demands in this study are assigned with a significantly large variance and it will disappear once mission is end, which can be regarded as a stochastic assault to the enemy. Fig. 3.6 provides an example of generated 20-day demands.

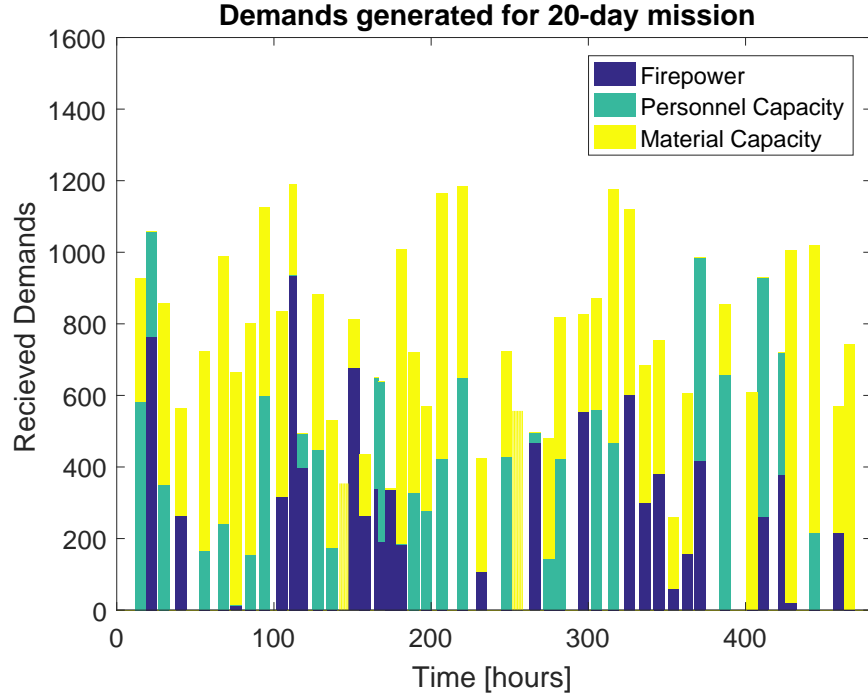


Figure 3.16: Stochastically generated demands for 20-day operation

Trigger matrix can be estimated through fitting of historical data. In this study, it is assumed to be

$$\mathbf{T} = \begin{bmatrix} 0.02 & 0.01 & 0.03 \\ 0.02 & 0.02 & 0.02 \\ 0.01 & 0.005 & 0.03 \end{bmatrix}.$$

The fluctuation of vehicle levels at battle field $\mathbf{w}(t)$ and demands $\mathbf{v}(t)$ follows

$$w_k \sim \mathcal{N}(0, 1^2), \quad (3.63)$$

$$v_q \sim \mathcal{N}(0, 10^2). \quad (3.64)$$

The planning horizon t_p is 24 hours and time required for ADR actions is the summation of the time required on processing each part of vehicles.

$$\tau_{ak} = \sum_i M_{v_k m_p} \tau_{am_i}, \quad (3.65)$$

$$\tau_{dk} = \sum_i M_{v_k m_p} \tau_{dm_i}, \quad (3.66)$$

$$\tau_{kk'} = \sum_i [(M_{v_k m_p} - M_{v_{k'} m_p})^+ \tau_{dm_p} + (M_{v_{k'} m_p} - M_{v_k m_p})^+ \tau_{am_p}] \quad (3.67)$$

where, time required for module assembly τ_{am} and disassembly τ_{dm} are 1 hour and 0.5 hour respectively. Lead time for resupply is 12 hour and vehicle relocation between battlefield and base is 1 hour. We assume the cost of action is proportional to the time required to accomplish the action. Action costs are used as a reference for costs in other types. In this study, cost for module resupply is assigned as 50 times of action cost. Cost assigned for overused and insufficient attributes is 100 times and 10000 times of action cost respectively.

3.6.1 Fleet Performance with SMPC

In this section, modular fleet operation history is selected as an example to show the improvement by using SMPC. Given sufficient resources, we first compare the vehicle level at battle field in Fig. 3.17.

As can be seen from the plots, vehicles levels change quickly in reacting to the demands. Once demands are perceived, vehicles are dispatched to field intensively and sent back to base once mission is over. By using MPC approach, the uncertainty

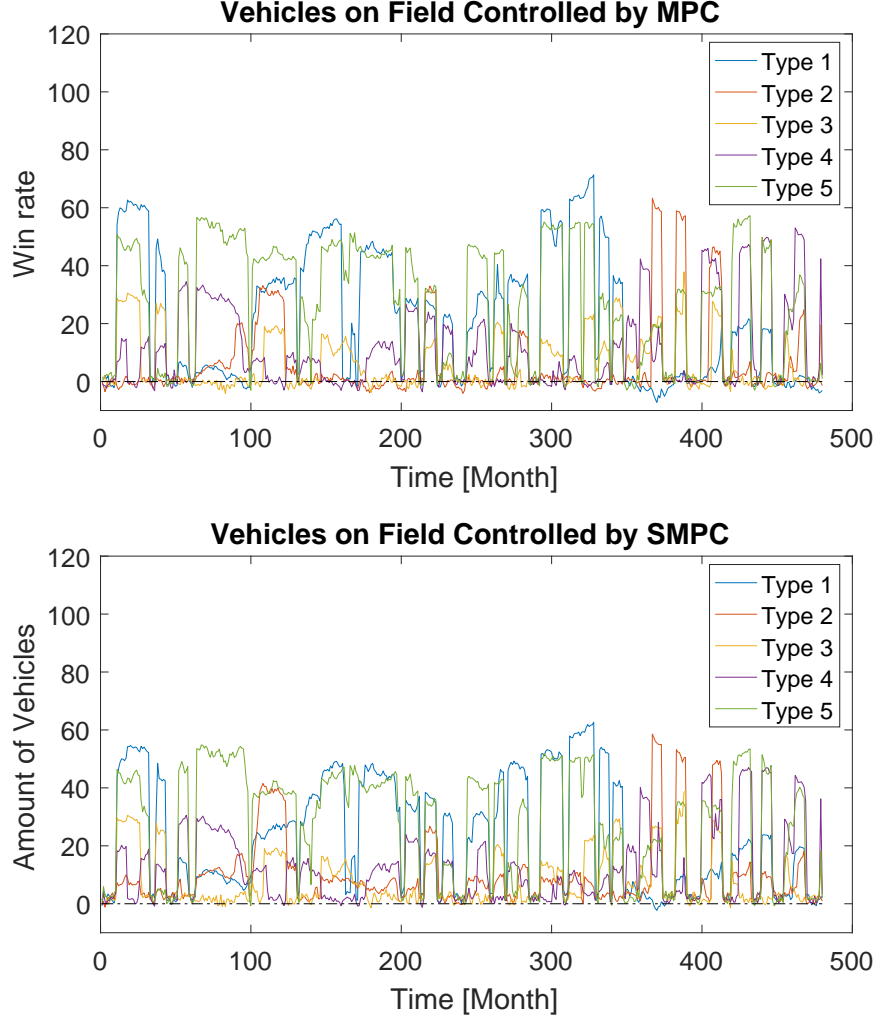


Figure 3.17: Comparison of fleet performance between MPC and SMPC

of vehicle stocks make the on-field vehicle level fluctuation around x-axis. Even MPC refresh its decisions at each time, the stochasticity in vehicle levels still dramatically incur stock insufficiency at battlefield, which cannot be compensated because of the delays in operation actions. In such a case, the MPC may not find feasible solutions because of the constraint violation, i.e., no negative stock level is allowed.

For the fleet managed by SMPC, all the vehicles stocks are mainly above zeros, except for several negative values occurring around Hour 10 and Hour 370. Compared to the length of mission, these insufficiencies are inconspicuous, which follows the idea of SMPC that constraints are allowed to be violated but in a tolerable probability.

Tightening constraints by SMPC in the fleet operation is equivalent to the safety level in the inventory management, to avoid constraint violations (backorders) based on the probability distributions of states (stock levels). The insufficiency of vehicles can also be further reduced by decreasing the tolerance level.

3.6.2 Fleet Comparison with Supply

Initially, each fleet is provided with 12 of vehicles in all types to battlefield. Resupplies are allowed to avoid system instability and minimize the amount of delayed attributes. After stability is almost guaranteed, SMPC outputs operational decisions to minimize the amount of resupplies for minimizing acquisition cost.

No unstable case is observed in 100 simulations with randomly generated demands. Because trigger matrix \mathbf{T} is known, SMPC can approximate the amount of demands to be triggered in the short future, thus, the amount of resources to be resupplied can also be easily evaluated. With unlimited resupplies, conventional fleet will always stay stable by acquiring required vehicles, the only reason of losing stability in modular fleet operation dues to the lack of capacity.

Fig. 3.6.2 compares the delayed attributes for both fleets. Based on the plots, both of fleets have a resource insufficiency period at the beginning of the operation, which incurs augmented demands by trigger mechanism in the following period. Once resupplies arrived, delayed demands of conventional fleet declined sharply to zeros, however, modular fleet takes a longer time to return to zeros, which origins from the time required in ADR actions to form a vehicle convoy from modules. After resupply arrives, delayed demands for both fleet are negligible.

The amount of resupplies are also compared in Fig. 3.6.2. Although vehicles are the only resources to be ordered by conventional fleet, we also regard conventional vehicles as a combination of 'conventional modules' to make comparison fair.

The results indicate a larger resource demanded by conventional fleet than by

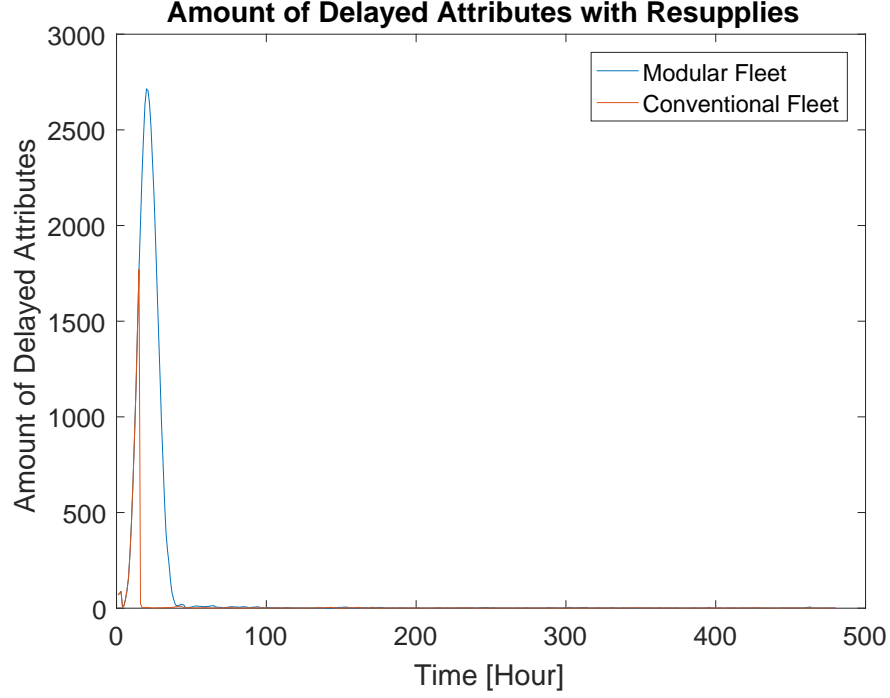


Figure 3.18: Comparison of delayed attributes between different fleets with unlimited resupply

modular fleet to avoid system instability and attribute insufficiency. Higher standard deviations are also observed in the resource requirements during conventional fleet operation, as well as the outliers of resupply. After fleet modularity, ADR actions facilitate resource sharing by swapping modules, thus promote the resource utility. The fleet reconfiguration can increase the flexibility in reacting to the demands to make the amount of supplies more predictable and stable.

3.6.3 Fleet Comparison without Supply

In this case, each fleet is provided by 30 of each type of vehicles to battlefield initially. Without any following resupplies, the fleet performance are compared in 100 randomly generated scenarios. As one of important metrics, the amount of delayed attributes is first compared in Fig. 3.6.3.

Based on the results, the mean value of delayed attributes of modular fleet is

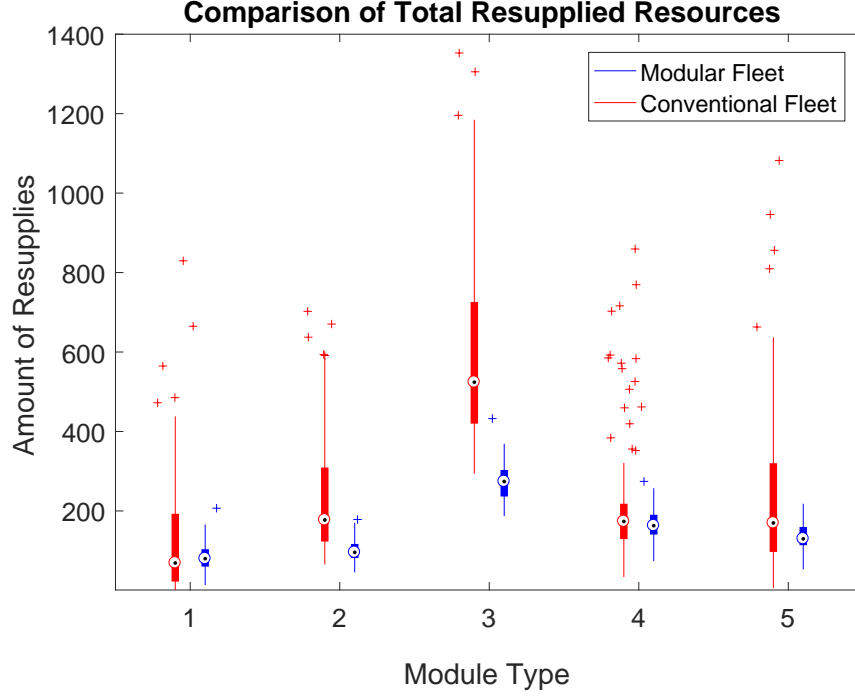


Figure 3.19: Comparison of resupplied resources between different fleets with resupply less than that of conventional fleet in both type of attributes. The value of outliers of conventional fleet is also higher than that of modular fleet, especially in material capacity. The capability of fleet reconfiguration enable modular fleet to real-timely morph itself to have a better convoy to fit the demands.

By gradually reducing the amount of initial stocks, the unstable margin can be sketch by checking the magnitude of demands. As shown in Fig. 3.21, the delayed demands increase with reducing initial supplies for both fleet. However, conventional fleet reacts more sensitively to the reduction in stocks, i.e., significant value of delayed material capacity at Hour 50. The triggered attributes of modular fleet are markedly less than that in conventional fleet, as well as the increment of delayed demands among adjacent vehicle stocks. Modular fleet can always morph itself to slow down the trigger or satisfy the demands quicker. As a sequence, conventional fleet loses its stability once the initially supplies reach to 27 vehicles, but modular fleet remains

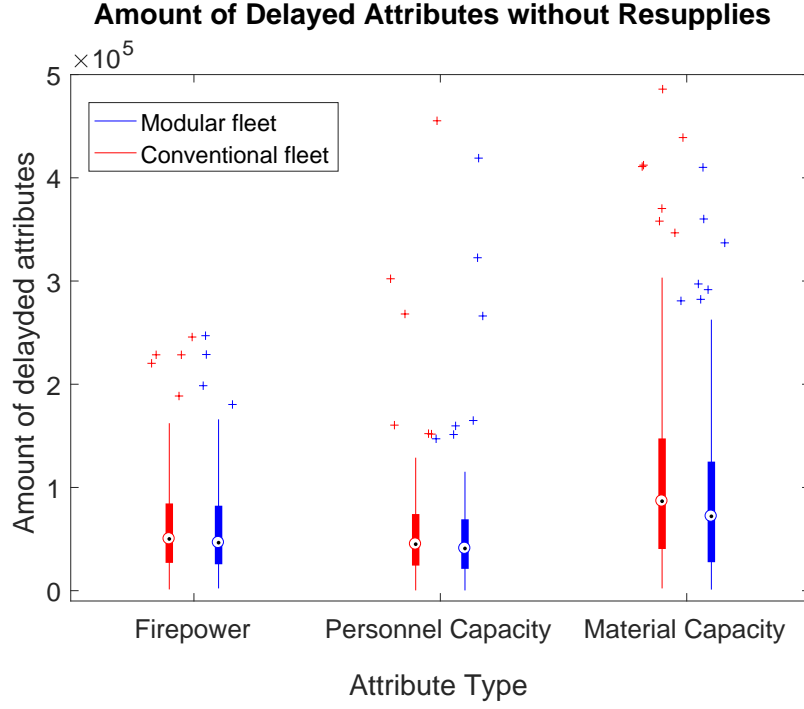


Figure 3.20: Comparison of suffered insufficiencies between different fleets without resupplies

stable.

Pie charts in Fig. 3.22 compare the capacity share for ADR actions. The plots indicate two different strategies in fleet operation: 1) once sufficient resources are given, fleet prefers to use reconfiguration actions frequently to rapidly group up a desired convoy; 2) once the resources are limited, SMPC decide to increase the proportion of disassembly action. Once the required modules for reconfiguration actions may not be available, as a solution, fleet has to disassemble existing vehicles to harvest necessary modules. Although it may also provide qualified convoy, there is no doubt that reconfiguration can save much more time than disassembly-reconfiguration-reassembly. The extended operation time incurs the slower responsiveness to demands and a higher triggered demand.

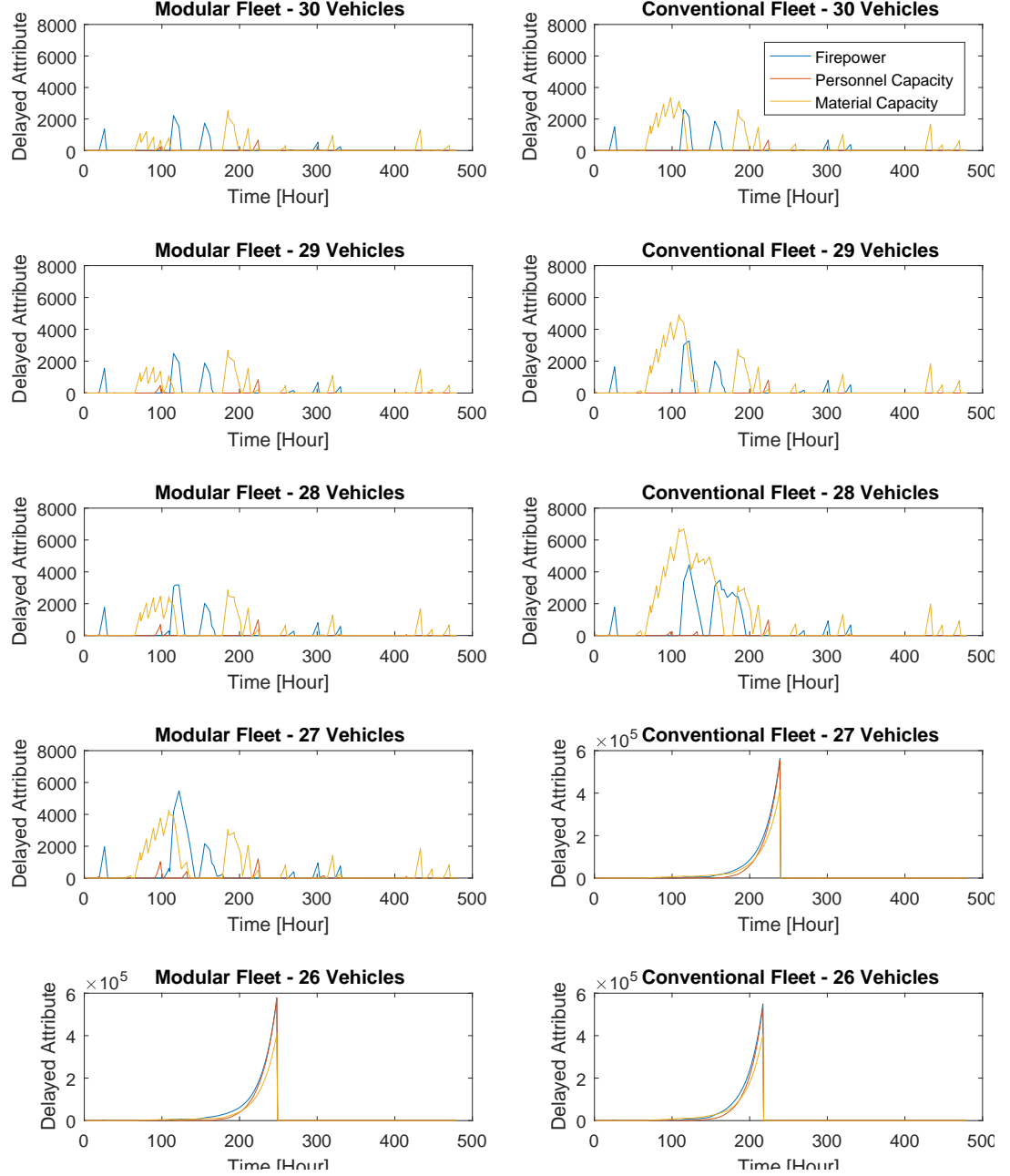


Figure 3.21: Comparison of attribute insufficiency and stability with changing initial supplies

3.6.4 ADR Action Capacity

ADR actions capacity determines the maximum work flow during modular fleet operation. It is one of the essential factors that dominates flexibility in modular fleet operation. Given 30 vehicles on battle field, we also vary the available capac-

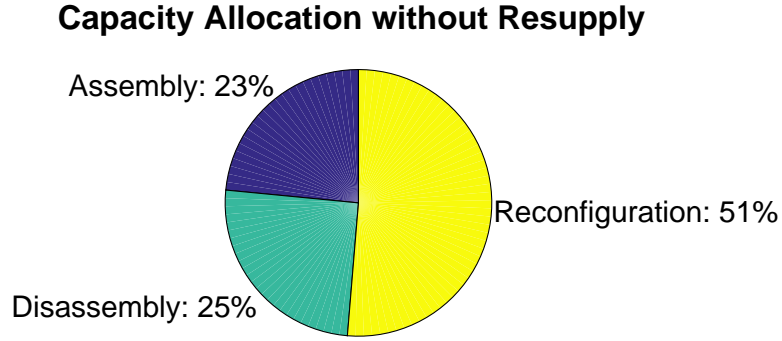
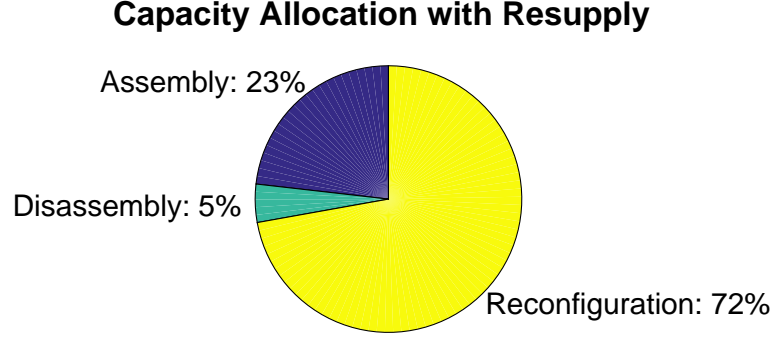


Figure 3.22: Capacity allocation in different available working capacity

ity to investigate its influences in fleet performance. After 100 randomly generated simulations, the averaged delayed attributes are plotted in Fig. 3.6.4.

A decreasing trend can be observed through the plots. A larger capacity always provides system a higher flexibility in reconfiguration. It allows multiple vehicles to be processed in the same time rather than queuing up. This result can also provide the decision support for designing the size of workshop and specifying the minimum assembly/disassembly speed that can be reach to a desired fleet performance.

3.7 Summary

In this section, a discrete dynamical system that represents the modular fleet operation is presented. An MPC method is also implemented to make real-time operational decisions according to the system status and upcoming field demands. Our

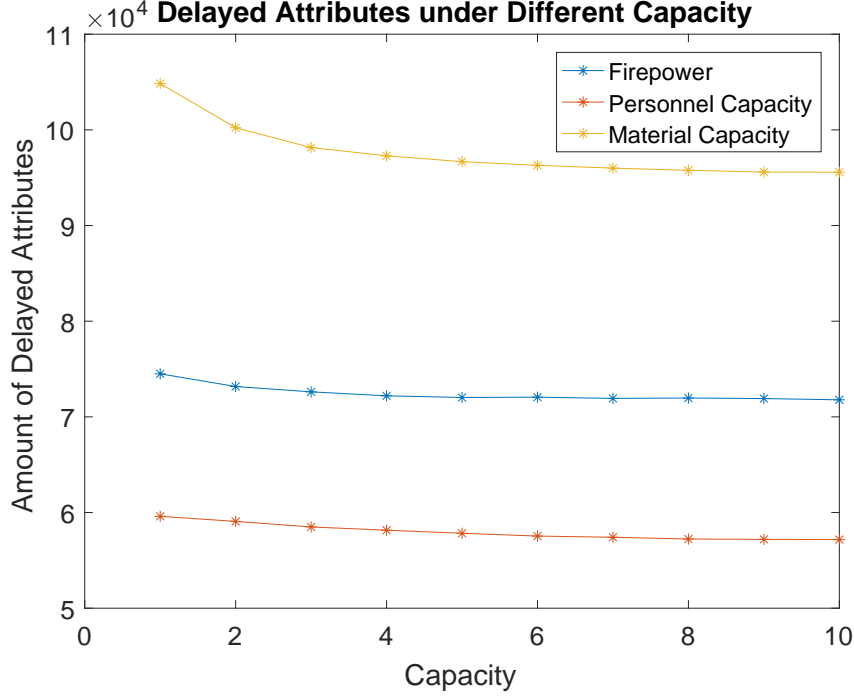


Figure 3.23: Amount of delayed attributes changing with different capacity

results indicate a significant boost in adaptability due to the flexibility gained from on-field ADR actions. Furthermore, we also consider the predictable uncertainty in fleet operation. To find the operation decisions, we first build a stochastic dynamical system to capture the dynamics in fleet operation with uncertainty, and then propose an approach to manage the dynamics of stochastic system through SMPC. Compared to MPC, SMPC boosts the robustness without sacrificing the computation efficiency and significantly reducing violations of constraints.

By contrasting the fleet performance, conventional fleet shows a higher needs of resources to avoid delayed attributes. If no supply is allowed, modular fleet performs a lower delayed attributes given similar level of resources. ADR actions endow modular fleet with additional flexibility in operation. Available ADR action capacity also closely connects to the fleet performance. The modular fleet exhibits better resilience and robustness when it comes to dealing with unexpected disruptions and noise.

CHAPTER IV

Intelligent Operation System

4.1 Problem Formulation

The mission scenario is created based on the Joint Tactical Transport System (JTTS) which is a vehicle demonstrator program in US Army Tank Automotive Research, Development and Engineering Center (TARDEC). Vehicle fleet operations is at a certain area, named as field. A field contains a main base and N camps. An example of field layout is shown in Fig. 4.1. Main base provides the space to receive and store the resources (vehicles and modules) supplied from global manufacturer. For modular fleet operation, base also provides the equipment for ADR actions. Camps are close to the battle field, which are the places to receive the demands.

In reacting to the demands, a convoy, which is a group of vehicles, is required to be customized and dispatched to camps to satisfy the demands. Operation actions also vary according to the type of fleets. For a conventional fleet, the operation actions to be determined include

1. vehicle resupply
2. vehicle relocation between different locations
3. vehicle convoy composition.

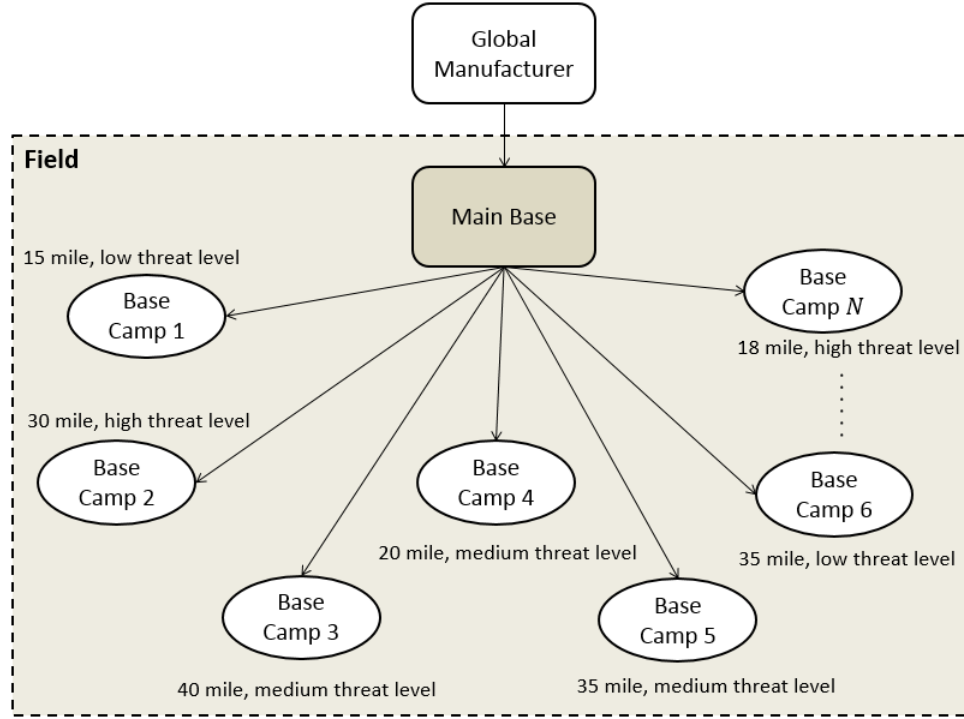


Figure 4.1: Mission scenario in JTTS

Once fleet modularity is considered, modules are the only resources required for operating a modular fleet. The actions to be determined include

1. module resupply
2. vehicle relocation between different locations
3. ADR actions
4. vehicle convoy composition.

Compared to conventional fleet, modular fleet needs to achieve transition from modules at supplier to vehicles at camps, which increases the difficulty in fleet operation. This process can be analogized as a completely dynamic supply chain, as shown in Fig. 4.2.

Although the similarity between the modular fleet operation and civilian applications is significant, there are marked differences between operating a military fleet

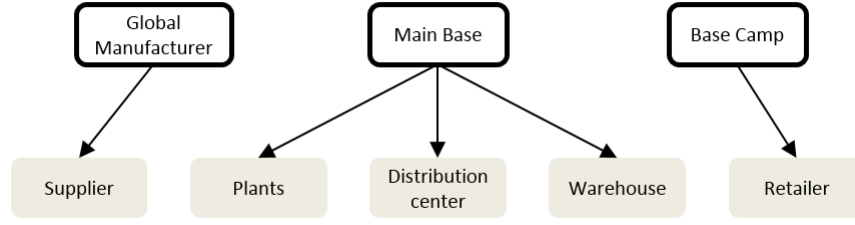


Figure 4.2: Analogy between the military fleet operation and supply chain

and a commercial fleet. For example, the enemy's actions are the main source of damage to the military fleet. The critical damage of modules/parts incur the loss of the resources. The uncritical damage also leads to vehicle repair or maintenance, which increases the uncertainty of vehicle usage. The stochasticity of damage creates additional complexity in management of a military fleet. Taking the modular fleet operation as an example, the operation process is summarized in Fig. 4.3.

To operate a highly modularized vehicle fleet in the complicated military mission scenario, it is required to make real-time decisions on vehicle dispatch, ADR action scheduling and module resupplies. A mathematical model is required to be formulated to analyze field situations and provide suggestions on operation decisions based on the current inventory status and camp demands. By realizing operation decisions into a military mission simulation model, fleet performance can be captured by using the indices including fleet readiness, operation cost, and total supplied resources.

4.2 Agent Based Model

To satisfy these requirements, an ABM is customized, which can be used for demand forecasting and operation decisions making. With consideration of computational efficiency and restriction of communication on the field, multiple types of agents are created to represent decision makers with different functionalities, in terms of camp agent, base agent, and supply agent. Each of them can communi-

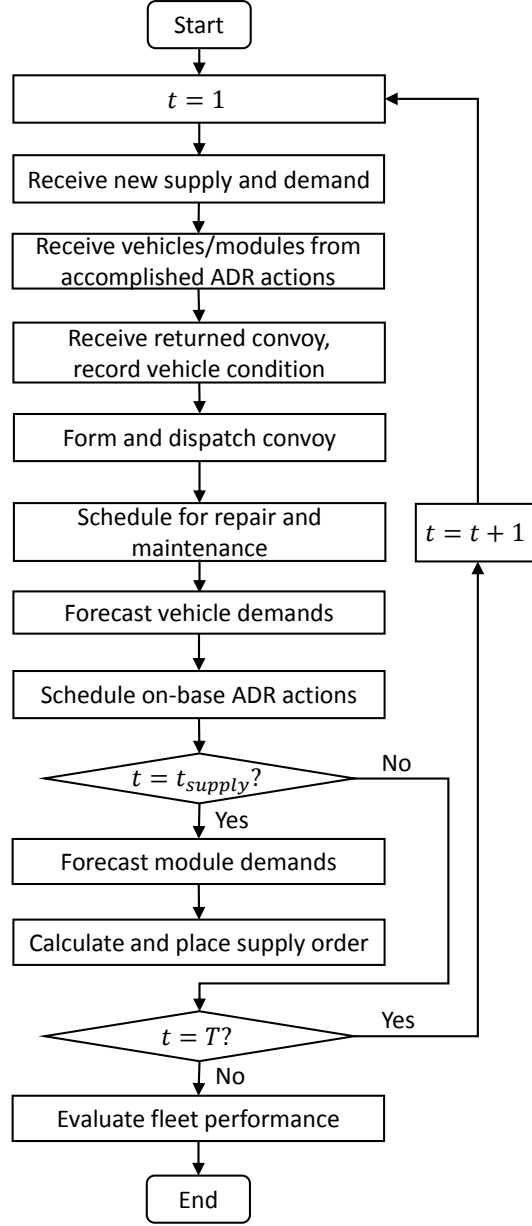


Figure 4.3: Process flow in modular fleet operation

cate with adjacent agents and make decisions in a centralized manner. According to the priority in perceiving the combat information, a hierarchical framework is formulated to connect a number of agents structured in a master-slave relationship. The control decision is generated from agent and sent downward to their subordinates. Correspondingly, status reports flow bottom-up to a higher level supervisory agents. Fig. 4.4 demonstrates the information flow among agents.

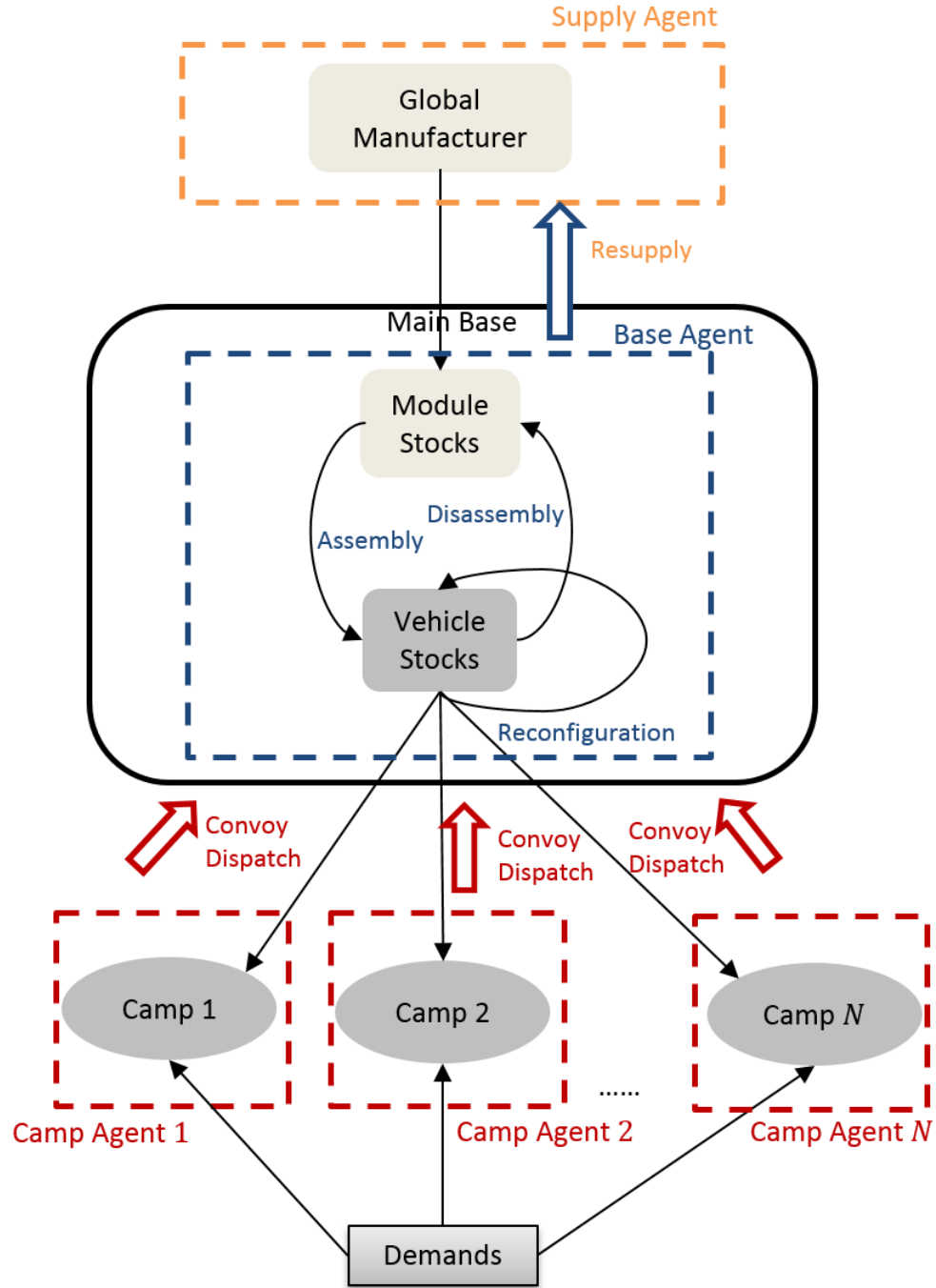


Figure 4.4: Diagram of the agent-based model for modular fleet operation

The agents make decisions in a predetermined direction. First, camp agent makes decisions on the convoy compositions and dispatch dues. Then, given the convoy dispatch schedule, base agent analyzes the inventory status and plans for the on-base

ADR actions and vehicle dispatch schedule to guarantee the vehicle sufficiency on camps. Finally, based on the usage of vehicles and modules, supply agent forecasts the future resource demands and calculates the resupply orders. Several assumptions are used to simplify this problem with reserving a reasonable level of model fidelity:

1. ABM only knows the demands one month ahead.
2. The records of actual resource consumption are accurate and accessible to all agents.
3. Insufficient modules will be back ordered.
4. Assembly/disassembly time for the vehicles is the sum of the time required for each module.
5. All the scheduled actions will be finished exactly on time.
6. Sequence of vehicle ADR actions is ignored.

4.2.1 Camp agent

As the upstream agent of the framework, camp agent needs to determine the dispatch order in reacting to the received demands assuming that all the orders are achievable by the downstream agents. Demands from military mission scenario are classified into two aspects, which are convoy requirements a_c and vehicle requirements a_v . Convoy requirements specify the requirement for overall convoy, which are generally additive attributes carried by vehicles, e.g., personnel/material capacity, firepower, etc. In contrast, vehicle requirements are mainly from environmental constraints, e.g., vehicle weight, tire type, threat level, and terrain capacity.

Taking camp n as an example, the formed convoy to camp n is denoted as $\mathbf{d}^{c_n}(t)$,

and the received convoy attribute requirements as a vector \mathbf{a} , where

$$\mathbf{d}^{c_n}(t) = [d_{v_1}^{c_n}(t), d_{v_2}^{c_n}(t), d_{v_3}^{c_n}(t), \dots, d_{v_{N_v}}^{c_n}(t)]^T \quad (4.1)$$

$$\mathbf{a} = [a_{c_1}, a_{c_2}, \dots, a_{c_a}, a_{v_1}, a_{v_2}, \dots, a_{v_b}]^T \quad (4.2)$$

To avoid the myopic decisions, the camp agent usually makes decisions not only for the current time, but also for the short future. The objective of convoy formation is to optimize fuel economy, demand fulfillment, acquisition cost, convoy weight, etc. The convoy dispatch problem can be generalized as an optimization problem:

$$\begin{aligned} & \underset{\mathbf{d}^{c_n}(t)}{\text{minimize}} && \sum_n \mathbf{w}_{c_n}^T \mathbf{d}^{c_n}(t), \\ & \text{subject to} && (a) f^{c_\alpha}(d_k^{c_n}(t)) \geq a_{c_\alpha}, \quad \forall \alpha \\ & && (b) f^{v_\beta}(\mathbf{d}^{c_n}(t)) \geq a_{v_\beta}, \quad \forall \beta, \end{aligned} \quad (4.3)$$

where, t_p is the planning horizon. \mathbf{w}_{c_n} is a vector represent interested costs for all types of vehicles. f^{c_α} is a scalar function to evaluate the convoy attributes of type α . f^{v_β} is a scalar function to evaluate the vehicle attributes of type β . Constraints (a)(b) ensure that both convoy and each vehicle of the convoy satisfy the requirements. In this study, referring to the objective used in previous research (*Bayrak et al.*, 2018), objective is considered as fuel economy and constrained by order fulfillment and terrain condition.

4.2.2 Base agent

Given convoy dispatch schedule, $\mathbf{d}^{c_n}(t)$ and the associated operation period $\mathbf{t}^{c_n}(t)$ to satisfy the demands, the expected number of vehicles operating in the battle field

can be denoted as $\mathbf{D}^{c_n}(t)$, which is the demand, as shown in Eqn. 4.4.

$$\mathbf{D}^{c_n}(t) = \sum_t \mathbf{d}^{c_n}(t), \quad \forall t \in \mathbf{t}^{c_n}(t). \quad (4.4)$$

Base agent aims to plan ADR actions to ensure all the dispatch decisions can be achieved. However, vehicle damage and maintenance commonly exist during operation, which require additional time to reuse the damaged vehicles. In other words, the time of vehicle recovery can be seen as an extension of field operation. And its stochastic occurrence raises the difficulties in estimating vehicle demands. Based on the fact that vehicle damage and maintenance only occur during convoy operation, additional vehicle usage highly dependent on the vehicle dispatch schedule. In this study, the demands are forecasted through Auto-regressive Exogenous (ARX) model (*Pandit and Wu, 1983*). Take the vehicle in type k as an example, the expression of ARX model is

$$\begin{aligned} e_k(t) = & a_1 e_k(t-1) + \dots + a_{na} e_k(t-na) + \\ & b_0 \sum_n d_k^{c_n}(t) + \dots + b_{nb} \sum_n d_k^{c_n}(t-nb), \end{aligned} \quad (4.5)$$

where, $e_k(t)$ is the extra vehicle usage from repair and maintenance at hour t . a_i and b_j are the parameters to be determined by fitting the historical data. na and nb specify the review horizon of historical data in additional usage and dispatch order. With continuously training by using latest field information, agent can real-timely adjust the model in reacting to the time-varying scenario and enemy's actions. To guarantee the robustness of production planning (*Ouelhadj and Petrovic, 2009*), the predicted hourly demands are compiled into the daily demands. The goal is to manage the inventory stocks to ensure the sufficiency of vehicles for the demand in the next day.

Thus, an effective management strategy is required to efficiently operate the fleet and rapidly respond to the stochasticity. In this study, two methods are proposed to achieve real-time inventory management. Two strategies are named as Empiricist which is driven by heuristic rules and Optimizer which is based on the mathematical optimization. Results are compared between different management strategies to show their impacts on the overall fleet performance.

4.2.2.1 Empiricist

Empiricist represents a rule-based approach. No firm schedule is generated in advance and all the decisions are made to the unfulfilled field demands and inventory status. Specifically, at the beginning of each day, Empiricist creates a set of spare vehicles \mathbf{s}^+ based on the vector $\mathbf{v}^+ = (\mathbf{s}_v - \mathbf{d}_v)^+$, and a remaining demand set \mathbf{s}^- based on the vector $\mathbf{v}^- = (\mathbf{d}_v - \mathbf{s}_v)^+$. Additional two sets are also defined for decision making: \mathbf{s}_a^- for the demands to be satisfied by assembly, \mathbf{s}_r^- for the demands to be satisfied by reconfiguration.

According to the popular sequencing rules used by the industry (*Nahmias and Cheng, 1993*), military operation (*Bayrak et al., 2018*) and suggestions from military experts, a rule-based strategy is created as a mixture of First-come, first-served (FCFS), Shortest processing time (SPT) and Earliest due date (EDD). The algorithm is demonstrated in Fig. 4.5.

4.2.2.2 Optimizer

Optimizer represents an optimization-based approach. The objective is to maximize fleet readiness with minimized operation costs, in terms of inventory holding cost, insufficiency cost, and ADR action cost. The actions to be determined are

1. Number of vehicles of type k to be assembled, o_{ak}
2. Number of vehicles of type k to be disassembled, o_{dk}

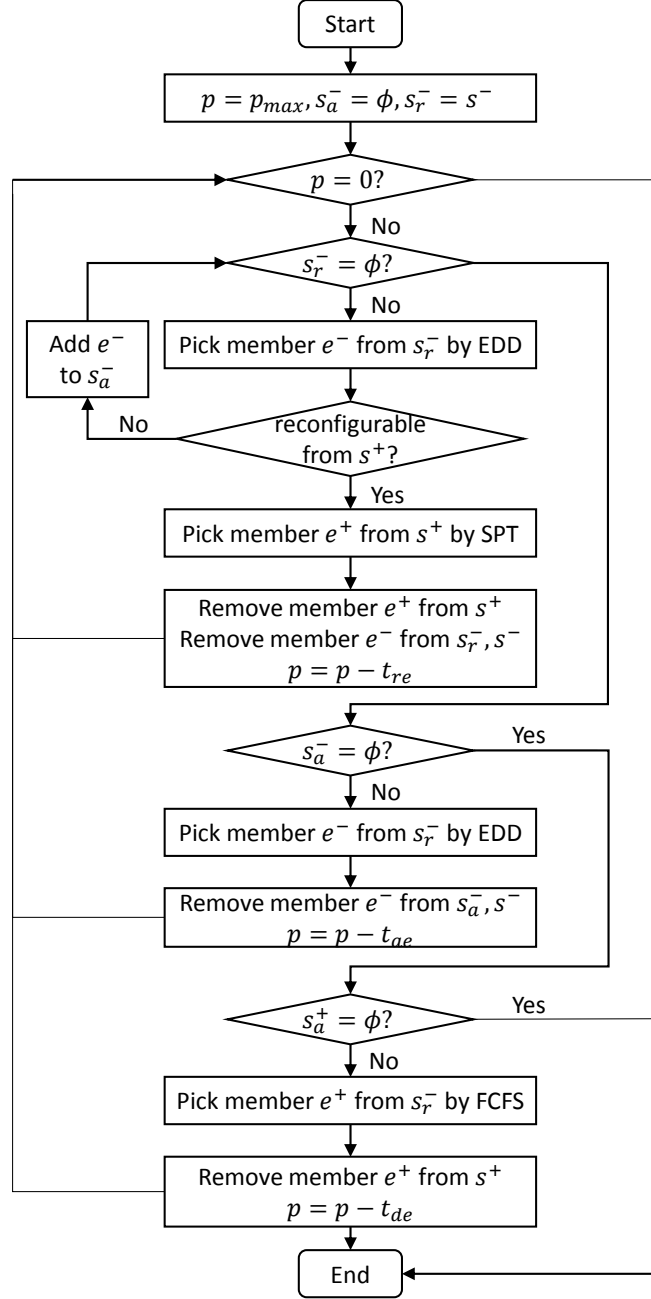


Figure 4.5: Rule-based algorithm for ADR action planning

3. Number of vehicles of type k to be reconfigured into type k' , $o_{kk'}$
4. Number of vehicles of type k to be dispatched from main base to camp n , $o_k^{c_n}$.

All the actions are assumed to require less than 24 hours to finish and can be accomplished in one day. Thus, by selecting the module/vehicle stocks as states, e.g.,

$s_{v_k}(t)$, system dynamics is dominated by Eqn. 4.6.

$$s_{v_k}(t) = s_{v_k}(t-1) + o_{ak}(t) - o_{dk}(t) - \sum_{k \neq k'} o_{kk'}(t) + \sum_{k \neq k'} o_{k'k}(t). \quad (4.6)$$

Thus, the fleet operation problem can be formulated by a linear programming model to optimize operation actions in the planning horizon t_p .

$$\begin{aligned} \min_o \quad & \sum_{t \in t_{plan}} \left[\sum_k [c_{ak} o_{ak}(t) + c_{dk} o_{dk}(t) \right. \\ & + \sum_{k' \neq k} o_{kk'}(t)] + \sum_k c_{hk} [s_{v_k}(t) - \sum_n o_k^{c_n}(t)] \\ & \left. + \sum_k \sum_n c_k^{c_n} (D_k^{c_n} - o_k^{c_n}(t)) \right] \\ \text{s.t.} \quad & (a) \sum_k [t_{ak} o_{ak}(t) + t_{dk} o_{dk}(t) \\ & + \sum_{k \neq k'} t_{kk'} o_{kk'}(t)] \leq p_{max}, \forall k, t \\ & (b) \sum_n o_k^{c_n}(t) \leq s_k(t), \forall k, n, t \\ & (c) o_k^{c_n}(t) \leq D_k^{c_n}(t), \forall k, n, t \\ & (d) o_{ak}(t), o_{dk}(t), o_{kk'}(t), o_k^{c_n}(t) \geq 0, \quad \forall k, t, \end{aligned} \quad (4.7)$$

where, c_{hk} corresponds to the hold cost of vehicle of type k . $c_k^{c_n}$ is the sufficiency cost of a vehicle of type k at camp n . Insufficiency cost may also vary among camps to reflect different tactical importance. t_{ak}, t_{dk} and $t_{kk'}$ are the time required to finish ADR actions, which is evaluated according to the complexity of actions. Constraint (a) claims that ADR actions should be scheduled under capacity threshold; (b) ensures the operation actions are strictly constrained by current stocks; (c) guarantees that the number of dispatched vehicles cannot exceed the dispatch order; (d) ensures all the decision variables are non-negative.

4.2.3 Supply agent

Given the schedule of operation actions, resupplies in proper schedules are important to ensure that fleets can operate smoothly without delays. Two inventory control strategies are commonly used in practice, which are optimal (Q,R) policy (*Nahmias and Cheng, 1993*) and order-up-to-level policy (*Silver et al., 2009*). In optimal (Q,R) policy, the inventory status is assumed to be reviewed continuously, once stocks reach to the reorder point s , an optimized resupply order Q is placed, which is calculated to minimize the expected cost of holding, setup and shortages. In the order-up-to-level policy, the inventory status is periodically reviewed which is closer to JTTS mission scenario. Once stocks are below the reorder point s , the resupply order is calculated as

$$o = S - IP, \quad (4.8)$$

where, IP is the inventory position, which is the sum of on-hand stocks and on-order stocks minus back-orders. S is the order-up-to-level, which is calculated by Eqn. 4.9

$$S = E(X) + k\sigma_X + (n - E(\tau))\mu, \quad (4.9)$$

where, τ records the length of time that the inventory position drops below the reorder point until the next review instant. μ is the demand rate during the review interval R . X is the total demands over the lead time $\tau + L$, with mean and standard deviation as $E(X)$ and σ_X respectively. n denotes the desired periods to reorder. k is the factor that determines the safety level, which is evaluated according to the target fill rate.

Previous methods require the estimation of μ in resupply order calculation. However, in this study, module damage heavily depends on the vehicle dispatch schedule, which leads to a dramatic change in μ among different review intervals. Given dispatch schedule and module usage history, the module demands $D_{m_i}(t+1, t+n)$ can be predicted in the next review interval and used as a substitution of μ in order-up-

to-level calculation.

$$S(t) = E(X) + k\sigma_X + D_{m_i}(t+1, t+n - E(\tau)). \quad (4.10)$$

The module usage consists of two parts, ADR actions and repair/maintenance. Modules for ADR actions usage D_m^{ADR} can be estimated according to the output from base agent. Taking modules in type i as an example, its usage can be estimated by

$$\begin{aligned} d_{m_i}^{ADR}(t) = & \sum_k M_{v_k m_i} (o_{ak}(t) - o_{dk}(t)) + \\ & \sum_{k \neq k'} (M_{v'_k m_i} - M_{v_k m_i}) (o_{kk'} - o_{k'k}). \end{aligned} \quad (4.11)$$

where, $M_{v_k m_i}$ indicates the number of module of type i carried by vehicle of type k . The predictions on the modules usage for repair and maintenance is similar as the approach for the extra vehicle unavailability. However, as a module might be used in multiple types of vehicles, the damage probability also varies for different vehicles. This fact is considered in module usage forecasting. Denote the set of vehicle types that contains modules in type i as ϕ_i , and the total number of vehicle types that contain module in type i as n_i , the forecasting of module usage of type i can be expressed as

$$\begin{aligned} d_{m_i}^c(t) = & p_1 d_{m_i}^c(t-1) + \dots + p_{np} d_{m_i}^c(t-np) + \\ & \sum_{\gamma \in \phi_i} \left[q_{1\gamma} \sum_n d_{v_\gamma}^{c_n}(t) + \dots + q_{nq\gamma} \sum_n d_{v_\gamma}^{c_n}(t-nq) \right], \end{aligned} \quad (4.12)$$

where, p_i and q_j are the parameters to be evaluated based on the historical data. np, nq are the corresponding review horizons. The total module usage can be calcu-

lated.

$$D_{m_i}(t+1, t+n-E(\tau)) = \max_{t+1 \leq \tau \leq t+n-E(\tau)} d_{m_i}^{ADR}(\tau) + \sum_{t+1}^{t+n-E(\tau)} d_{m_i}^c(t). \quad (4.13)$$

As the resupply lead-time is assumed to be significantly shorter than review interval, thus, all resupply orders are received at reorder point. Combining with Eqn. 4.8, 4.9, 4.13, the resupply order can be calculated as

$$o_{m_i}(t) = D_{m_i}(t+1, t+n-E(\tau)) + k\sigma_X - s_{m_i}(t) + s_{bm_i}(t). \quad (4.14)$$

ABMs are created for decision making of both fleets, where camp agent and supply agent perform in a similar way. The main difference locates at the bases agent. Conventional fleet has no choice in executing ADR actions to achieve fleet reconfiguration. In other word, conventional fleet can be regarded as a special type of modular fleet which has 0 available working time. The model is built in *MATLAB*, which has over 20 parameters and over 30 sub-functions, it owns a high flexibility to change the scenarios and vehicle designs.

4.3 Case Study

In the application, mission is provided as transporting required supply materials from a main base to several camps following the battlefield requirements. Based on the supply requirements, 12 types of existing military trucks are selected to accomplish transportation missions, where 16 type of modules are created by disintegrating the conventional vehicles. According to the functionality and interface of modules, 18 types of modular vehicles are designed as substitutes of conventional vehicles.

Table 4.1: Comparison of total operational costs between fleets

Cost	Conv. Fleet	Mod. Fleet (Empiricist)	Mod. Fleet (Optimizer)
Holding	42289	26402	29695
Backorder	233	114110	1766
Assembly	N/A	7633	13766
Disassembly	N/A	2433	4670
Reconf.	N/A	46	2370
Total	42522	150624	52267

4.3.1 JTTS scenario

Based on the suggestions from experts of Army, the virtual costs and time constants are $c_{hk} = 0.5$, $c_{bk}^{cn} = 100$, $\forall n, k$ and $c_{ami} = 1$, $c_{dmi} = 0.5$, $t_{ami} = 1$, $t_{dmi} = 0.5$, $\forall i$. The settings of the operation system are $\bar{p} = 60$, $k = 3$ and $n = 30$. For ARX parameters, $na = np = 3$, $nb = nq = 24$ are assigned according to the time requirement of operation actions. After 100 realizations, the mean value of total operation cost between modular fleet and conventional fleet are firstly compared in Tab. 4.1.

Based on the comparison, the expenditure from inventory holding for conventional fleet is markedly higher than that for modular fleet, because modular fleet can release space by ADR actions. However, these additional operation actions incur a higher cost in backorders, especially for empiricist. Compared to Empiricist, Optimizer can yield a better schedule to significantly reduce backorders and total operational costs.

The total amount of supplies is also of interest, since there is no existing modular military fleet that can be used as a reference for cost evaluation. The conventional vehicles are also disintegrated into 18 types of 'conventional modules' to make fair comparison. Thus, the metric used is the sum of back-ordered modules and supplied

modules. Fig. 4.6 compares the total supplies ordered from both fleets.

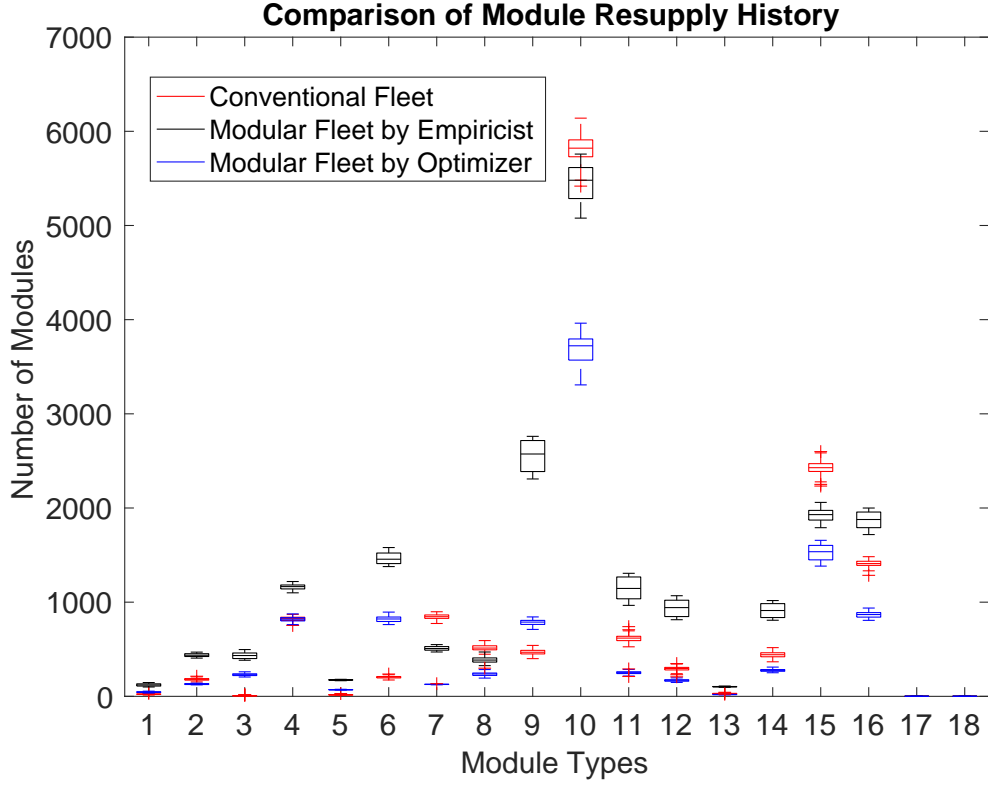


Figure 4.6: Comparison of total resupplied resources

According to the plot, fleet modularity leads to around 40 percentage reduction in resupplied resources of armor (type 16 - type 18), tire and suspension (type 8 - type 10). One reason is the ease of repairing and maintaining modular vehicles, which significantly reduces unavailability periods of vehicles. Furthermore, the savings can also be interpreted as the pooling effect from the component sharing. It has been proven that a higher commonality designed in the product family can lead to lower supplies in satisfying the same service level (*Gerchak et al.*, 1988). In this study, vehicles share a large proportion of modules, which increase module utility rate through ADR actions. With time-varying demands, modular fleet can reshape itself rapidly to satisfy the demands without ordering all the necessary vehicles.

In addition, the pooling effect also explains the different strategy for power train

resupply (type 6 - 8). Modular fleet prefer to order modules of type 6; conventional fleet prefer to order modules of type 7. By checking the mapping between modular vehicles and modules, the powertrain of type 6 owns a much higher commonality than modules of type 7 among modular vehicles. This fact makes the agents prefer to use and order powertrain of type 6 to promote module utility rate and the speed of fleet reconfiguration.

Compared to conventional fleet, the standard deviations of supplies are higher during modular fleet operation, which can be explained by the extra echelon level brought by fleet modularization (*Liao and Chang, 2010*). According to the analogy in Fig. 4.2, conventional fleet operates as a single-tier supply chain, where the supplier offers all the required vehicles (products) to the base(retailer) directly. For modular fleet, workshop (manufacturer) is required to convert modules to vehicles, which makes the system become more vulnerable to the stochastic damage and maintenance.

Based on the plot, modular fleet managed by different approaches exhibits total distinct performances: the fleet controlled by heuristic rules even requires more resupplies than the conventional fleet; the fleet controlled by optimization shows a much lower needs in resources. Similar results exist in fleet readiness comparison which is shown in Fig. 4.7. Dramatically high backorders occur in the operation managed by Empiricist, in term of both mean value and standard deviation. The results show a considerable impact from operation strategy on modular fleet performance. With improper operation strategies, modular fleet may acquire more resources and suffer a much higher insufficiency.

To highlight the differences between Optimizer and Heuristics, the number of actions on different types of vehicles for different strategies is shown in Fig. 4.8. The total operation time on field is used to represent the vehicle usage. Firstly, it can be observed that ADR actions are driven by the requirement of vehicles. Taking vehicles of type 11 as an example, they are mainly reconfigured from other types of vehicles

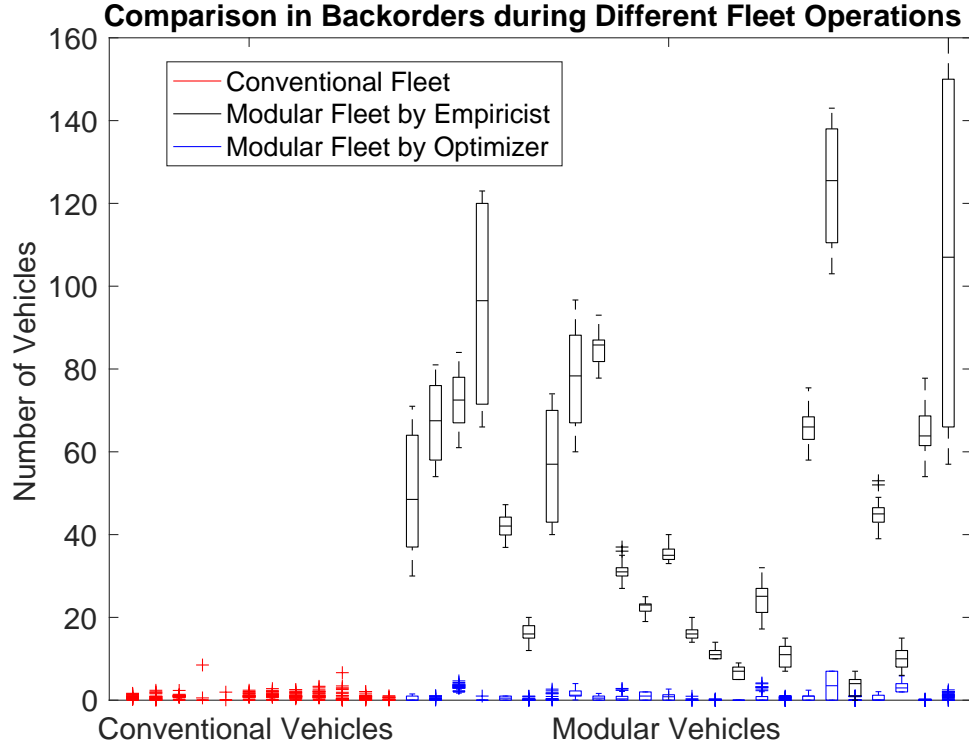


Figure 4.7: Readiness comparison between different fleets

rather than assembled from modules. By checking the composition of each vehicle, vehicles of type 11 are found to be easily reconfigured from other modular vehicles, but they are hard to be reconfigured to other modular vehicles.

Optimizer outputs an interesting operation strategy: the vehicle of type 11 is used as a temporary state between a more complex vehicle and modules. These transient periods are shown to be sufficient to accomplish the insignificant orders for this type of vehicle. Optimizer can properly use the field and vehicle information, and yield operation strategy which may not be easily perceived.

Compared to Empiricist, more disassembly and reconfiguration actions emerges under the management of Optimizer. Reconfiguration actions can release spare modules to guarantee the responsiveness to the stochasticity from operation. Optimizer can also timely sense the redundant vehicles, and convert them to other vehicles

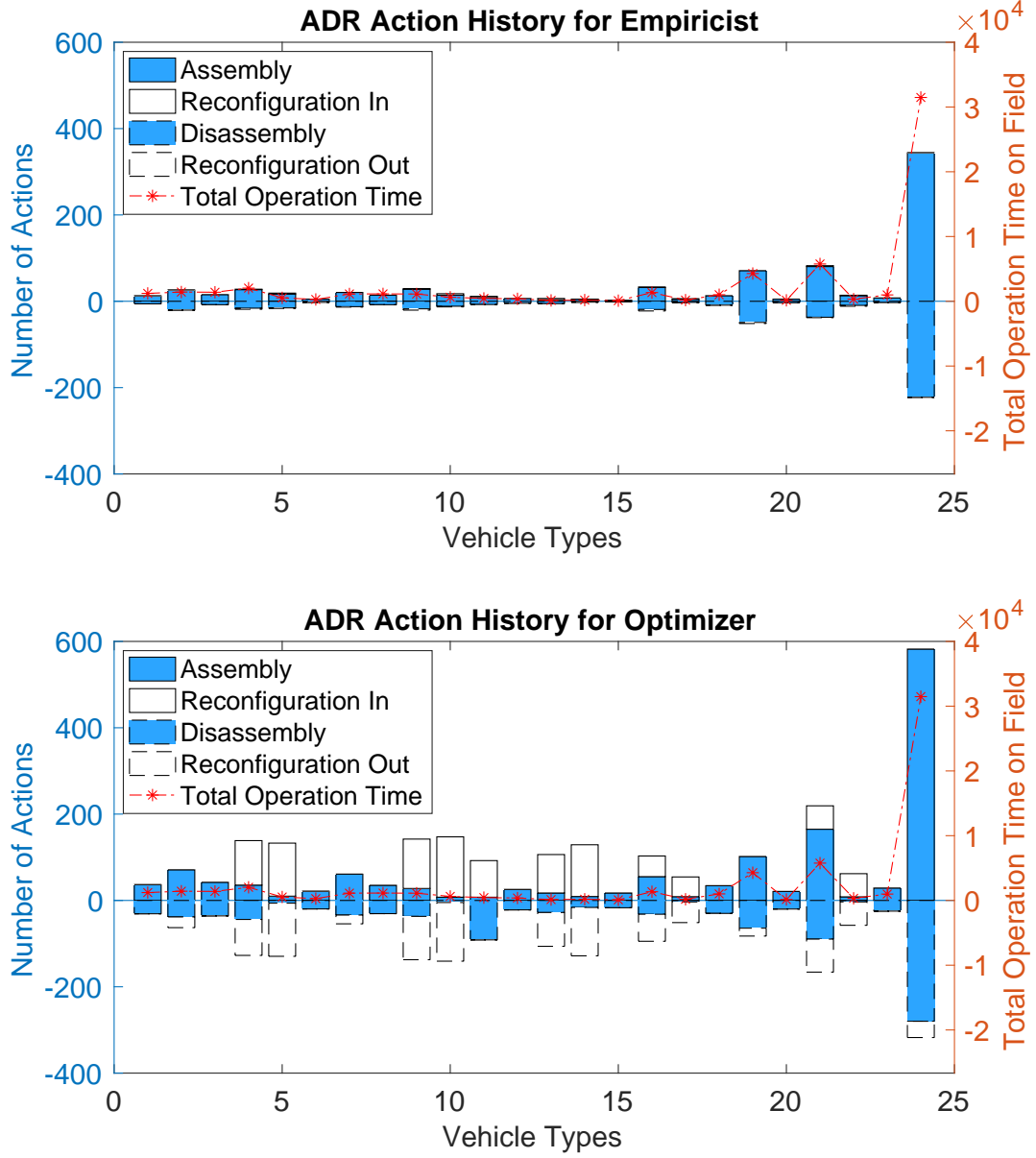


Figure 4.8: Comparison of ADR actions on different types of vehicles between Empiricist and Optimizer

that can be used in the future, which promotes resource utility rate and operation effectiveness.

As a summary, the fleet managed by Empiricist performs a build-to-order manner, i.e., provides custom-built vehicles in a minimal lead time (*Holweg and Jones, 2001*). It is proven to be a way to restrict redundancy as the vehicles would only be

assembled to the received demands (*Holweg et al.*, 2005). However, in modular fleet operation, the key of managing whole vehicle fleet to satisfy the long-term goal of the operation, other than try the utmost to satisfy every received demands. Optimizer can successfully balance the operation actions with consideration of future impacts of current decisions, thus yield more considerate plans.

4.3.2 Sensitivity Analysis

It is known that sufficient production capacity gives the company the required ability to meet demands in the marketplace (*Krasnikov and Jayachandran*, 2008). Similarly, the ADR action capacity is essential for the modular fleet to perform in a desired way. ADR action capacity impacts on the module supplies, base infrastructure designs and personnel requirements, which is an important metric to evaluate fleet performance and budget. Because of the sophistication in the fleet operation and field demands, it is intractable to theoretically evaluate the minimal ADR actions capacity requirements (?). Therefore, taking Optimizer as an example, a parametric study is conducted on the influences from ADR actions. Based on the data from 100 realizations, Fig. 4.9 demonstrates the impacts from ADR capacity.

Increasing ADR capacity shrinks the mean backorders occurred during the fleet operation, which indicates a better fleet responsiveness in reacting to the demands. Once ADR capacity reaches 60 hours, the average number of backorders become steady at 6 vehicle · hours. Compared to the total operation time during the year, e.g., more than 50000 vehicle · hours, these amount of backorders become unobtrusive. The remaining backorders are induced by the stochasticity in operation, which could be eliminated by setting up vehicle safety stocks base on the demand fluctuation (*Nahmias and Cheng*, 1993).

Two typical stages exist in the changing of average vehicle stocks, which is separated by a certain ADR action capacity (36 hours in this case). This capacity is

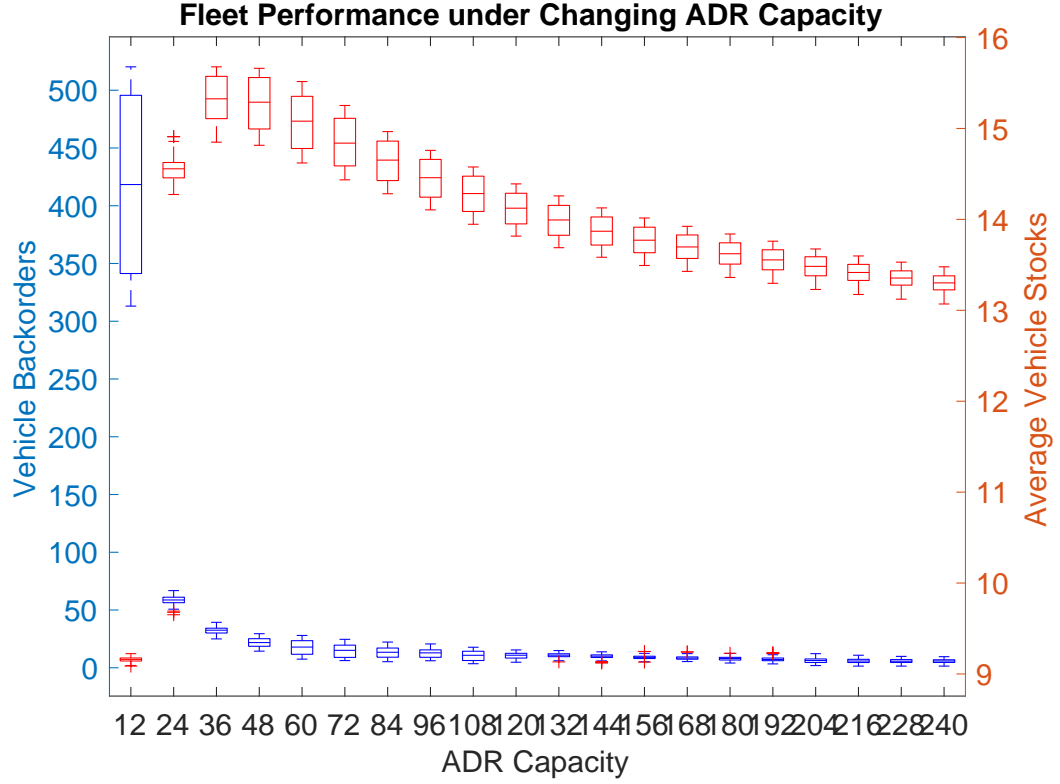


Figure 4.9: Modular fleet performances under diverse ADR capacities

named as separation capacity. Before it, vehicle assembly is the mainstream of on-field actions, because the agents spend most of the capacity in grouping up convoy to satisfy the demands. Thus, vehicle stocks and total resupplied resources are all at maximum once the ADR action capacity reaches the separation capacity. Once vehicle sufficiency is guaranteed, the superfluous capacity is applied to reduce the total operational cost, e.g., disassemble the vehicles to save inventory holding cost. As a sequence, stock level keeps decreasing towards the steady level.

At the separation capacity, higher amplitude fluctuation in both backorders and vehicle stocks is observed. As capacity allocation is sensitive to the stochasticity from fleet operation once the capacity is just right, a subtle change in operation may require long-term actions to recover, which leads to a different fleet behavior. However, the highest amount of fluctuations in backorders exist in the lowest ADR

action capacity, because the operation system does not even have the ability to satisfy the deterministic demands, let alone resist to the operation stochasticity. Once the capacity is enough, e.g., 120 hours, system always has spare capacity to deal with stochasticity from operation, which leads to a small variation of fleet performance.

4.4 Model Implementation

In this study, ABMs are proposed to solve operation problem of military vehicle fleets. The model can be used as a decision support tool to real-time yield operation decisions, and the model also has the capability to allow the decision makers adjust the system parameters based on different scenarios, e.g., changed terrain conditions, and tactical importance of different areas. The model provides the platform to trade off two of the major cost drivers, acquisition costs and operation costs. It also creates operational data for design engineers to improve vehicle designs.

For example, this study shows a trade-off between major fleet performance metrics, which can be used into modular vehicle design. For example, with a higher level of vehicle modularity, the number of components are more likely to be shared by other vehicles, which amplifies the pooling effect and lead to more reductions in required resources. The burden from high fleet modularity is the reduction of responsiveness in reacting to the changes of demand. A larger number of modules usually leads to a longer time in assembly and disassembly. By using the model, designers can quantify the impacts from modularity on practical mission scenarios and improve vehicle designs accordingly.

For JTTS scenario as an example, it is found that low-demand vehicles, i.e., type 11, with high commonality are formed mainly through reconfigurations, as shown in Fig. 4.8. However, high-demand vehicles, e.g., type 24, are mainly processed by assembly and disassembly actions. Combining with research of common component design problem (*Thonemann and Brandeau, 2000*), the resulted recommendations for

the modular fleet design in JTTS scenario are:

1. For low-demand vehicles, high commonality is suggested to increase pooling effect and reduce resource supply.
2. For high-demand vehicles, the design needs to focus on ADR action time reduction.

Although the model was inspired by and applied to a military mission scenario, it is applicable to the scheduling of other types of scenarios, for which demand requirements are measurable. Field demands can be treated as a type of product-as-a-service (*Mathieu, 2001*), where, the service is accomplished by operating a vehicle convoy on field. Thus, the model can be easily generalized into the civilian applications, i.e., scheduling of reconfigurable machine system (?).

4.5 Summary

The major aim of the work is to provide an efficient approach that would enable Army to gain competitive edge from fleet modularity, by integrating and coordinating the operation actions for improved awareness and responsiveness to the stochasticity in the fleet operation. To achieve this aim, a multi-layer hierarchical agent-based model is formulated, and embedded it into a high fidelity simulation environment. The model is capable to automatically output optimal dispatching, planning and resupply decisions in reacting to the current system status and dynamic field demands.

The model is implemented in the JTTS transportation mission scenario. Results show that the total resupplied resources for operating a modularized vehicle system can be reduced by more than 40 percent. The comparison between Empiricist and Optimizer reveals the importance of operation strategy on fleet performance. A sensitivity analysis is also performed to show a close correlation between the fleet performance and available ADR action capacity.

A number of interesting branches of future research remain. In this study, field demands are highly certain, especially when the time is close to the deadline. The stochasticity mainly exists in the fleet operation, i.e., repair, maintenance, which can reflect the operation for a logistical mission scenario. However, unpredictability is one of the typical features of the field demands once competing against an intelligent enemy is considered. The commander always needs to make and adjust their decisions in reacting to enemys actions under time-varying combat environment (*Army*, 2007). To fully exploit potentialities of modular fleet, one would have to formulate the competition model between the modular fleet and conventional fleet to explore the additional advantages from fleet modularity.

CHAPTER V

Attack-Defender Game

5.1 Game Formation

During the armed conflicts with Iraq and Afghanistan, the U.S. faced supply shortages due to exogenous supply chain disruptions (*Xu et al.*, 2016). This example is modeled as a competition between two military forces. In this game, each force is composed of a fleet of vehicles, i.e., fleet red and fleet blue. The goal of fleet operation is to satisfy the supply demands which randomly appear at battle fields. Each demand contains the due time, required materials, personnel, and target fleet to accomplish the supply. In order to satisfy the demand, each dispatched convoy, which are vehicles selected from fleet, needs to own enough capacities and firepower to guarantee the safety of transportation. For convenience, all demands are automatically converted into the attribute requirements for the convoy, i.e., fire power, water capacity, etc. The demands received at time t are denoted as $\mathbf{r}(t)$. According to the due time of demand, attributes required to be satisfied at a future time $t + \tau$ can be obtained as $\mathbf{d}^x(t + \tau) = [d_1^x(t + \tau), d_2^x(t + \tau), \dots, d_{N_a}^x(t + \tau)]^T$, where $d_e^x(t + \tau)$ represents the attributes of type e to be satisfied before time $t + \tau$. Matrix $\mathbf{D}^x(t)$ can also be created and updated to record the demands to be satisfied in the planning horizon T_p , i.e., $\mathbf{D}^x(t) = [\mathbf{d}^x(t + 1), \mathbf{d}^x(t + 2), \dots, \mathbf{d}^x(t + T_p)]$. Correspondingly, attributes carried by on-field convoy at time t are $\mathbf{v}^x(t) = [v_1^x(t), v_2^x(t), \dots, v_{N_a}^x(t)]^T$.

The fleet targeted by demand becomes defender, with the goal of delivering a qualified convoy to field on time. The other becomes attacker automatically, which aims to disrupt the delivery of defender by dispatching an attacker convoy. In other words, each demand initializes a supply task for one fleet and attack task for another. Based on the demand, the role of the player dynamically changes correspondingly. Common rules of the game are assumed to be known to the players: delivery of a qualified convoy to the assigned battlefield makes the defender win and attacker lose. Because the damage from the attacker can reduce the attributes of the convoy, defender may lose the game even if a qualified convoy is dispatched.

The resulted conflict between convoys is denoted as an event and assume the dispatch decision made from attacker and defender are simultaneous, which specifies the game as simultaneous-move. To simplify the problem, both fleets are assumed to be able to simultaneously sense the demands regardless of its target. Thus, given the same probability to be selected as target fleet, each fleet will play equivalent times of defender or attacker to guarantee the fairness of the game. Fig. 5.1 illustrates the convoy competition in a multiple battle fields scenario.

Following the assumptions from previous research (*Azaiez and Bier, 2007; Wang and Bier, 2011*), both attacker and defender are modeled as rational and strategic. Based on the simulation results, 10 types of dispatch strategies are summarized for attacker and defender respectively, as shown in Tab.5.1. For an attacker, firepower is the only attribute needed to win. Thus, the strategy are represented by k_a times of required firepower for enemy. For defender, they need to guarantee the delivered convoy to satisfy the demands with consideration of attribute losses from the enemy's attack. The strategy is a mixture of decisions in selecting of safety coefficients of fire power k_a and capacity k_c . All the dispatch orders that are less than the requirements are clustered as strategy 1, which means the defender gives up the game to save force once a strong enemy convoy is predicted.

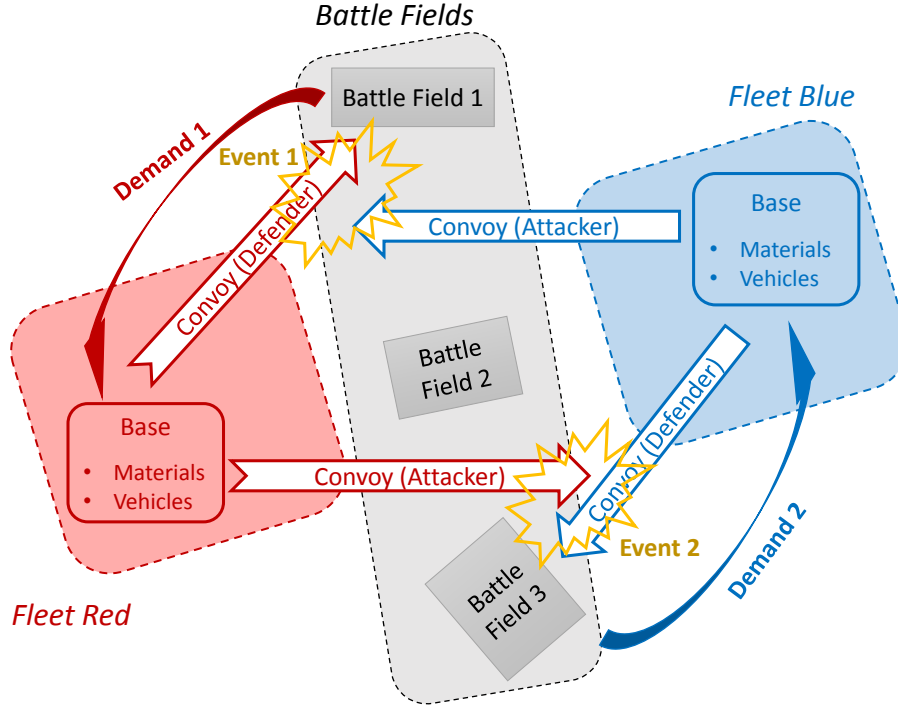


Figure 5.1: Mission scenario created for attacker-defender game

Table 5.1: Dispatch strategies for attacker and defender

Attack Strategy	1	2	3	4	5
Range of k_a	$[0, 0.5)$	$[0.5, 1)$	$[1, 1.5)$	$[1.5, 2)$	$[2, 2.5)$
Attack Strategy	6	7	8	9	10
Range of k_a	$[2.5, 3)$	$[3, 3.5)$	$[3.5, 4)$	$[4, 4.5)$	$[4.5, \infty)$
Defense Strategy	1	2	3	4	5
Range of k_a	$[0, 1)$	$[1, 1.5)$	$[1, 1.5)$	$[1, 1.5)$	$[1.5, 2)$
Range of k_c	$[0, 1)$	$[1, 1.5)$	$[1.5, 2)$	$[2, \infty)$	$[1, 1.5)$
Defense Strategy	6	7	8	9	10
Range of k_a	$[1.5, 2)$	$[1.5, 2)$	$[2, \infty)$	$[2, \infty)$	$[2, \infty)$
Range of k_c	$[1.5, 2)$	$[2, \infty)$	$[1, 1.5)$	$[1.5, 2)$	$[2, \infty)$

In this study, the amount of damage is based on the comparison of the fire power carried by different convoys. The probability of damage of type i component for team red p_{di}^r and team blue p_{di}^b are,

$$p_{di}^r = \tanh(k_{di} \frac{v_f^b}{v_f^r}) \quad (5.1)$$

$$p_{di}^b = \tanh(k_{di} \frac{v_f^r}{v_f^b}), \quad (5.2)$$

where v_f^x is the amount of firepower carried by convoy x . k_{di} represents the damage factor for a component in type i . Each component in the convoy gets damage stochastically based on the calculated probability.

In order to create a fair game, the amount of supplies are constrained and all damaged resources are assumed to be recoverable. Thus, the amount of resources for both fleets are constant, but the conditions of resources are dynamic. Vehicle damage is penalized by a long waiting time to recover. Recovery strategy for damaged vehicles is to replace all the damaged components by healthy ones. Once fleet modularity is considered, disassembly becomes another option in dealing with a badly damaged vehicle.

Several assumptions are also used to simplify this problem by reserving a reasonable level of fidelity:

1. Each fleet can accurately observe and record the damage occurred in its convoy.
2. Each fleet can accurately observe the composition of enemy's convoy in every event, i.e., number and types of vehicles.
3. Convoy will return to base immediately after task.
4. Mission success will be reported to both fleets once it is completed.
5. No other type of vehicle damage is considered besides attack.

6. All components are recoverable.
7. Damage independently occurs based on the probability of damage.
8. Inventory status is updated every hour and accessible to all the agents.
9. The sequence of vehicle assembly, disassembly, and recovery is ignored.

5.2 Attacker-Defender Model

With consideration of computational load and battlefield decision making process, an agent-based model is presented to automatically yield adaptive tactics and real-timely plan for operational actions accordingly. To simplify the notation, the approach is described from the standpoint of fleet blue, and proposed an approach to beat the enemy, i.e., fleet red. Three types of agents models are created to perform different functionalities. The decisions making process is then achieved by the cooperation of three types of agents. The interconnections are shown in Fig. 5.2.

1. **Inference Agent:** analyze enemy's historical behaviors, forecast enemy's future actions.
2. **Dispatch Agent:** optimize dispatch order based on inference.
3. **Base Agent:** optimally plan for the operation actions to satisfy the dispatch order.

5.2.1 Inference Agent

As simultaneous-move is considered, it is critical to forecast the enemy's actions to counter. As combat resources and workshop capacity are limited, it is possible to possible to get cues from the enemy's historical dispatch actions in inference. For example, if enemy dispatched a significant size of convoy in the short past, it is

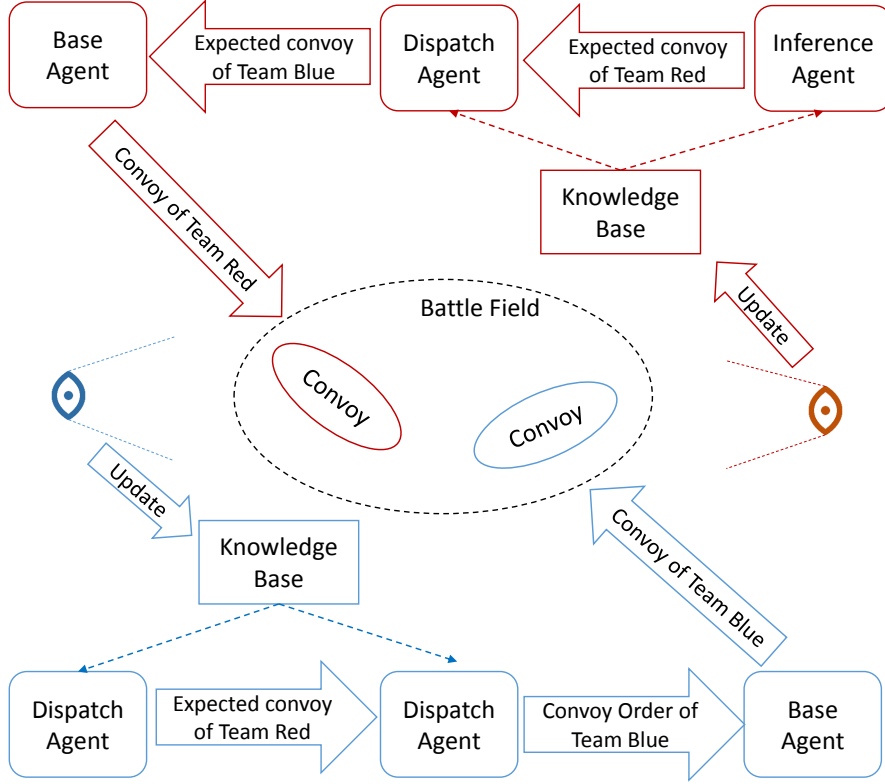


Figure 5.2: Agent based model and its information flow in attacker-defender game

possible to conclude that the enemy is not capable of grouping up a strong force in the short future. Meanwhile, existing damage in enemy's resources can also be analyzed by comparing the fire power of dispatched convoys in historical events. The amount of damage is also useful for decision maker to infer enemy's available forces.

The information that can be used for inference is very limited, including demand records $\mathbf{D}(t)$, our previous dispatch, $\mathbf{v}^b(\tau)$, and enemy's previous dispatches $\mathbf{v}^r(\tau)$. Dispatch decisions depend on optimization algorithm, inference and personality of commander, which leads to a remarkable nonlinearity in modeling the decision making process. Meanwhile, as commander needs to adjust its strategy after learning from the enemy. It requires the prediction model capable of exchanging information from outsides, and make corrections once needed. Techniques from artificial intelligence are adopted to solve this problem.

Recurrent neural network is a well known machine learning model that could capture the dependencies of time-series data (*Mikolov et al.*, 2010). Compared to the neural network, it can memorize a certain period of historical data and analyze its influence on the future. Long short-term memory model networks is one of the popular RNN, which is capable of learning long-term dependencies which is widely implemented in the neural language without gradient vanishing problem in RNN. In this study, a variant model of RNN - long short term memory (LSTM) is implemented as the predictive model to capture the correlations in enemy's sequential decisions. The model is widely used in forecasting based on sequential data, including, stock market price (*Chen et al.*, 2015; *Di Persio and Honchar*, 2016; *Fischer and Krauss*, 2017), traffic (*Ma et al.*, 2015; *Zhao et al.*, 2017), etc. In this study, the inference of enemy's strategy is modeled as a classification problem, where each class corresponds to a strategy. The inputs $\mathbf{x}^{(m)}$ of training data is the record for each event, including enemy's dispatched convoy, our dispatched convoy and received demand, which is a time-series data recording all the information during review horizon T_b . The outputs $y^{(m)}$ of the training example is the actual dispatch strategy adopted by the enemy. The architecture of LSTM used in this study is shown in Fig. 5.3.

The status of LSTM at time t are described by input gate \mathbf{i}_t , forget gate \mathbf{f}_t , output gate \mathbf{o}_t , cell state \mathbf{c}_t and activation at i^{th} hidden layer \mathbf{a}_t^i . The forward propagation can be described by

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{a}_{t-1}, \mathbf{x}_t^{(m)}] + \mathbf{b}_i), \quad (5.3)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{a}_{t-1}, \mathbf{x}_t^{(m)}] + \mathbf{b}_f), \quad (5.4)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{a}_{t-1}, \mathbf{x}_t^{(m)}] + \mathbf{b}_o), \quad (5.5)$$

$$\mathbf{c}_t = \mathbf{i}_t * \tanh(\mathbf{W}_c[\mathbf{a}_{t-1}, \mathbf{x}_t^{(m)}] + \mathbf{b}_c) + \mathbf{f}_t * \tanh(\mathbf{W}_c[\mathbf{a}_{t-2}, \mathbf{x}_{t-1}^{(m)}] + \mathbf{b}_c), \quad (5.6)$$

$$\mathbf{a}_t^0 = \mathbf{o}_t * \tanh(\mathbf{c}_t), \quad (5.7)$$

$$\mathbf{a}_t^1 = \mathbf{W}_1 \mathbf{a}_t^0 + \mathbf{b}_1, \quad (5.8)$$

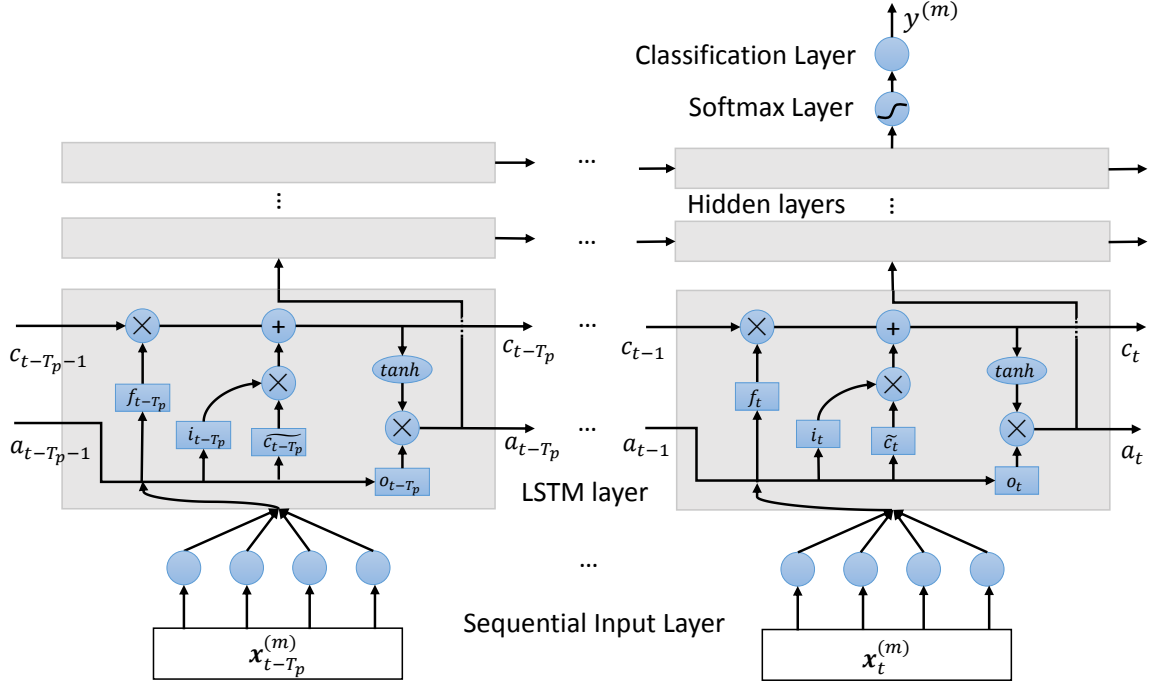


Figure 5.3: LSTM architecture used for analyze enemy's historical behaviors and forecast enemy's future actions

$$\mathbf{a}_t^2 = \mathbf{W}_2 \mathbf{a}_t^1 + \mathbf{b}_2, \quad (5.9)$$

\vdots

$$\mathbf{a}_t^{N_h} = \mathbf{W}_{N_h} \mathbf{a}_t^{N_h-1} + \mathbf{b}_{N_h}, \quad (5.10)$$

$$\mathbf{p}^{(m)} = \sigma_m(\mathbf{a}_t^{N_h}), \quad (5.11)$$

where, \mathbf{W} , \mathbf{b} are weights and biases to be obtained through training. σ is the sigmoid function. N_h is the number of hidden layer. σ_m is the softmax function. $\mathbf{p}^{(m)}$ records the estimated probability of each class based on inputs from training sample m and weights of the model, i.e., $\mathbf{p}^{(m)} = [p_1^{(m)}, p_2^{(m)}, \dots, p_{N_c}^{(m)}]^T$. The loss function is represented by a cross entropy equation:

$$E^m = - \sum_{i=1}^{N_c} \gamma_i \log(p_i^{(m)}), \quad (5.12)$$

where, γ_i is a binary indicator (0 or 1), with value 1 if class label i is the correct classification and 0 otherwise. Thus, the training of the model is to minimize the sum of entropy of the training set to find the best model parameters through backward propagation (*Hecht-Nielsen*, 1992), i.e.,

$$\min_{\mathbf{w}, \mathbf{b}} \sum_m E^m, \quad (5.13)$$

thus, enemy's behavior can be forecasted through Eq.5.14.

$$\bar{y}^r = \{i \mid 1 \leq i \leq N_c, \forall 1 \leq j \leq N_c : p_i^{(m)} \geq p_j^{(m)}\}. \quad (5.14)$$

Based on the predicted enemy's strategy \bar{y}^r , the possible enemy's dispatch order $\bar{\mathbf{v}}^r$ can be calculated by using the upper bound of strategy of Tab. 5.1.

5.2.2 Dispatch Agent

The goal of convoy dispatch is to determine the desired attributes that need to be carried by our convoy to maximize the win rate. Based on game formation, a convoy with higher attributes, especially in fire power, indicates a higher chance to win. However, as resources are limited, the less vehicles are ordered, the higher chance that the order can be achieved by the inventory planner. The convoy dispatch should be carefully planned to guarantee the win rate of current mission without overdrawing resources.

To avoid the overuse or underutilization of available attributes, it is important to have an accurate evaluation of our current forces before making dispatch decisions. Available attributes in the future depend on inventory status and on-base action scheduling. As time delays exist in operations and future demands are only partially observable. It is hard to infer the actual planning that will be made in the short future, which may change the available attributes at dispatching time totally. This

difficulty becomes one of the challenges in placing the dispatch order. Even our convoy order can be achieved by base, it is still uncertain for them to determine the probability that this convoy can win. Because all the players do not know how the vehicle gets damaged, it requires players to speculate the damage mechanism based on the experience.

To resolve this problem, the estimation of event success is disintegrated in two parts, which are feasibility of order p_f^b and conditional success rate if order is feasible p_s^b . The probability of winning an event for convoy blue p_w^b can be calculated.

$$p(\text{win}) = p(\text{win}|\text{order is feasible})p(\text{order is feasible}) \quad (5.15)$$

$$p_w = p_s^b p_f^b.$$

5.2.2.1 Feasibility

Based on the historical records, the feasibility of order can be flagged by 0(infeasible) and 1 feasible based on the comparison between dispatch order $\mathbf{do}^b(t)$ and actually dispatched convoy $\mathbf{o}_v^b(t)$. The order is denoted as feasible if $\mathbf{do}^b(t) \leq \mathbf{M}_{va}\mathbf{o}_v^b(t)$, is infeasible otherwise. As optimizations are implemented in operation planning, the relationship between factors and feasibility is complex and nonlinear. A neural network model (*Hagan et al.*, 1996; *Atsalakis et al.*, 2018; *Rezaee et al.*, 2018) is implemented to capture these nonlinear inter-connections, as shown in Fig. 5.4. The outputs of training set is the feasibility of the order (1 for feasible, 0 for infeasible).

With enough training, the model is capable of evaluating the feasibility of dispatch order across diverse operation situations. To capture the changing of inventory operation strategy, the model is periodically retrained based on the latest operation information. The relationship between model inputs and feasibility rate can be de-

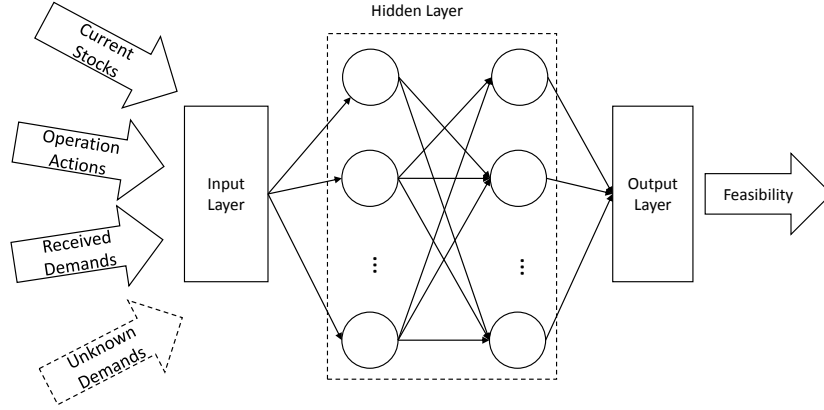


Figure 5.4: Neural network model for feasibility training

scribed by Eq. 5.16,

$$p_f = f_f(\mathbf{do}^b(t), \mathbf{s}^b(t), \delta \mathbf{s}^b(t+1), \dots, \delta \mathbf{s}^b(t+T_p), \mathbf{D}^b(t)), \quad (5.16)$$

where, $\mathbf{s}^b(t)$ records the number of vehicles and component stocks on base at time t . $\delta \mathbf{s}(t+\tau)$ is the changes in inventory stocks at time $t+\tau$ from unfinished actions.

5.2.2.2 Conditional Success Rate

Vehicle damage plays an important role in determining the success of a mission. However, it is driven by a stochastic process and will vary according to the changes in terrain, operational preparations and soldiers' reactions. It requires the model to be able to capture the complexity in the damage mechanism. Combining with the nonlinearity in predicating the success, another neural network model is adopted for success rate forecasting, as shown in Fig. 5.5.

Similarly, the outputs of training set are success reports of previous events (1 for success, 0 for fail). Given forecasted enemy's convoy attributes $\bar{\mathbf{v}}^r(t)$, the trained model will yield conditional win rate for a certain dispatch order and mission requirements. The model is capable of capturing the changes in damage mechanism by

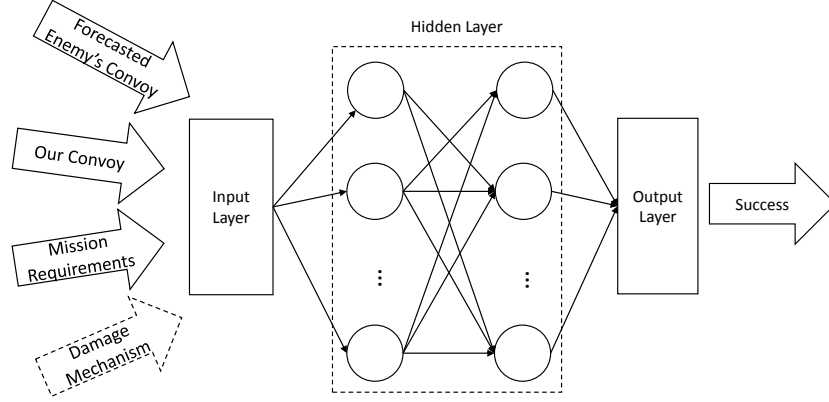


Figure 5.5: Neural network model for success rate training

continuously feeding in the latest combat information and results. By denoting the trained neural network model for success as f_s^b , the probability of success can be also calculated.

$$p_s^b = f_s^b(\mathbf{do}^b(t), \bar{\mathbf{v}}^r(t), \mathbf{D}(t)). \quad (5.17)$$

5.2.2.3 Optimization

For each dispatch order $\mathbf{do}^b(t)$, the above approach provides the way to estimate the probability of success and feasibility based on predicted enemy's behavior, demand information and inventory status. An optimization model can be used to seek the optimal dispatch order to maximize win rate or minimize failure rate, i.e., $J = 1 - p_b^w$. Combining with Eqn. 5.16, 5.17, a nonlinear programming model can be formulated to seek the optimal dispatch order.

$$\begin{aligned} \min_{\mathbf{do}^b(t)} \quad & 1 - [f_f(\mathbf{do}^b(t), \mathbf{s}^b(t), \delta \mathbf{s}^b(t+1), \dots, \delta \mathbf{s}^b(t+T_p), \mathbf{D}(t))][f_s^b(\mathbf{do}^b(t), \bar{\mathbf{v}}^r(t), \mathbf{D}(t))] \\ \text{s.t.} \quad & (a) \ \mathbf{do}^b(t) \geq 0, \end{aligned} \quad (5.18)$$

where, $\mathbf{do}^b(t)$ is the decision variable that specifies the desired attributes to be carried by the convoy. Thus, the number of decision variables is the number of attribute types ($3 \sim 10$). However, because of the non-convexity in objective function, it is intractable to get global optimum by the gradient-based approach. In this study, a pattern search technique is implemented to yield optimal dispatch decisions.

As the minimized failure rate can be any value in the range of $[0, 1]$, dispatch agent should be capable of giving up the mission once a very high failure rate is calculated. There is also a stream of literature studying risk preferences in repeated and evolutionary games (*Roos and Nau, 2010; Lam and Leung, 2006; Zhang et al., 2018*). The ϵ_f is defined as a customizable parameter to represent the minimal failure rate that can be tolerated, which filters the dispatch order $\mathbf{do}^b(t)$ as

$$\mathbf{do}^b(t) = \begin{cases} = \mathbf{0}, & \text{for } (1 - p_w) > \epsilon_f, \\ = \mathbf{do}^b(t) & \text{for } (1 - p_w) \leq \epsilon_f. \end{cases} \quad (5.19)$$

Thus, a convoy can be dispatched only when commander is confident enough. In this study, risk aversion behavior is purely related to ϵ which is constant during operation. As a future work, it is also interesting to vary ϵ to seek an advanced fleet operation strategy, i.e., combination of risk-prone and risk-averse (*Roos and Nau, 2010*).

5.2.3 Base Agent

Based on the behavior analysis from inference agent and dispatch order suggestion from dispatch agent, base agent is the one to plan operational actions to accomplish the orders. Li et al. proposed a model predictive control based approach to real-time schedule the operation actions in reacting to the received demands. However, they did not consider the possible damage that occurs during the fleet operation (*Li and Epureanu, 2018*). In this section, I further their research by considering

the possible damage during fleet operation and manage the inventory based on the resulting diverse conditions. For convenience, the notation of operation actions for fleet x is simplified from \mathbf{o}^x to \mathbf{o} in this section as no enemy is considered.

As resources for each player are limited and repairable, it is important to schedule the operation actions properly to recover damaged resources and increase utility rate. It is also essential to allocate the capacity properly to balance between order satisfaction and damage recovery. In this section, military fleet operation is first modeled as a time-varying dynamical system. Then, a model predictive control is proposed to manage system dynamics thus achieving operation management.

5.2.3.1 Dynamical System

The dynamics in fleet operation is mainly located at the changes of inventory stocks and remaining demands, in terms of

1. Vehicle stocks, $\mathbf{I}_v = [I_{v_1}, I_{v_2}, \dots, I_{N_v}]$
2. Module/component stocks, $\mathbf{I}_m = [I_{m_1}, I_{m_2}, \dots, I_{N_m}]$
3. Damaged vehicles, $\mathbf{I}_{dv} = [I_{dv_1}, I_{dv_2}, \dots, I_{N_{dv}}]$
4. Damaged components, $\mathbf{I}_{dc} = [I_{dc_1}, I_{dc_2}, \dots, I_{N_c}]$
5. Unsatisfied demands, $\mathbf{I}_a = [I_{a_1}, I_{a_2}, \dots, I_{N_a}]$.

Although healthy vehicles and damaged vehicles are recorded in the similar structures, their meanings are totally different. For healthy stocks and damaged components, the subscript of variable is the type of vehicle/component; the value of variable indicates the number. For damaged stocks, the subscript is the index of damaged vehicles, which is created based on the vehicle receive date. Binary values are used to represent the status of the vehicle, where, 1 represents that the damaged stocks remain to be repaired; 0 indicates the damaged stock is recovered or not received yet.

Vehicle type $\mathbf{v}_{lt}(t)$, number of damaged components $\mathbf{v}_{ldc}(t)$ and healthy components $\mathbf{v}_{lc}(t)$ in the damaged vehicle of type l are also recorded as a reference for the repair. These data are time variant as the number and type of damaged vehicles keep changing with newly occurred damage and vehicle recovery. These data structure can bypass the numerous states incurred from diverse vehicle damage patterns, as the vehicles with different damage are distinguished as different damaged vehicles. A state will be created for each newly arrived damaged vehicle and removed once the damaged vehicle is recovered, i.e., state value changes from 1 to 0.

Vehicle conditions are reported to base agent as one of the inputs, all inputs to the system are summarized as

1. Returning healthy vehicles, $\mathbf{a}_r = [a_{r_1}, a_{r_2}, \dots, a_{r_{N_v}}]$,
2. Returning damaged vehicles, $\mathbf{a}_d = [a_{d_1}, a_{d_2}, \dots, a_{d_{N_{dv}}}]$,
3. Dispatch order from dispatch agent, $\mathbf{do} = [do_1, do_2, \dots, do_{N_a}]$.

Based on the characteristics of fleet operation, the operational actions to be determined are also distinct. For conventional fleet, the operation actions include

1. Convoy dispatch, $\mathbf{o}_v = [o_{v_1}, o_{v_2}, \dots, o_{v_{N_v}}]$
2. Recovery of damaged vehicle, $\mathbf{o}_{dr} = [o_{dr_1}, o_{dr_2}, \dots, o_{dr_{N_{dv}}}]$
3. Recovery of damaged component, $\mathbf{o}_c = [o_{c_1}, o_{c_2}, \dots, o_{N_c}]$.

The dynamics of vehicle stocks of type k , component stocks of type i , damaged vehicle of index l , damaged components of type i , and remaining attributes of type h are shown by Eq. 5.20, 5.21, 5.22, 5.23, 5.24 respectively

$$I_{v_k}(t+1) = I_{v_k}(t) - o_{v_k}(t) + a_{r_k}(t) + \sum_l v_{lkt}(t) o_{dr_l}(t - \tau_{v_l}), \quad (5.20)$$

$$I_{c_i}(t+1) = I_{c_i}(t) + o_{c_i}(t - \tau_{c_i}) - \sum_l v_{lc_i}(t) o_{dr_l}(t), \quad (5.21)$$

$$I_{dv_l}(t+1) = I_{dv_l}(t) - o_{dr_l}(t) + a_{d_l}(t), \quad (5.22)$$

$$I_{dc_i}(t+1) = I_{dc_i}(t) - o_{c_i}(t) + \sum_l v_{ldc_i}(t) o_{dr_l}(t - \tau_{dr_l}), \quad (5.23)$$

$$I_{a_h}(t) = do_h(t) - \sum_k M_{v_k a_h} o_{v_k}(t). \quad (5.24)$$

By introducing fleet modularity, several additional operation actions are available, in terms of

1. Vehicle assembly, $\mathbf{o}_a = [o_{a_1}, o_{a_2}, \dots, o_{a_{N_v}}]$
2. Vehicle disassembly, $\mathbf{o}_d = [o_{d_1}, o_{d_2}, \dots, o_{d_{N_v}}]$
3. Vehicle reconfiguration, $\mathbf{o}_r = [o_{12}, o_{13}, \dots, o_{N_v N_v - 1}]$
4. Damaged vehicle disassembly, $\mathbf{o}_{dd} = [o_{dd_1}, \dots, o_{dd_{N_{dv}}}]$.

With consideration of these actions, dynamic equations become

$$\begin{aligned} I_{v_k}(t+1) = & I_{v_k}(t) - o_{v_k}(t) + a_{r_k}(t) + \sum_{k' \neq k} o_{k'k}(t - \tau_{k'k}) - \sum_{k' \neq k} o_{kk'}(t) \\ & + \sum_l v_{lkt}(t) o_{dr_l}(t - \tau_{dr_l}) + o_{a_k}(t - \tau_{a_k}) - o_{d_k}(t), \end{aligned} \quad (5.25)$$

$$\begin{aligned} I_{c_i}(t+1) = & I_{c_i}(t) + o_{c_i}(t - \tau_{c_i}) - \sum_k M_{v_k c_i} [o_{a_k}(t) - o_{d_k}(t - \tau_{d_k}) \\ & + \sum_{k' \neq k} o_{k'k}(t) - \sum_{k' \neq k} o_{kk'}(t - \tau_{kk'})] + \sum_l v_{lc_i}(t) o_{dd_l}(t - \tau_{dd_l}), \end{aligned} \quad (5.26)$$

$$I_{dv_l}(t+1) = I_{dv_l}(t) - o_{dr_l}(t) - o_{dd_l}(t) + a_{d_l}(t), \quad (5.27)$$

$$I_{dc_i}(t+1) = I_{dc_i}(t) - o_{c_i}(t) + \sum_l v_{ldc_i}(t) o_{dr_l}(t - \tau_{dr_l}) + \sum_l v_{ldc_i}(t) o_{dd_l}(t - \tau_{dd_l}) \quad (5.28)$$

$$I_{a_h}(t) = do_h(t) - \sum_k M_{v_k a_h} o_{v_k}(t). \quad (5.29)$$

Because of the delays in operation actions, current inventory stocks might be influenced by previously-determined actions. In other words, the current actions may impact the stock level in the future. Thus, the states of the system are defined by all inventory statuses that might be influenced by current actions, $\mathbf{s}(t)$, i.e.,

$$\mathbf{s}(t) = [\mathbf{I}_t(t), \mathbf{I}_{t+1}(t), \mathbf{I}_{t+2}(t), \mathbf{I}_{t+3}(t) \dots \mathbf{I}_{t+\tau_{max}}(t)]^T. \quad (5.30)$$

Input matrices \mathbf{B}_τ are used to connect the current actions at time t to inventory level at a later time $t + \tau$. Furthermore, damage on stocks keep changing along with time. The matrices that connect to previous states \mathbf{A} , actions \mathbf{B}_τ and inputs \mathbf{C} are also time-varying matrices. Thus, the system dynamics for both fleets can be written as

$$\mathbf{I}(t+1) = \mathbf{I}(t) + \sum_{\tau} [\mathbf{B}_\tau^c(t) \mathbf{o}^c(t-\tau)] + \mathbf{C}^c(t) \mathbf{a}(t-\tau), \quad (5.31)$$

$$\mathbf{I}(t+1) = \mathbf{I}(t) + \sum_{\tau} [\mathbf{B}_\tau^m(t) \mathbf{o}^m(t-\tau)] + \mathbf{C}^m(t) \mathbf{a}(t-\tau), \quad (5.32)$$

where

$$\mathbf{o}^c(t) = [\mathbf{o}_v(t), \mathbf{o}_{dr}(t), \mathbf{o}_c(t)], \quad (5.33)$$

$$\mathbf{o}^m(t) = [\mathbf{o}_v(t), \mathbf{o}_{dr}(t), \mathbf{o}_c(t), \mathbf{o}_a(t), \mathbf{o}_d(t), \mathbf{o}_r(t), \mathbf{o}_{dd}(t)]. \quad (5.34)$$

Thus, a state space model can be created to record the influence from the actions at a single time point to the states in the short future, as shown in Eq. 5.35

$$\mathbf{s}(t+1) = \mathbf{A}(t) \mathbf{s}(t) + \mathbf{B}(t) \mathbf{u}(t) + \mathbf{C}(t) \mathbf{a}(t), \quad (5.35)$$

$$\mathbf{A}(t) = \begin{bmatrix} \mathbf{0}_{n_s(t) \times n_s(t)} & \mathbf{I}_{n_s(t) \times n_s(t)} & \mathbf{0}_{n_s(t) \times n_s(t)} & \dots & \mathbf{0}_{n_s(t) \times n_s(t)} \\ \mathbf{0}_{n_s(t) \times n_s(t)} & \mathbf{0}_{n_s(t) \times n_s(t)} & \mathbf{I}_{n_s(t) \times n_s(t)} & \dots & \mathbf{0}_{n_s(t) \times n_s(t)} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{0}_{n_s(t) \times n_s(t)} & \mathbf{0}_{n_s(t) \times n_s(t)} & \mathbf{0}_{n_s(t) \times n_s(t)} & \dots & \mathbf{I}_{n_s(t) \times n_s(t)} \\ \mathbf{0}_{n_s(t) \times n_s(t)} & \mathbf{0}_{n_s(t) \times n_s(t)} & \mathbf{0}_{n_s(t) \times n_s(t)} & \dots & \mathbf{I}_{n_s(t) \times n_s(t)} \end{bmatrix}, \quad (5.36)$$

$$\mathbf{B}(t) = [\mathbf{B}_0(t), \mathbf{B}_1(t), \mathbf{B}_2(t), \dots, \mathbf{B}_{\tau_{max}}(t)]^T, \quad (5.37)$$

$$\mathbf{C}(t) = [\mathbf{C}_0(t), \mathbf{C}_1(t), \mathbf{C}_2(t), \dots, \mathbf{C}_{\tau_{max}}(t)]. \quad (5.38)$$

5.2.3.2 System Control

The goal of system control is to meet the received dispatch orders on time. In the decision making process, the predictions of future system are always involved. For example, given several dispatch orders, one may want to know what are the influences from satisfying one order on others. Compared to the classical control methodologies, e.g., PID control, MPC makes better use of future information and adapts to the system changes (*Li and Epureanu, 2017b*). This section is separated into two parts, future state prediction and optimization of operation decisions.

Future State Prediction

Because of time delays in the operation actions, the operation decisions made at the current time have to guarantee the match between the attributes of the dispatched convoy and ordered attributes. Given

1. current system states $\mathbf{s}(t) = [\mathbf{s}(t+1), \mathbf{s}(t+2), \mathbf{s}(t+3), \dots, \mathbf{s}(t+t_p)]^T$
2. operation actions in the future $\underline{\mathbf{q}}_t = [\mathbf{o}(t), \mathbf{o}(t+1), \mathbf{o}(t+2), \dots, \mathbf{o}(t+t_p-1)]^T$

3. system input, $\underline{\mathbf{a}}_t = [\mathbf{a}(t), \mathbf{a}(t+1), \mathbf{a}(t+2), \dots, \mathbf{a}(t+t_p-1)]^T$,

The future system states $\underline{\mathbf{s}}_{t+1}$ are predictable by iteratively substituting Eq. 5.35.

Thus, the $\underline{\mathbf{s}}_{t+1}$ can be expressed as a function of $\underline{\mathbf{q}}_t$

$$\underline{\mathbf{s}}_{t+1} = \mathbf{P}(t)\mathbf{s}(t) + \mathbf{H}(t)\underline{\mathbf{q}}_t + \mathbf{G}(t)\underline{\mathbf{a}}_t \quad (5.39)$$

$$\mathbf{P}(t) = [\mathbf{A}(t), \mathbf{A}^2(t), \mathbf{A}^3(t), \dots, \mathbf{A}^n(t)], \quad (5.40)$$

$$\mathbf{H}(t) = \begin{bmatrix} \mathbf{B}(t) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}(t)\mathbf{B}(t) & \mathbf{B}(t) & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{A}(t)^{n-1}\mathbf{B} & \mathbf{A}^{n-2}(t)\mathbf{B}(t) & \dots & \mathbf{B}(t) \end{bmatrix}, \quad (5.41)$$

$$\mathbf{G}(t) = \begin{bmatrix} \mathbf{C}(t) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}(t)\mathbf{C}(t) & \mathbf{C}(t) & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{A}(t)^{n-1}\mathbf{B} & \mathbf{A}^{n-2}(t)\mathbf{C}(t) & \dots & \mathbf{C}(t) \end{bmatrix}, \quad (5.42)$$

with $\mathbf{P}(t)$ being the matrix that connects the future system outputs with current system states, and $\mathbf{H}(t), \mathbf{G}(t)$ being the matrix that connects the system outputs with the future operation actions and inputs respectively. Although the dynamical system is changing along with time, it is assumed as constant at each decision making time. The system keeps updating to ensure the optimization is based on accurate system status.

Cost Function

The optimization of fleet operation originates from two facts: 1. convoy with insufficient attributes suffer a remarkable risk in losing the mission; 2. convoy with redundant attributes can also deteriorate the overall fleet performance from utility reduction. Furthermore, several operational costs that may be significant in the real-world fleet operation are also considered. As a summary, the costs of interest are,

1. Attribute redundancy cost, c_{o_h}
2. Attribute insufficiency cost, c_{u_h}
3. ADR action cost, $c_{a_k}, c_{d_k}, c_{kk'}$
4. Recovery cost, $c_{c_i}, c_{dr_l}, c_{dd_l}$
5. Inventory holding cost, $c_{hc_i}, c_{hv_k}, c_{hdc_i}, c_{hdv_l}$

Therefore, the cost function is shown as

$$\begin{aligned}
 J = & \sum_{\tau=t}^{\tau=t+T_p-1} \left[\sum_h c_{o_h} I_{a_h}^+(\tau) + \sum_h c_{u_h} I_{a_h}^-(\tau) + \sum_l (c_{dr_l} o_{dr_l}(\tau) + c_{dd_l} o_{dd_l}(\tau)) \right. \\
 & \left. + \sum_k [c_{a_k} o_{a_k}(\tau) + c_{d_k} o_{d_k}(\tau)] + \sum_{k' \neq k} c_{kk'} o_{kk'}(\tau) \right] + \sum_i c_{c_i} o_{c_i}(\tau) \\
 & + \sum_{\tau=t+1}^{\tau=t+T_p} \left[\sum_i c_{hc_i} I_{c_i}(\tau) + \sum_k c_{hv_k} I_{v_k}(\tau) + \sum_i c_{hdc_i} I_{dc_i}(\tau) + \sum_l c_{hdv_l} o_{dv_l}(\tau) \right],
 \end{aligned} \tag{5.43}$$

where $I_{a_h}^+(\tau), I_{a_h}^-(\tau)$ are non-negative variables created to remove the nonlinearity, which satisfies that

$$I_{a_h}(\tau) = I_{a_h}^+(\tau) - I_{a_h}^-(\tau). \tag{5.44}$$

The insufficient and redundant attributes during the planning horizon are recorded as $\underline{I}_{\vec{a}}^+$ and $\underline{I}_{\vec{a}}^-$ respectively. The holding costs and actions related costs are also

aggregated as \underline{c}_h and \underline{c}_q . By substituting in Eq.5.39, a mixed-integer programming model is created to optimize operational decisions

$$\begin{aligned}
& \min_{\underline{I}_a^+, \underline{I}_a^-, \underline{q}_t} \quad \underline{c}_o \underline{I}_a^+ + \underline{c}_u \underline{I}_a^- + \underline{c}_h \underline{X}_s [\underline{P}(t) \underline{s}(t) + \underline{H}(t) \underline{q}_t] \\
& \text{s.t.} \quad (a) \quad \underline{q}_t \geq 0 \text{ and integer} \\
& \quad \quad (b) \quad \underline{X}_s \underline{s}_{t+1} \geq 0 \\
& \quad \quad (c) \quad \sum \underline{o}(t) \leq \bar{P}, \quad \forall t \\
& \quad \quad (d) \quad \underline{I}_a^+ - \underline{I}_a^- = \underline{X}_a [\underline{P}(t) \underline{s}(t) + \underline{H}(t) \underline{q}_t + \underline{G}(t) \underline{a}_t] \\
& \quad \quad (e) \quad o_{dl}(t) + o_{dr}(t) \leq 1, \quad \forall t,
\end{aligned} \tag{5.45}$$

where, $\underline{X}_s, \underline{X}_a$ are index of inventory stocks and remaining dispatch orders in states respectively. Constraint (a) ensures that all operational decisions are non-negative and integer; (b) indicates that the amount of inventory stocks are non-negative; (c) ensures that the on-base ADR actions are always constrained by the maximum action capacity \bar{P} ; (d) preserves the balance between auxiliary variables and remaining orders to be satisfied; (e) specifies that each damaged vehicle can only be recovered by one recovery strategy. As cost function and constraints are linear and the number of decision variables is huge, a cutting-plan is implemented first to reduce the decision space and then use the integer programming solver to get the solution. Time required for decision making of each time point is less than 1 second for operating 5 types of modular vehicles with planning horizon as 12 hours.

5.3 Numerical Illustrations

In this section, numerical illustrations are provided in a generalized mission scenario to study the different impacts of modularity on fleet performance. In general, it may be difficult to estimate the parameters accurately. However, it may be possible

to get reasonable estimates for these parameters by using expert judgments and data from the existing literature. In this study, the resources provided to fleet operation is constant and equal, which can be imagined as a competition of two fleets at an isolated island. One of them is conventional fleet; the other is modular fleet. Initially, ten of each type of vehicle and component are provided to both fleets. Demands randomly occur at battle field based on Poisson distribution with time interval as 10 hours. Demands include personnel capacity d_p , material capacity d_m and fire power d_f , which are generated based on Gaussian distribution as shown in Eq. 5.46, 5.47, 5.48. Because of the lack of diversity in the existing designs of modular vehicles, five types of modular vehicles as well as six resulted modules are borrowed from (*Li and Epureanu, 2017b*). The attributes carried by each vehicle are summarized in Tab. 5.2.

$$d_p \sim \mathcal{N}(40, 15) \quad (5.46)$$

$$d_m \sim \mathcal{N}(50, 20) \quad (5.47)$$

$$d_f \sim \mathcal{N}(30, 10) \quad (5.48)$$

Table 5.2: Mapping between vehicles and attributes

Vehicle Type	1	2	3	4	5
Fire power	1	3	8	0	6
Material capacity	2	6	2	2	8
Personnel capacity	4	1	0	10	5

The costs of insufficiency and redundancy are created based on the heuristic rules. For example, convoy usually suffers a high risk of failure once attributes of dispatched

convoy are less than ordered. Thus, the cost for attribute insufficiency is assigned much higher than attribute redundancy, i.e., $c_{oh} \gg c_{uh}, \forall h$. The costs for operation actions are created based on the difficulties and time required.

The time required for module assembly and disassembly are assigned as constant vectors τ_{ma}, τ_{md} . Vehicle assembly/disassembly time τ_{va}, τ_{vd} is calculated by summing up all the time required for its components. Similarly, for repair and reconfiguration, all the actions required to process each individual components in the vehicle are firstly summed. It is also assumed that the interface between components are well-designed to achieve quick vehicle reconfiguration, where assembly and disassembly time for all types of components are 1 hour and 0.5 hour respectively. On-base ADR actions are assumed to be proceeded in a generalized work station, thus, the number of stations determines the amount of available capacity. In this study, the number of available work stations for both fleets is assigned as 10.

A discrete event model is used to simulate fleet competition for three years. The mission is separated into two parts, which are stochastic stage (1st year) and learning stage (2nd and 3rd year). In stochastic stage, dispatch agent randomly picks up a dispatch strategy based on Tab.5.1 and passes this decisions to base agent. First-year operations generate time-series data, including combat history, feasibility records, etc., which are important inputs for the learning model. Training of learning model starts at the beginning of learning stage, where inference agent and dispatch agent make decisions based on the historical enemy's behavior. Learning models are also updated monthly to ensure they reflect the enemy's latest behavior.

5.4 Fleet Comparison

In this section, the fleet performance is compared between modular fleet and conventional fleet in different stages. As one of the important metrics in measuring the fleet performance, the probability of win is firstly compared based on the results from

multiple simulation, which is shown in Fig.5.6. According to the plot, conventional fleet outperforms modular fleet at the stochastic stage. However, once the intelligence of agent is introduced, i.e., both fleets entered learning stage, modular fleet gradually receives more wins of the game. A well separation can also be noticed along with the learning stage, which indicates a solid leading position of modular fleet during the learning stage. To explain these results, the attributes carried by the actual dispatched convoy between fleets are compared, as well as the estimation accuracy in inferring enemy, order success and feasibility.

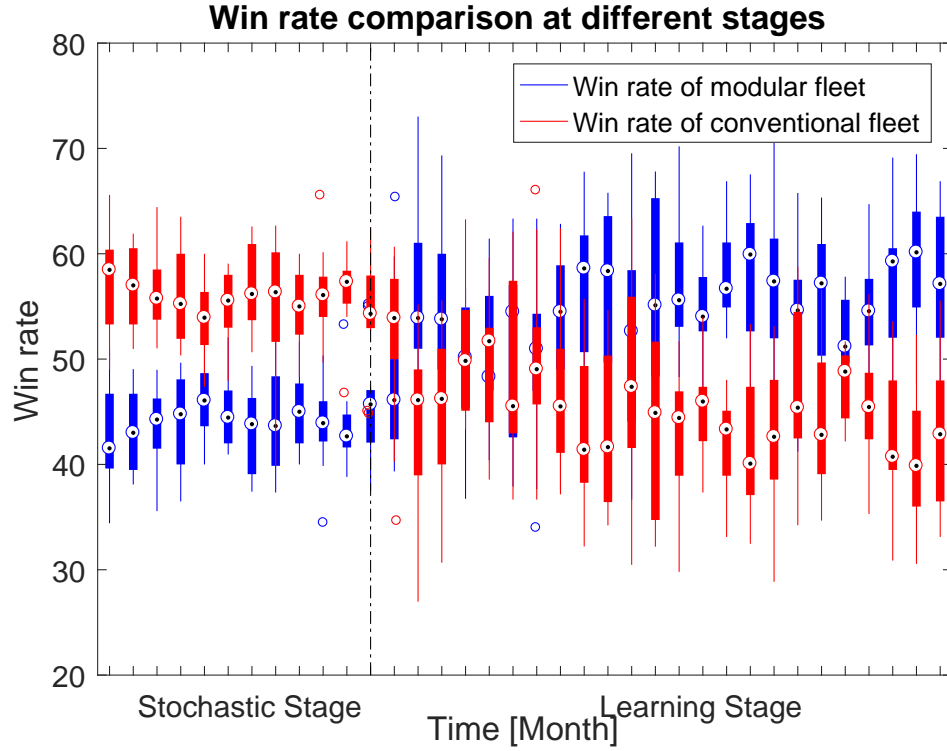


Figure 5.6: Comparison of win rate between modular fleet and conventional fleet

During the stochastic stage, dispatch agents from both teams place dispatch orders based on randomly selected strategy. The way of strategy selection and order achievement are also equivalent for both fleets. In order to explain the better performance of conventional fleet, it is necessary to explore the accuracy of both fleets in satisfying dispatch order. Mismatched attributes dramatically change the fleet per-

formance: convoy with insufficient attributes may significantly raise the failure rate and damage; convoy with redundant attributes may increase the win rate slightly, but it can also contribute to the insufficiency in the short future because of the limitation of resources. Thus, the amount of overused and insufficient convoy attributes during every month is calculated to compare the dispatch accuracy in Fig. 5.7.

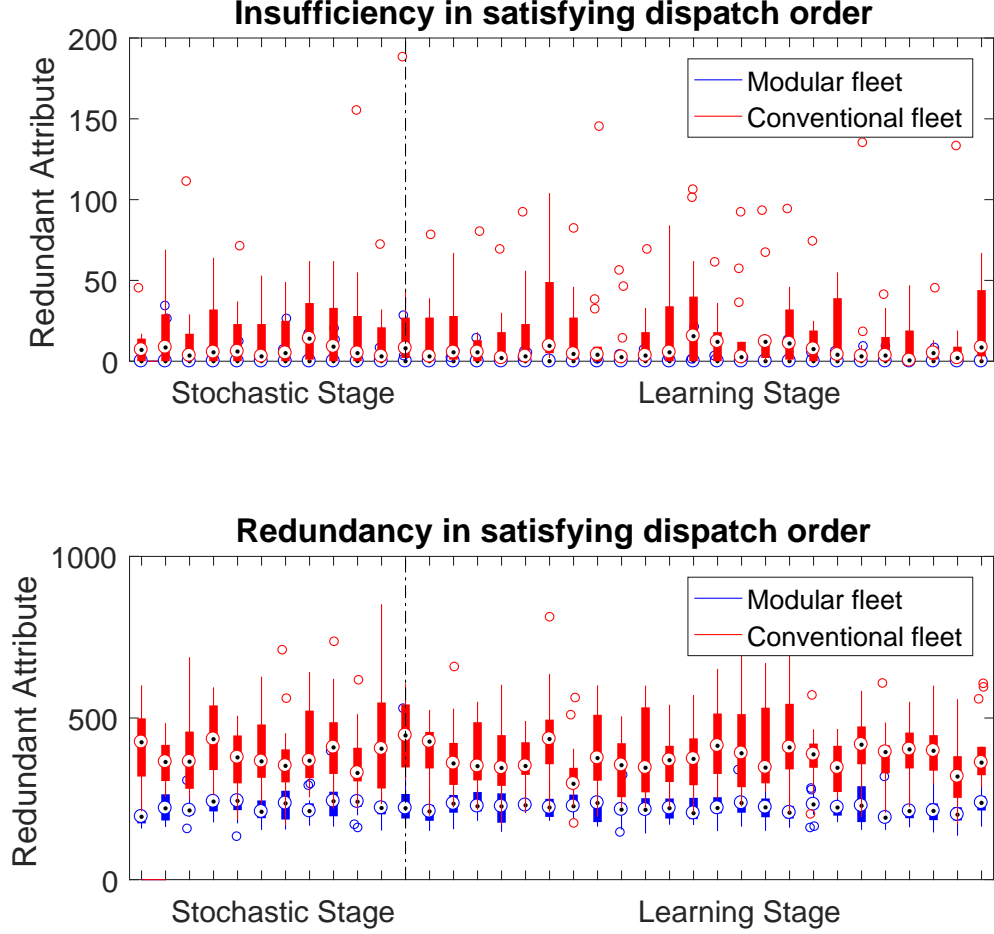


Figure 5.7: Comparison of mismatched convoy attributes

Compared to the modular fleet, conventional fleet suffers remarkable redundancy. The higher redundancy comes from the rigidity of conventional fleet operation. As we know, modular fleet can real timely reconfigure itself to fit the dispatch order, however, conventional fleet can only wait for vehicles returned from field or recovery. This limitation is proven to hamper the ability of conventional fleet in satisfying

the dispatch order. Once proper vehicles are scarce, conventional fleet has to use improper vehicles with little desired attributes to avoid the insufficiency. This rigidity in fleet operation is beneficial in improving the success rate during the stochastic stage, because its opponent cannot be aware of the unexpected additional attributes. However, once the opponent start to study the behavior, this advantage no longer exists. From the failures at stochastic stage, modular fleet realize that conventional fleet intents to dispatch convoy with superfluous attributes. As a solution, they increase the attributes of dispatch order correspondingly. These redundant attributes are powerless in reacting to a well-armed enemy's convoy to make the conventional fleet stay the lead.

Besides the better understanding of enemy's behavior, intelligent agents also improve their understandings of the game along with time. To address the changes, the ability of convoy is denoted as maximum attributes can be achieved at each month, and compare the ability between both fleets in Fig.5.8. After entering the learning stage, both fleets raise the ability of convoy in all types of attributes, especially in firepower. Once modular fleet realize the importance of fire power to win the event by model f_s^b , the combat vehicles are rapidly formed from reconfiguration to boost the available fire power.

The swift reconfiguration of modular fleet lead to a dramatic increase of damage to the enemy in the first few months of learning stage, as shown in Fig.5.9. Although conventional fleet intend to increase the fire power to fight back, the limitation in vehicle structure results in a lower upper limit in the convoy ability. Thus, the difference in ability makes conventional fleet suffer higher damage from more dispatch, which forces the conventional fleet to operate in sub-healthy conditions for a long time.

The strategies used in learning stage are also distinct between two fleets. Fig.5.10 compares the proportion of strategies adopted by different fleets. After learning of game, both fleets prefer to select the defense strategy with large amount of firepower

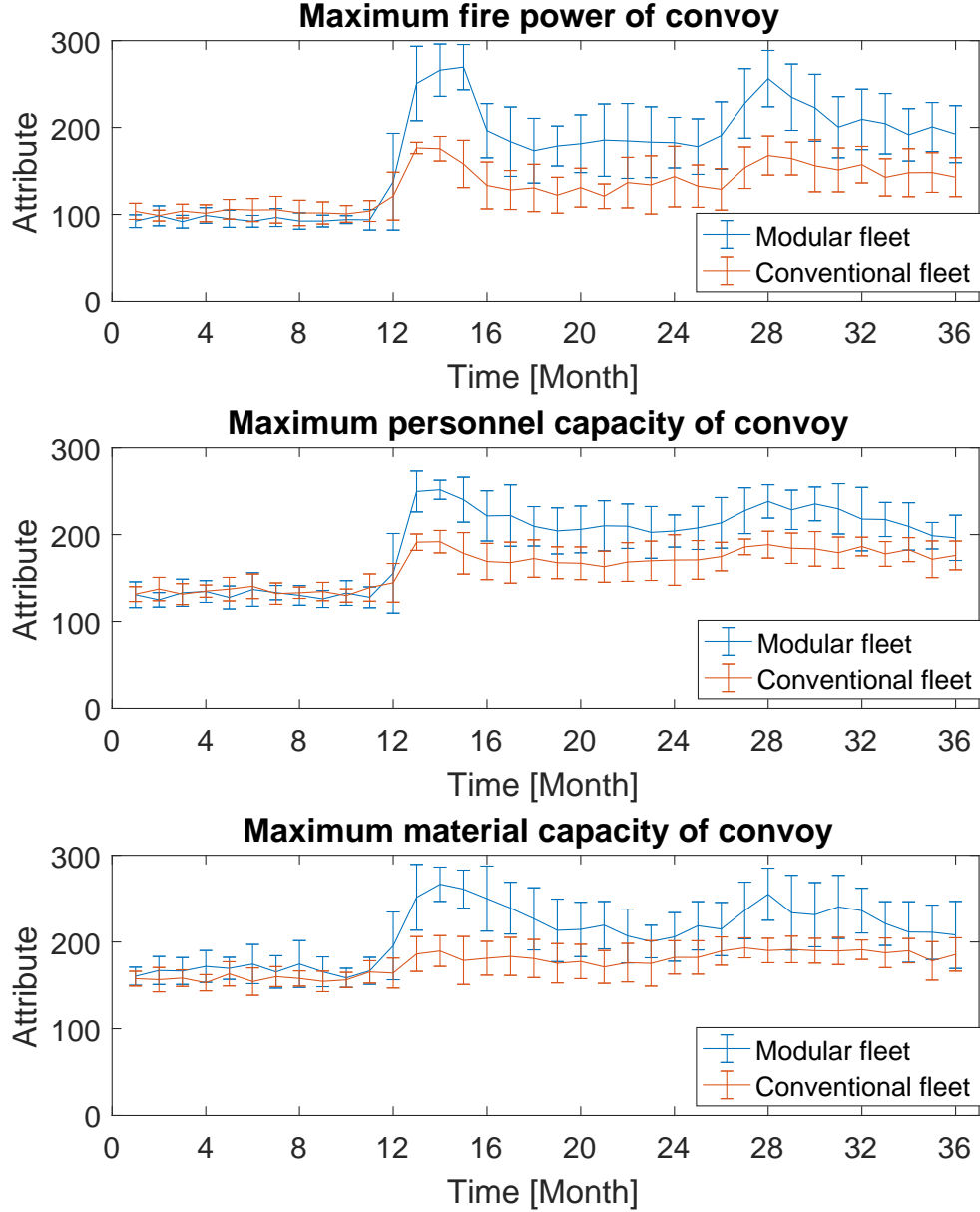


Figure 5.8: Maximum value of attributes carried by the convoy in different months and fair amount of capacity, i.e., strategy 8,9. Because of the flexibility of fleet structure, modular fleet can be easily adapted to the vehicle damage and enemy's behavior, which leads to a better balance between different types of vehicles to perform a stronger strategy. The defense strategy selection also impacts the attack strategy. Compared to modular fleet, conventional fleet is much more likely to give up mission because of resources insufficiency. This weakness makes the modular fleet confident

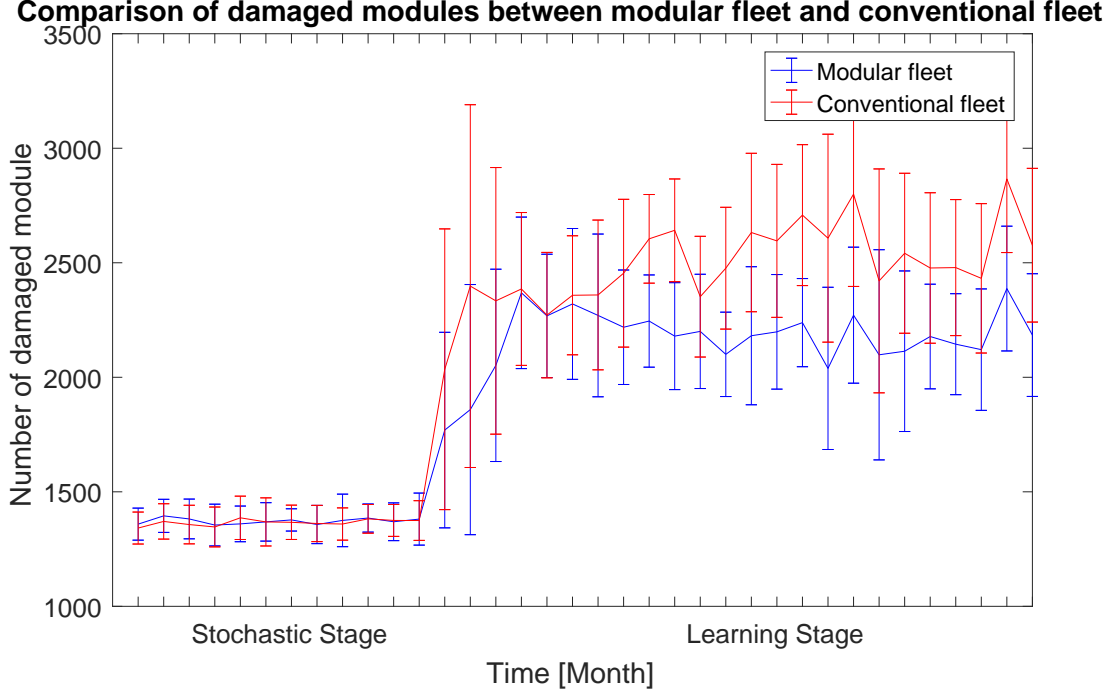


Figure 5.9: Modules damage occurs at each month

in dispatching little or even no combat vehicles to win the game. As an evidence, the proportion of strategy 1 used by modular fleet is much higher than that by conventional fleet. Meanwhile, modular fleet is more capable of performing aggressive strategies, i.e., strategy 8,9,10, more often than conventional fleet once a strong enemy is sensed.

To further investigate the improved performance of modularized fleet, the inference accuracy are also compared between different fleets. The mean square error (MSE) between forecasted and actual convoy attributes is denoted as the metric to quantify the inference accuracy. As can be seen from the comparison in Fig.5.11. Inference errors are significantly high at the beginning of the learning stage, because agents are trained by the data from stochastic dispatch, which contributes little to forecasting the behavior of a rational competent. Along with the learning process, more combat and operation records, which are based on trained ABM, are generated, in which

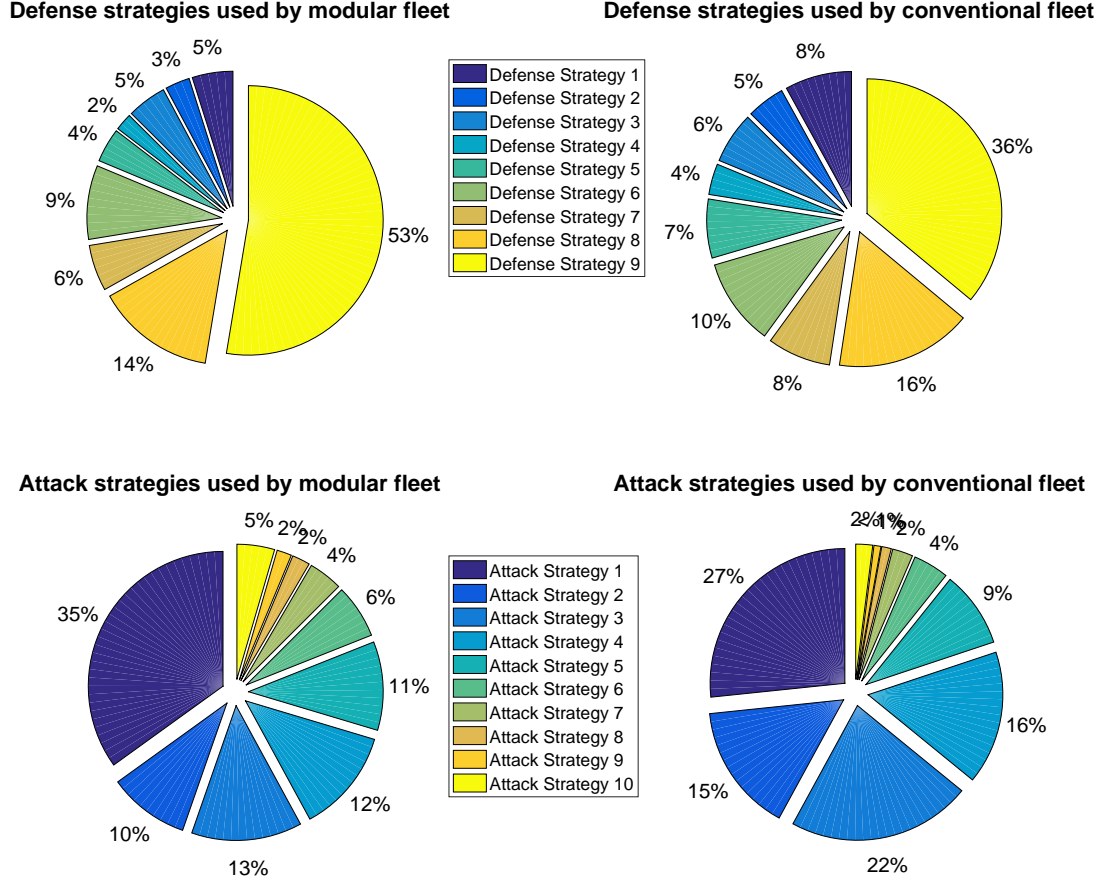


Figure 5.10: Proportion of defense strategy at different stages

enemy's behaviors are more explainable. As a result, inference errors are significantly reduced in the following four months. However, the inference error keep fluctuating during the rest of the learning stage, because both fleets keep checking and countering the other's behavior.

The results also show that it is easier to infer the strategy of conventional fleet than modular fleet, especially in the attribute of firepower. It originates from the higher freedom in decision making after fleet modularity. As a defender, fleet usually needs to prepare a convoy with all types of attributes to satisfy the demands. With limited vehicle stocks, decision maker of conventional fleet has constrained choice of strategy. However, for the modular fleet, they could vary the dispatch strategy by real-time vehicle reconfiguration, i.e., reconfigure cargos to combat vehicles to achieve

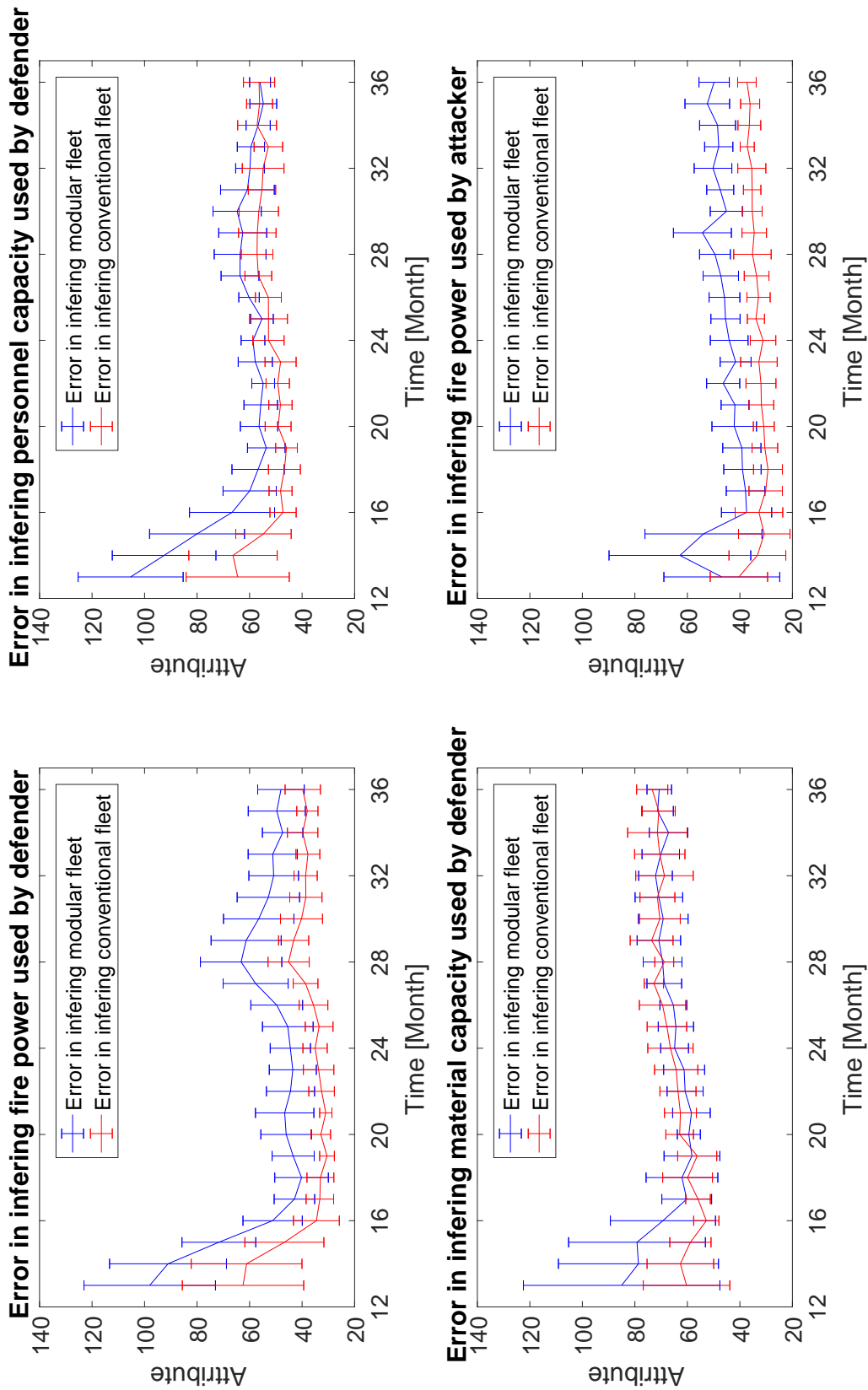


Figure 5.11: Comparison of inference accuracy between modular fleet and conventional fleet

the switch from Strategy 4 to Strategy 8.

However, the burden of modularity is also significant, which is the high acquisition of capacity. According to Fig.5.12, modular fleet always requires more machines than conventional fleet because of additional ADR actions. It can also be observed that machine requirements are increased significantly once entering the learning stage, which comes from damage from smarter strikes by enemy. The higher losses in conventional fleet also shrink the difference in machine usage at learning stage. In this study, the fleet performance is only tested at a certain capacity, studies investigating on the influence of capacity can be found in the following literature (*Li and Epureanu, 2017b, 2018*).

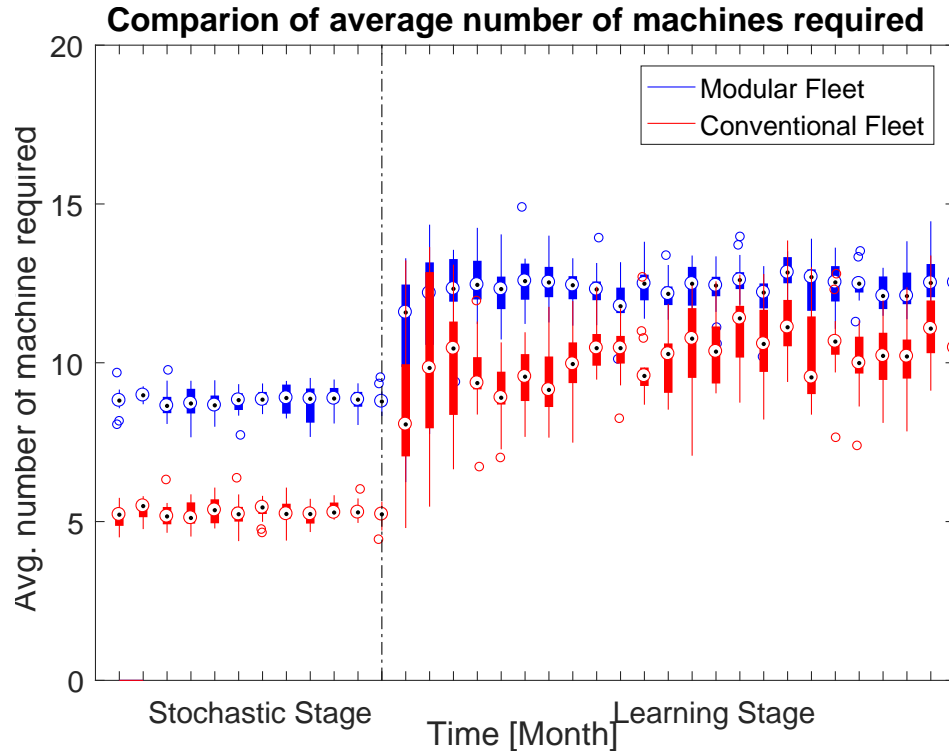


Figure 5.12: Comparison of machine requirements between modular fleet and conventional fleet

5.5 Summary

In this section, the benefits and burdens from fleet modularization are investigated by simulating an attacker-defender game between modular fleet and conventional fleet. The fleet competition is simulated for three years which are divided into stochastic stage and learning stage. By contrasting the simulation results from two fleets, it is found that conventional fleet stays the lead when both fleets select strategies stochastically; modular fleet outperforms conventional fleet once intelligence of decision maker is considered. With additional operational flexibility from on-base ADR actions, modular fleet exhibits a better adaptability in reacting to enemy's actions, higher upper limit in convoy formation and a more significant unpredictability from the additional flexibility in operation.

CHAPTER VI

Real-time management of RMS

6.1 Model Description

In this section, a negotiation algorithm is formulated to reallocate demands and resources among multiple lines. Then, a planning model for individual autonomous RMS is formulated to yield real-time decisions under highly stochastic demands. The model structure and associated approach is shown in Fig. 6.1.

6.1.1 Negotiation Among Lines

Given limited module stocks in a company, task and resource allocations are important to guarantee effective processing and resource utility rate. However, the

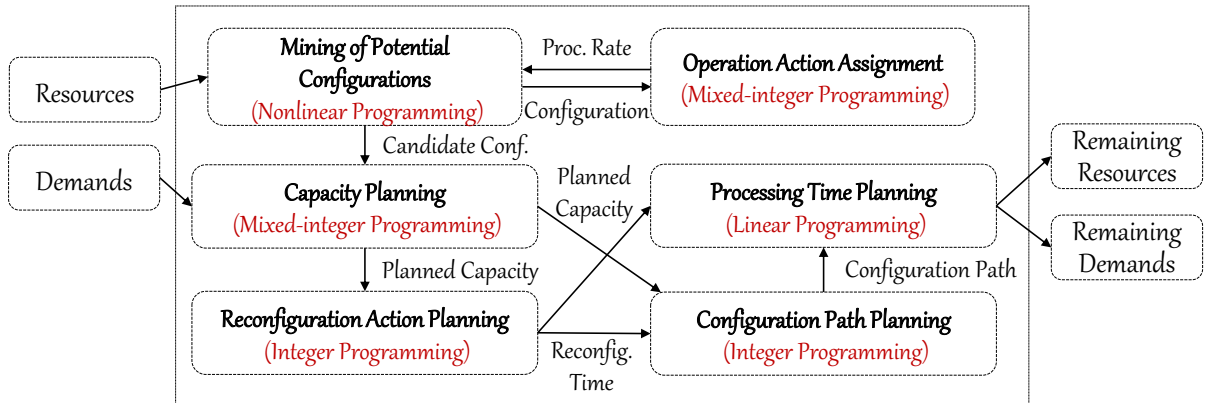


Figure 6.1: Structure of optimization models designed for RMS operation

complexity from RMS operation makes the simultaneous optimization of allocations intractable. To address this issue, each individual RMS (i.e., each line) is modeled as an intelligent agent, which negotiates with others if the line cannot fulfill the received demand alone. Denote by \mathbf{f}_u the unfulfilled orders (details in Section 3.7). A multi-step bidding process is created for negotiation, summarized as

Initialize: Distribute new received demands by heuristic rules. Tenderee Selection: Each agent predicts its unfulfilled demands by using \mathbf{f}_u . The one with the highest value becomes the tenderee, and the others bidders. Bidding Document: The tenderee is responsible to provide possible options to reduce delayed demands, namely Offer all unachievable demands and available modules. Request new modules to boost capacity. Bidding: Each bidder evaluates every option provided in the bidding document. The option with lowest unfulfilled demands becomes the bid to tenderee. Evaluation: The bidder with lowest unfulfilled demands wins. Demands and stocks are relocated based on bidding document. The process stops if one of these conditions are satisfied: Total unfulfilled demands are unchanged. Total unfulfilled demands is zero. Otherwise, the process returns to step 2 above.

The algorithm converges in finite iterations since the delayed demand is finite. The computational cost of the algorithm is in the bidding part, where each line accesses the options through \mathbf{f}_u . However, since bidders are independent, parallel computations can be used to improve the scalability of the proposed method. Next, based on the negotiated demands and resources, the real-time management of each RMS and formation of \mathbf{f}_u are obtained.

1. Initialize: Distribute new received demands by heuristic rules.
2. Tenderee Selection: Each agent predicts its unfulfilled demands by using \mathbf{f}_u .

The one with the highest value becomes the tenderee, and the others bidders.

3. Bidding Document: The tenderee is responsible to provide possible options to

reduce delayed demands, namely Offer all unachievable demands and available modules. Request new modules to boost capacity.

4. Bidding: Each bidder evaluates every option provided in the bidding document. The option with lowest unfulfilled demands becomes the bid to tenderee.
5. Evaluation: The bidder with lowest unfulfilled demands wins. Demands and stocks are relocated based on bidding document. The process stops if one of these conditions are satisfied: Total unfulfilled demands are unchanged. Total unfulfilled demands is zero. Otherwise, the process returns to step 2 above.

6.1.2 Operation Action Assignment

To customize an RMS as to provide almost the exact capacity and functionality that satisfies the demand [2], the marginal capacity of each potential configuration is calculated. At each stage t_s , only one machine variant v_{t_s} is allowed to be used. Given the processing operation op_{k,t_s} and the number of machines n_{t_s} , the processing rate of part k at stage t_s is $spr_{k,t_s} = \mathbf{f}_c(k, v_{t_s}, n_{t_s})\mathbf{op}_{k,t_s}$, where f_c is a vector function to calculate the stage capacity for all operations, \mathbf{op}_{k,t_s} contains an array of binary variables $\mathbf{op}_{k,t_s,i}$, which indicates whether the operation i is selected (1) or not (0). The stage with the minimum rate determines the rate of the line, $pr_k = \min_{t_s} spr_{k,t_s}$. The processing rates of the line for all parts $\mathbf{pr}(t)$ is denoted as a measure of capacity at time t . By using s_{k,t_s} as an auxiliary variable representing the superfluous capacity, an optimization model is created to find the marginal capacity as

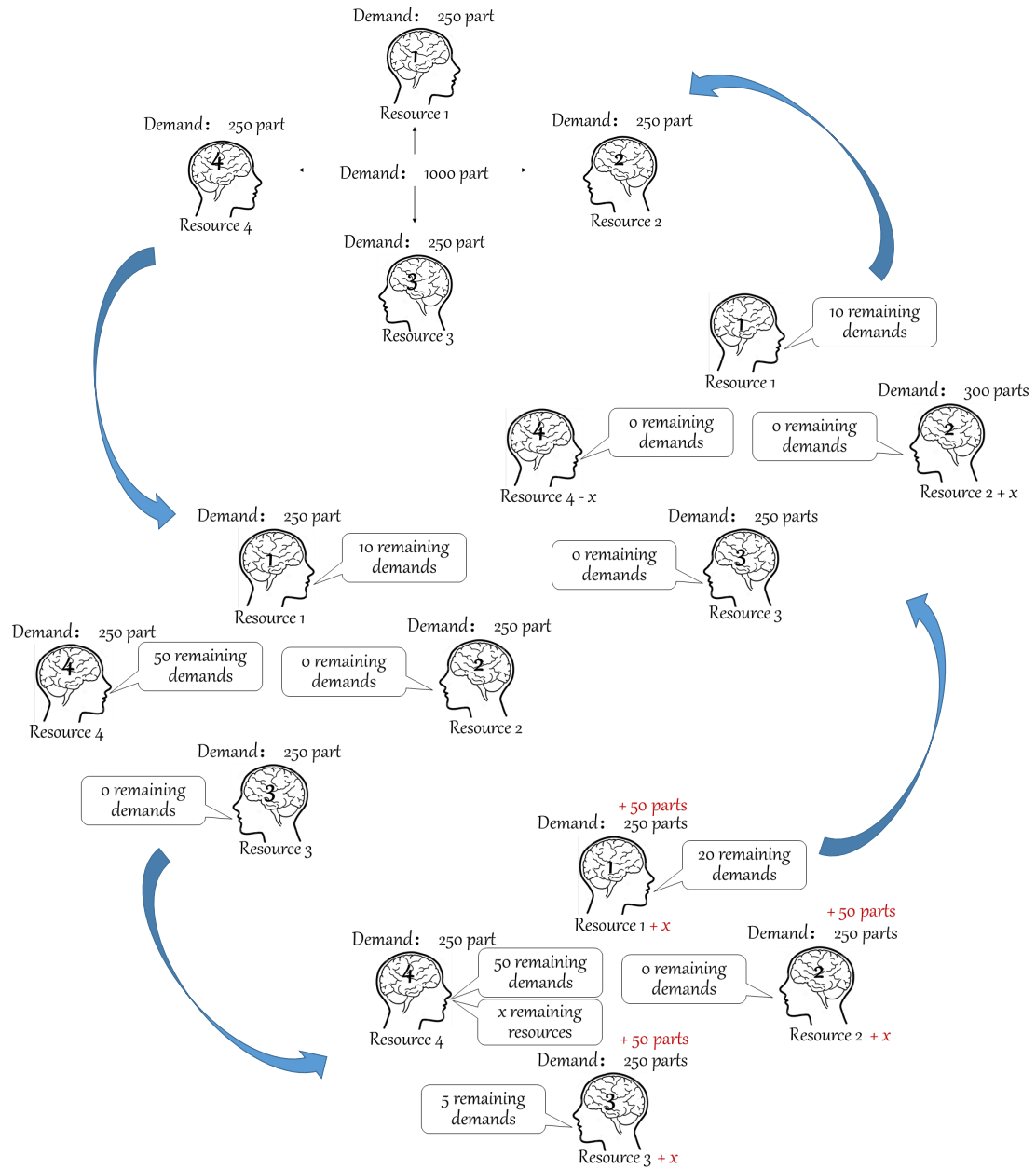


Figure 6.2: Negotiation among lines base on bidding process

$$\begin{aligned}
& \min_{s_k, \mathbf{op}} \quad -pr_k \\
& \text{s.t.} \quad (a) \quad \mathbf{mp}_{k,i}^T \mathbf{op}_{k,t_s} \leq \sum_{\tau_s=1}^{\tau_s=t_s-1} \mathbf{mp}_{k,i}^T \mathbf{op}_{k,\tau_s}, \quad \forall t_s \\
& \quad (b) \quad \sum_{t_s} \mathbf{M}_{ck} \mathbf{op}_{k,t_s} = \mathbf{1} \\
& \quad (c) \quad pr_k = spr_{k,t_s} - s_{k,t_s} = spr_{k,t'_s} - s_{k,t'_s}, \quad \forall t'_s \\
& \quad (d) \quad \sum_i op_{k,t_s,i} \leq 1, op_{k,t_s,i} \in \{0, 1\}, s_{k,t_s} \geq 0, \quad \forall t_s, i.
\end{aligned} \tag{6.1}$$

where, $\mathbf{mp}_{k,i}$, $\mathbf{mc}_{k,i}$ are vectors that specify the precedence and type of operations for action i . $\mathbf{1}$ is a column vector with all entries as 1. Constraint (a) ensures the operation actions occur in the required sequence; (b) ensures the completeness of the processing job; (c) ensures that $spr_{k,1} - s_{k,1}$ is equal to the capacity of the line; (d) bounds the feasible ranges of decision variable. Each RMS configuration is uniquely defined by the vectors \mathbf{v} and \mathbf{n} that record machine variants and numbers. This programming model in Eq. (1) allows us to link an RMS configuration to a maximum capacity, i.e., $pr_k = f_{pr}(k, \mathbf{v}, \mathbf{n})$. Only the operation that gives the maximal RMS capacity is used to increase resource utility rate. Thus, a configuration of the RMS is represented by a machine layout.

6.1.3 Mining of Potential Configurations

Given a set of available resources, potential RMS configurations are explored to find the configurations with high capacity using the following nonlinear integer programming model

$$\begin{aligned}
& \max_{\mathbf{v}, \mathbf{n}} \quad \sum_k w_k f_{pr}(\mathbf{v}, \mathbf{n}) \\
& \text{s.t.} \quad (a) \ f_{m_p}(\mathbf{v}, \mathbf{n}) \leq \mathbf{s}_{m_p} \\
& \quad \quad (b) \ 1 \leq v_{ts} \leq N_v, 0 \leq n_{ts} \leq N_n \text{ and integer, } \forall t_s,
\end{aligned} \tag{6.2}$$

where, w_k is the weight factor for part k , which can be dynamically tuned to shift the priority of mining when reacting to demands. f_{m_p} is a nonlinear function that calculates the required modules of type p for a configuration. $s_{m_p}(t)$ is the available resources in type p on the shop floor at time t . Constraint (a) ensures that the required modules are less than the stocks, and (b) bounds the feasible ranges of the decision variables. N_v is the number of machine variants. N_n is the maximum number of machines allowed per stage.

The optimization model in Eqn. (1) which defines f_{pr} makes the cost function in Eqn. 6.2 nonlinear and nonconvex. Combined with the nonlinearity from constraint (a), it is intractable to find the global optimal solutions efficiently. Note that the goal of RMS configuration mining is to seek a group of near-optimal potential configurations with high similarity to facilitate reconfiguration. Thus, a genetic algorithm (GA) is used to generate first and then to select a population of near-optimal configurations with high similarity. However, increasing the number of machines enlarges the size of possible layouts, and GAs cannot efficiently output all solutions. Also, module requirements for the GA-generated solutions are not strictly equal to the existing stocks. Hence, some configurations are just candidate solutions for lower stock cases.

To boost computational efficiency, a configuration library which stores all explored configurations is created, which includes a configuration index, machine layout, required modules and capacity. With continuous updates during planning, the library

provides the knowledge-base for real-time decision making.

6.1.4 Capacity Planning

Capacity planning assigns capacity at each time step during of the planning horizon to minimize unsatisfied demands. The first priority is to select candidate capacities, which are the capacities in the library that satisfy module requirements. Although candidate capacities provide the design space, the daily processing time $\mathbf{pt}(t)$ according to the reconfiguration time $rt(t)$ spent each day to determine the amount of delayed jobs is still needed. However, the nonlinearity of the problem and the high dimensionality of the decisions make it intractable to optimize for all variables at the same time. This problem is decoupled by approximating the processing time and reconfiguration time as $\bar{\mathbf{pt}}(t)$ and $\bar{rt}(t)$ by heuristics. Thus, a binary integer programming is formulated to minimize the expected total delayed demand during a planning horizon t_p as

$$\begin{aligned}
& \min_{\substack{\delta \mathbf{pr}, \\ \mathbf{s}_{o/b, n_d}}} \sum_{n_d} [\mathbf{c}_o^T \mathbf{s}_{o, n_d} + \mathbf{c}_b^T \mathbf{s}_{b, n_d}] + \sum_{t=1}^{t+t_p} \mathbf{c}_{\delta \mathbf{pr}}^T \delta \mathbf{pr}(\tau) \\
& \text{s.t.} \quad (a) \sum_{\tau=t+1}^{t_d, n_d} [\mathbf{pr}(\tau) \circ \bar{\mathbf{pt}}(\tau) - \mathbf{d}(\tau)] = \mathbf{s}_{o, n_d} - \mathbf{s}_{b, n_d}, \forall n_d \\
& \quad (b) \sum \mathbf{pr}(\tau) = 1, \forall \tau \\
& \quad (c) \mathbf{pr}(\tau + 1) = \mathbf{pr}(\tau) + \mathbf{M}_{\delta \mathbf{pr}} \delta \mathbf{pr}(\tau), \forall \tau \\
& \quad (d) s_{o, n_d, j}, s_{b, n_d, j} \geq 0, \forall n_d, \forall j \\
& \quad (e) \delta c_\alpha(\tau), c_\alpha(\tau) \in \{0, 1\}, \forall \tau, \forall \alpha,
\end{aligned} \tag{6.3}$$

where $\delta \mathbf{pr}(\tau)$ is a binary vector indicating the switch of capacity with cost $\mathbf{c}_{\delta \mathbf{pr}}$. $\mathbf{M}_{\delta \mathbf{pr}}$ records the changes of capacity under reconfiguration actions. n_d is the index of demand in the planning horizon. t_{d, n_d} is the due time of the n_{dth} demand. $\mathbf{d}(\tau)$ records all the received demands with due time as time index τ . $\mathbf{s}_{o, n_d}, \mathbf{s}_{b, n_d}$ are variables that

represent overused and insufficient capacity of the $n_{d^{th}}$ demand, with costs c_o and c_b . Constraint (a) states the balance between remaining demands and auxiliary variables. (b) comes from our assumption that RMS reconfiguration occurs only once for each time index τ . (c) states the balance of capacity at different times. (d) and (e) specify the type of decision variables. The proposed optimization model reveals a trade-off between the utility rate of capacity and the reconfiguration costs. The model is able to adjust the reconfiguration frequency based on the cost.

6.1.5 Reconfiguration Action Planning

There are multiple configurations that achieve the same RMS capacity. Before deciding the RMS configuration path, it is necessary to calculate the minimum time required to switch configurations. The minimal reconfiguration time is calculated by using Li et al.s methodology (*Li and Epureanu, 2017b*), and formulate the optimization problem as

$$\begin{aligned}
& \min_{\substack{\mathbf{oa}, \mathbf{od}, \\ \mathbf{oc}, \mathbf{ol}}} \sum_{t_s} \mathbf{t}_a^T \mathbf{oa}_{t_s} + \mathbf{t}_d^T \mathbf{od}_{t_s} + \mathbf{t}_c^T \mathbf{oc}_{t_s} + \mathbf{t}_l^T \mathbf{ol}_{t_s} \\
& \text{s.t.} \quad (a) \quad \mathbf{tc}'_{t_s} = \mathbf{tc}_{t_s} + \mathbf{oa}_{t_s} - \mathbf{od}_{t_s} + \mathbf{M}_c \mathbf{oc}_{t_s} + \mathbf{M}_l \mathbf{ol}_{t_s}, \quad \forall t_s \\
& \quad (b) \quad \sum_{t_s} \mathbf{m}_p (\mathbf{oa}_{t_s} - \mathbf{od}_{t_s} + \mathbf{M}_c \mathbf{oc}_{t_s}) \leq s_{amp}(t), \quad \forall p \\
& \quad (c) \quad \mathbf{oa}_{t_s}, \mathbf{od}_{t_s}, \mathbf{oc}_{t_s}, \mathbf{ol}_{t_s} \geq 0 \text{ and integer}, \quad \forall t_s,
\end{aligned} \tag{6.4}$$

where $\mathbf{tc}_{t_s}, \mathbf{tc}'_{t_s}$ record machine layouts before and after reconfiguration. $\mathbf{oa}_{t_s}, \mathbf{od}_{t_s}, \mathbf{oc}_{t_s}, \mathbf{ol}_{t_s}$ are decision variables for assembly, disassembly, reconfiguration and relocation. $\mathbf{M}_c, \mathbf{M}_l$ are matrices representing changes of machines due to reconfiguration and relocation actions. m_p records modules of type p required by each machine. $s_{amp}(t)$ is the number of free modules of type p at time index t . Constraint (a) ensures that the reconfiguration is complete. (b) guarantees that the reconfiguration is achieved by the given free modules. (c) ensures all decisions are non-negative inte-

gers. Times required rt for reconfiguration actions are obtained by enumerating all combinations of qualified candidates.

6.1.6 Configuration Path Planning

In configuration path planning, the goal is to select candidate configurations that effectively provide the target capacity $\mathbf{pr}(t)$. First, the changes of capacity is represented as a dynamical system

$$\begin{aligned}\mathbf{c}(t+1) &= \mathbf{c}(t) + \mathbf{M}_{\delta\mathbf{c}}\delta\mathbf{c}(t), \\ \mathbf{cpr}(t) &= \mathbf{M}_{\mathbf{pr}}\mathbf{c}(t) + \mathbf{M}_{\delta\mathbf{pr}}\delta\mathbf{c}(t),\end{aligned}\tag{6.5}$$

where $\mathbf{c}(t)$ is a binary vector that indicates the selected candidate configuration with capacity $\mathbf{cpr}(t)$. $\mathbf{M}_{\delta\mathbf{c}}$ records how the configurations change under reconfiguration actions. $\mathbf{M}_{\mathbf{pr}}$ stores the capacity of all candidate configurations. $\mathbf{M}_{\delta\mathbf{pr}}$ records the expected capacity loss from reconfiguration actions. Given all $\delta\mathbf{c}(t)$ over the planning horizon, the corresponding capacity is calculated by iteratively substituting Eqn. (5). An integer programming is formulated to minimize the total reconfiguration time

$$\begin{aligned}\min_{\delta\mathbf{c}} \quad & \sum_{\tau=t+1}^{t+t_p} \mathbf{rt}^T \delta\mathbf{c}(\tau) \\ \text{s.t.} \quad & (a) \ \mathbf{cpr}(\tau) = \mathbf{pr}(\tau), \ \forall \tau \\ & (b) \ \sum \mathbf{c}(\tau) = 1, \ \forall \tau \\ & (c) \ \delta c_{\alpha}(\tau), c_{\alpha}(\tau) \in \{0, 1\}, \ \forall \tau, \forall \alpha.\end{aligned}\tag{6.6}$$

Once the configuration path is found, the exact reconfiguration time $rt(t)$ spent during each time index is determined also. That may differ from the estimated $\bar{rt}(t)$ used in section 3.4. It is possible to train a neural network to improve estimations based on records. The training set includes inventory status, received demands,

current configuration and calculated processing time.

6.1.7 Processing Time Planning

Because of inevitable errors in estimating the reconfiguration time and unexpected changes on the shop floor, i.e., fluctuations in the processing speed of each machine, the actual processing time should be adjusted at each time index τ given the determined path, to minimize unsatisfied demands as

$$\begin{aligned}
& \min_{pt_k} \quad \sum_{\tau=t+1}^{t+t_p} \sum_k -\gamma^t pt_k(\tau) pr_k(\tau) \\
& \text{s.t.} \quad (a) \quad \sum_{\tau=t+1}^{t_d, n_d} pt_k(\tau) pr_k(\tau) \leq \sum_{\tau=t+1}^{t_d, n_d} d_k(\tau), \quad \forall n_d, \forall k \\
& \quad (b) \quad \left[\sum_k pt_k(\tau) + rt(\tau) \right] \leq \bar{t}, \quad \forall \tau \\
& \quad (c) \quad pt_k(\tau) \geq 0, \quad \forall \tau,
\end{aligned} \tag{6.7}$$

where the decision variable $pt_k(\tau)$ represents the production time for part k at time index τ . γ is a discount factor that encourages the system to process jobs. Constraint (a) limits the amount of processed parts to be less than the demand. (b) bounds the total working time by a threshold \bar{t} . (c) ensures that decisions are non-negative. Thus, \mathbf{f}_u used in section 3.1 is formed as a function of the module stock $\mathbf{s}_m(t)$ and demands in the planning horizon $\mathbf{D}(t)$, which contains models described in 3.2-3.7, namely

$$\mathbf{f}_u(\mathbf{s}_m(t), \mathbf{D}(t)) = \sum_{n_d} \left[\sum_{\tau=t+1}^{t_d, n_d} (d(\tau) - \mathbf{pt}(\tau) \circ \mathbf{pr}(\tau)) \right]. \tag{6.8}$$

Table 6.1: Machines' composition and their average processing rate

Machine Variant									
Module Type	1	2	3	4	5	6	7	8	9
1	3	3	3	3	4	0	0	0	0
2	1	2	3	4	1	0	0	0	0
3	0	0	0	0	0	1	2	3	4
4	1	1	1	1	1	0	0	0	0
5	0	0	0	0	0	1	1	1	1
\bar{or}_1	11	22	33	44	12	2	4	5	7
\bar{or}_2	7	15	23	30	11	8	17	26	35

6.2 Sample Results

In this section, the proposed model is implemented to simulate the real-time operation of a processing company to demonstrate how the capacity of RMS matches the demands in real-time, and how the teaming between lines results in a better performance. The example from Youssef et al. (*Youssef and ElMaraghy, 2007*) is used in this study, where two types of parts with multiple features require operation actions to process. There are 9 types of machines with different operation speeds, which are formed by 5 types of modules as shown in Table 1.

where, \bar{or}_k represents the averaged processing rate for part k by each machine

6.2.1 Real-time Capacity Management

In this example, the manufacturing firm initially invests in a single processing line by using 50 modules of Type 1 and 20 modules of Types 2, 3, 4 and 5. The time interval between stochastically arriving demands is approximately 10 days, and the average completion time is one week. The magnitude of demands is generated based

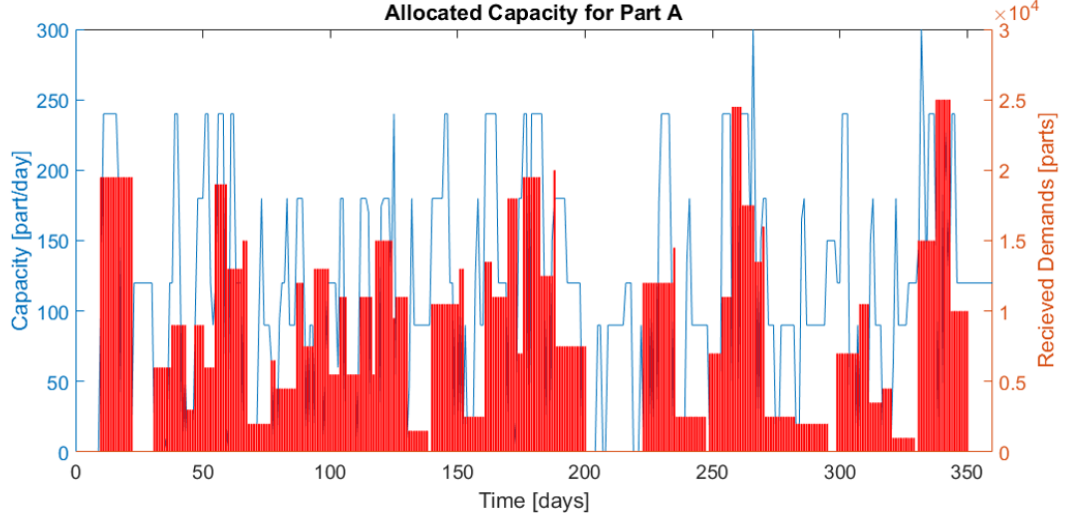


Figure 6.3: Matching between capacity of RMS for part 1 and its demand

on Gaussian distributions, i.e.,

$$d_k \sim \mathcal{N}(5000, 5000). \quad (6.9)$$

After simulating over one year, the averaged computation time required for daily decision is 0.42 seconds. Figure 2 shows the incomplete orders and their impact on capacity.

No delayed demand was found in the simulation. Additionally, the capacity for Part B always followed the trend of the demand, as shown in Figure 2. However, there are situations where the capacity for Part A is significant even without receiving a demand, e.g., between Day 200 and Day 220. During such periods, most of the capacity is allocated to process Part B, but the system decides to keep some superfluous capacity for Part A so that it can respond to changes in demand. Considering the high cost of customizing a configuration, the model outputs a compromised strategy to balance the reconfiguration cost and overused capacity.

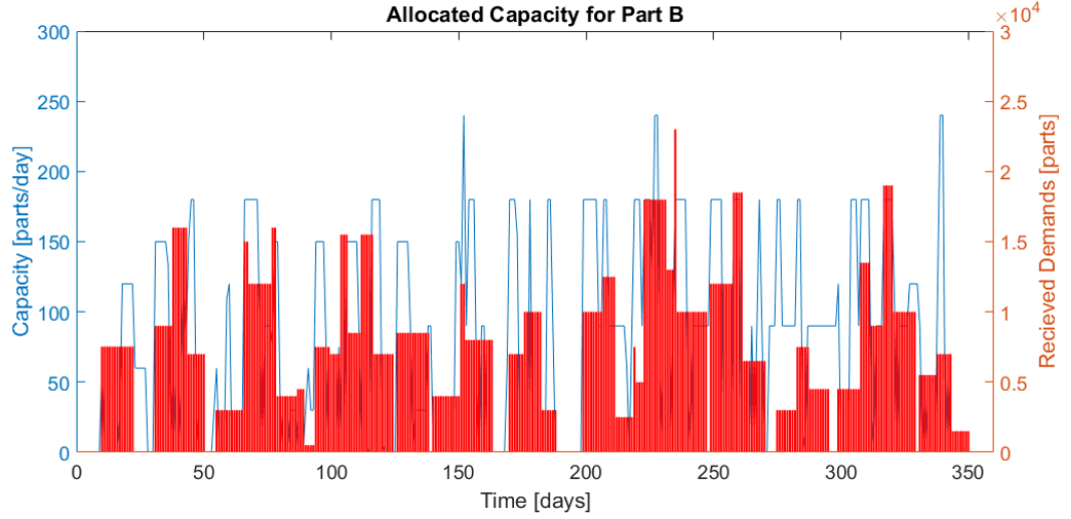


Figure 6.4: Matching between capacity of RMS for part 2 and its demand

6.2.2 Teamming Among Lines (i.e., among distinct RMS)

As an example, consider the case where the demand settings are the same, and the firm allocates resources initially to the two processing lines as follows for Types 1-5: (a) line 1: 30, 10, 10, 10, 10, and (b) line 2: 20, 10, 10, 10, 10. Initially, the received demands are allocated evenly to each line. Because of the limited resources, one RMS may not accomplish the assigned task alone. Negotiation is required to minimize the remaining demands, which increase the time required for decision making to 2.79 seconds.

Figure 3, 4 shows an example of task and resource relocations respectively, which are triggered by the sudden occurrence of intensive demands. The results shown in Figure 4 demonstrate the performance boosts through negotiation between lines. In the figure, the total capacity in the case without demand and module sharing is limited to 360 parts/hour. However, once the negotiation is considered, the maximum capacity can reach up to 420 parts/hour. This improvement in capacity eliminates most of the delayed demands that occur otherwise.

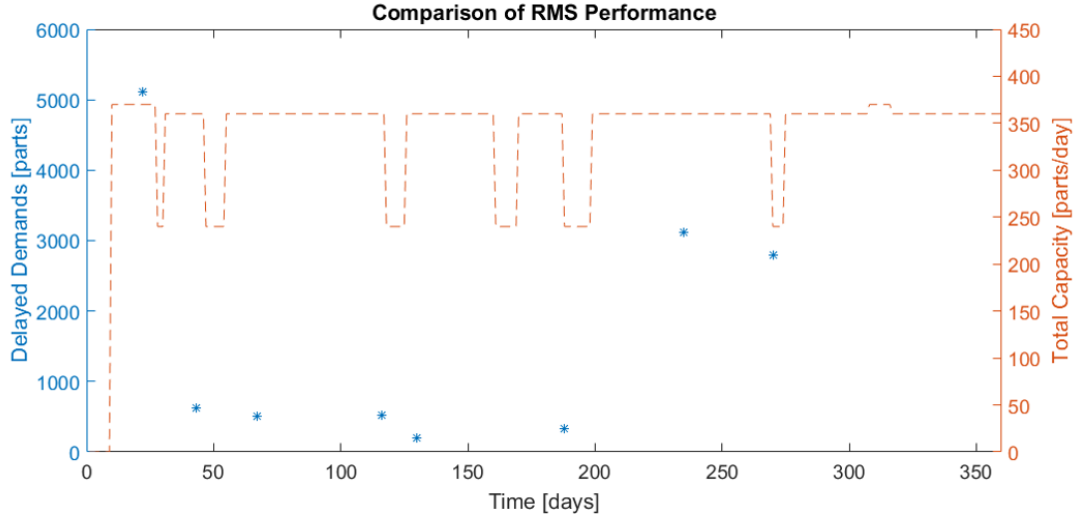


Figure 6.5: Matching between capacity of RMS for part 1 and its demand

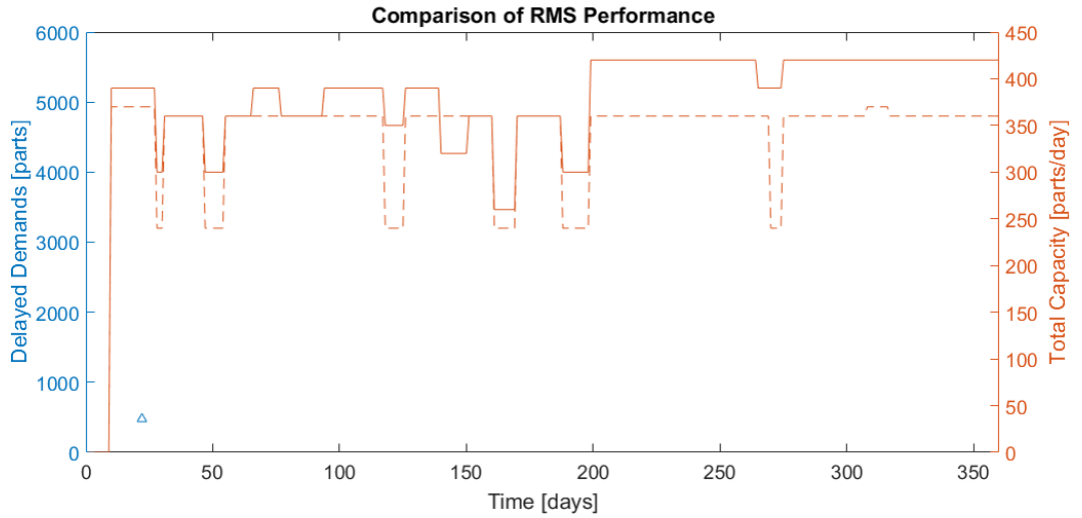


Figure 6.6: Matching between capacity of RMS for part 2 and its demand

6.3 Summary

In this section, the potential of autonomous reconfigurable manufacturing systems for highly stochastic and rapidly changing demands is discovered. An integrated model is formulated for the decentralized management of such teams of autonomous RMS in real-time under a limited inventory. A negotiation algorithm is proposed for resource and demand relocation between multiple lines to increase the readiness and

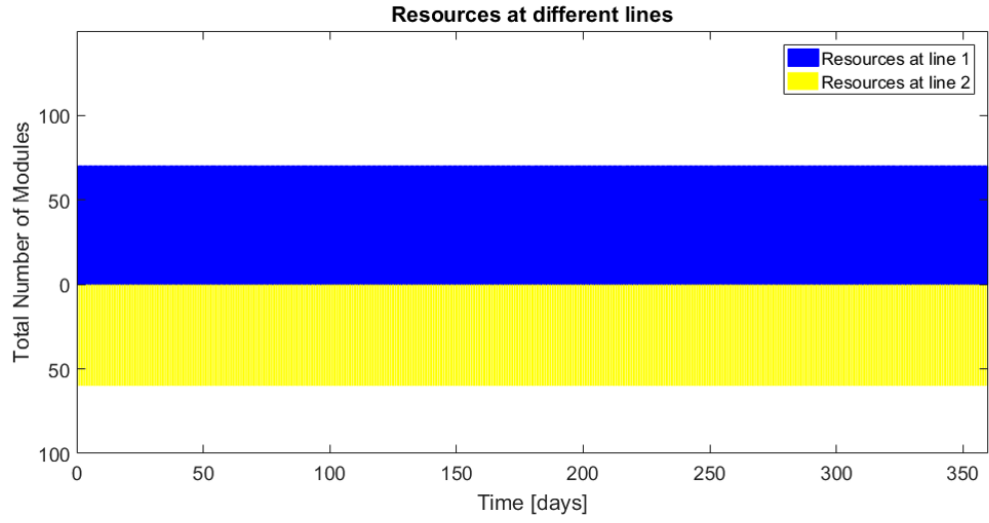


Figure 6.7: Module level of both lines without negotiation

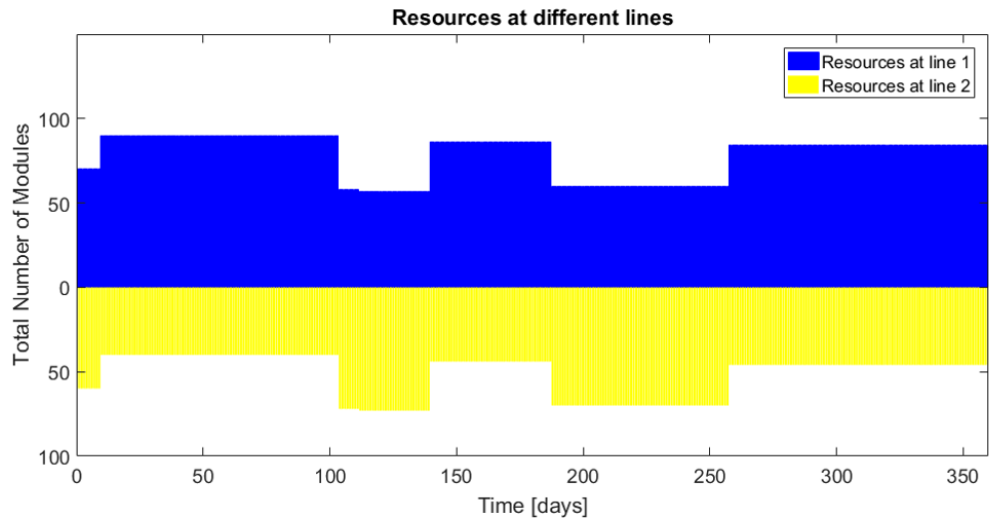


Figure 6.8: Module level of both lines with negotiation

utility rate of the overall system.

CHAPTER VII

Conclusions

7.1 Summary

This dissertation has demonstrated the intelligent fleet operation systems created for evaluating the benefits and burdens of modularity in military fleet operation. Three mission scenarios are created to compare the fleet performance with different emphasizes. The fleet operation system is presented as a discrete dynamical system. Because of the ADR actions, the number of system inputs of modular fleet is larger than that of conventional fleet. Thus, the problem is converted to evaluate the benefits of these additional freedom in control inputs to the system performance.

First, the fleet performance are compared in a simplified scenario - commander needs to operate the fleet to guarantee the vehicle readiness at battlefields. An MPC method have been implemented to make real-time operational decisions according to the system status and upcoming field demands. The results indicate a significant boost in adaptability due to the flexibility gained from modularity. Then, uncertainty in demands and damage are considered, which are modeled as Gaussian random variables with known distribution. SMPC is implemented by stochastic formulation of constrains and objective function. Simulation results reveals that conventional fleet demands more resources to avoid delayed attributes. Once resources are limited, modular fleet performs a higher robustness in reacting to the demand uncertainty.

The operation model is also implemented in a realistic logistic military mission scenario, where vehicle usage and damage are considered to be dependent on the demands, and the quality of vehicle dispatch is evaluated in a more complicated manner. To ensure the computational efficiency, the decisions are distributed to three types of agents: resupply agent, base agent and dispatch agent. Operational decisions are calculated through agent collaboration and real-time optimization. Results show that the total resupplied resources for operating a modularized vehicle system can be reduced by more than 40 percent due to pooling effect. And modular fleet performs distinct operational strategies under changing ADR capacity.

To further improve the model fidelity, an attacker-defender game between two intelligent and adversarial forces is simulated, which can evolve their combat strategies based on the enemy's behaviors. To achieve that, inference agent are added by implementing a deep learning model - LSTM. Dispatch agent is updated by a multi-objective optimization model which maximize the win rate with consideration of order feasibility. By contrasting the simulation results from two fleets, it is found that conventional fleet stays the leader when both fleets selecting strategies stochastically; modular fleet outperforms conventional fleet once intelligence of decision maker is considered. The freedom of operation enables the modular fleet to perform an improved adaptability to the changes of enemy and an augmented complication to be inferred.

Finally, the operation model is extended to civilian application - management of reconfigurable manufacturing systems. A negotiation algorithm among lines is created for distributing the stochastic demands and exchanging machines or modules. A group of optimization models are designed for adaptively configuring these lines and machines to minimize the missing demands under a limited inventory of configurable components.

7.2 Summary of Contributions

This dissertation offers an innovative analytical and methodological approach in operation management of a fleet of autonomous vehicles/robots. To achieve the real-time decision making in reacting to the stochastically arrived demands, a new stochastic MPC framework is created for managing the dynamics of systems, where each of them representing impacts of operational actions on the configuration of fleet. Through analyzing the results from diverse logistic mission scenarios, boosts in adaptability and robustness are observed once fleet modularity is introduced.

Concerning the evolutionary decision making process during military operation, a novel intelligent agent based model is created to simulate the human-like decision making process by combining the real-time optimization and artificial intelligence techniques. The model is capable to guide the action of autonomous vehicles through analyzing field information, forecasting enemy's decisions and planning for operational actions. The model offers the capability of performing a competition between two intelligent and adversarial military commanders. By formulating a warfare between conventional fleet and modular fleet, several pristine advantages brought from fleet modularity are disclosed: increased capability in satisfying the orders and augmented difficulty of being inferred by enemy.

The model also has significant implications in solving civilian operation problems, i.e., real-time management of RMSs. To resolve the computational complexity in real-time management of multiple RMSs, new optimization models are created to collaboratively configure the systems and machines in reacting to the demands under a limited inventory of configurable components. A bidding process is also customized to efficiently and effectively reallocate demands and resources among multiple RMSs for seeking a better team performance.

7.3 Limitation of the Study

Although this research was carefully prepared, I am still aware of its limitations and shortcomings.

First of all, the type of vehicles and attributes used in Chapter 5 is limited, which incurs the incomplete operational strategies.

Second, the way of modeling vehicle damage is simplified to convoy firepower comparison. However, many other factors are also critical to affect the damage probability, i.e., convoy layout, vehicle teaming strategy, which can be additional decisions to be made by operation model.

Third, the learning model is based on the assumption that all forces operate based on limited resources. Once resupplies are allowed, more training samples are needed for fleets to infer the other's strategy with consideration of possible reinforce ordered by enemy. Scout vehicles can be introduced for information acquisition.

In addition, introducing new/obsoleting old modules in both modular fleet and RMS are not in consideration. One of the advantage from modularity is its rapid response to the technology update, which is not emphasized here.

7.4 Future Work

Many different analyses, theoretical proofs and parametric studies have been left for the future due to lack of time (i.e. for Chapter 4, the experiments with real data are usually very time consuming, requiring even days to finish a single run). Future work concerns deeper analysis of fleet operation strategy, new proposals to try different methods or modeling approach. The following ideas could be achieved in the future:

1. In Chapter 3, the stability constraint, Eqn. 3.55, is created to avoid system instability. Based on the results from multiple realizations, this condition is

shown to be the sufficient and necessary condition for stability in a deterministic system. Once uncertainty is considered, the condition becomes the sufficient condition. Proofs for these statements can be done in the future.

2. In Chapter 4, a lot of parametric studies can be conducted. For example, the number of available working stations is fixed at 10. It is interesting to vary the number of stations and observe the changes of fleet performance. The modular fleet may lose the tactical advantages under insufficient capacity. Similar studies can also be performed by varying planning horizon, assembly time, repair time, frequency of demand, initial resource level and etc.
3. In Chapter 4, if a equilibrium can be proven from the competition of two fleets, the benefits from modularity become intuitive. However, as the decision making process is dominated by machine learning techniques and optimizations, and the resulted operational actions are numerous, operation strategies are obscure and hard to be summarized. Simplifications are needed before finding the equilibrium.
4. In Chapter 5, new actions can be added to increase the universality of the model, in terms of new component design, machine maintenance and repair.
5. Finally, a model that simulates the detailed vehicle operation actions, i.e., movement control, teaming and etc, are necessary for a better estimation of vehicle damage and mission success. Moreover, teaming among vehicles (movable) and modules (mainly static) could offer additional and flexible combat strategies for a highly modularized vehicle fleet.

BIBLIOGRAPHY

BIBLIOGRAPHY

- Adhau, S., M. Mittal, and A. Mittal (2013), A multi-agent system for decentralized multi-project scheduling with resource transfers, *International journal of production economics*, 146(2), 646–661.
- Adhitya, A., R. Srinivasan, and I. A. Karimi (2007), A model-based rescheduling framework for managing abnormal supply chain events, *Computers & Chemical Engineering*, 31(5), 496–518.
- Anosike, A., and D. Zhang (2009), An agent-based approach for integrating manufacturing operations, *International Journal of Production Economics*, 121(2), 333–352.
- Army, U. (2007), Fm 5-0: Army planning and orders production.
- Asl, F. M., and A. G. Ulsoy (2003), Stochastic optimal capacity management in reconfigurable manufacturing systems, *CIRP Annals-Manufacturing Technology*, 52(1), 371–374.
- Atsalakis, G. S., I. G. Atsalaki, and C. Zopounidis (2018), Forecasting the success of a new tourism service by a neuro-fuzzy technique, *European Journal of Operational Research*, 268(2), 716–727.
- Azab, A., and H. ElMaraghy (2007), Mathematical modeling for reconfigurable process planning, *CIRP Annals-Manufacturing Technology*, 56(1), 467–472.
- Azaiez, M. N., and V. M. Bier (2007), Optimal resource allocation for security in reliability systems, *European Journal of Operational Research*, 181(2), 773–786.
- Bayrak, A. E., et al. (2018), A system-of-systems approach to the strategic feasibility of modular vehicle fleets, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.
- Bemporad, A., and M. Morari (1999), Robust model predictive control: A survey, in *Robustness in identification and control*, pp. 207–226, Springer.
- Bensmaine, A., M. Dahane, and L. Benyoucef (2014), A new heuristic for integrated process planning and scheduling in reconfigurable manufacturing systems, *International Journal of Production Research*, 52(12), 3583–3594.

- Bichi, M., G. Ripaccioli, S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky (2010), Stochastic model predictive control with driver behavior learning for improved powertrain control, in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 6077–6082, IEEE.
- Böhnlein, D., K. Schweiger, and A. Tuma (2011), Multi-agent-based transport planning in the newspaper industry, *International Journal of Production Economics*, *131*(1), 146–157.
- Camacho, E. F., and C. A. Bordons (2012), *Model predictive control in the process industry*, Springer Science & Business Media.
- Ceryan, O., and Y. Koren (2009), Manufacturing capacity planning strategies, *CIRP annals*, *58*(1), 403–406.
- Chen, J. Y., and M. J. Barnes (2013), Human-agent teaming for multi-robot control: A literature review, *Tech. rep.*, ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD.
- Chen, K., Y. Zhou, and F. Dai (2015), A lstm-based method for stock returns prediction: A case study of china stock market, in *Big Data (Big Data), 2015 IEEE International Conference on*, pp. 2823–2824, IEEE.
- Dasch, J. M., and D. J. Gorsich (2016), Survey of modular military vehicles: benefits and burdens, *Tech. rep.*, Army Tank Automotive Research, Development and Engineering Center (TARDEC) Warren United States.
- Di Cairano, S., D. Bernardini, A. Bemporad, and I. V. Kolmanovsky (2014), Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management., *IEEE Trans. Contr. Sys. Techn.*, *22*(3), 1018–1031.
- Di Persio, L., and O. Honchar (2016), Artificial neural networks approach to the forecast of stock market price movements, *International Journal of Economics and Management Systems*, *1*, 158–162.
- Dresher, M. (1961), Games of strategy: theory and applications, *Tech. rep.*, RAND CORP SANTA MONICA CA.
- D’Souza, K., et al. (2016), An integrated design approach for evaluating the effectiveness and cost of a fleet, *The Journal of Defense Modeling and Simulation*., *13*(4), 381–397.
- Duffie, N. A., and V. V. Prabhu (1994), Real-time distributed scheduling of heterarchical manufacturing systems, *Journal of Manufacturing Systems*., *13*(2), 94–107.
- Dunbar, W. B., and S. Desa (2007), Distributed mpc for dynamic supply chain management, in *Assessment and future directions of nonlinear model predictive control*, pp. 607–615, Springer.

- Evers, L., A. I. Barros, H. Monsuur, and A. Wagelmans (2014), Online stochastic uav mission planning with time windows and time-sensitive targets, *European Journal of Operational Research*, 238(1), 348–362.
- Fischer, T., and C. Krauss (2017), Deep learning with long short-term memory networks for financial market predictions, *European Journal of Operational Research*.
- Fukushima, H., and R. R. Bitmead (2003), Robust constrained model predictive control using closed-loop prediction, in *American Control Conference, 2003. Proceedings of the 2003*, vol. 3, pp. 2511–2516, IEEE.
- Gerchak, Y., M. J. Magazine, and A. B. Gamble (1988), Component commonality with service level requirements, *Management science*, 34(6), 753–760.
- Giannakis, M., and M. Louis (2011), A multi-agent based framework for supply chain risk management, *Journal of Purchasing and Supply Management*, 17(1), 23–31.
- Hagan, M. T., H. B. Demuth, M. H. Beale, et al. (1996), *Neural network design*, vol. 20, Pws Pub. Boston.
- Hausken, K., and G. Levitin (2009), Minmax defense strategy for complex multi-state systems, *Reliability Engineering & System Safety*, 94(2), 577–587.
- Hausken, K., and J. Zhuang (2011), Defending against a terrorist who accumulates resources, *Military Operations Research*, pp. 21–39.
- He, N., D. Zhang, and Q. Li (2014), Agent-based hierarchical production planning and scheduling in make-to-order manufacturing system, *International Journal of Production Economics*, 149, 117–130.
- Hecht-Nielsen, R. (1992), Theory of the backpropagation neural network, in *Neural networks for perception*, pp. 65–93, Elsevier.
- Herroelen, W., and R. Leus (2005), Project scheduling under uncertainty: Survey and research potentials, *European journal of operational research*, 165(2), 289–306.
- Holweg, M., and D. Jones (2001), The build-to-order challenge: can current vehicle supply systems cope?, *Manufacturing Operations and Supply Chain Management: The Lean Approach*, Thomson Learning, London, pp. 362–72.
- Holweg, M., F. K. Pil, et al. (2005), The second century: reconnecting customer and value chain through build-to-order moving beyond mass and lean in the auto industry, *MIT Press Books*, 1.
- Jones, A. T., and C. R. McLean (1986), A proposed hierarchical control model for automated manufacturing systems, *Journal of Manufacturing Systems*, 5(1), 15–25.

- Jose, V. R. R., and J. Zhuang (2013), Technology adoption, accumulation, and competition in multi-period attacker-defender games, *Military Operations Research*, 18(2), 33–47.
- Julka, N., R. Srinivasan, and I. Karimi (2002), Agent-based supply chain management1: framework, *Computers & Chemical Engineering*, 26(12), 1755–1769.
- Kardes, E., and R. Hall (2005), Survey of literature on strategic decision making in the presence of adversaries.
- Koren, Y., and M. Shpitalni (2010), Design of reconfigurable manufacturing systems, *Journal of manufacturing systems*, 29(4), 130–141.
- Koren, Y., U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel (1999), Reconfigurable manufacturing systems, *CIRP Annals-Manufacturing Technology*, 48(2), 527–540.
- Koren, Y., X. Gu, and W. Guo (2017), Reconfigurable manufacturing systems: Principles, design, and future trends, *Frontiers of Mechanical Engineering*.
- Krasnikov, A., and S. Jayachandran (2008), The relative impact of marketing, research-and-development, and operations capabilities on firm performance, *Journal of marketing*, 72(4), 1–11.
- Kumar, R., M. J. Wenzel, M. J. Ellis, M. N. ElBsat, K. H. Drees, and V. M. Zavala (2018), A stochastic model predictive control framework for stationary battery systems, *IEEE Transactions on Power Systems*, 33(4), 4397.
- Lam, K.-m., and H.-f. Leung (2006), Formalizing risk strategies and risk strategy equilibrium in agent interactions modeled as infinitely repeated games, in *Pacific Rim International Workshop on Multi-Agents*, pp. 138–149, Springer.
- Landa, M. d. (1991), *War in the age of intelligent machines*, Zone Books.
- Li, X., and B. I. Epureanu (2017a), Intelligent agent-based dynamic scheduling for military modular vehicle fleets, in *IIE Annual Conference. Proceedings*, pp. 404–409, Institute of Industrial and Systems Engineers (IISE).
- Li, X., and B. I. Epureanu (2017b), Robustness and adaptability analysis of future military modular fleet operation system, in *ASME 2017 Dynamic Systems and Control Conference*, pp. V002T05A003–V002T05A003, American Society of Mechanical Engineers.
- Li, X., and B. I. Epureanu (2018), An agent-based approach for optimizing modular vehicle fleet operation, *Review of International Journal of Production Economics*.
- Li, X., A. E. Bayrak, B. I. Epureanu, and Y. Koren (2018), Real-time teaming of multiple reconfigurable manufacturing systems, *CIRP Annals*.

- Liao, T. W., and P.-C. Chang (2010), Impacts of forecast, inventory policy, and lead time on supply chain inventory: a numerical study, *International Journal of Production Economics*, 128(2), 527–537.
- Linnéusson, G., A. H. Ng, and T. Aslam (2018), Towards strategic development of maintenance and its effects on production performance by using system dynamics in the automotive industry, *International Journal of Production Economics*, 200, 151–169.
- Lynch, J. (2015), On strategic unpredictability, *Modern War Institute at West Point*.
- Ma, X., Z. Tao, Y. Wang, H. Yu, and Y. Wang (2015), Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, *Transportation Research Part C: Emerging Technologies*, 54, 187–197.
- Maidstone, R. (2012), Discrete event simulation, system dynamics and agent based simulation: Discussion and comparison, *System*, pp. 1–6.
- Maione, B., and D. Naso (2001), Evolutionary adaptation of dispatching agents in heterogeneous manufacturing systems, *International Journal of Production Research*, 39(7), 1481–1503.
- Makssoud, F., O. Battaïa, and A. Dolgui (2013), Reconfiguration of machining transfer lines, in *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*, pp. 339–353, Springer.
- Mathieu, V. (2001), Service strategies within the manufacturing sector: benefits, costs and partnership, *International Journal of Service Industry Management*, 12(5), 451–475.
- Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. Scokaert (2000), Constrained model predictive control: Stability and optimality, *Automatica*, 36(6), 789–814.
- Meng, Q., Z. Li, H. Liu, and J. Chen (2017), Agent-based simulation of competitive performance for supply chains based on combined contracts, *International Journal of Production Economics*, 193, 663–676.
- Mikolov, T., M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur (2010), Recurrent neural network based language model, in *Eleventh Annual Conference of the International Speech Communication Association*.
- Nahmias, S., and Y. Cheng (1993), *Production and operations analysis*, vol. 2, Irwin Homewood, IL.
- Negenborn, R. R., B. De Schutter, and J. Hellendoorn (2008), Multi-agent model predictive control for transportation networks: Serial versus parallel schemes, *Engineering Applications of Artificial Intelligence*, 21(3), 353–366.

- Onggo, B. S., and M. Karatas (2016), Test-driven simulation modelling: A case study using agent-based maritime search-operation simulation, *European Journal of Operational Research*, 254(2), 517–531.
- Ouelhadj, D., and S. Petrovic (2009), A survey of dynamic scheduling in manufacturing systems, *Journal of scheduling*, 12(4), 417.
- Pandit, S. M., and S.-M. Wu (1983), *Time series and system analysis, with applications*, John Wiley & Sons.
- Paulson, E. C., I. Linkov, and J. M. Keisler (2016), A game theoretic model for resource allocation among countermeasures with multiple attributes, *European Journal of Operational Research*, 252(2), 610–622.
- Powell, R. (2007), Defending against terrorist attacks with limited resources, *American Political Science Review*, 101(3), 527–541.
- Qu, T., H. Chen, D. Cao, H. Guo, and B. Gao (2015), Switching-based stochastic model predictive control approach for modeling driver steering skill, *IEEE Transactions on Intelligent Transportation Systems*, 16(1), 365–375.
- Reaidy, J., P. Massotte, and D. Diep (2006), Comparison of negotiation protocols in dynamic agent-based manufacturing systems, *International journal of production economics*, 99(1), 117–130.
- Rezaee, M. J., M. Jozmaleki, and M. Valipour (2018), Integrating dynamic fuzzy c-means, data envelopment analysis and artificial neural network to online prediction performance of companies in stock exchange, *Physica A: Statistical Mechanics and its Applications*, 489, 78–93.
- Ripaccioli, G., D. Bernardini, S. Di Cairano, A. Bemporad, and I. Kolmanovsky (2010), A stochastic model predictive control approach for series hybrid electric vehicle power management, in *American Control Conference (ACC)*, 2010, pp. 5844–5849, IEEE.
- Roos, P., and D. Nau (2010), Risk preference and sequential choice in evolutionary games, *Advances in Complex Systems*, 13(04), 559–578.
- Ryu, K., and M. Jung (2003), Agent-based fractal architecture and modelling for developing distributed manufacturing systems, *International Journal of Production Research*, 41(17), 4233–4255.
- Sabuncuoglu, I., and M. Bayız (2000), Analysis of reactive scheduling problems in a job shop environment, *European Journal of operational research*, 126(3), 567–586.
- Scattolini, R. (2009), Architectures for distributed and hierarchical model predictive control—a review, *Journal of process control*, 19(5), 723–731.

- Sharda, B., and N. Akiya (2012), Selecting make-to-stock and postponement policies for different products in a chemical plant: A case study using discrete event simulation, *International Journal of Production Economics*, 136(1), 161–171.
- Shinkman, P. (2014), Trashed: Us gear in afghanistan to be sold, *Scrapped, US News*.
- Silver, E. A., H. Naseraldin, and D. P. Bischak (2009), Determining the reorder point and order-up-to-level in a periodic review system so as to achieve a desired fill rate and a desired average time between replenishments, *Journal of the Operational Research Society*, 60(9), 1244–1253.
- Spicer, P., and H. J. Carlo (2007), Integrating reconfiguration cost into the design of multi-period scalable reconfigurable manufacturing systems, *Journal of Manufacturing Science and Engineering*, 129(1), 202–210.
- Swaminathan, J. M., S. F. Smith, and N. M. Sadeh (1998), Modeling supply chain dynamics: A multiagent approach, *Decision Sciences*, 29(3), 607–632.
- Thonemann, U. W., and M. L. Brandeau (2000), Optimal commonality in component design, *Operations Research*, 48(1), 1–19.
- Tsao, M., R. Iglesias, and M. Pavone (2018), Stochastic model predictive control for autonomous mobility on demand, *arXiv preprint arXiv:1804.11074*.
- Ulrich, K. (1994), Fundamentals of product modularity, in *Management of Design*, pp. 219–231, Springer.
- Van Hessem, D., and O. Bosgra (2006), Stochastic closed-loop model predictive control of continuous nonlinear chemical processes, *Journal of Process Control*, 16(3), 225–241.
- Veselý, V., D. Rosinová, and M. Foltin (2010), Robust model predictive control design with input constraints, *ISA transactions*, 49(1), 114–120.
- Vieira, G. E., J. W. Herrmann, and E. Lin (2003), Rescheduling manufacturing systems: a framework of strategies, policies, and methods, *Journal of scheduling*, 6(1), 39–62.
- Wang, C., and V. M. Bier (2011), Target-hardening decisions based on uncertain multiattribute terrorist utility, *Decision Analysis*, 8(4), 286–302.
- Wang, W., and Y. Koren (2012), Scalability planning for reconfigurable manufacturing systems, *Journal of manufacturing systems*, 31(2), 83–91.
- Xu, J., J. Zhuang, and Z. Liu (2016), Modeling and mitigating the effects of supply chain disruption in a defender–attacker game, *Annals of Operations Research*, 236(1), 255–270.

- Youssef, A. M., and H. A. ElMaraghy (2007), Optimal configuration selection for reconfigurable manufacturing systems, *International Journal of Flexible Manufacturing Systems*, 19(2), 67–106.
- Yu, L., S. Wang, and K. K. Lai (2009), An intelligent-agent-based fuzzy group decision making model for financial multicriteria decision support: The case of credit scoring, *European journal of operational research*, 195(3), 942–959.
- Zhang, J., J. Zhuang, and V. R. R. Jose (2018), The role of risk preferences in a multi-target defender-attacker resource allocation game, *Reliability Engineering & System Safety*, 169, 95–104.
- Zhao, Z., W. Chen, X. Wu, P. C. Chen, and J. Liu (2017), Lstm network: a deep learning approach for short-term traffic forecast, *IET Intelligent Transport Systems*, 11(2), 68–75.
- Zhuang, J., V. M. Bier, and O. Alagoz (2010), Modeling secrecy and deception in a multiple-period attacker–defender signaling game, *European Journal of Operational Research*, 203(2), 409–418.