Article

War Game between Two Matched Fleets with Goal Options and Tactical Optimization

Zhi-Xiang Jia † **and Jean-Fu Kiang** *,†

Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan
* Correspondence: jfkiang@ntu.edu.tw; Tel.: +886-2-33663661
† These authors contributed equally to this work.

Abstract: A war game between two matched fleets of equal size and capability is designed and simulated in this work. Each fleet is composed of a carrier vessel (CV), a guided missile cruiser (CG), and two guided-missile destroyers (DDGs). Each vessel is equipped with specific weapons, including fighters, missiles, and close-in weapon system (CIWS), to carry out tactical operations. The maneuverability, maximum flying distance, and kill probability of different weapons are specified. Three goal options, a defense option and two more aggressive ones, are available to each fleet. A particle-pair swarm optimization (P^2SO) algorithm is proposed to optimize the tactical parameters of both fleets concurrently according to their chosen options. The parameters to be optimized include take-off time delay of fighters, launch time delay of anti-ship missiles (ASHMs), and initial flying directions of fighters and ASHMs, respectively. All six possible contests between options are simulated and analyzed in terms of payoff, impact scores on CV, CG, DDG, and the number of lost fighters. Some interesting outlier cases are inspected to gain some insights on this game.

Keywords: war game; goal option; tactical optimization; particle-pair swarm optimization (P^2SO) algorithm

Citation: Jia, Z.-X.; Kiang, J.-F. War Game between Two Matched Fleets with Goal Options and Tactical Optimization. *AI* **2022**, *3*, 890–930. https://doi.org/10.3390/ai3040054

Academic Editor: Gianni D'Angelo

Received: 16 August 2022
Accepted: 4 November 2022
Published: 14 November 2022

1. Introduction

War-gaming is a technique for exploring complex problems [1] (p. 7), is a tool for exploring decision-making possibilities in an environment with incomplete and imperfect information [1] (p. 3). War game participants may make decisions and take actions in a game that even they would not have anticipated, if not for the game environment [1] (p. 3). War-gaming provides structured but intellectually liberating safe-to-fail environments to help explore what works and what does not, at relatively low cost, and is usually umpired or adjudicated [2] (p. 5). Military wargaming was largely abandoned after World War II and was revived in the early 1970s, inspired by commercial wargames [3] (p. 11).

A wargame is a combination of 'game', history and science [3] (p. 13). It offers unique perspectives and insights that complement other forms of analysis or training on decision-making in complex contexts when faced with a dynamic opponent [2] (p. 11). The value of the war game is that decisions are not constrained by safety, rules of engagement, real-world territorial boundaries, or training objectives. It is different from a training exercise, which uses real forces [1] (p. 4). In a scenario-based warfare model, the outcome and sequence of events affect and are affected by the decisions made by the players [4]. War games are a pre-conflict way to test strategy. Experiential war games provide value to game participants, while experimental war games provide value through the testing of plans [1] (p. 2).

There has been an age-old debate about whether war games are a simulated exercise on military confrontation as realistically as possible, or merely a game to be played and enjoyed for its own sake [5] (p. 16). NATO defines a war game as a simulation of military operation by whatever means, using specific rules, data, methods, and procedures [2] (p. 5).

Ever booming PC capabilities facilitate computer war-games for entertainment, which also drive a tide of viewing war games as a simulation, like other genres of games [3] (p. 7).

War games can be used to inform decisions by raising questions and insights, not to provide a definitive answer [2] (p. 12). More robust conclusions can be drawn by playing multiple games, perhaps with different scenarios and starting conditions [2] (p. 12).

War games immerse participants in an environment with the required level of realism to improve their decision-making skills [2] (p. 5). A simulation might provide the engine that determines outcomes, but is not the war game [2] (p. 7). The elements of a war game include aims and objectives, an immersive gameplay scenario, data and sources to build the setting, scenario and episodes within the war game, simulation of computer-assisted, computerized or manual type to execute episodes, robust rules and procedures, adjudication to determine the outcomes, analysis reliant on data and outcomes to help understand what has happened during a war game and consolidate the benefits of war-gaming [2] (pp. 7–8).

In this work, we design and simulate a war game between two matched fleets with equal size and equal capability. Both fleets are equipped with the same weaponry and follow the same rules of engagement. Each fleet has three strategical goal options to choose, and the tactical parameters are optimized accordingly. Multiple games are then played between the two fleets, and the simulation results are analyzed statistically to gain insights and lessons. In this section, we will review references on tactical maneuvers and episodes that are indispensable to the implementation of game simulation. Typical episodes in the proposed game scenarios include target assignment, path planning, interception, chase between pursuer(s) and evader(s), and simultaneous attack. War games between two opposing parties will be reviewed in more details in the next section of Related Works.

Tactical maneuvering between pursuers and evaders has been widely studied in the literature. In Reference [6], a multiagent pursuit-evasion (MPE) game was developed to derive an optimal distributive control strategy for each agent. A graph-theoretic method was used to model the interaction between agents. A mini–max strategy was developed to derive robust control strategies at reduced complexity rather than solving the Hamilton–Jacobi–Isaacs (HJI) equations of Nash equilibrium. In Reference [7], a guidance law for a pursuer to chase multiple evaders was developed. The pursuer ran after all evaders over a given period, then picked its target evader labeled the highest priority. The guidance law significantly minimized the control effort of the pursuer, but this method does not work if an interceptor is launched against the pursuer. In Reference [8], an autonomous maneuver decision-making method for two cooperative unmanned aerial vehicles (UAVs) in air combat was proposed. The situation or threat was assessed from the real-time positions of target and UAVs. The training was conducted in a 13-dimensional air-combat state space, supported with 15 optional action commands. Both the real-time maneuver gain and the win–lose gain were included in the reward function. A hybrid autonomous maneuver decision strategy in air combat was demonstrated on dual-UAV olive formation, capable of obstacle avoidance, formation, and confrontation. In Reference [9], a particle swarm optimization algorithm was enhanced with simulated annealing for path planning of unmanned surface vehicle (USV), but only static obstacles were considered. Chase between pursuer(s) and evader(s) is a common episode in a war game. In this work, different types of intercepting missiles are fired to chase enemy fighters and different types of attacking missiles.

Interception of a moving target is an important element of defense tactics. In Reference [10], an integrated guidance and control (IGC) design method was proposed to guide a suicide unmanned combat air vehicle (UCAV) against a moving target during the terminal attack stage. A trajectory linearization control (TLC) and a non-linear disturbance observer (NDO) were used to control the accuracy and stability, respectively, of the system. In Reference [11], a swarm of multi-rotor UAVs were dispatched to intercept a moving target while maintaining a predetermined formation. An enhanced three-dimensional pure-pursuit guidance law was used for controlling the trajectories of UAVs, a Kuhn–Munkres

(KM) optimal matching algorithm was used to avoid redundancy or failure in searching for the look-ahead time, and a virtual force-based algorithm was used to avoid collision between UAVs. The proposed method claimed to decrease the length of UAV trajectories, facilitate the interception process, and enhance the formation integrity. In Reference [12], an optimal midcourse trajectory planning based on a Gauss pseudo-spectral method (GPM) was proposed to intercept a hypersonic target, considering the path, terminal and physical constraints. Proportional guidance was used in the terminal phase to meet the challenges on high maneuverability of target, unpredictable trajectory, and detection error. In Reference [13], a team of cooperative interceptors were dispatched from different intercept angles against a moving target in a linear quadratic zero-sum two-party differential game. The cooperative guidance law was used for maneuvering the interceptors to reduce the evasion probability of the target. The proposed guidance law was claimed better than the optimal-control-based cooperative guidance law when the target is unpredictable. In Reference [14], an extended differential geometric guidance law (EDGGL) was developed for guiding a missile to intercept an arbitrarily maneuvering target without foreknowing their motions. A modified EDGGL was developed to determine the direction of missile acceleration without prior knowledge of the target acceleration. Interception of a moving target is a critical tactical action in our game scenario, including the use of missiles to bust enemy fighters or missiles, as well as close-in weapon system (CIWS) as the last defense measure of ships.

Simultaneous attack has been considered an effective tactical movement. In Reference [15], a model-predictive-control (MPC) cooperative guidance law was developed to synchronize the states of all the missiles to achieve simultaneous attack. The salvo coordination was based on the consensus of ranges and range rates of individual missiles to the static target rather than impact time or time-to-go. In Reference [16], an impact-time-control guidance (ITCG) law was developed for hypersonic missiles to simultaneously hit a static target. The ITCG law was applied in the vertical plane, based on the proportional navigation guidance (PNG) law. In Reference [15,16], simultaneous attack became more difficult to achieve if many missiles were used to intercept the attacking missiles, forcing them to reroute. In this work, kill probability of intercepting missile, time gap between consecutive missile launches, and transit time of CIWS engagement are potential weak points that can be broken through with salvo, which will be verified in the simulations.

Target assignment among friendly agents may significantly affect the outcome of a game. In Reference [17], a static weapon target assignment (SWTA) problem was studied to minimize the target lethality and the total missile cost, constrained by the kill probability of missile to target. A combinatorial optimization problem was formed and solved by using an improved artificial fish swarm algorithm-improved harmony search (IAFSA-IHS). In Reference [18], a problem of assigning several cooperative air combat units to attack different targets was studied. Each air combat unit could be assigned to sequentially attack different targets, with specific weapon for each target. A large-scale integer programming problem was formulated, with constraints on flight route, damage cost, weapon cost, and execution time. Target or task assignment is always required to initiate a tactical operation with a team of agents. In this work, target assignment is guided by the goal option, and each agent takes action based on a set of optimal tactical parameters.

Path planning is typically practiced by an agent before taking tactical operation on a target. In Reference [19], a real-time path planning method was developed for an UAV to fly through a battlefield, under threats from multiple radar stations and radar-guided surface-to-air missiles (SAMs). A threat netting model was applied to the radars by sharing the UAV information. A model predictive control (MPC) problem was formed by integrating the threat netting model and the SAM models. The best control scheme in the state space of UAV was searched by minimizing an objective function in terms of normal acceleration, prediction horizon, distance cost, and threat cost. In Reference [20], the impact time for an interceptor to hit a static target within the field-of-view (FOV) was controlled by using a guidance strategy of deviated pursuit (DP) in the early stage, and the terminal lateral acceleration was set to zero in the later stage based on a barrier Lyapunov function. Real-time cooperative tuning parameters of multiple interceptors were adopted to achieve simultaneous attack on the static target. In Reference [21], an episodic memory Q-network model was applied to train a drone swarm for military confrontation. A local-state target-threat assessment method was designed by using expert knowledge, with a more efficient dynamic weight adjustment strategy applied on the episodic memory. In this work, path planning is conducted by adjusting the initial flying angles of fighters and anti-ship missiles (ASHMs). Their actual flying paths are determined by the interaction with enemy agents on their path and the engagement rules that apply. The ideal salvo on specific enemy target is not planned explicitly. By setting different initial flying angles, we will be able to observe salvo events and verify their efficacy in attacking a heavily defended enemy ship.

A practical agent makes trade-off between attacking target to gain some score and avoiding counterattack to survive. In Reference [22], an evasive maneuver strategy was proposed for one of two opponent unmanned combat air vehicles (UCAVs) to evade high-performance beyond-visual-range (BVR) air-to-air missiles (AAMs) fired upon each other. A multi-objective optimization problem was formed in a three-level decision space of maneuver time, type, and parameters. A hierarchical multi-objective evolutionary algorithm (HMOEA) was developed, by combining air combat experience and maneuver decision optimization method, to find the Pareto-optimal strategies. Similar situations take place all the time in the games of our work. For example, a fighter tries to launch missiles against an enemy ship while avoiding missiles launched from nearby ships, or a fighter tries to launch AAM against an enemy fighter while avoiding AAMs fired from the latter.

Similar episodes in these references have been included in our game design. In reciprocal, some practical factors considered in our work may also be incorporated into these works, as summarized in Table 1, to enhance the simulation realism. For example, the impact method and action [17–19], evading maneuver of agents [10,11], kill probability [6,7,19,22], maximum flying distance of agents [18–20], and counterattack of evader or target [6,7,10–17,19,20,22].

Table 1. Suggested practical factors for relevant works in Section 1.

Practical Factor	References
impact method and action	[17–19]
evading maneuverability	[10,11]
kill probability	[6,7,19,22]
maximum flying distance	[18–20]
counterattack of evader or target	[6,7,10–17,19,20,22]

2. Related Works

In this section, we will review the literature on war games between two parties of agents. There are always an attack force and a defense force in a war game. Each force is formed to achieved a strategical goal, which is implemented via constituent agents and weapons. Each agent or weapon follows a tactical plan to act and react with its own capability, constraint and rules of engagement. Different parties may pursue different goals in different settings to compete with their opponent. Asset-guarding games have been widely discussed. In Reference [23], several teams of interceptors cooperatively protected a non-maneuvering asset by intercepting maneuverable and non-cooperative attacking missiles. A combinatorial optimization method was first applied to assign each team of interceptors to simultaneously attack one of the missiles. Then, a cooperative guidance law based on linear quadratic dynamic game theory was derived to control the interceptors to hit the missiles as far away as possible from the asset. In Reference [24], an optimal deployment of missile defense system was proposed to maximize the survivability of the protected assets. A single shot kill probability (SSKP) model for an interceptor was developed by considering the geometry of scenario, trajectories of interceptor and missile, and error models of interceptor and radar. Then, the survivability distribution of protected assets was computed to determine an optimal deployment of the missile defense system.

Target-missile-defender (TMD) games have been widely studied since the cold-war era. In Reference [25], a homing guidance law with control saturation was proposed for a missile to hit a moving target while avoiding a cooperative defender fired from the latter. A cooperative augmented proportional navigation (APN) was adopted by the defender to minimize its acceleration loading. A performance index based on the miss distance to the target and power consumption was proposed. An analytical solution was derived, considering control saturation and minimum power consumption, to guide the missile as close as possible to the target and to evade the defender by at least a specific miss distance. The guidance law was claimed to achieve high precision and reliability, compared to the optimal differential game guidance (ODGG) and combined minimum effort guidance (CMEG). In Reference [26], an optimal online defender launch-time selection method in a TMD game was proposed. A missile tried to hit an evading target while the latter picked an optimal moment to launch a defender against the missile. An autonomous switched-system optimization problem was formed and a deep neutral network (DNN) was proposed to solve the problem with two launch strategies, wait-and-decide or assess-and-decide. This method could deliver accurate prediction of optimal launch-time with little computational load.

Proper task assignment plan for agents can effectively facilitate the tactical operation. In Reference [27], a dynamic task assignment problem was studied between an attack force and a defense force of UCAVs, with each UCAV equipped with different combat platforms and weapons. The defense force cooperatively transported military supplements from its base to the battlefront, while the attack force tried to bash the defense force. A predator–prey particle swarm optimization (PP-PSO) algorithm was developed to search for a mixed Nash equilibrium as the optimal assignment scheme. In Reference [28], an antagonistic game WTA (AGWTA) in a cooperative aerial warfare (CAW) was studied. A non-cooperative zero-sum game was played between two teams of fighters, with fire assault and electronic interference to weaken the opponents and preserve themselves. A decomposition co-evolution algorithm (DCEA)-AGWTA was proposed to derive a non-cooperative Nash equilibrium (NCNE) strategy, by dividing the problem into several subproblems and redistributing the objective functions of both teams to different subproblems. In Reference [29], a group of fighters were assigned to attack an integrated air defense system (IADS). Payoff functions on both parties were defined in terms of value, threat, attacking probability, jamming probability, weapon consumption, and decision vectors for attacking and jamming. An improved chaotic particle swarm optimization (I-CPSO) algorithm was developed to optimize the decision vectors of both parties. Different constraints on both parties were studied, including weapon consumption, radius of non-escape zone, and jamming distance. Each fighter was assigned off-line, without improvising when the situation changed. However, the attacking and jamming probabilities were adjusted over a given interval, leading to unnecessary complexity. In practice, a single value of probability inducted from previous observations is enough to simulate the scenarios. In Reference [30], an algorithm was proposed to simulate a within-visual-range (WVR) air combat between two opponent teams of UCAVs. Each team was divided into several groups, with cooperation only among UCAVs in the same group. Then, different groups were assigned to engage designate enemy groups, with the optimal tactics based on achieving Nash equilibrium.

In Reference [31], force allocation in the confrontation of UAV swarms was analyzed by using Lanchester law and Nash equilibrium. The UAVs were divided into groups to fight in separate battlefields, regarded as a Colonel Blotto game. A method inspired by the double oracle algorithm was proposed to search for the best force allocation. In Reference [32], a close formation of multiple missiles was studied in attacking a ship defended with multi-layered defense system. Such formation was intended to camouflage as a single object when observed far away and took the target ship by surprise at close range. In Reference [33], a description framework of cooperative maritime formation was built on an event graph model, to describe the operations of formation and the evolution of battlefield dynamics. Event-driven task scheduling theory and methods were then proposed to formulate an optimal combat plan.

Denial of intrusion into a designate region is a strategical goal in many war games. In Reference [34], a state-feedback saddle-point strategy was developed on a pursuit-evasion game (PEG), with two defenders trying to deny a designate area from being intruded by a faster invader. In Reference [35], a reach-avoid game between an attacker and two defenders bound within a rectangular domain was studied. The optimal strategies for both parties were derived under different conditions that the attacker moved in an attacker dominance region (ADR), a defender dominance region (DDR), and a barrier, respectively. In Reference [36], a high-dimensional subspace guarding game was studied, with an attacker trying to enter a target subspace protected by two cooperative faster defenders. An attack subspace (AS) method was proposed to form optimal strategies for both defenders and attacker to reach a saddle-point equilibrium, in favor of the defenders. In Reference [37], a multi-player reach-avoid differential game in 3D space was studied, with two cooperative pursuers protecting a target by capturing an evader moving at the same speed. A barrier surface that enclosed a winning subspace was computed, and the saddle-point strategies for agents were derived by using an Isaacs' method and a Hamilton–Jacobi–Isaacs (HJI) equation. In Reference [38], a planar multiagent reach-avoid game was studied, with an evader targeting at a fixed point while avoiding capture by multiple pursuers. A non-linear state feedback strategy based on a risk metric was developed for the evader to reach the target, against pursuers moving under a semi-cooperative strategy. The strategy can be switched by taking different control laws in different state-subspaces.

In Reference [39], a non-zero-sum Hostility Game with four players was studied, in which blue players intended to navigate freely under the obstruction of red players. A parallel algorithm was proposed to form strategies of reaching Nash equilibrium between blue players and red players. In Reference [40], a coastal defense problem of deploying unmanned surface vehicle (USV) to intercept enemy targets entering a warning area was studied. The defense objective function included energy consumption for sailing, contingent cost upon opponent capabilities, and reward of successful interception. A hierarchically distributed multi-agent defense system was proposed on a central and distributed system and cooperative agents, which improved the decision-making efficiency and interception rate over conventional centralized architecture.

Missile combat between battleships in navel warfare has been studied. In Reference [41], a salvo model was developed for exploratory analysis on warship capability and comparison between two naval forces in a naval salvo warfare. The warship capability was composed of combat power and staying power, where the former included the number of force units, number of missiles, scouting effectiveness, training deficiency, enemy distraction chaff, enemy alertness, enemy seduction chaff, and enemy evasion. The percentage of out-of-action units between two forces was compared, without prior knowledge of how and where the warships would fight. The staying power was valuable because a force with weaker combat power but stronger staying power might win the game in the end. In Reference [42], rescheduling of target assignment was made for ship-to-air missiles (SAMs) launched from a naval task group (TG) to intercept incoming air-to-ship missiles (ASMs). The combat condition may suddenly change due to, for example, destruction of an ASM, breakdown of a SAM system, or an ASM pop out of nowhere, hence rescheduling of SAMs should be made at regular intervals after the initial scheduling. A biobjective mathematical model was built to efficiently maximize the probability of successful defense and minimize the difference between rescheduling and the initial scheduling. Small-size problem was solved by using an augmented ε-constraint method and large-size problem was solved by running two fast heuristics procedures to obtain non-dominated solutions.

In Reference [43], a multi-ship cooperative air defense model against incoming missiles was proposed, which considered missile launch rate, launch direction, flight speed, as well as ship interception rate, interception range, number of fire units. The cooperation among ships was modeled via task assignment. The defense system was designed and analyzed with analytical models, and the penetration probability was estimated with the queuing theory. In Reference [44], fleet modularity was evaluated via a game theoretical model of the competition between autonomous fleets in an attacker–defender game. Heuristic operational strategies were obtained through fitting a decision tree on simulation results of an intelligent agent-based model. A multi-stage game theoretical model was also built to identify the Nash equilibrium strategy based on military resources and the outcomes of past decisions. In Reference [45], a framework of consulting air combat of cooperative UAV fleet was studied by using matrix game theory, negotiating theory, and U-solution. The best paring of joint operations between own agents and enemy agents was computed, considering the optimal tactics and situational assessment. NVIDIA graphics processor was used to improve the computing efficiency in solving the equations of motion, consultation, situational assessment, and searching for optimal strategies. In Reference [46], offense/defense confrontation between two UAV swarms of the same size and capability on an open sea was studied. The UAVs aimed to attack the enemy aircraft carrier, while protecting own aircraft carrier from enemy UAVs. Each UAV made independent decisions based on the behavioral rules, detection radius, and enemy position. A distributed auction-based algorithm was used to guide the UAVs in making their real-time decisions with limited communication capability. The algorithm parameters were optimized to facilitate target allocation and elimination.

There are other relevant issues in the literature. In Reference [47], a maneuvering plan for beyond-visual-range (BVR) engagement was developed, with two opponent aircrafts taking actions of pursuit, head-on attack, or flee. The goal was to prevent an aircraft from entering the missile attack zone of the opponent, reduce meaningless maneuver of itself, and lure the opponent to its missile attack zone. A long short-term memory-deep Q network (LSTM-DQN) algorithm was proposed for situational awareness and action decision-making. A reward function was defined to account for the threat of enemy missile, reduction in meaningless maneuver, and the boundary of battlefield. In Reference [48], a mean-field multi-agent reinforcement learning (MARL) method was developed, by modeling the interaction among a large number of agents as a single agent playing against a multi-agent system. A mean-field Q-learning algorithm and a mean-field actor-critic algorithm were applied to a mixed cooperative–competitive battle game, with two teams of agents fighting each other by using different reinforcement learning algorithms. The goal of each team was to collaborate with the allied agents to neutralize as many opponents as possible. In Reference [49], a multi-UAV cooperative air combat was studied, with a team of UAVs against their targets. The maneuver decision model of each UAV was built on an actor-critic bidirectional recurrent neural network (BRNN), taking all the UAV states concurrently to achieve cooperation, and the reward value was computed with the results in target assignment and air combat situation assessment.

In Reference [50], an adversarial intelligence and decision-making (RAID) program of the Defense Advanced Research Projects Agency (DARPA) was developed to estimate the enemy's movements, positions, actions, goals, and deceptions in urban battles. Approximate game-theoretic and deception-sensitive algorithms were used to acquire real-time estimation for the commander to execute and modify tactical operations more effectively. In Reference [51], an intelligent agent-based model was proposed to optimize fleets of modular military vehicles in order to meet the dynamic demands on real-time response to adversarial actions. A game was played between a modular intelligent fleet and a conventional intelligent fleet, in which the former predicted its adversary's actions from historical data and optimized its own dispatch decision. In Reference [52], a partially observable asynchronous multi-agent cooperation challenge (POAC) was built for testing the performance of different multi-agent reinforcement learning (MARL) algorithms in a war game between two armies. Each army was composed of heterogeneous agents, each acting on observations and asynchronously cooperating with other agents. The POAC could be configured to meet various requirements such as a human–AI model and a self-play model.

In Reference [53], a multi-agent deep deterministic policy gradient (MADDPG) algorithm was built to improve the capability of an UAV swarm in confronting another swarm. A rule-coupled method was proposed to effectively increase the winning rate and reduce the required action time. In Reference [54], a multi-agent deep reinforcement learning (MADRL) method was used to conduct UAV swarm confrontation. Two non-cooperative game models based on MADRL were built to analyze the dynamic game process and acquire the Nash equilibrium. In Reference [55], confrontation between two UAV swarms in a territory-defense scenario was studied, in which the UAVs maneuvered by searching for the Nash equilibrium from the measured cost functions without explicit expressions. Each UAV followed second-order fully-actuated dynamics and focused on minimizing the coalition cost.

In Reference [56], a real-time strategy (RTS) was applied on a bot to navigate an unfamiliar environment with a multi-agent potential field, in order to search and attack enemies. In Reference [57], a hierarchical multi-agent reinforcement learning framework was proposed to train an AI model in a traditional wargame played on a hexagon grid. A high-level network was used for cooperative agent assignment, and a low-level network was used for path planning. A grouped self-play approach was proposed to enhance the AI model in contending various enemies. In Reference [58], a distributed interaction campaign model (DICM) was proposed for campaign analysis and asset allocation of geographically distributed naval and air forces. The model accounted for the uncertainty of enemy plan, the factors that boost the force and the factors that disrupt enemy command, control, and surveillance. In Reference [59], a stochastic diffusion model for two opponent forces in an air combat was proposed to study the trade-off between logistics system and combat success, including factors like lethality, endurance, non-linear effect of logistics, and uncertainty of measurement. In Reference [60], a high-fidelity simulator was used to optimize the tactical formation of autonomous UAVs in a BVR combat against the other UAV team, considering uncertainties, such as firing range and position error of the enemy.

Similar settings in these references have been included in our work. In reciprocal, some practical factors considered in our work may also be incorporated into these works, as summarized in Table 2, to enhance the simulation realism. For example, the impact method and action [28,29,41,42], evading maneuver of agents [26,27,48], kill probability [23,30,35,36,41,48,49], and maximum flying distance of agents [30,38,48]. The boundary of battlefield in [35,47] can be removed, the number of agents in [25,26,34–38] can be increased, and the optimization scheme in [23,24,26,38,47] can be extended to both parties.

Table 2. Suggested practical factors for relevant works in Section 2.

Practical Factor	References
impact method and action	[28,29,41,42]
evading maneuverability	[26,27,48]
kill probability	[23,30,35,36,41,48,49]
maximum flying distance	[30,38,48]
remove boundary of battlefield	[35,47]
increase number of agents	[25,26,34–38]
optimization on both parties	[23,24,26,38,47]

In this work, a war game between two matched fleets of equal size and equal capability is developed and played by simulations. The results are analyzed to gain some insights on the risks of different strategical options. The rules of engagement and tactical maneuvers of agents are delineated to set up a credible game environment. The constraint of matched fleets maybe too tight, but the game scenarios and simulation results are already complicated and enlightening. This work will focus on the comparison of goal options and the optimization of tactical parameters to achieve the goal.

To preserve the complexity of the game, three types of vessels (carrier vessel, guided missile cruiser, guided-missile destroyer), four types of missiles (air-to-air, air-to-ship, ship-to-ship, ship-to-air) are included. The numbers of fighters and missiles are large enough to manifest the complicated game scenario, yet not too large to muddle adjudication and lessons that can be learned. The characteristics of the agents are comparable to those in the literature to make the games close to realism, including weaponry parameters, route planning, evading maneuver, and maximum flying distance of fighters and missiles. The imperfect kill probability of missiles against different targets is critical to cast uncertainty in each individual game. With all the stages prepared, we will be able to play games between these two matched fleets, each can pick one out of three distinct goal options. Each fleet will then optimize its tactical parameters, including take-off time delay of fighters, launch time delay of anti-ship missiles (ASHMs), and initial flying directions of fighters and ASHMs. The optimal parameters depend not only the goal option of own fleet, but also that of the opponent fleet. Hence, a particle-pair swarm optimization (P^2SO) algorithm is proposed to concurrently search the optimal tactical parameters for both fleets. The optimal parameters derived with the P^2SO algorithm are then used to play multiple games, and the outcomes of all games are analyzed in terms of the payoff distribution and the cumulative distribution of impact scores on carrier vessel (CV), guided missile cruiser (CG), guided-missile destroyer (DDG), and lost fighters. Some interesting outlier games are further inspected to gain more understanding of the tactical operations.

The war game proposed in this work is aimed to mimic real-life wars to some extents, hence appear complex at the first glance. However, all the games are played by the rules, the statistics of multiple games are compact for assessment and some insights could be learned. The most complex and confusing parts maybe the game rules, which are inevitable for the following reasons. Firstly, we try to incorporate more practical factors, such as evasion, flying distance, kill probability, goal options, take-off/landing time gap, missile launching time gap, diverse attack orientation and method, alert radius, attack range, counterattack, impact score, cost, and so on. The simulated scenarios become diverse as many practical factors are included. Secondly, we try to design a matched game, without dominant factors or overwhelming agents, in order to observe subtle nuances that may affect the outcome of a game.

The rest of this work is organized as follows. All the game rules are presented in Section 3, the principle and implementation scheme of the proposed P^2SO algorithm are elaborated in Section 4. Since the game scenarios are complicated, different goal options are carried out by different sets of rules, and the game rules in different stages of a game and the engagement rules for different vessels are prepared in separate subsections for clarity of presentation.

We propose the P^2SO algorithm in Section 4 to optimize the objective function specified in Section 3.1 via playing multiple games, following the rules specified in Sections 3.3–3.7. The optimal tactical operations thus obtained are used to play another set of games, and the payoff specified in Section 3.2 is computed at the end of each game for statistical analysis.

All six possible contests between goal options are simulated and analyzed in Section 5, summarized and compared in Section 6. A few interesting outlier cases are further inspected in Section 7, more retrospective discussions are presented in Section 8, and some conclusions are drawn in Section 9.

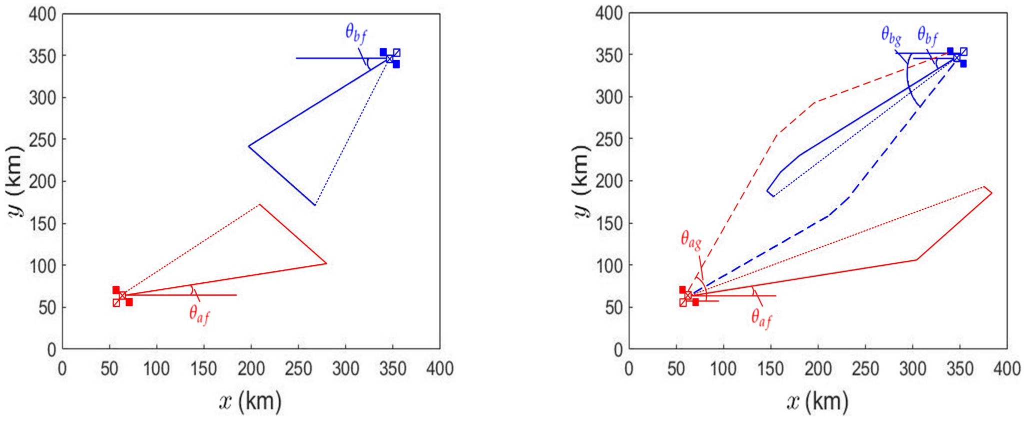
Table 3 lists all the symbols used in this work, in an alphabetical order, for the convenience of the readers.

Table 3. List of symbols.

Symbol	Definition	Reference
(c_1, c_2)	acceleration constants	(12)
(c_3, c_4)	acceleration constants	(16)
D_{aam}	maximum AAM flying distance	Table 4
D_{ashm}	maximum ASHM flying distance	Table 6
D_{asm}	maximum ASM flying distance	Table 4
D_f	maximum fighter flying distance	Table 4
D_{sam}	maximum SAM flying distance	Table 5
$\bar{G}_a$	global best position of fleet A	(12)
$\bar{G}_b$	global best position of fleet B	(16)
$I_{\max}$	maximum iteration number	Table 9
M_{ashm}	ASHM maneuverability	Table 6
M_{asm}	ASM maneuverability	Table 4
M_f	fighter maneuverability	Table 4
M_{sam}	SAM maneuverability	Table 5
N_p	number of particle pairs	Table 9
p_{aamh}	AAM kill probability against high-speed target	Table 4
p_{aaml}	AAM kill probability against low-speed target	Table 4
p_{samh}	SAM kill probability against high-speed target	Table 5
p_{saml}	SAM kill probability against low-speed target	Table 5
$\bar{P}_{an}$	best position of fleet A in particle-pair n	(12)
P_{ashm}	position of own ASHM	(8)
P_{asm}	position of own ASM	(6)
$\bar{P}_{bn}$	best position of fleet B in particle-pair n	(16)
P_f	position of own fighter	(5)
P_{sam}	position of own SAM	(7)
P_m, P_t	position of a missile or target	(10)
Q	position of enemy SAM or AAM	(5)
(r_1, r_2)	random numbers over $[0, 1]$	(12)
(r_3, r_4)	random numbers over $[0, 1]$	(16)
R_{ashm}	ASHM alert radius	Table 6
R_{asm}	ASM alert radius	Table 4
R_{ci}	maximum firing range of CIWS	Table 7
R_{ddg}	DDG alert radius	Table 5
R_f	fighter alert radius (against missile)	Table 4
R_i	impact radius of all missiles	Table 8
R_{sam}	SAM alert radius	Table 5
t	progress time	(9)
t_a	anticipated intercept time on target	(10)
t_{gf}	take-off/landing time gap	Table 4
t_{gm}	minimum time gap between missile launches	Table 8
t_p	CIWS preparation time	Table 7
t_r	remaining flight time of fighter	-
t_s	time instant for CIWS to open fire	(9)
T_{fp}	CIWS firing period	(9)
$\bar{U}_t$	velocity vector of target	(10)
$\bar{V}_{an}$	particle velocity of fleet A in particle-pair n	(12)
$\bar{V}_{an}^*$	old particle velocity of fleet A in particle-pair n	(12)
$\bar{V}_{bn}$	particle velocity of fleet B in particle-pair n	(16)
$\bar{V}_{bn}^*$	old particle velocity of fleet B in particle-pair n	(16)
v_{aam}	AAM speed	Table 4
v_{ashm}	ASHM speed	Table 6
v_{asm}	ASM speed	Table 4
v_f	fighter speed	Table 4
v_m	speed of missile	(10)
v_{sam}	SAM speed	Table 5
w_v	weight on particle velocity	(12)
$\bar{X}_{an}$	particle position of fleet A in particle-pair n	(11)
$\bar{X}_{an}^*$	old position of fleet A in particle-pair n	(12)

Table 3. *Cont.*

Symbol	Definition	Reference
X_{ashm}	anticipated intercept point of own ASHM	(8)
X_{asm}	anticipated intercept point of own ASM	(6)
$\bar{X}_{bn}$	particle position of fleet B in particle-pair n	(17)
$\bar{X}^*_{bn}$	old position of fleet B in particle-pair n	(16)
X_f	anticipated intercept point of own fighter	(5)
X_{sam}	anticipated intercept point of own SAM	(7)
δ_{anfm}	take-off time delay of fighter m of fleet A in particle-pair n	(11)
$\delta_{ang\ell}$	launch time delay of ASHM ℓ of fleet A in particle-pair n	(11)
δ_{bnfm}	take-off time delay of fighter m of fleet B in particle-pair n	(18)
$\delta_{bng\ell}$	launch time delay of ASHM ℓ of fleet B in particle-pair n	(19)
Δt	simulation time step	Table 8
Δt_{ci}	time interval if missile in R_{ci}	Table 7
θ_{af}	initial flying angle of fighters from CV-A	Figure 1
θ_{anf}	initial flying angle of fighters of fleet A in particle-pair n	(11)
θ_{ang}	initial flying angle of ASHMs of fleet A in particle-pair n	(11)
θ_{bf}	initial flying angle of fighters from CV-B	Figure 1
△	symbol of fighter	Figure 19
●	symbol of AAM	Figure 19
⋆	symbol of ASM	Figure 19
Hexagram	symbol of ASHM	Figure 19
ø	symbol of SAM	Figure 19
×	symbol of intercept point	Figure 19
⊠	symbol of CV	Figure 1
⧄	symbol of CG	Figure 1
solid □	symbol of DDG	Figure 1



(**a**) (**b**)

Figure 1. Examples of game scenario, (**a**) defense option, (**b**) attack option, red marks for fleet A and blue marks for fleet B, square with cross: CV, square with slash: CG, solid square: DDG, ———: fighter forward path, ⋯⋯: fighter return path, — — —: ASHM path.

3. Game Rules

The problem in this work is to evaluate possible risk of contest between two matched fleets, each is allowed to choose among three goal options, followed by optimizing the tactical parameters to fulfill the chosen goal option. The tactical parameters include take-off time delay of fighters, launch time delay of anti-ship missiles (ASHMs) and initial flying directions of fighters and ASHMs. The outcome of a game is boiled down to payoffs for both fleets. The commanding officer will pick a goal option, and the payoffs summarized from the war games suggest potential risk to take. The probability distribution of payoffs reveals more nuances for an experienced commanding officer to speculate.

Figure 1 shows examples of game scenario between two fleets, with defense option and attack option, respectively. Each fleet is composed of a carrier vessel (CV), a guided missile cruiser (CG) and two guided missile destroyers (DDGs). The distance between two CVs is 400 km. Each DDG is 10 km beside the CV and the CG is 10 km behind the CV.

Each CV carries 10 fighters, each fighter can carry air-to-ship missiles (ASMs) and air-to-air missiles (AAMs). Note that ASM and ASHM are anti-ship missiles launched from fighters and CG, respectively. The ASMs are used to attack enemy vessels and the AAMs are used to intercept enemy fighters, AAMs and ship-to-air missiles (SAMs). Each fighter carries 4 AAMs under a defense option, or 2 ASMs plus 2 AAMs under an attack option. The speeds of fighter, ASM and AAM are v_f, v_{asm}, and v_{aam}, respectively, and their maximum flying distances are D_f, D_{asm} and D_{aam}, respectively. Each fighter must return to the CV before running out of fuel. The minimum time gap between consecutive take-offs or landings of fighters is t_{gf}. A CG carries 20 ASHMs to attack enemy vessels. The speed of ASHM is v_{ashm} and its maximum flying distance is D_{ashm}. Each DDG carries 30 SAMs to intercept enemy fighters, ASHMs, ASMs, and SAMs. The speed of SAM is v_{sam} and its maximum flying distance is D_{sam}. The minimum time gap between missile launches from the same fighter or vessel is t_{gm}. The symbols, as listed in Tables 4–8, are chosen to comply with the features of the indicated parameters.

Table 4. Parameters of CV and fighters.

Parameter	Symbol	Value
CV cost	-	USD 6.2 B
fighter speed	v_f	1 mach
fighter cost	-	USD 66.9 M
maximum fighter flying distance	D_f	1000 km
fighter alert radius (against missile)	R_f	50 km
fighter maneuverability	M_f	0.3
take-off/landing time gap	t_{gf}	60 s
ASM speed	v_{asm}	2 mach
ASM cost	-	USD 1.67 M
maximum ASM flying distance	D_{asm}	400 km
ASM alert radius	R_{asm}	40 km
ASM maneuverability	M_{asm}	0.45
AAM speed	v_{aam}	1.2 mach
AAM cost	-	USD 1.09 M
maximum AAM flying distance	D_{aam}	160 km
AAM kill probability against high-speed target	p_{aamh}	0.55
AAM kill probability against low-speed target	p_{aaml}	0.7

Table 5. Parameters of DDG.

Parameter	Symbol	Value
DDG cost	-	USD 679.6 M
DDG alert radius	R_{ddg}	200 km
SAM speed	v_{sam}	2 mach
SAM cost	-	USD 4.32 M
maximum SAM flying distance	D_{sam}	200 km
SAM alert radius	R_{sam}	50 km
SAM maneuverability	M_{sam}	0.45
SAM kill probability against high-speed target	p_{samh}	0.6
SAM kill probability against low-speed target	p_{saml}	0.75

Table 6. Parameters of CG.

Parameter	Symbol	Value
CG cost	-	USD 1 B
ASHM speed	v_{ashm}	2 mach
ASHM cost	-	USD 4 M
maximum ASHM flying distance	D_{ashm}	800 km
ASHM alert radius	R_{ashm}	50 km
ASHM maneuverability	M_{ashm}	0.45

Table 7. Parameters of CIWS.

Parameter	Symbol	Value
maximum firing range	R_{ci}	5.5 km
firing period	T_{fp}	13 s
preparation time	t_p	2 s
time interval if missile in R_{ci}	Δt_{ci}	2 s

Table 8. Other parameters.

Parameter	Symbol	Value
minimum time gap between missile launches	t_{gm}	15 s
simulation time step	Δt	15 s
impact radius of all missiles	R_i	50 m

Before a game is played, each fleet chooses a goal option, based on which to search for the optimal tactical parameters, including take-off time delay of fighters, launch time delay of ASHMs, and initial flying directions of fighters and ASHMs, in order to achieve the highest possible payoff. Then, a game is played and the outcomes on both fleets are recorded for analysis.

3.1. Goal Options and Objective Functions

Three goal options are considered in this work. A defensive goal option 1 aims to minimize the enemy threat by shooting down the intruding fighters and missiles. Option 2 aims to wreck havoc on enemy CV to significantly cripple the anti-ship threat from enemy fighters. Option 3 aims to paralyze the enemy fleet, including its anti-ship and defending capabilities.

Each option is carried out with an optimal set of tactical parameters acquired by maximizing a corresponding objective function defined as

$$\begin{aligned}\text{objective function of option 1} &= 0.1 \times (\text{number of enemy fighters lost}) \\ &-(\text{total impact scores on own CV}) - 0.2 \times (\text{total impact scores on own CG}) \\ &-0.1 \times (\text{total impact scores on own DDG})\end{aligned} \tag{1}$$

$$\text{objective function of option 2} = (\text{total impact scores on enemy CV}) \tag{2}$$

$$\begin{aligned}\text{objective function of option 3} &= (\text{total impact scores on enemy CV}) \\ &+0.2 \times (\text{total impact scores on enemy CG}) + 0.1 \times (\text{total impact scores on enemy DDG})\end{aligned} \tag{3}$$

Option 1 aims to deter enemy threat by shooting down the intruding fighters. Hence, the objective function of option 1 is set to the value accrued from hitting enemy fighters minus the value lost in the hit of own ships. The weighting coefficients on number of enemy fighters lost, total impact scores on own CV, total impact scores on own CG and total impact scores on own DDG are proportional to the cost of one fighter or the construction cost of ships listed in Tables 4–6. Option 2 aims to strike only the enemy CV, hence the objective function is proportional to the number of hits on the enemy CV. Option 3 aims to inflict wreck on all enemy ships, hence the objective function is a sum over hits on all enemy ships, with the weighting coefficients on total impact scores on enemy CV, total impact scores on enemy CG, and total impact scores on enemy DDG proportional to their respective construction cost. Intangible values of fighters or ships are not considered in setting the weighting coefficients. The optimal tactical parameters, obtained by applying the P^{2}SO algorithm, are used to play multiple games and derive the statistics of payoff on both fleets.

3.2. Payoff

At the end of each game, payoff is counted on each fleet as

$$\begin{aligned}\text{payoff} = {} & 620 \times (\text{total impact scores on own CV}) + 100 \times (\text{total impact scores on own CG}) \\ & +67.96 \times (\text{total impact scores on own DDG}) + 66.9 \times (\text{number of own fighters lost}) \\ & +1.67 \times (\text{numbers of own ASM fired on targets or sunk with vessels}) \\ & +1.09 \times (\text{numbers of own AAM fired on targets or sunk with vessels}) \\ & +4.32 \times (\text{numbers of own SAM fired on targets or sunk with vessels}) \\ & +4 \times (\text{numbers of own ASHM fired on targets or sunk with vessels})\end{aligned} \tag{4}$$

where the total impact scores on own CV is set to 10 if it is greater than 10 (CV is sunk), the total impact scores on own CG is set to 8 if it is greater than 8 (CG is sunk), and the total impact scores on own DDG is set to 8 if it is greater than 8 (DDG is sunk). The weighting coefficients are proportional to the construction cost of the assets listed in Tables 4–6. The ammunition cost of CIWS is neglected.

3.3. Rules on CV and Fighters

Fighters take off from CV to engage enemy CV, CG, DDGs, and fighters. If option 1 is taken, all the fighters will try to intercept enemy fighters and missiles. If option 2 is taken, all the fighters will attack the enemy CV. If option 3 is taken, the enemy CV, CG, DDG 1, and DDG 2 will be targeted by 3, 3, 2, and 2 fighters, respectively. The fighters take off in a repeated sequence to target DDG 1, DDG 2, CG, and CV, until all the fighters are launched.

Figure 1a shows an example of game scenario with both fleets taking defense option 1. A fighter from CV-A flies north-east at an initial angle θ_{af} measured from the east. It flies 90% of the path from CV-A to the middle line between the two CVs, turns parallel to the middle line and flies for 100 km, then returns to CV-A along a straight path. A fighter in CV-B follows the same rule except its initial position is CV-B and its initial angle is θ_{bf} measured from the west.

Figure 1b shows an example of game scenario with both fleets taking attack option 2 or 3. A fighter from CV-A flies north-east at an initial angle θ_{af} measured from the east. It reaches the middle line first, turns in the direction parallel to the line connecting the two CVs and flies 80% of the distance measured from its position on the middle line to the center point between two CVs, then turns straight towards CV-B. A fighter in CV-B follows the same rule except its initial position is CV-B and its initial angle is θ_{bf} measured from the west. When the distance between a fighter and the enemy CV is shorter than 170 km, the fighter fires its first ASM that flies along a straight path to its designate target. Meanwhile, the fighter turns and flies parallel to the middle line until all the ASMs are fired, then returns to its CV along a straight path.

If an enemy SAM or AAM, labeled as Q, flies within the alert radius R_f of the fighter, sited at P_f, with anticipated intercept point at X_f, then the fighter fires an AAM against it. At the same time, the velocity vector of the fighter is adjusted to

$$\begin{aligned}& \text{Adjusted velocity vector of fighter} \\ & = v_f \frac{M_f \times (\text{unit vector of } \overline{QX_f}) + (1 - M_f) \times (\text{unit vector of } \overline{P_f X_f})}{\left| M_f \times (\text{unit vector of } \overline{QX_f}) + (1 - M_f) \times (\text{unit vector of } \overline{P_f X_f}) \right|}\end{aligned} \tag{5}$$

where v_f is the fighter speed, M_f is the fighter maneuverability. The AAM tries to intercept the enemy missile at an intercept point or tailgates the enemy missile if no intercept point is available.

Under option 1, when a fighter detects an enemy fighter within $R_f + 30$ km and an intercept point is available, it will fire an AAM to attack the enemy fighter. If the intercept point is lost track of during pursuit, the AAM will tailgate the targeted enemy fighter. Each enemy fighter can be marked by at most two AAMs, and each enemy SAM and AAM can be marked by only one AAM. If a fighter finds itself being targeted by an enemy missile which is not marked by any AAM, it will fire an AAM against it. When a fighter successfully evades enemy missile, which means no enemy missile within range R_f and no intercept point is found, the fighter will return to its own CV along a straight path.

Under option 2 or 3, when a fighter detects any enemy fighter within $R_f + 55$ km, it will fire an AAM to intercept it if an intercept point is found. If the intercept point is lost track of during pursuit, the AAM will tailgate the marked enemy fighter. Each enemy fighter, SAM and AAM can be marked by only one AAM. If a fighter finds itself being targeted by enemy missile which is not marked by any AAM, it will fire an AAM against it. When a fighter successfully evades enemy missile, which means no enemy missile within range R_f and no intercept point is found, the fighter will fly in a straight path towards the enemy CV if it is farther than 170 km away, or fly parallel to the middle line to fire ASMs if the enemy CV is within 170 km of range.

An ASM on its way to the designate vessel regularly checks if any SAM is fired to intercept itself. If an ASM, sited at P_{asm} detects an enemy SAM at Q within an alert radius R_{asm} and the intercept point X_{asm} lies between the ASM and its designate vessel, then the ASM will adjust its velocity vector to

$$\begin{aligned}&\text{Adjusted velocity vector of ASM}\\&= v_{asm}\frac{M_{asm}\times(\text{unit vector of }\overline{QX_{asm}})+(1-M_{asm})\times(\text{unit vector of }\overline{P_{asm}X_{asm}})}{\left|M_{asm}\times(\text{unit vector of }\overline{QX_{asm}})+(1-M_{asm})\times(\text{unit vector of }\overline{P_{asm}X_{asm}})\right|}\end{aligned}\tag{6}$$

where v_{asm} is the ASM speed, M_{asm} is the ASM maneuverability. When an ASM successfully evades enemy SAM or no intercept point lies ahead, it will approach its target vessel along a straight path. When a fighter uses up all the ASMs, it will return to its own CV along a straight path. If a fighter finds itself being targeted by enemy missile on the return path, it will fire AAM against it.

Under any circumstance, if it takes longer than $t_r - 3t_{gf}$ for a fighter to fly straight back to its own CV, where t_r is the remaining flight time of the fighter, it will return to its own CV along a straight path. When a fighter finds itself being targeted by enemy missile on its return path, it will fire an AAM against it and remain its original course. When a fighter returns to its own CV earlier than the landing time of the last fighter plus t_{gf}, it will wait around CV for its turn to land.

3.4. Rules on DDG

Under any circumstance, any enemy fighter, ASHM, ASM, or SAM within the alert radius R_{ddg} of a DDG will trigger the latter to fire SAM for interception. Each enemy fighter, ASHM, ASM, and SAM can be marked by only one SAM. If a SAM, sited at P_{sam} finds an enemy missile (AAM or SAM) at Q is fired to intercept itself, which means the enemy missile is within the alert radius R_{sam} of the SAM with intercept point at X_{sam}, the SAM will adjust its velocity vector to

$$\begin{aligned}&\text{Adjusted velocity vector of SAM}\\&= v_{sam}\frac{M_{sam}\times(\text{unit vector of }\overline{QX_{sam}})+(1-M_{sam})\times(\text{unit vector of }\overline{P_{sam}X_{sam}})}{\left|M_{sam}\times(\text{unit vector of }\overline{QX_{sam}})+(1-M_{sam})\times(\text{unit vector of }\overline{P_{sam}X_{sam}})\right|}\end{aligned}\tag{7}$$

where v_{sam} is the speed of SAM, M_{sam} is the maneuverability of SAM.

3.5. Rules on CG

A CG is equipped with ASHMs to attack enemy CV, CG, and DDGs. Under option 1, no ASHM will be fired. Under option 2, all the ASHMs will be fired against the enemy CV. Under option 3, the enemy CV, CG, DDG 1, and DDG 2 will be targeted with 6, 6, 4, and 4 ASHMs, respectively. The launch sequence is repeated as two ASHMs for enemy DDG 1, two for enemy DDG 2, two for enemy CG, two for enemy CV, until all the ASHMs are fired. The flying path of an ASHM is similar to that of a fighter, as shown in Figure 1b. It reaches the middle line first, turns in parallel to the line connecting the two CVs and flies 80% of the distance measured from its position on the middle line to the center point between two CVs, then flies straight towards the designate vessel.

If an ASHM at P_{ashm} find itself marked by an enemy SAM at Q within the alert radius R_{ashm} of the ASHM, with intercept point at X_{ashm}, the ASHM will adjust its velocity vector to

$$\begin{aligned}&\text{Adjusted velocity vector of ASHM}\\&= v_{ashm}\frac{M_{ashm}\times(\text{unit vector of }\overline{QX_{ashm}})+(1-M_{ashm})\times(\text{unit vector of }\overline{P_{ashm}X_{ashm}})}{\left|M_{ashm}\times(\text{unit vector of }\overline{QX_{ashm}})+(1-M_{ashm})\times(\text{unit vector of }\overline{P_{ashm}X_{ashm}})\right|}\end{aligned} \tag{8}$$

where v_{ashm} is the speed of ASHM, M_{ashm} is the maneuverability of ASHM. When the ASHM evades enemy SAM, it will fly straight to its target vessel.

3.6. Rules on CIWS

Each vessel is equipped with a CIWS to defend against enemy missiles at close range. If an enemy missile flies within the maximum firing range R_{ci} of a vessel, its CIWS will engage the former. The CIWS can mark only one target at a time and the ammunition can last to the end of the game. The kill probability of CIWS is modeled as

$$\text{kill probability of CIWS} = \frac{1-e^{-(t-t_s)/T_{fp}}}{1-e^{-1}}, \quad t_s \le t \le t_s + T_{fp} \tag{9}$$

where t is the progress time, T_{fp} is the firing period, and t_s is the time to open fire. At $t = t_s + T_{fp}$, the CIWS halts for a period of t_p to prepare its next engagement. Then, the kill probability of (9) is resumed. Table 7 lists the parameters of CIWS and Table 8 lists other parameters used in the simulations.

3.7. Other Rules

Since the vessels move much slower than fighters or missiles, the vessels are approximated as static objects during the game period of about 3600 s. If a missile, sited at P_m, moving at speed v_m, tries to intercept a target, sited at P_t, moving at velocity vector $\bar{U}_t$, the anticipated intercept time t_a is computed by solving

$$|P_t + \bar{U}_t t_a - P_m| - v_m t_a = 0 \tag{10}$$

and the anticipated intercept point is at $P_t + \bar{U}_t t_a$.

A fighter or a missile takes its first priority to react to the nearest enemy missile with the anticipated intercept point falling within the alert radius of the former. Each enemy fighter or enemy missile can not be marked by more than one missile of the same kind, except an enemy fighter under option 1. If a missile or fighter is hit by one of the two intercept missiles, the other one will turn to pursue the nearest enemy fighter or missile of the same kind. If no other enemy fighter or missile is nearby, the intercept missile keeps flying until fuel is burned out.

Impact scores of 1 and 2 are gained if an ASM and an ASHM, respectively, hits a vessel. If the score on a CV is higher than 4, it can no longer launch fighters and the return fighters will dive to the water. If the score on a CV is higher than 9, the CV is sunk, bringing the carried missiles and fighters to the bottom. If the score on a CG or DDG is higher than 4, it can no longer launch missiles. If the score on a CG or DDG is higher than 7, the CG or DDG is sunk, bringing all the carried missiles to the bottom.

4. Particle-Pair Swarm Optimization

The P^{2}SO algorithm proposed in this work is designed to search for the optimal solution through a bunch of particle pairs. Each particle pair is consisted of particles A and B, each having its position and velocity. The position of particle A(B) represents a candidate set of tactical parameters for fleet A(B). In each iteration, the velocity of each particle is updated first, which is then used to update the particle position. The velocity is updated in terms of the current particle position, particle's best position and the global best position (among all the particle positions). By doing so, the updated particle position tends to be closer the particle's best position (local optimum) or the global best position (global optimum).

After updating the position of a particle pair, a game is played to acquire an objective function for each particle, which is then used to update the particle's best position and the global best position. The other particle pairs also update their positions, play a game and acquire an objective function to update their best position and the global best position.

Several iterations are taken to approach the global optimum. The performance of each particle pair keeps improving iteration by iteration. Finally, the global best position is mapped to the optimal tactical parameters of each fleet to play another set of games, of which the results are analyzed to gain some insights.

The parameters of each fleet include the take-off time delay of each fighter, the launch time delay of each ASHM and the initial flying angles of fighters and ASHMs, respectively. Denote the particle position of fleet A in the nth particle pair as

$$\bar{X}_{an} = (\delta_{anf1}, \cdots, \delta_{anf10}, \delta_{ang1}, \cdots, \delta_{ang20}, \theta_{anf}, \theta_{ang}) \tag{11}$$

where $0 \leq \delta_{anfm} \leq 60$ is the take-off time delay of the mth fighter, $0 \leq \delta_{ang\ell} \leq 60$ is the launch time delay of the ℓth ASHM, $10^\circ \leq \theta_{anf} \leq 80^\circ$ is the initial flying angle of fighters, and $10^\circ \leq \theta_{ang} \leq 80^\circ$ is the initial flying angle of ASHMs. The particle position $\bar{X}_{bn}$ of fleet B is defined in the same manner as (11), with subscript a changed to b.

A swarm of N_p particle pairs are generated, each with random initial position and velocity. The P^{2}SO algorithm is applied to update the particle positions, based on the goal options chosen by both fleets and the associated objective functions. The best positions of the nth particle pair are denoted as ($\bar{P}_{an}$, $\bar{P}_{bn}$), and the global best positions of all the particle pairs are denoted as ($\bar{G}_a$, $\bar{G}_b$).

In each iteration of the P^{2}SO algorithm, the velocity of the nth particle for fleet A is updated first as

$$\bar{V}_{an} \leftarrow w_v \bar{V}^*_{an} + c_1 r_1 (\bar{P}_{an} - \bar{X}^*_{an}) + c_2 r_2 (\bar{G}_a - \bar{X}^*_{an}) \tag{12}$$

where $\bar{X}^*_{an}$ and $\bar{V}^*_{an}$ are the previous particle position and velocity, respectively, w_v is the weighting factor on particle velocity, c_1 and c_2 are acceleration constants, r_1 and r_2 are random variables of uniform distribution over $[0, 1]$. Its position is updated as

$$\bar{X}_{an} \leftarrow \bar{X}^*_{an} + \bar{V}_{an} \tag{13}$$

The take-off time of fighters and launch time of ASHMs from fleet A are then set to

$$\begin{aligned}&\text{take-off time of the first fighter of fleet A in the } n\text{th particle pair} \leftarrow \delta_{anf1}\\&\text{take-off time of the } m\text{th fighter of fleet A in the } n\text{th particle pair}\\&\leftarrow \text{(take-off time of the } (m-1)\text{th fighter of fleet A in the } n\text{th particle pair)} + t_{gf} + \delta_{anfm}\end{aligned} \tag{14}$$

$$\begin{aligned}&\text{launch time of the first ASHM of fleet A in the } n\text{th particle pair} \leftarrow \delta_{ang1}\\&\text{launch time of the } \ell\text{th ASHM of fleet A in the } n\text{th particle pair}\\&\leftarrow \text{(launch time of the } (\ell-1)\text{th ASHM of fleet A in the } n\text{th particle pair)} + t_{gm} + \delta_{ang\ell}\end{aligned} \tag{15}$$

Similarly, the velocity of the nth particle for fleet B is updated as

$$\bar{V}_{bn} \leftarrow w_v \bar{V}^*_{bn} + c_3 r_3 (\bar{P}_{bn} - \bar{X}^*_{bn}) + c_4 r_4 (\bar{G}_b - \bar{X}^*_{bn}) \tag{16}$$

where c_3 and c_4 are acceleration constants, r_3 and r_4 are random variables of uniform distribution over $[0,1]$. Its position is updated as

$$\bar{X}_{bn} \leftarrow \bar{X}^*_{bn} + \bar{V}_{bn} \tag{17}$$

The take-off time of fighters and launch time of ASHMs from fleet B are then set to

$$\begin{aligned}
&\text{take-off time of the 1st fighter of fleet B in the } n\text{th particle pair} \leftarrow \delta_{bnf1} \\
&\text{take-off time of the } m\text{th fighter of fleet B in the } n\text{th particle pair} \\
&\leftarrow (\text{take-off time of the } (m-1)\text{th fighter of fleet B in the } n\text{th particle pair}) + t_{gf} + \delta_{bnfm}
\end{aligned} \tag{18}$$

$$\begin{aligned}
&\text{launch time of the 1st ASHM of fleet B in the } n\text{th particle pair} \leftarrow \delta_{bng1} \\
&\text{launch time of the } \ell\text{th ASHM of fleet B in the } n\text{th particle pair} \\
&\leftarrow (\text{launch time of the } (\ell-1)\text{th ASHM of fleet B in the } n\text{th particle pair}) + t_{gm} + \delta_{bng\ell}
\end{aligned} \tag{19}$$

Then, a game is played with the updated particle positions $\bar{X}_{an}$ and $\bar{X}_{bn}$. Both fleets operate according to the rules described in the last section. At the end of the game, each fleet counts its objective function defined in (1), (2) or (3) to update $\bar{P}_{an}$, $\bar{G}_a$, $\bar{P}_{bn}$ and $\bar{G}_b$. The procedure is repeated for all the particle-pairs to complete one iteration. The P^{2}SO algorithm is completed after a specified number of iterations. The procedure of the P^{2}SO algorithm is listed in Algorithm 1, and is described below.

Input: Weighting factor of velocity w_v, acceleration constants $\{c_1, c_2, c_3, c_4\}$, maximum iteration number $I_{\max}$, particle-pair number N_p and game parameters in Tables 4–8.

Output: $\bar{G}_a$, $\bar{G}_b$.

step 1: Randomly initialize the position and velocity of the nth particle-pair.

step 2: Use the nth particle-pair to compute take-off time of fighters and launch time of ASHMs for both fleets, and play a game. Then update ($\bar{P}_{an}$, $\bar{P}_{bn}$) and ($\bar{G}_a$, $\bar{G}_b$).

step 3: Repeat step 1 and step 2 for all the N_p particle-pairs.

step 4: The velocity and position for fleet A in the nth particle-pair are updated with (12) and (13), respectively. The take-off time of fighters and launch time of ASHMs from fleet A are updated with (14) and (15), respectively.

step 5: The velocity and position for fleet B in the nth particle-pair are updated with (16) and (17), respectively. The take-off time of fighters and launch time of ASHMs from fleet B are updated with (18) and (19), respectively.

step 6: Play a game with take-off time of fighters and launch time of ASHMs from fleet A in step 5, take-off time of fighters and launch time of ASHMs from fleet B in step 6, as well as θ_{anf}, θ_{ang}, θ_{bnf}, and θ_{bng} .

step 7: Use the resulting objective functions of fleets A and B in step 7 to update ($\bar{P}_{an}$, $\bar{P}_{bn}$) and ($\bar{G}_a$, $\bar{G}_b$).

step 8: Repeat step 4 to step 7 for all the N_p particle-pairs.

step 9: Increment the iteration index. If the index is smaller than $I_{\max}$, go to step 4. Otherwise, the algorithm is completed.

Algorithm 1 Pseudocode of P^{2}SO algorithm

```
Initialize: Particles best objective functions $J_{pa} = 0, J_{pb} = 0$
            Global best objective functions $J_{ga} = 0, J_{gb} = 0$
Input: Game parameters (Table 4–8) and P$^2$SO algorithm parameters (Table 9)
Output: $\bar{G}_a$, $\bar{G}_b$

  for $n$ = 1:$N_p$ do
1.    Randomly initialize $\bar{X}_{an}, \bar{X}_{bn}, \bar{V}_{an}, \bar{V}_{bn}$
2.    Use $\bar{X}_{an}, \bar{X}_{bn}$ to compute take-off time of fighters and launch time of ASHMs for fleet
      A and B, respectively, with (14), (15), (18), (19).
      Play a game with the resulting take-off time of fighters, launch time of ASHMs, $\theta_{anf}$
      and $\theta_{ang}$ in $\bar{X}_{an}$, $\theta_{bnf}$ and $\theta_{bng}$ in $\bar{X}_{bn}$ and game parameters to obtain objective
      functions $J_a$ and $J_b$.
      $J_{pan} \leftarrow J_a, J_{pbn} \leftarrow J_b$
      $\bar{P}_{an} \leftarrow \bar{X}_{an}, \bar{P}_{bn} \leftarrow \bar{X}_{bn}$
      if $J_a > J_{ga}$
         $J_{ga} \leftarrow J_a$ ; $\bar{G}_a \leftarrow \bar{X}_{an}$
      end if
      if $J_b > J_{gb}$
         $J_{gb} \leftarrow J_b$ ; $\bar{G}_b \leftarrow \bar{X}_{bn}$
      end if
3. end for
  for $i$ = 1:$I_{\max}$ do
      for $n$ = 1:$N_p$ do
4.       $\bar{V}_{an}$ and $\bar{X}_{an}$ are updated with (12) and (13), respectively
         Take-off time of fighters and launch time of ASHMs of fleet A are updated with
         (14) and (15)
5.       $\bar{V}_{bn}$ and $\bar{X}_{bn}$ are updated with (16) and (17), respectively
         Take-off time of fighters and launch time of ASHMs of fleet B are updated with
         (18) and (19)
6.       Play a game with the updated take-off time of fighters, launch time of ASHMs,
         $\theta_{anf}, \theta_{ang}, \theta_{bnf}, \theta_{bng}$ and game parameters to obtain objective functions $J_a$ and $J_b$
7.       if $J_a > J_{pan}$
            $J_{pan} \leftarrow J_a$ ; $\bar{P}_{an} \leftarrow \bar{X}_{an}$
            if $J_a > J_{ga}$
               $J_{ga} \leftarrow J_a$ ; $\bar{G}_a \leftarrow \bar{X}_{an}$
            end if
         end if
         if $J_b > J_{pbn}$
            $J_{pbn} \leftarrow J_b$ ; $\bar{P}_{bn} \leftarrow \bar{X}_{bn}$
            if $J_b > J_{gb}$
               $J_{gb} \leftarrow J_b$ ; $\bar{G}_b \leftarrow \bar{X}_{bn}$
            end if
         end if
8.    end for
9. end for
```

Table 9 lists the P^{2}SO parameters used in the simulations. The P^{2}SO algorithm concurrently updates the optimal tactical parameters of both fleets. The global best position of one fleet is determined from its objective functions over all games, and both fleets usually receive their global best positions from different particle-pairs. The global best position of one fleet is unknown to the other fleet before a new game, but the outcome of the game may affect both global best positions, which in turn will affect the next game. The simulation results with the optimal tactical parameters of P^{2}SO will be presented in the next section.

The parameters listed in Tables 4–8 are roughly estimated with available data collected from public sources, such as Wikipedia. The parameters listed in Table 9 are tried out and fine-tuned over many runs of simulations, based on the convergence performance, computational time, and the optimization outcomes. For example, increasing the number of particle pairs (N_p) and the maximum iteration number ($I_{\max}$) of the P^{2}SO algorithm may generate tactical parameters that deliver better performance, but the computational load is also increased. The speed, flying distance, alert radius, kill probability and maneuverability of agents are properly chosen to design a matched game without dominant factors or overwhelming agents in order to observe subtle nuances that may affect the outcome of a game. The simulation time step (Δt) is adjusted to ensure the games are played smoothly with reasonable CPU time. The choice of minimum time gap between missile launches (t_{gm}) is also constrained by the simulation time step.

Table 9. Parameters of P^{2}SO algorithm.

Parameter	Symbol	Value
weight on particle velocity	w_v	1
acceleration constants	c_1, c_2, c_3, c_4	2
number of particle pairs	N_p	40
maximum iteration number	$I_{\max}$	60

5. Simulations on Contest between Options

The simulations are run on a PC with i7-3.0 GHz CPU and 32 GB memory, in MATLAB R2019a code. The game is staged as in Figure 1. Each fleet is composed of one CV, one CG, and two DDGs. The two CVs are deployed at the north-east and south-west corners, respectively, separated by 400 km and facing each other. The DDGs are 10 km beside the CV and the CG is 10 km behind the CV. The P^{2}SO algorithm is applied to optimize the tactical parameters, including take-off time delay of fighters, launch time delay of ASHMs and initial flying angles of fighters and ASHMs, for both fleets according to their chosen goal option among (1)–(3). The optimized tactical parameters are then used to play 100 or 200 games. In each game, each fleet dispatches its fighters and ASHMs against the enemy assets, and defends itself against incoming enemy weaponry with available missiles, following the game rules presented in Section 3. Each game takes calendar time of 3600 s. At the end of a game, the payoff of each fleet is computed with (4). Finally, the outcomes of these games are analyzed statistically.

In short, the P^{2}SO algorithm in Section 4 is applied to optimize the objective function specified in Section 3.1 via playing multiple games, following the rules specified in Sections 3.3–3.7. The optimal tactical operations thus obtained are used to play another set of games, and the payoff specified in Section 3.2 is computed at the end of each game for statistica analysis.

With three goal options available to each fleet, there are six possible contest scenarios between the two fleets. In this section, each contest scenario will be simulated and analyzed statistically. General observations will be summarized in Section 5 and a few interesting outlier cases will be investigated in Section 6 to gain more understanding of the tactical operations.

5.1. Option 2 for Fleet A versus Option 3 for Fleet B

Figure 2 shows the convergence of P^{2}SO objective functions, with option 2 for fleet A versus option 3 for fleet B, which is abbreviated as A2/B3. The objective function of fleet A converges to 11, which is higher than 6.4 of fleet B because fleet A targets solely on the enemy CV, which has the highest value among all vessels.

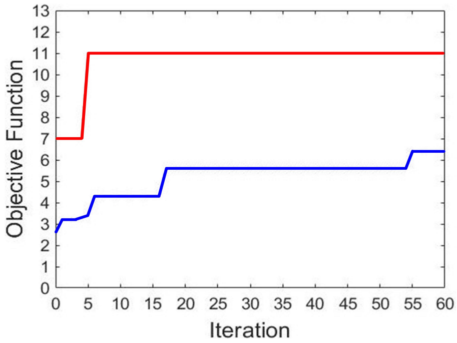


Figure 2. Convergence of P^{2}SO objective functions, option 2 for fleet A versus option 3 for fleet B (A2/B3), ———: fleet A, ———: fleet B.

It is observed that the objective function first converges to certain level and remains unchanged for a few iterations before jumping to a higher level. There exists uncertainty in the value of objective functions because the outcomes a game, which are used to update the particle pair, manifest fluctuations caused by the imperfect kill probability of missiles, and CIWS.

The proposed P^{2}SO algorithm is aimed to update the optimal solution by using the outcome of the latest game, hence it does not converge to a steady optimal solution and stay therein. On the other hand, conventional multi-objective PSO algorithms [61–63] converge to a steady optimal solution theoretically, even constrained by multiple objective or fitness functions. With the proposed P^{2}SO algorithm, each fleet in the game can always update its optimal tactical parameters by incorporating the outcomes of the latest game, which is more compatible to the real-life scenarios.

Figure 3a shows the cumulative distribution (CDF) of payoff over 100 games with one P^{2}SO run, namely, the global best positions in the final iteration are used to simulate 100 separate games. The curve of fleet B shows more obvious steps because fleet A targets only CV-B, and each impact on CV-B contributes a significant payoff value to fleet B. On the other hand, the curve of fleet A is smoother than that of fleet B because all four vessels of fleet A are targeted, resulting in more possible levels of payoff value. The payoff of fleet B is higher than that of fleet A because all the missiles from fleet A target solely on CV-B. The lowest possible payoffs of fleets A and B are USD 1 B and USD 1.6 B, respectively. The median payoffs of fleets A and B are USD 1.9 B and USD 3.9 B, respectively. The payoff of fleet A is always lower than USD 6.2 B, is below USD 3 B in 80 games and is lower than the median payoff (USD 3.9 B) of fleet B in 96 games.

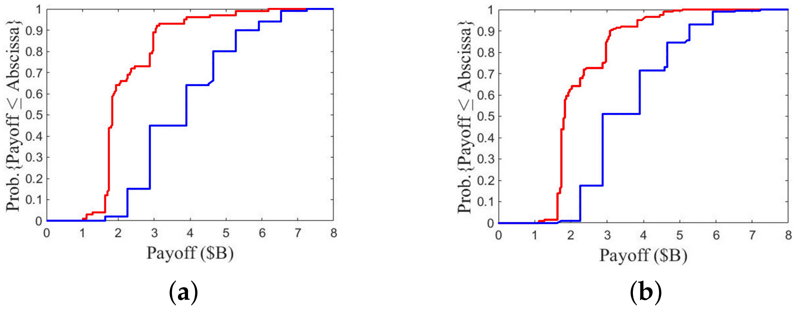


Figure 3. (A2/B3) Cumulative distribution of payoff with one P^{2}SO run, (**a**) over 100 games, (**b**) over 200 games; ———: fleet A, ———: fleet B.

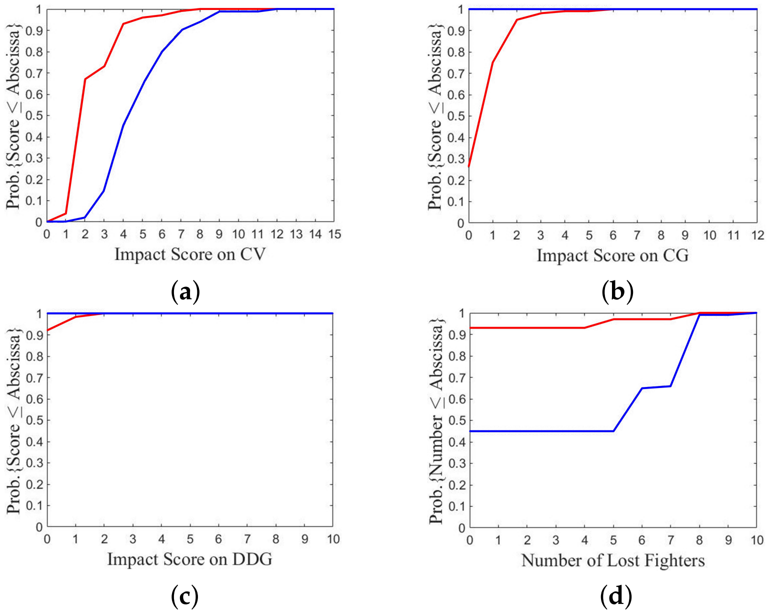
Figure 3b shows the CDF of payoff over 200 games with one P^{2}SO run, namely, the same set of parameters are used to simulate 200 separate games. The curves look similar to those in Figure 3a, with most payoff values lying between USD 1.5 B and USD 5 B. The difference between two fleets is slightly smaller, especially at payoff between USD 2.2 B and USD 5.2 B.

Figure 4a shows the CDF of impact score on CV over 100 games with one P^{2}SO run. It is observed that the impact score on CV-B is higher than that on CV-A in most games because all the ASMs and ASHMs from fleet A are used to attack the CV-B while those from fleet B are dispersed to attack all the vessels of fleet A. The lowest and highest scores on CV-B are 2 and 12, respectively. The lowest and highest scores on CV-A are 1 and 8, respectively, lower than their counterparts on CV-B. The score on CV-A is lower than 3 in 67 games, while that on CV-B is lower than 3 in only 2 games and higher than 5 in 37 games.

Figure 4b shows the CDF of impact score on CG over 100 games. The score on CG-B is always 0 because it is not targeted by fleet A. The lowest and highest scores on CG-A are 0 and 6, respectively. The score on CG-A is 0 in 26 games and 1 in 49 games possibly because it is deployed behind the other three vessels.

Figure 4c shows the CDF of impact score on DDG over 100 games, encompassing 200 scores on each fleet. The score on DDG-B is always 0 since they are not targeted by fleet A. The scores on DDG-A are 0 in 184 games, 1 in 14 games and 2 in 2 games, which are relatively low because fighters and ASHMs from fleet B are dispersed to attack all four vessels of fleet A. Figure 4d shows the CDF of lost fighters over 100 games. Fewer fighters of fleet A are lost compared to fleet B in most games. Fleet A loses no fighters in 93 games. Fleet B loses more than 5 fighters in 54 games and loses all 10 fighters in 1 game.

Figure 5a shows the CDF of impact score on CV over 200 games with one P^{2}SO run. The curves are similar to those in Figure 4a, the difference between two curves become smaller, and the percentage of games with score between 4 and 6 slightly increases.



(**a**) (**b**) (**c**) (**d**)

Figure 4. (A2/B3) Cumulative distribution of impact score on (**a**) CV (**b**) CG, (**c**) DDG, (**d**) fighters; over 100 games with one P^{2}SO run, ———: fleet A, ———: fleet B.

Figure 5b shows the CDF of impact score on CG over 200 games. The curve of fleet A is similar to that in Figure 4b, with score 0 in 60 games and 1 in 110 games. The highest score is 5, as compared with the highest score of 6 in Figure 4b. The difference can be accounted for by the statistical consequence of kill probability.

Figure 5c shows the CDF of impact score on DDG over 200 games, encompassing 400 scores on DDGs of each fleet. The curve of fleet A is similar to that in Figure 4c, the highest score becomes 3 and the percentage of score 0 slightly decreases. Figure 5d shows the CDF of lost fighters over 200 games. The curves are similar to those in Figure 4d, with fleet A losing at most 5 fighters.

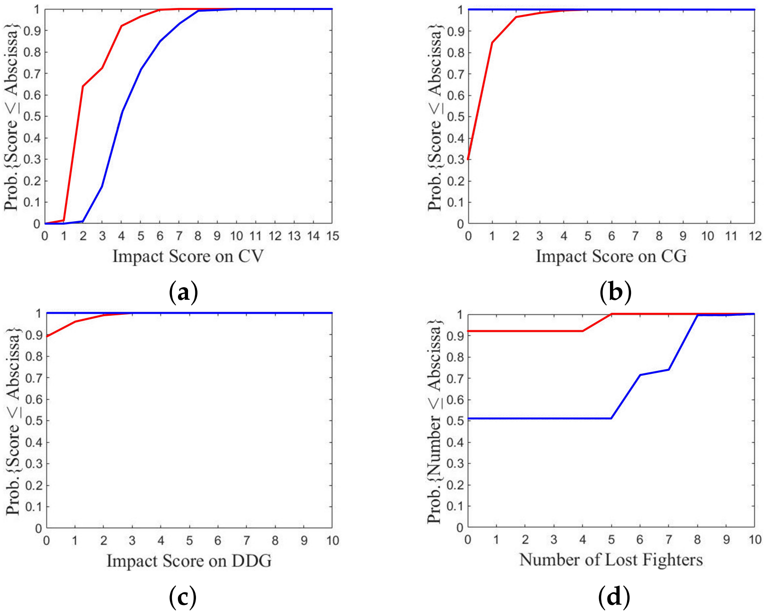


Figure 5. (A2/B3) Cumulative distribution of impact score on (**a**) CV (**b**) CG, (**c**) DDG, (**d**) fighters; over 200 games with one P^2SO run, ——: fleet A, ——: fleet B.

Figure 6a shows the CDF of payoff over 200 games with two P^2SO runs, where the parameters from each run are used to simulate 100 separate games. The curves are similar to their counterparts in Figures 3a,b, respectively, implying the payoff distributions obtained with both P^2SO runs are similar. The curves become smoother because more samples of payoff value are counted in. The payoff difference between the two curves in Figure 6a is smaller than that in Figures 3a,b, especially between percentiles 45% and 75%.

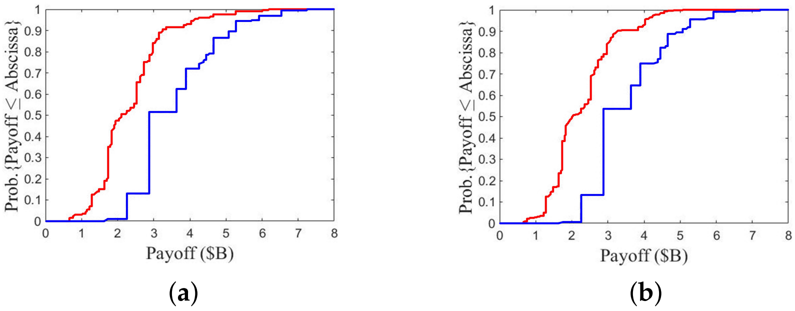
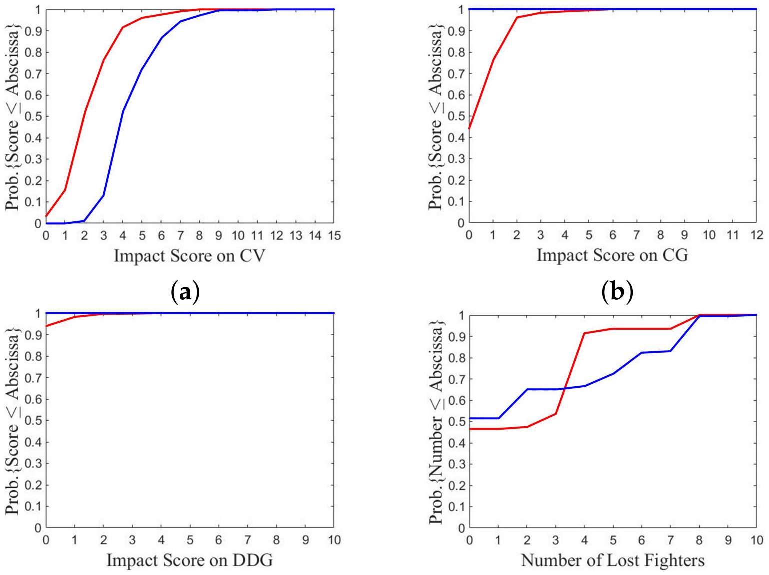


Figure 6. (A2/B3) Cumulative distribution of payoff with two P^2SO runs, (**a**) over 200 games, (**b**) over 400 games; ——: fleet A, ——: fleet B.

Figure 6b shows the CDF of payoff over 400 games with two P^2SO runs, where the parameters from one run are used to simulate 200 separate games. The curves are similar to their counterparts in Figure 6a.

By comparing Figure 3 and 6, it is observed that the payoffs from both P^2SO runs reveal similar distribution, although the global best positions of the two P^2SO runs may not be close.

Figure 7a shows the CDF of impact score on CV over 200 games with two P^2SO runs. The curves are similar but smoother than their counterparts in Figure 5a. The lowest score on CV-A is 0.



(**a**) (**b**)

(**c**) (**d**)

Figure 7. (A2/B3) Cumulative distribution of impact score on (**a**) CV (**b**) CG, (**c**) DDG, (**d**) fighters; over 200 games with two P^{2}SO runs, ——: fleet A, ——: fleet B.

Figure 7b shows the CDF of impact score on CG over 200 games. The curve of fleet A is similar to its counterparts in Figures 4b and 5b. The impact score on CG-A is 0 in 90 games, but the percentage is higher than those in Figures 4b and 5b.

Figure 7c shows the CDF of impact score on DDG over 200 games. The curve of fleet A is similar to its counterparts in Figures 5c and 4c. The score on DDG-A is 3 in Figure 5c, which is slightly higher than that in Figure 7c.

Figure 7d shows the CDF of lost fighters over 200 games. The curves are obviously different from those in Figure 5d. Different take-off time and flying angle of fighters from both fleets may result in different scenarios of missile chasing fighter. Note that the number of lost fighters is not considered in the objective function for optimizing the tactical parameters. Thus, different P^{2}SO runs may lead to different numbers of lost fighters. Fleet A loses no fighters in 90 games, 4 fighters in 80 games, and up to 8 fighters in 16 games, which is higher than 5 in Figure 5d. Fleet B loses fewer fighters as compared to Figure 5d.

Figure 8a shows the CDF of impact score on CV over 400 games with two P^{2}SO runs. The curves are similar to those in Figure 7a, except the probability of high impact score slightly decreases.

Figure 8b shows the CDF of impact score on CG over 400 games. The curve of fleet A is similar to its counterparts in Figures 7b and 5b. However, the curve appears smoother since more samples are counted in. The score on CG-A is 0 in 184 games, at higher percentage than those in Figures 7b and 5b.

Figure 8c shows the CDF of score on DDG over 400 games, encompassing 800 scores for each fleet. The curve of fleet A is similar to that in Figure 7c. The score on DDG-A is 0 in 736 incidents and 380 incidents in Figure 7c.

Figure 8d shows the CDF of lost fighters over 400 games. The curves are similar to that in Figure 7d.

In this contest scenario, the simulation results with one versus two P^{2}SO runs and with 100 games versus 200 games per run are compared. In general, the distributions of payoff and impact score are insensitive to these two numbers. Thus, only one P^{2}SO run and 100 games per run will be simulated in the other five contest scenarios.

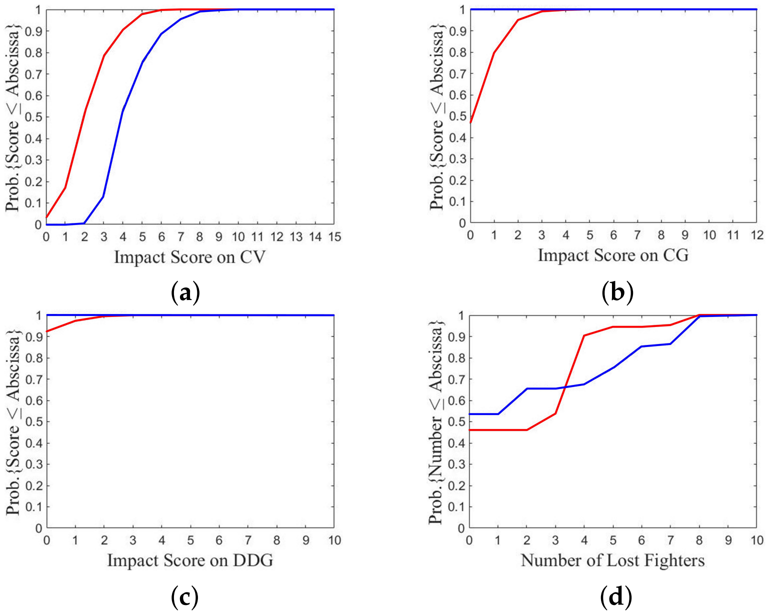


Figure 8. (A2/B3) Cumulative distribution of impact score on (**a**) CV (**b**) CG, (**c**) DDG, (**d**) fighters; over 400 games with two P^{2}SO runs, ——: fleet A, ——: fleet B.

5.2. Option 1 for Fleet A versus Option 1 for Fleet B

In the A1/B1 contest scenario, the P^{2}SO objective functions of both fleets are close to each other and quickly converge with iteration to the maximum value of 1. Figure 9a shows the CDF of payoff over 100 games with one P^{2}SO run. The two curves are close to each other since both fleets take the same option. The payoff value is relatively low because no vessels are targeted. The minimum payoff of both fleets is USD 0.24B. Fleet A takes USD 0.38 B payoff in 74 games, its maximum payoff is USD 0.45 B, slightly lower than USD 0.58 B of fleet B.

Figure 9b shows the CDF of lost fighters over 100 games. The two curves are close to each other. Both fleets lose at least 3 fighters and at most 6 or 8 fighters, respectively. Fleet B loses more fighters in most games. Fleet A loses 5 fighters in 74 games. Note that more fighters are lost in A1/B1 scenario than in A2/B3 scenario, because each fighter is marked by at most two AAMs when the opponent chooses option 1 and marked by one AAM otherwise.

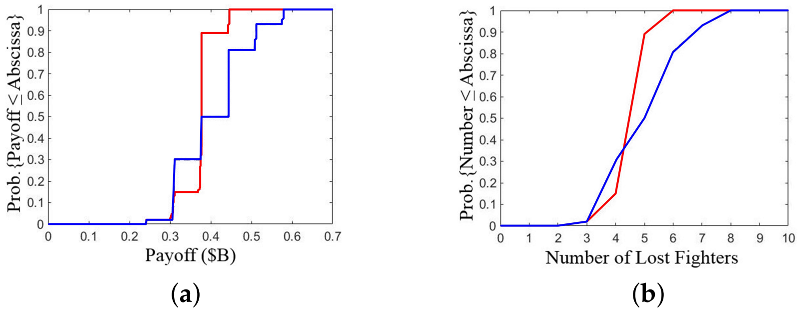
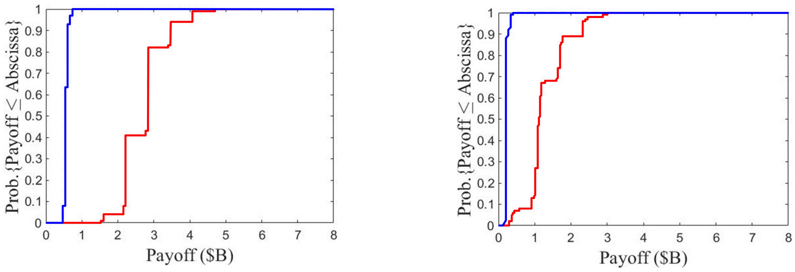


Figure 9. (A1/B1) Cumulative distribution of (**a**) payoff, (**b**) lost fighters, with one P^{2}SO run over 100 games, ——: fleet A, ——: fleet B.

5.3. Option 1 for Fleet A versus Option 2 for Fleet B

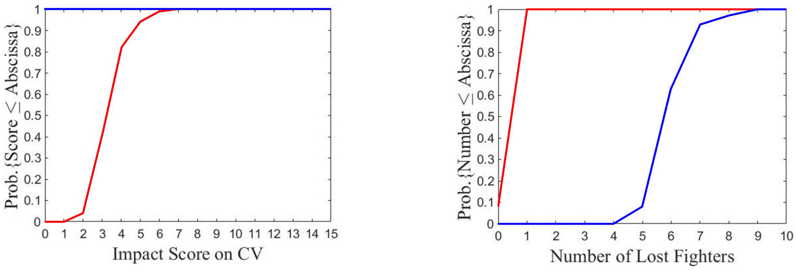
Figure 10a shows the CDF of payoff over 100 games with one P^{2}SO run. The payoff of fleet B is relatively low, ranging from USD 0.4 B to USD 0.7 B, since no vessels are targeted by fleet A, and the payoff is attributed to the lost fighters, launched ASHMs, ASMs, and AAMs. Fleet A takes payoff from USD 1.5 B to USD 4.7 B, takes USD 2.2 B in 36 games and USD 2.8 B in 40 games, attributed to the score on CV-A.

Figure 11a shows the CDF of impact score on CV over 100 games with one P^{2}SO run. The score on CV-B is always 0 since it is not targeted. The score on CV-A ranges from 2 to 7, is 3 or 4 in 77 games.



(**a**) (**b**)

Figure 10. Cumulative distribution of payoff, (**a**) A1/B2, (**b**) A1/B3, with one P^{2}SO run over 100 games, ——: fleet A, ——: fleet B.



(**a**) (**b**)

Figure 11. (A1/B2) Cumulative distribution of (**a**) impact score on CV and (**b**) lost fighters, over 100 games with one P^{2}SO run, ——: fleet A, ——: fleet B.

Figure 11b shows the CDF of lost fighters over 100 games. Fleet A loses only one fighter in 91 games and fleet B loses 5 to 9 fighters. Fleet B loses many more fighters in all the games because fleet A takes option 1 and its fighters can launch more AAMs to intercept the enemy fighters.

5.4. Option 1 for Fleet A versus Option 3 for Fleet B

Figure 10b shows the CDF of payoff over 100 games with one P^{2}SO run. The payoff of fleet B varies from USD 0.2 B to USD 0.4 B because its vessels are not targeted. Fleet A takes payoff from USD 0.3 B to USD 3 B, and takes about USD 1 B in 59 games.

Figure 12a shows the CDF of impact score on CV over 100 games with one P^{2}SO run. It is observed that CV-B is never targeted. The score on CV-A is 0 in 8 games, 1 in 61 games, and the highest score is 4.

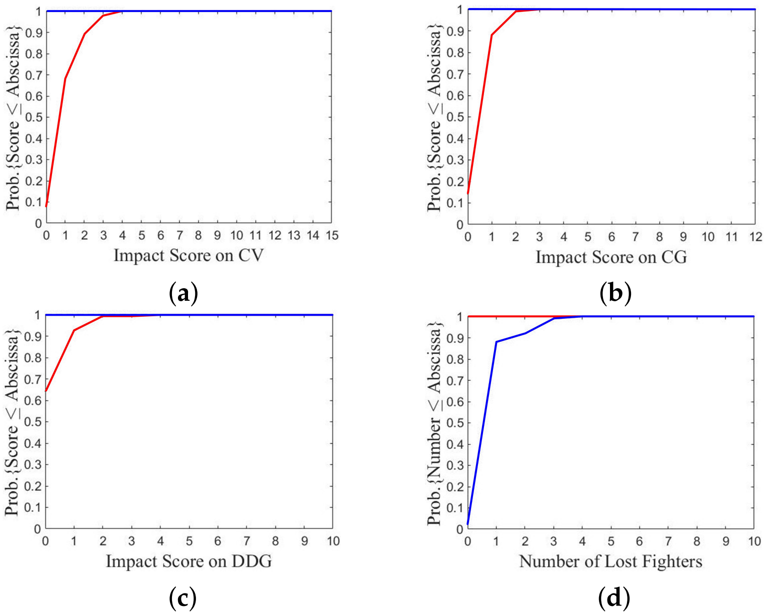


Figure 12. (A1/B3) Cumulative distribution of impact score on (**a**) CV (**b**) CG, (**c**) DDG, (**d**) fighters; over 100 games with one P^{2}SO run, ——: fleet A, ——: fleet B.

Figure 12b shows the CDF of impact score on CG over 100 games. The score on CG-A is 0 in 15 games, 1 in 74 games, and the highest score is 3. The score on CG-B is 0 because it is not targeted.

Figure 12c shows the CDF of impact score on DDG over 100 games. The score on DDG-B is always 0 since they are not targeted. The score on DDG-A is 0 in 130 incidents and 1 in 54 incidents.

Figure 12d shows the CDF of lost fighters over 100 games. It is observed that all the fighters of fleet A are intact. In contrast, fleet B loses 1 fighter in 86 games and loses 4 fighters in 1 game since more AAMs are fired upon them.

5.5. Option 3 for Fleet A versus Option 3 for Fleet B

Figure 13a shows the CDF of payoff over 100 games with one P^{2}SO run. Both fleets take similar payoff in most games because they choose the same option. Fleet A take payoff from USD 1.1 B to USD 5.5 B, and takes around USD 1.2 B in 36 games. Fleet B takes payoff from USD 1 B to USD 6.4 B, and takes around USD 1.8 B in 36 games.

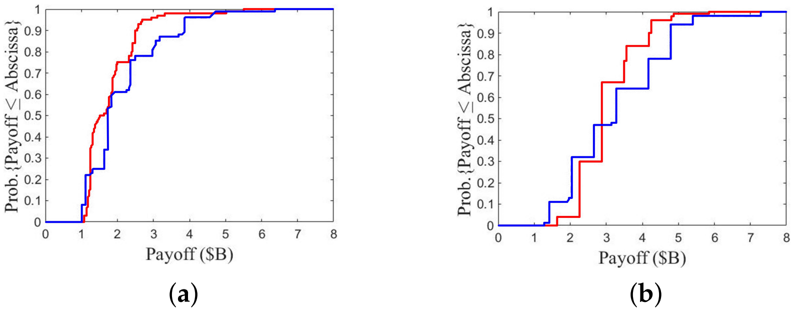


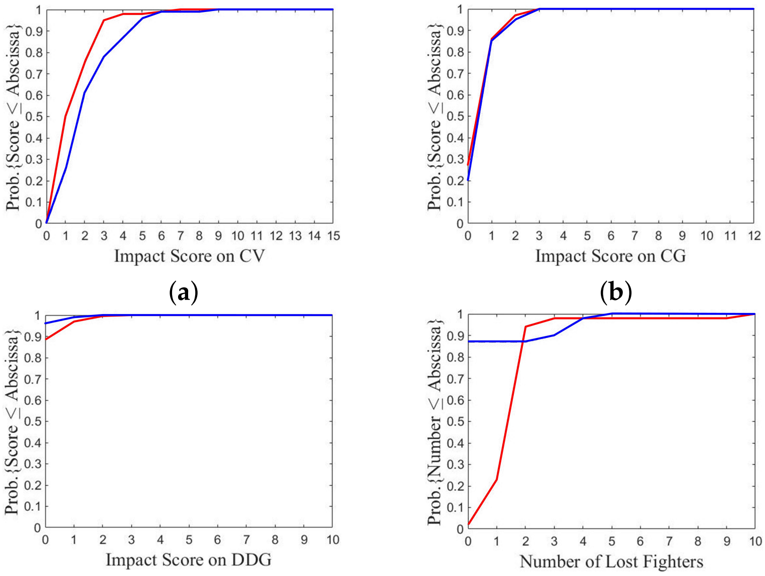
Figure 13. Cumulative distribution of payoff, (**a**) A3/B3, (**b**) A2/B2, with one P^{2}SO run over 100 games, ——: fleet A, ——: fleet B.

Figure 14a shows the CDFs of impact score on CV-A and CV-B are close to each other. The highest scores on CV-A and CV-B are 7 and 9, respectively, and both have the lowest score of 1. The score on CV-A is lower than 4 in 95 games and that on CV-B is lower than 4 in 78 games.

Figure 14b shows the CDFs of score on CG-A and CG-B are similar. Both fleets take the highest score of 3 and take score of 1 in most games. CG-A takes no hit in 28 games and CG-B takes no hit in 20 games.

Figure 14c shows the CDFs of score on DDG-A and DDG-B are similar, and both take the highest score of 3. DDG-A takes no hit in 178 incidents and DDG-B takes no hit in 192 incidents.

Figure 14d shows the CDF of lost fighters over 100 games. It is observed that fleet A loses more fighters in most games. Fleet A loses 0 fighter in 2 games, 2 fighters in 71 games, and all the 10 fighters in 4 games. In contrast, fleet B loses 0 fighter in 88 games and up to 5 fighters in 4 games. It can be attributed to the optimal tactical parameters of both fleets, which will be investigated further in Section 6.



(**a**) (**b**)

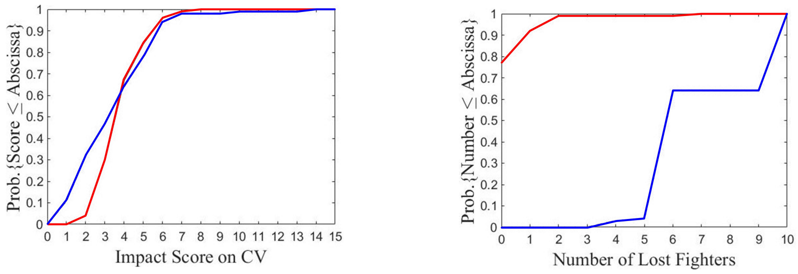
(**c**) (**d**)

Figure 14. (A3/B3) Cumulative distribution of impact score on (**a**) CV (**b**) CG, (c) DDG, (**d**) fighters; over 100 games with one P^{2}SO run, ———: fleet A, ———: fleet B.

5.6. Option 2 for Fleet A versus Option 2 for Fleet B

Figure 13b shows that both fleets have similar CDF of payoff since they choose the same option. Both curves show a few big steps which are attributed to the score on CV. Fleet A takes payoff from USD 1.6 B to USD 5.8 B, and takes USD 2.9 B in 36 games. Fleet B takes payoff from USD 1.3 B to USD 7.3 B, and the distribution of payoff is uniform.

Figure 15a shows the CDFs of impact score on CV are close to each other, especially at higher scores. The score on CV-A ranges from 2 to 8, and has a uniform distribution between 2 and 4 in 64 games. The score on CV-B ranges from 1 to 14, and has a uniform distribution between 1 and 6 in 82 games.



(**a**) (**b**)

Figure 15. (A2/B2) Cumulative distribution of (**a**) impact score on CV and (**b**) lost fighters, over 100 games with one P^{2}SO run, —: fleet A, —: fleet B.

Figure 15b shows the CDF of lost fighters. It is observed that fleet B loses more fighters in most games. Fleet A loses 0 fighter in 78 games, 1 fighter in 13 games, 2 fighters in 8 games, and 7 fighters in 1 game. Fleet B loses 4 to 10 fighters, 6 fighters in 60 games, and 10 fighters in 36 games.

5.7. Comparison of Median Payoff

Table 10 lists the median payoffs on both fleets, summarized from the contests between options in Sections 5.1–5.6. Since both fleets have equal size and equal capability, their payoffs are swapped if both fleets exchange options with each other, with slight difference due to the combined effects of too many factors involved in the game simulations. It is observed that the difference of payoffs between both fleets is smaller when they choose the same option than different ones. The fleet choosing option 1 suffers higher payoff than its opponent, and the fleet choosing option 2 suffers lower payoff than its opponent. Table 10 can serve as a quick hint to decision-makers, but the CDF is indispensable in making grave decision.

Table 10. Median payoffs of option contest (USD B).

A/B	Option 1	Option 2	Option 3
Option 1	0.378/0.409	2.836/0.536	1.115/0.202
Option 2	0.536/2.836	2.874/3.275	1.832/3.895
Option 3	0.202/1.115	3.895/1.832	1.570/1.732

6. Summary and Comparison

In the last section, it is observed that the highest payoff is inflicted upon a fleet if the opponent fleet chooses option 2 to solely target the most valuable CV. If option 1 is chosen, the own payoff is slightly reduced and more enemy fighters will be lost. If both fleets choose the same option, their payoffs are generally in the same level. If option 3 is chosen, the impact score on DDGs is low because fewer fighters and ASHMs are dispatched to attack them.

In general, different P^{2}SO runs deliver similar trend of distributions in payoff and impact score. However, some optimal initial flying angle makes the evasion from enemy missiles more difficult while some makes it easier. This will lead to some interesting results, which will be investigated in the next section.

At first, compare the results between A2/B3 and A1/B2, as shown in Figure 16. One of the two fleets chooses option 2, while the other chooses either defense option 1 or aggressive option 3. Figure 16a,b show the CDF of payoff over 100 games with one P^{2}SO run. The payoff of fleet A in Figure 16b is lower than that of fleet B in Figure 16a. The lowest, median, and highest payoffs of fleet A in Figure 16b are USD 1.5 B, USD 2.8 B, and USD 4.7 B, respectively, while the lowest, median, and highest payoffs of fleet B in Figure 16a are USD 1.6 B, USD 3.9 B, and USD 7.2 B, respectively. It suggests that choosing a defense option inflicts lower payoff than choosing a more aggressive one.

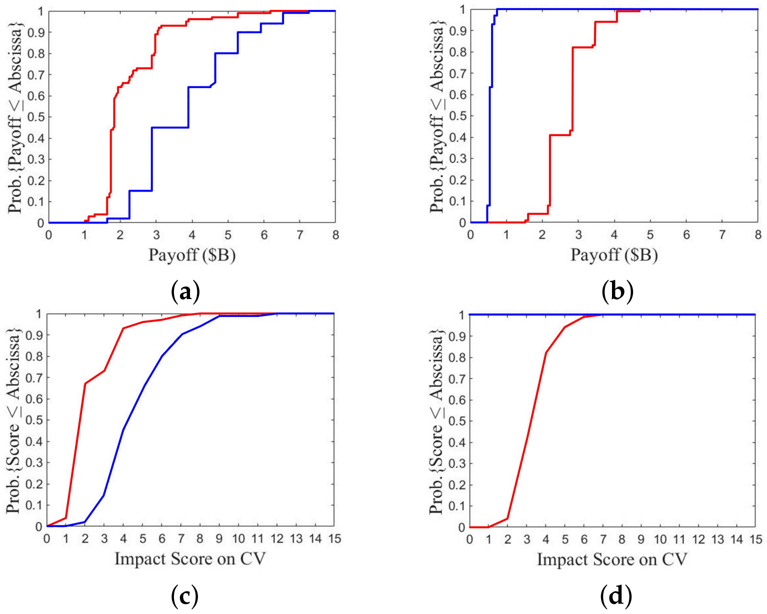


Figure 16. Comparison between A2/B3 and A1/B2 over 100 games with one P^{2}SO run. Cumulative distribution of payoff, (**a**) A2/B3, (**b**) A1/B2. Cumulative distribution of impact score on CV, (**c**) A2/B3, (**d**) A1/B2. ———: fleet A, ———: fleet B.

Figure 16c,d show the CDF of impact score on CV. The curve of fleet A in Figure 16d is steeper than that of fleet B in Figure 16c, which suggests that choosing a defense option takes lower impact score on the own CV.

Figure 17 shows the comparison between A1/B3 and A2/B3, where fleet B chooses option 3 while fleet A chooses a defense option (A1) and a more aggressive option (A2), respectively. The payoff of fleet A in Figure 17a is significantly lower than that in Figure 17b. Figure 17a shows that the lowest, median, and highest payoffs of fleet A are USD 0.3 B, USD 1.1 B, and USD 3 B, respectively, and the payoff is lower than or equal to USD 1 B in 13 games. Figure 17b shows that the lowest, median and highest payoffs of fleet A are USD 1 B, USD 1.9 B, and USD 6.2 B, respectively, and the payoff is higher than or equal to USD 1 B in all games. Fleet A takes payoff lower than USD 2 B in 89 games if option 1 is chosen and in 63 games if option 2 is chosen. It suggests that choosing option 1 inflicts lower payoff than choosing a more aggressive one.

Figure 17c,d show the CDF of impact score on CV. The score on CV-A in Figure 17c is slightly lower than that in Figure 17d. The score on CV-A is lower than 3 in 90 games if option 1 is chosen and in 68 games if option 2 is chosen.

Figure 17e,f show the CDF of impact score on CG. The highest score on CG-A is 3 in Figure 17e and 6 in Figure 17f. The score on CG-A is lower than 3 in 99 games if option 1 is chosen and 96 games if option 2 is chosen.

The simulations are run with MATLAB R2019a on a PC with i7-3.0 GHz CPU and 32 GB memory. It takes about 13–16 CPU hours to play a typical game with one fleet on aggressive option and the other fleet on aggressive or non-aggressive option, and about 8–10 CPU hours with both fleets on non-aggressive option. The CPU time with aggressive option versus non-aggressive option is not as low as expected, even though ASHMs are not launched from the non-aggressive fleet and almost no SAMs are fired from the aggressive fleet, because the fighters from the non-aggressive fleet fire many more AAMs, requiring longer CPU time to track the aftermath.

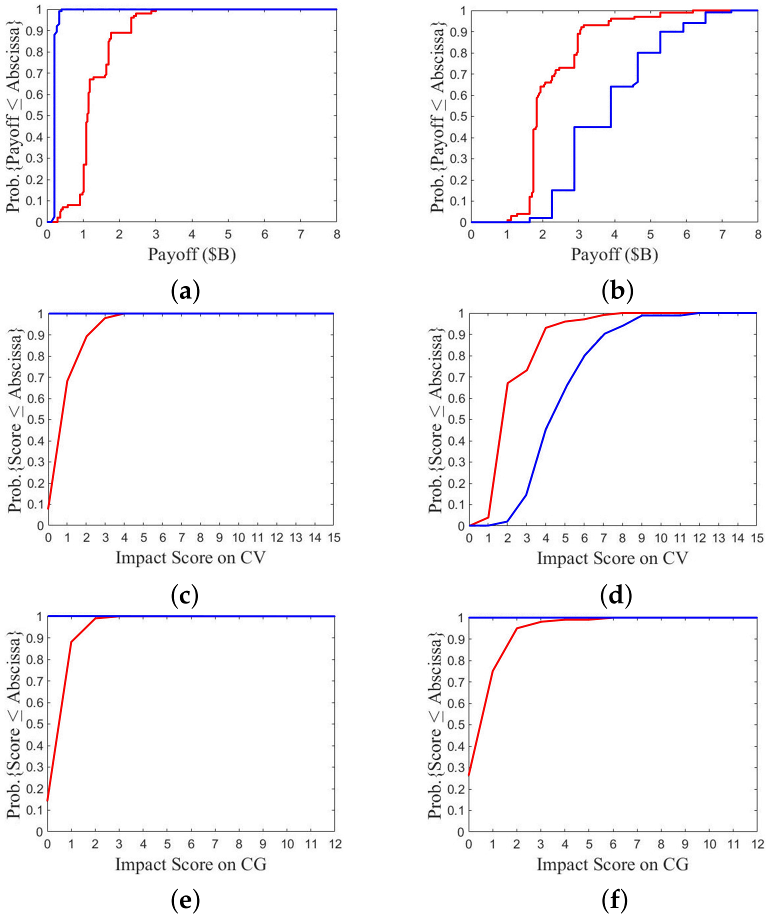


Figure 17. Comparison between A1/B3 and A2/B3 over 100 games with one P^{2}SO run. CDF of payoff, (**a**) A1/B3, (**b**) A2/B3. CDF of impact score on CV, (**c**) A1/B3, (**d**) A2/B3. CDF of impact score on CG, (**e**) A1/B3, (**f**) A2/B3. ——: fleet A, ——: fleet B.

7. Investigation on Interesting Outlier Cases

7.1. A1/B1

Figure 18 shows the outcome of a specific P^{2}SO run on A1/B1 over 100 games. Figure 18a shows that both objective functions have similar magnitude and converge to the maximum value of 1 quickly. Figure 18b shows the payoff of fleet B is much higher than that of fleet A in most games, which is beyond expectation under A1/B1 scenario. Fleet A takes payoff from USD 0.03 B to USD 0.51 B, and USD 0.18 B in 48 games. Fleet B takes payoff from USD 0.24 B to USD 0.65 B, and USD 0.51 B or USD 0.57 B in 70 games. Figure 18c shows fleet B loses more fighters than fleet A in most games. Fleet B loses 4 to 9 fighters, with 7 or 8 lost in 70 games, while fleet A loses 0 to 7 fighters, with 2 lost in 50 games.

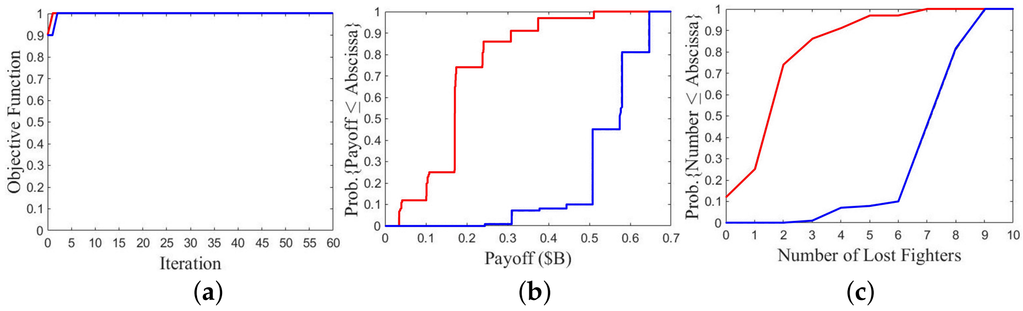


Figure 18. Specific P^{2}SO run on A1/B1 over 100 games, (**a**) convergence of objective functions, (**b**) CDF of payoff, (**c**) CDF of lost fighters; ——: fleet A, ——: fleet B.

The cause can be traced back to the global best positions of this specific P^{2}SO run that the initial flying angle of fighters is 22.315° from fleet A and 62.569° from fleet B. With these initial flying angles, most AAMs from fleet A close in on their targeted fighters more quickly and more difficult to evade. On the other hand, most AAMs from fleet B take longer time to reach their targeted fighters, easier to burn out of fuel before hitting the targets.

Figure 19 demonstrates a specific game on A1/B1, with an AAM from fleet A intercepting a fighter from fleet B. The symbols defined in the caption apply to all the cases elaborated in this section. Figure 19a shows that at $t = 960$ s, a fighter from fleet A detects an enemy fighter and launches an AAM to intercept the latter. Figure 19b shows that at $t = 1005$ s, the marked fighter detects the AAM and begins to evade, while the AAM closes in on the marked fighter at high speed. Figure 19c shows that at $t = 1050$ s, the AAM flies close to the marked fighter. Figure 19d shows that at $t = 1080$ s, the AAM successfully hits the target, 120 s after launching.

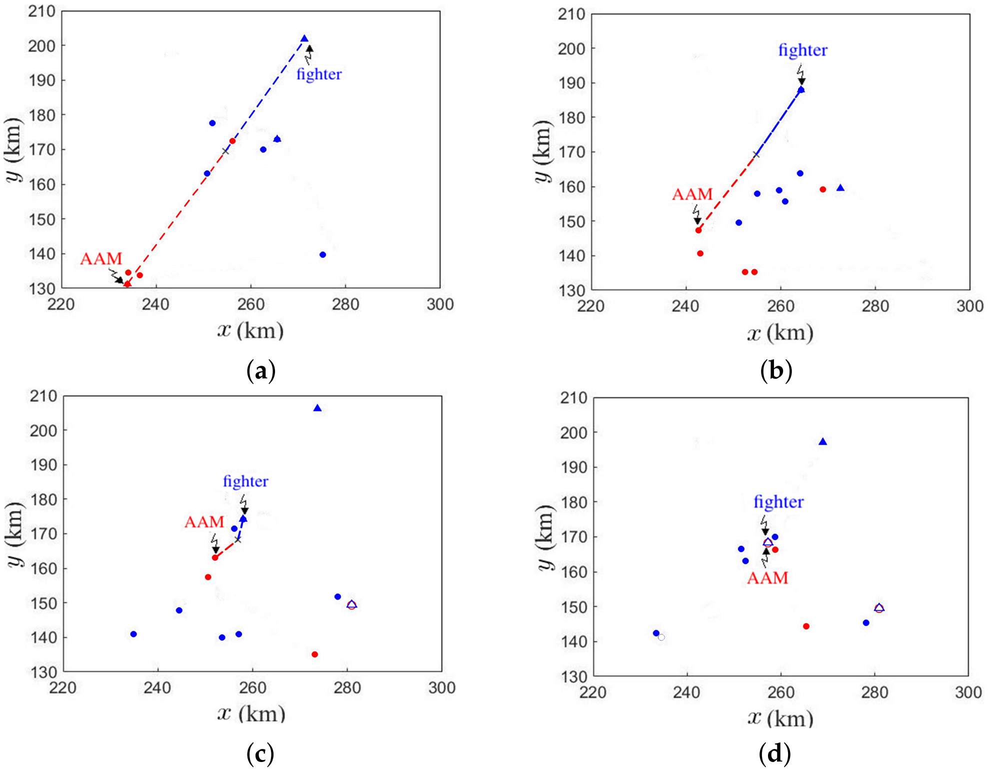
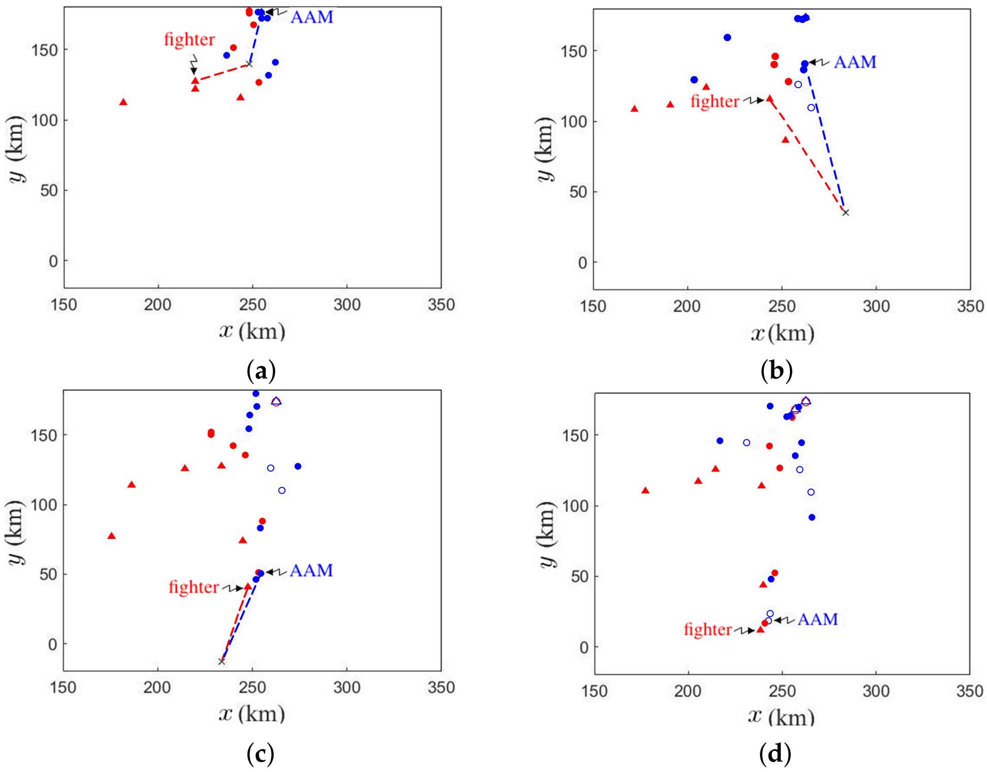


Figure 19. Specific game on A1/B1, an AAM from fleet A is intercepting a fighter from fleet B, (**a**) $t = 960$ s, (**b**) $t = 1005$ s, (**c**) $t = 1050$ s, (**d**) $t = 1080$ s; red marks for fleet A and blue marks for fleet B, blank symbols: dead assets, △: fighter, •: AAM, ×: intercept point, − − −: projected flight path.

Figure 20 demonstrates a specific game with an AAM from fleet B intercepting a fighter from fleet A. Figure 20a shows that at $t = 675$ s, a fighter from fleet B detects an enemy fighter and launches an AAM against it. The relative speed between the AAM and the target fighter is lower than that if they fly head on. Figure 20b shows that at $t = 765$ s, the fighter changes its cause to evade while the AAM flies towards a farther intercept point than that in Figure 20a. Figure 20c shows that at $t = 990$ s, even though the AAM is very close to the marked fighter, but the relative speed is too low to hit the latter. Figure 20d shows that at $t = 1080$ s, the AAM burns out fuel and the fighter survives.

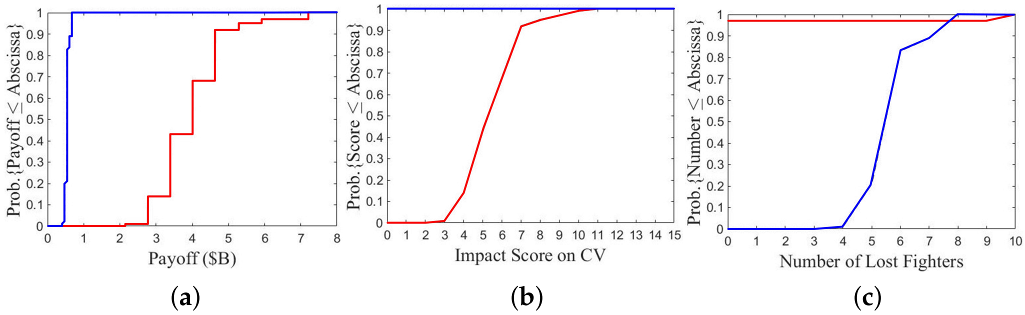


(**a**) (**b**) (**c**) (**d**)

Figure 20. Specific game on A1/B1, an AAM from fleet B is intercepting a fighter from fleet A, (**a**) $t = 675$ s, (**b**) $t = 765$ s, (**c**) $t = 990$ s, (**d**) $t = 1080$ s.

7.2. A1/B2

The P^{2}SO objective function of fleet A quickly converges to 1 as option 1 is chosen. The objective function of fleet B converges to a high value of 11 since CV-A is targeted. Figure 21 shows the outcome of a specific P^{2}SO run on A1/B2 over 100 games. Figure 21a shows that fleet B takes payoff from USD 0.4B to USD 0.7B, while fleet A takes payoff from USD 2.2 B to USD 7.2 B, takes higher than USD 3.4 B in 57 games, much higher than anticipated under option 1. Figure 21b shows that the impact score on CV-A ranges from 3 to 11, and the score is 5 to 7 in 78 games. Figure 21c shows that fleet A loses no fighter in 97 games, while fleet B loses 4 to 8 fighters, and 6 fighters in 61 games.



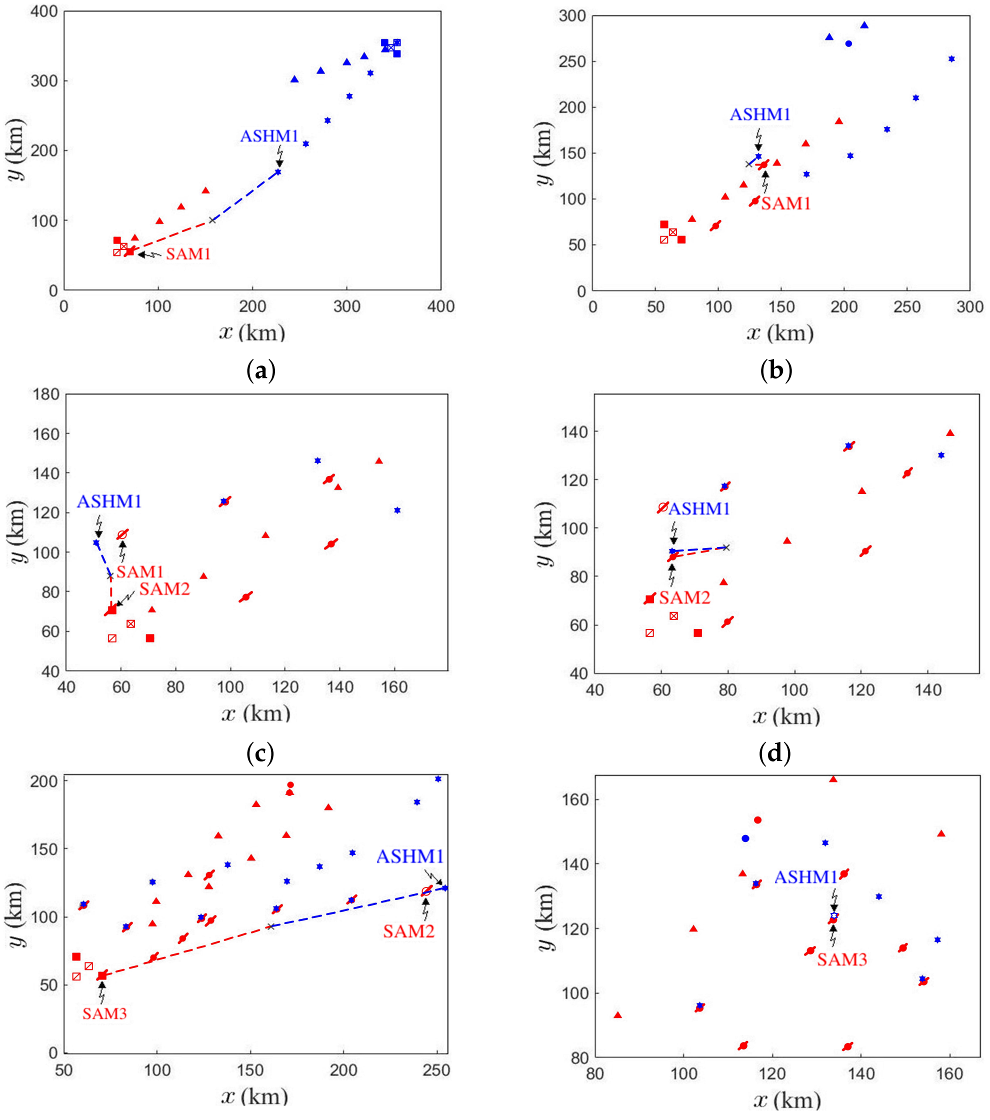
(**a**) (**b**) (**c**)

Figure 21. Specific P^{2}SO run on A1/B2 over 100 games, cumulative distribution of (**a**) payoff; (**b**) impact score on CV, (**c**) lost fighter; ——: fleet A, ——: fleet B.

A closer examination reveals that the optimal flying angle of ASHMs from CG-B make them difficult to intercept with SAMs from fleet A. The first SAM fired to intercept an approaching ASHM often burns out of fuel in the pursuit, and another SAM has to be fired to intercept the same ASHM. With the optimal flying angles of fighters from both fleets, the AAMs fired by fighters from fleet A chase away the fighters from fleet B, but later either burn out of fuel or are hit by AAM from the marked fighter.

The fighters from fleet A chase the fighters from fleet B far away from the alert radius of DDG-A. Meanwhile, DDG-A quickly uses up all SAMs against the approaching ASHMs. Later, the fighters from fleet B fly towards CV-A after evading the AAMs from fleet A, and each fighter launches two ASMs when it reaches the attacking range. At this stage, all SAMs are used up and CV-B becomes a sitting duck, with CIWS as its last gate keeper. With proper timing, one of ASMs penetrates the CIWS and hits CV-A.

Figure 22 shows the flight path of an ASHM from CG-B. Figure 22a shows that at $t = 390$ s, one DDG-A fires the first SAM to intercept an approaching ASHM. Figure 22b shows that at $t = 570$ s, the first SAM is still chasing the ASHM. Figure 22c shows that at $t = 705$ s, the first SAM burns out of fuel, and the DDG-A fires a second SAM to intercept the same ASHM. Figure 22d shows that at $t = 735$ s, the second SAM is still chasing the ASHM. Figure 22e shows that at $t = 1020$ s, the second SAM also burns out of fuel, and the DDG-A fires a third SAM to intercept the ASHM. Figure 22f shows that at $t = 1260$ s, the ASHM burns out of fuel. By firing three SAMs to intercept one ASHM, all SAMs will be exhausted quickly.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 22. Specific game on A1/B2, an ASHM from CG-B evading SAMs from fleet A, (**a**) $t = 390$ s, (**b**) $t = 570$ s, (**c**) $t = 705$ s, (**d**) $t = 735$ s, (**e**) $t = 1020$ s, (**f**) $t = 1260$ s. Symbols are referred to Table 3.

Figure 23 shows the scenario of fighters from fleet B attacking CV-A after DDG-A uses up all SAMs. The circle in Figure 23a shows that at $t = 1020$ s, four fighters from fleet B are chased away to the upper left region by AAMs fired by fighters from fleet A. Meanwhile, several SAMs are fired upon ASHMs from CG-B. The circle in Figure 23b shows that at $t = 1200$ s, the AAMs fail to intercept the enemy fighters because burning out of fuel or hit by the marked fighters. The circle in Figure 23c shows that at $t = 1890$ s, three fighters return from the upper left region, firing six ASMs to attack CV-A. DDG-A now has no spare SAMs to intercept these ASMs. Figure 23d shows that at $t = 2,145$ s, CV-A is hit by ASMs.

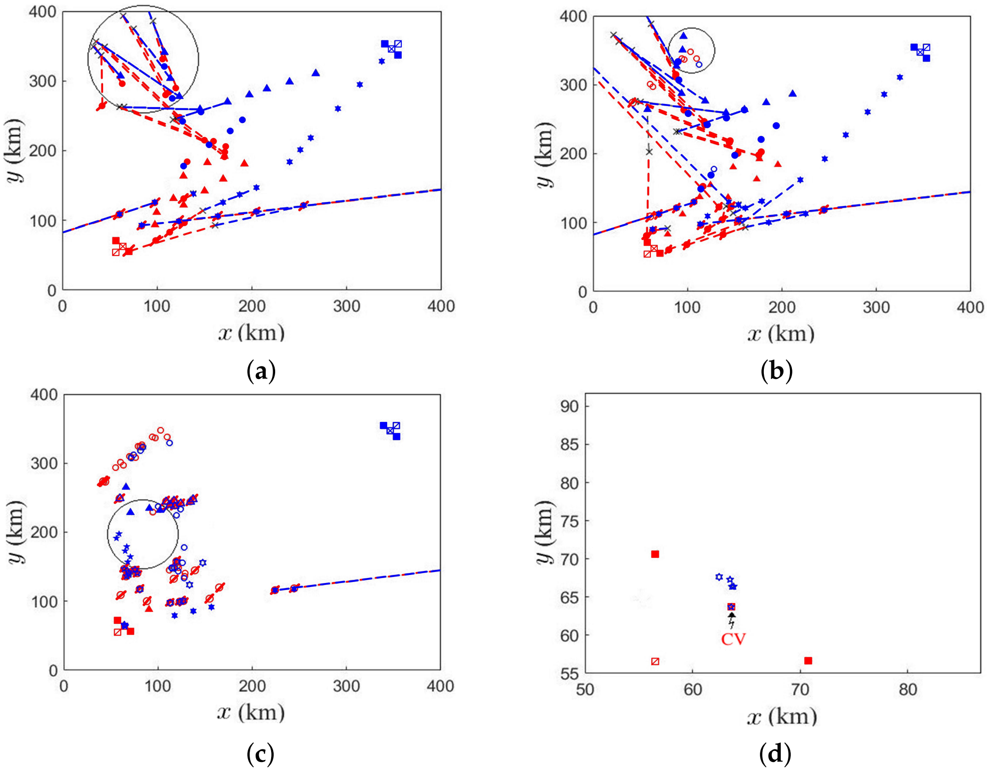


Figure 23. Specific game on A1/B2, fighters from fleet B attacking CV-A, (**a**) $t = 1020$ s, (**b**) $t = 1200$ s, (**c**) $t = 1890$ s, (**d**) $t = 2145$ s. Symbols are referred to Table 3.

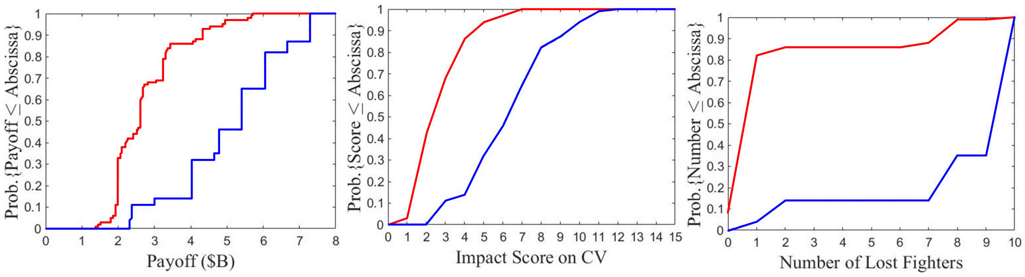
7.3. A2/B3

The P^{2}SO objective functions of fleets A and B quickly converge to 10 and 7, respectively. The objective function of fleet A is higher because it focuses the attack on CV-B. Figure 24 shows the results of a specific P^{2}SO run. Figure 24a shows that the payoff of fleet A looks normal, but that of fleet B is higher than expectation. The lowest, median, and highest payoffs of fleet B are USD 2.3 B, USD 5.4 B, and USD 7.3 B, respectively, and the payoff is higher than USD 4B in 87 games. Figure 24b shows the impact score on CV-B ranges from 3 to 12, and is higher than 6 in 56 games. Figure 24c shows that fleet A loses only one fighter in 82 games.

When the ASHMs and fighters from fleet A fly near DDG-B1, SAMs are fired to intercept the fighters. When DDG-B1 runs out of SAMs, the ASHMs from CG-A lure the SAMs fired from DDG-B2 to burn out of fuel in chasing them. In the later stage of the game, a few ASMs from fleet A survive the SAMs or no SAMs are spared for them, the remaining ASMs and ASHMs on the fly can simultaneously attack CV-B.

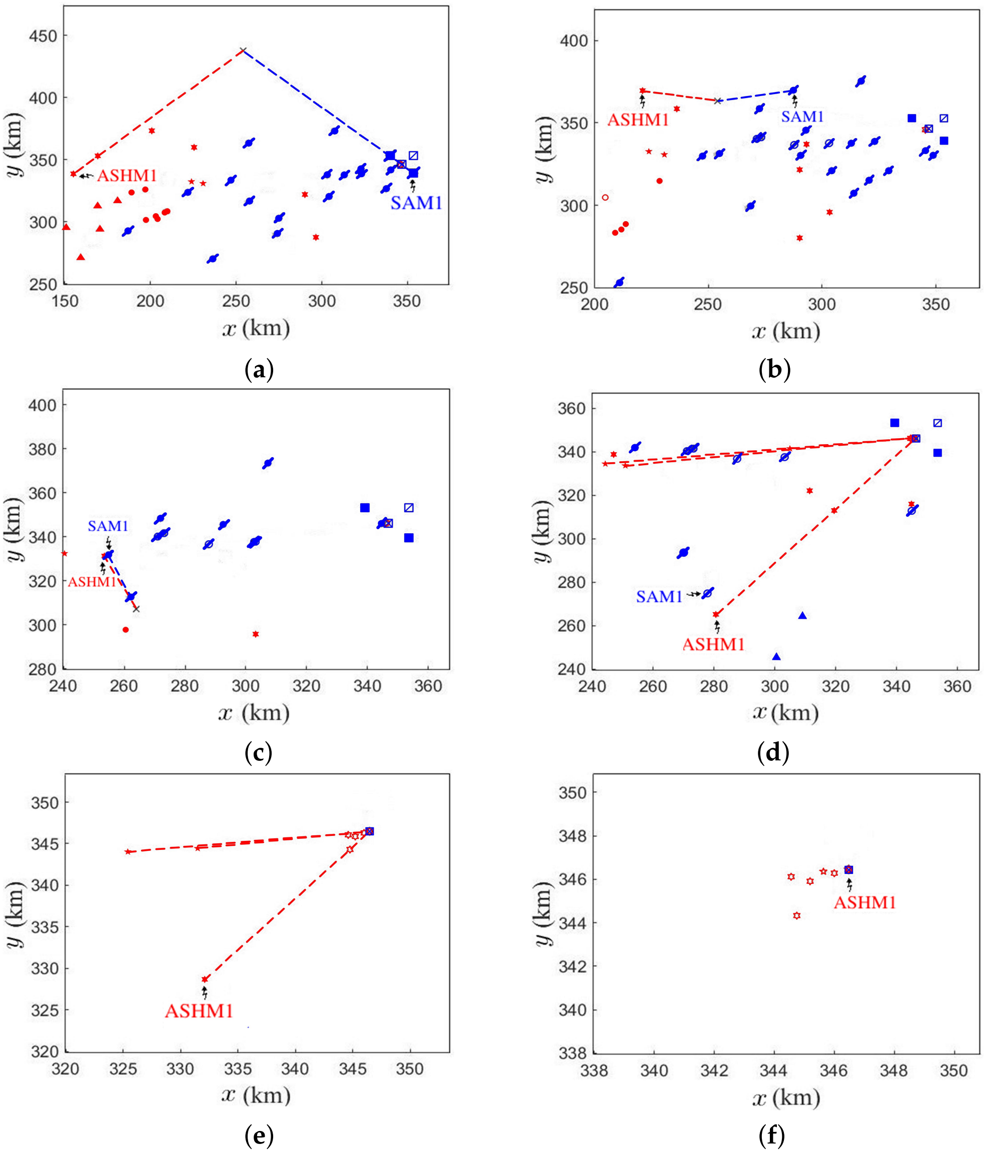
Figure 25 shows the move of an ASHM from CG-A when DDG-B1 runs out of SAMs. Figure 25a shows that at $t = 1320$ s, a SAM is fired by DDG-B2 to intercept an ASHM from CG-A. Figure 25b shows that at $t = 1440$ s, the SAM moves closer to the ASHM. Figure 25c shows that at $t = 1530$ s, the SAM moves to the proximity of ASHM, but the intercept point is far away. Figure 25d shows that at $t = 1635$ s, the SAM burns out of fuel. No SAM is available to intercept the ASHM, and the latter turns to fly straight towards

CV-B. Figure 25e shows that at $t = 1755$ s, the ASHM closes in on CV-B and coordinates simultaneous attack with two near-by ASMs. Figure 25f shows that at $t = 1800$ s, CV-B is hit by one ASM and the ASHM.



(**a**) (**b**) (**c**)

Figure 24. Specific P^{2}SO run on A2/B3 over 100 games, cumulative distribution of (**a**) payoff, impact score on (**b**) CV, (**c**) lost fighters; ——: fleet A, ——: fleet B.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 25. Specific game on A2/B3, an ASHM from CG-A evading a SAM from DDG-B and later attacking CV-B, (**a**) $t = 1320$ s, (**b**) $t = 1440$ s, (**c**) $t = 1530$ s, (**d**) $t = 1635$ s, (**e**) $t = 1755$ s, (**f**) $t = 1800$ s. Symbols are referred to Table 3.

7.4. A3/B3 with 100% Kill Probability

The distributions of payoff, impact scores on different vessels and number of lost fighters are affected by the optimal tactical parameters, as well as the kill probability of missiles and CIWS. To highlight the effects of optimal tactical parameters, we set the kill probability of missile and CIWS to 100% in an A3/B3 contest scenario. It is observed that the two objective functions are close to each other and have the same value over more than 30 iterations. The objective function of fleet B increases after the 58th iteration when the its global best position is updated according to the previous game results. The objective functions can be further increased as the iteration moves on.

By using the results of a specific P^{2}SO run to play 100 games, with the kill probability of missiles and CIWS set to 100%, it is observed that the payoff and impact scores of both fleets remain the same game after game. For fleet A, the payoff is USD 1.86 B, the impact scores on CV, CG, DDG and fighter are 2, 1, 0, and 2, respectively. For fleet B, the payoff is USD 0.66 B, the impact scores on CV, CG, DDG and fighter are 0, 2, 0, and 1, respectively. In all the previous cases where the kill probability is less than 100%, the distributions of payoff and impact scores of each fleet spread over certain interval, revealing uncertainty of missile or CIWS in hitting its target. When the kill probability is set to 100%, each hit by missile or CIWS on a target within its impact radius or firing range will succeed. Thus, the outcome of a game is solely determined by the tactical parameters optimized with the P^{2}SO algorithm.

A large payoff difference between two fleets is mainly attributed to the impact score on CV-A by ASMs. CV-A is hit by two enemy ASMs while CV-B remains intact. The optimal initial flying angles of fighters from fleets A and B are 78.603° and 71.614°, respectively. In the early stage of the game, all the ASHMs from both fleets are marked and intercepted by enemy SAMs. In the later stage, only ASMs are used to attack the enemy CV while all the enemy SAMs are used up, leaving CIWS to defend the vessels. If the two ASMs fired by a fighter are about equally distant from the target vessel, the latter can be hit. However, the initial flying angle of fighters from fleet A does not meet the condition for two ASMs from a fighter to simultaneously reach the target vessel, the two ASMs are sequentially hit by the CIWS. In contrast, the initial flying angle of fighters from fleet B renders two ASMs fired by a fighter to simultaneously reach the target vessel, leaving one of the two ASMs to penetrate the CIWS and hit CV-A.

A scenario of two ASMs from the same fighter of fleet B attacking CV-A is closely examined. At $t = 1830$ s, two ASMs at a separation of 6.15 km close in on CV-A. The CIWS hits ASM2 at $t = 1845$ s, but is not ready to engage ASM1. Then, ASM1 hits CV-A at $t = 1860$ s.

Another scenario of two ASMs from a fighter of fleet A attacking CV-B is also closely examined. At $t = 1935$ s, two ASMs at a separation of 7.11 km close in on CV-B. ASM2 is hit by the CIWS of CV-B at $t = 1950$ s. At $t = 1965$ s, ASM1 is also intercepted by CIWS before hitting CV-B. In this case, the separation between two ASMs is wide enough for the CIWS to sequentially engage them.

7.5. Lessons Learned

By inspecting the specific game on A1/B1, if an incoming interceptor or missile moves in opposite direction to that of its target, the former is likely to hit the latter. On the contrary, if an incoming interceptor or missile moves far from opposite to the moving direction of its target, the latter is likely to evade alive.

From the specific game on A1/B2, if an agent turns out to attract many enemy interceptors, the defense capability of the enemy will be weakened in the later stage of the game and become vulnerable to the follow-up attacking agents.

In the specific game on A2/B3, attacking agents can be coordinated effectively to have one group luring and exhausting enemy SAMs from DDGs, followed by another group to close in on the target.

The specific game on A3/B3 with 100% kill probability demonstrates the possibility of firing two ASMs from the same fighter to break through the defense of enemy CIWS, by adjusting the flying angle of fighter for the two ASMs to reach the target simultaneously.

8. Retrospective Discussion

In this work, we try to design a matched game, without dominant factors or overwhelming agents, in order to observe subtle nuances that may affect the outcome of a game. Thus, we make straightforward assumptions that two fleets have equal size and capability, that is, both fleets are the same in number of vessels and weapons, fleet formation, rules of engagement, cost of vessels and weapons, kill probability, optional goals, and objective functions.

The speed of ships is approximated as zero since they move much slower than fighters and missiles, and the fighters and missiles are assumed to move at constant speed in 2D space. The costs of assets are the construction cost or replacement cost.

The weaknesses of this work are directly related to the aforementioned assumptions, which can be remedied by designing the game and algorithms for two fleets with different sizes and capabilities, and extending the game scenario to 3D space. However, this will increase the complexity of game designs and the difficulty of analysis on game outcomes. This work takes a trade-off between complex reality and simple principles of operation, aiming to gain some lessons and insights through the war games.

We have found no comparable game designs or approaches in the literature. A major reason is that our approach integrates many types of episodes in a versatile game scenario. The game rules in each type of episode are clearly specified and each type of episode can find compatible works in the literature, such as evading and pursuing. However, a complete game composed of different types of episode is rare in the literature, ruling out possible performance comparison with existing literature. Instead, we focus on the game results and their implications, which are essential to the purpose of conventional war games.

We do not use analytical methods, such as game-theoretic method and Lanchester's equations, to derive the solution for comparison, because there are too many factors speculated in the game scenario. The interaction among a large number of agents and the uncertainty aroused by imperfect kill probability create numerous unexpected outcomes, as presented in Sections 5–7.

Machine learning methods are versatile, but they require tremendously long training time due to so many factors just mentioned. With all these considerations, we propose a heuristic-oriented variant of P^{2}SO algorithm to search for the optimal tactical parameters under given goal options from both fleets. The optimization process requires the outcome of tactical operation after playing a game between both fleets, which adds another dimension of complexity to the game design.

We do not know how the commanding officer of a fleet will act or react on site. They may avoid head-on clash by taking alert poise (close to option 1), or focus on the major target of CV while reducing loss to own fleet (close to option 2), or take an all-out assault at all costs (close to option 3). In this work, three options are devised for contest, and the outcomes of games are presented in terms of payoffs on both fleets, in statistical forms. The simulation results and their statistics should be useful to the commanding officer or decision-makers.

This work tries to encompass as many real-life factors as possible, including prototypical episodes of chase between pursuer(s) and evader(s), interception, simultaneous attack, target assignment, path planning, defense, CIWS as last line of defense, and so on. The strategy or goal-setting of each fleet is boiled down to three goal options, defensive, taking major targets at affordable cost, and all-out at all cost. The tactical parameters required to implement each option, given opponent's option, are optimized by using the proposed P^{2}SO algorithm, which is an extension of conventional PSO algorithm, with affordable computer resources.

An important war game between two fleets has been designed, focusing on the contest between three strategical options and the associated tactical optimization. Many useful prototypical episodes reported in the literature are seamlessly knitted into this work. Since intangible factors beyond these two fleets themselves are capricious, a fair play will be to presume both fleets have equal size and equal capability, and the values of these assets are their tangible construction cost. Assuming all other conditions the same, we can compare the efficacy of these three goal options against one another, as well as the tactical optimization tailored to implement the chosen goal option. The payoffs of both fleets, based on their construction cost, are used to assess the risk of taking one option against the other.

An attempt to design a closer to real-life war game may run the risk of including too many real-life factors or constraints, such as intangible value of fleet or its constituents. The simulation outcomes and their interpretations may become speculative and ambiguous, gaining some insights but losing the clear picture.

9. Conclusions

A war game has been designed and simulated between two matched fleets of equal size and capability. Each fleet is composed of one CV, one CG, and two DDGs, launching fighters and different kinds of missiles to attack the opponent. Three goal options are considered by each fleet, and the tactical parameters are optimized with the proposed P^{2}SO algorithm, including take-off time delay of fighters, launch time delay of ASHMs, initial flying angles of fighters and ASHMs, respectively. Six contests between goal options are simulated and analyzed statistically. The simulation results show that if a fleet chooses defense option 1, its own payoff is lower than if choosing more aggressive option, and the opponent usually loses more fighters. If a fleet chooses option 2, it can inflict the highest payoff to its opponent since all the weapons are targeted on the enemy CV. Some interesting outlier cases with unexpected payoff have also been inspected. It is demonstrated that the initial flying angle may facilitate or impede the evasion of fighters or ASHMs from enemy missiles, inflicting unexpected payoffs to both fleets.

In this work, we propose a method to optimize the tactical operation based on a designate goal or strategic option, which was not mentioned in the literature. The P^{2}SO algorithm is proposed to find two optimal tactical plans for two opponent fleets, respectively, with their objective functions updated according to the outcomes of previous games played between them. Such an interactive update approach was not found in the literature. In order to imitate actual combat scenarios more closely, one may implement variable speed or other practical factors to the agents in the future studies. One may also develop other optimization algorithms to acquire better tactical plans.

Author Contributions: Conceptualization, J.-F.K.; methodology, J.-F.K.; software, Z.-X.J.; validation, Z.-X.J. and J.-F.K.; formal analysis, Z.-X.J. and J.-F.K.; investigation, Z.-X.J. and J.-F.K.; resources, J.-F.K.; data curation, Z.-X.J.; writing—original draft preparation, Z.-X.J.; writing—review and editing, J.-F.K.; visualization, Z.-X.J. and J.-F.K.; supervision, J.-F.K.; project administration, J.-F.K.; funding acquisition, J.-F.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding and the article processing charges (APC) was funded by Ministry of Science and Technology, Taiwan, under contract MOST 109-2221-E-002-169.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. War Gaming Department. *War Gamers' Handbook: A Guide for Professional War Gamers*; War Gaming Department, U.S. Naval War College: Newport, RI, USA, 2015.
2. Ministry of Defence. *Wargaming Handbook*; Development, Concepts and Doctrine Centre, Ministry of Defence: London, UK, 2017.
3. Dunnigan, J.F. *The Complete Wargames Handbook*, 2nd ed.; William Morrow and Company: New York, NY, USA, 2005.
4. Ministry of Defence. *Red Teaming Guide*, 2nd ed.; Development, Concepts and Doctrine Centre, Ministry of Defence: London, UK, 2013.
5. Priestley R.; Lambshead, J. *Tabletop Wargames*; Pen & Sword Military: South Yorkshire, UK, 2016.
6. Lopez, V.G.; Lewis, F.L.; Wan, Y.; Sanchez, E.N.; Fan, L. Solutions for multiagent pursuit-evasion games on communication graphs: Finite-time capture and asymptotic behaviors. *IEEE Trans. Autom. Control* **2020**, *65*, 1911–1923. [CrossRef]

7. Turetsky, V.; Weiss, M.; Shima, T. Minimum effort pursuit guidance with delayed engagement decision. *J. Guid. Control Dyn.* **2019**, *42*, 2664–2670. [CrossRef]
8. Hu, J.; Wang, L.; Hu, T.; Guo, C.; Wang, Y. Autonomous maneuver decision making of dual-UAV cooperative air combat based on deep reinforcement learning. *Electronics* **2022**, *11*, 467. [CrossRef]
9. Dong, J.; Chen, X.; Zhang, J.; Li, Z. Global path planning algorithm for USV based on IPSO-SA. In Proceedings of the Chinese Control Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 2614–2619. [CrossRef]
10. Zhou, H.; Zhao, H.; Huang, H.; Zhao, X. Integrated guidance and control design of the suicide UCAV for terminal attack. *J. Syst. Eng. Electron.* **2017**, *28*, 546–555.
11. Wang, X.; Tan, G.; Dai, Y.; Lu, F.; Zhao, J. An optimal guidance strategy for moving-target interception by a multirotor unmanned aerial vehicle swarm. *IEEE Access* **2020**, *8*, 121650–121664.
12. Zhou, J.; Shao, L.; Wang, H.; Zhang, D.; Lei, H. Optimal midcourse trajectory planning considering the capture region. *J. Syst. Eng. Electron.* **2018**, *29*, 587–600. [CrossRef]
13. Vitaly, S.; Tal, S. Cooperative differential games guidance laws for imposing a relative intercept angle. *J. Guid. Control Dyn.* **2017**, *40*, 2465–2480.
14. Huang, J.; Zhang, H.; Tang, G.; Bao, W. Extended differential geometric guidance law for intercepting maneuvering targets. *J. Syst. Eng. Electron.* **2018**, *29*,1046–1057.
15. Kang, S.; Wang, J.; Li, G.; Shan, J.; Petersen, I.R. Optimal cooperative guidance law for salvo attack: An MPC-based consensus perspective. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 2397–2410.
16. Zhu, C.; Xu, G.; Wei, C.; Cai, D.; Yu, Y. Impact-time-control guidance law for hypersonic missiles in terminal phase. *IEEE Access* **2020**, *8*, 44611–44621. [CrossRef]
17. Li, Z.; Chang, Y.; Kou, Y.; Yang, H.; Xu, A.; Li, Y. Approach to WTA in air combat using IAFSA-IHS algorithm. *J. Syst. Eng. Electron.* **2018**, *29*, 519–529. [CrossRef]
18. Ruan, C.; Zhou, Z.; Liu, H.; Yang, H. Task assignment under constraint of timing sequential for cooperative air combat. *J. Syst. Eng. Electron.* **2016**, *27*, 836–844.
19. Fu, X.; Gao, X. Effective real-time unmanned air vehicle path planning in presence of threat netting. *J. Aerosp. Info. Syst.* **2014**, *11*, 170–177.
20. Mukherjee, D.; Kumar, S.R. Field-of-view constrained impact time guidance against stationary targets. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 3296–3306.
21. He, J.; Yang, J. Dynamic gain military game algorithm based on episodic memory. In Proceedings of the Intational Conference Computer Engineering Application (ICCEA), Kunming, China, 25–27 June 2021; pp. 30–36. [CrossRef]
22. Yang, Z.; Zhou, D.; Piao, H.; Zhang, K.; Kong, W.; Pan, Q. Evasive maneuver strategy for UCAV in beyond-visual-range air combat based on hierarchical multi-objective evolutionary algorithm. *IEEE Access* **2020**, *8*, 46605–46623.
23. Sin, E.; Arcak, M.; Packard, A.; Philbrick, D.; Seiler, P. Optimal assignment of collaborating agents in multi-body asset-guarding games. *arXiv* **2020**, arXiv:2005.12226. [CrossRef]
24. Na, H.; Lee, J.I. Optimal arrangement of missile defense systems considering kill probability. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 972–983.
25. Guo, H.; Fu, W.; Fu, B.; Chen, K.; Yan, J. Smart homing guidance strategy with control saturation against a cooperative target-defender team. *J. Syst. Eng. Electron.* **2019**, *30*, 366–383. [CrossRef]
26. Shalumov, V. Online launch-time selection using deep learning in a target-missile-defender engagement. *J. Aerosp. Info. Syst.* **2019**, *16*, 224–236.
27. Duan, H.; Li, P.; Yu, Y. A predator-prey particle swarm optimization approach to multiple UCAV air combat modeled by dynamic game theory. *IEEE/CAA J. Autom. Sin.* **2015**, *2*, 11–18. [CrossRef]
28. Pan, Q.; Zhou, D.; Tang, Y.; Li, X. A novel antagonistic weapon-target assignment model considering uncertainty and its solution using decomposition co-evolution algorithm. *IEEE Access* **2019**, *7*, 37498–37517.
29. Li, Q.; Yang, R.; Feng, C.; Liu, Z. Approach for air-to-air confrontment based on uncertain interval information conditions. *J. Syst. Eng. Electron.* **2019**, *30*, 100–109. [CrossRef]
30. Chae, H.J.; Choi, H.L. Tactics games for multiple UCAVs within-visual-range air combat. In Proceedings of the AIAA Information Systems—Infotech@Aerospace Conference, Kissimmee, FL, USA, 10 January 2018; p. 0645.
31. Ji, X.; Zhang, W.; Xiang, F.; Yuan, W.; Chen, J. A swarm confrontation method based on Lanchester law and Nash equilibrium. *Electronics* **2022**, *11*, 896.
32. An, Y.-Y.; Park, K.-K.; Ryoo, C.-K. A study of close-formation approach attack tactics of multiple anti-ship missiles. In Proceedings of the International Conference Mechanical Aerospace Engineering (ICMAE), Budapest, Hungary, 10–13 July 2018; pp. 362–365.
33. Gong, J.; Zhang, X.; Liu, Y.; Zhang, X. Event graph based warship formation air defense scheduling model and algorithm. In Proceedings of the International Conference Dependable Systems Applications (DSA), Yinchuan, China, 5–6 August 2021; pp. 572–580.
34. Fu, H.; Liu, H.H.-T. An isochron-based solution to the target defense game against a faster invader. *IEEE Control Syst. Lett.* **2022**, *6*, 1352–1357. [CrossRef]
35. Yan, R.; Shi, Z.; Zhong, Y. Reach-avoid games with two defenders and one attacker: An analytical approach. *IEEE Trans. Cybern.* **2019**, *49*, 1035–1046.

36. Yan, R.; Shi, Z.; Zhong, Y. Guarding a subspace in high-dimensional space with two defenders and one attacker. *IEEE Trans. Cybern.* **2020**, *52*, 3998–4011.
37. Garcia, E.; Casbeer, D.W.; Pachter, M. Optimal strategies for a class of multi-player reach-avoid differential games in 3D space. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4257–4264. [CrossRef]
38. Selvakumar, J.; Bakolas, E. Feedback strategies for a reach-avoid game with a single evader and multiple pursuers. *IEEE Trans. Cybern.* **2021**, *51*, 696–707. [CrossRef]
39. Ganzfried, S.; Laughlin, C.; Morefield, C. Parallel algorithm for approximating Nash equilibrium in multiplayer stochastic games with application to naval strategic planning. *arXiv* **2020**, arXiv:1910.00193. [CrossRef]
40. Zhang, S.; Ran, W.; Liu, G.; Li, Y.; Xu, Y. A multi-agent-based defense system design for multiple unmanned surface vehicles. *Electronics* **2022**, *11*, 2797. [CrossRef]
41. Hughes, W.P., Jr. A salvo model of warships in missile combat used to evaluate their staying power. *Nav. Res. Logist.* **1995**, *42*, 267–289. [CrossRef]
42. Silav, A.; Karasakal, O.; Karasakal, E. Bi-objective missile rescheduling for a naval task group with dynamic disruptions. *Nav. Res. Logist.* **2019**, *66*, 596–615. [CrossRef]
43. Ma, Z.; Wu, K.; Liu, Z. Multi-ship cooperative air defense model based on queuing theory. *arXiv* **2022**, arXiv:2205.07820. [CrossRef]
44. Li, X.; Mitra, M.; Epureanu, B.I. Analysis of the synergy between modularity and autonomy in an artificial intelligence based fleet competition. In Proceedings of the NDIA Michigan Chapter's Ground Vehicle Systems Engineering And Technology Symposium (GVSETS), Novi, MI, USA, 13–15 August 2019. [CrossRef]
45. Kung, C.-C. Study on consulting air combat simulation of cluster UAV based on mixed parallel computing framework of graphics processing unit. *Electronics* **2018**, *7*, 160. [CrossRef]
46. Shahid, S.; Zhen, Z.; Javaid, U.; Wen, L. Offense-defense distributed decision making for swarm vs. swarm confrontation while attacking the aircraft carriers. *Drones* **2022**, *6*, 271. [CrossRef]
47. Hu, D.; Yang, R.; Zuo, J.; Zhang, Z.; Wu, J.; Wang, Y. Application of deep reinforcement learning in maneuver planning of beyond-visual-range air combat. *IEEE Access* **2021**, *9*, 32282–32297. [CrossRef]
48. Yang, Y.; Luo, R.; Li, M.; Zhou, M.; Zhang, W.; Wang, J. Mean field multi-agent reinforcement learning. *arXiv* **2020**, arXiv:1802.05438.
49. Zhang, J.; Yang, Q.; Shi, G.; Lu, Y.; Wu, Y. UAV cooperative air combat maneuver decision based on multi-agent reinforcement learning. *J. Syst. Eng. Electron.* **2021**, *32*, 1421–1438. [CrossRef]
50. Ownby, M.; Kott, A. Reading the mind of the enemy: Predictive analysis and command effectiveness. *arXiv* **2016**, arXiv:1607.06759. [CrossRef]
51. Li, X.; Epureanu, B.I. Analysis of fleet modularity in an artificial intelligence-based attacker-defender game. *arXiv* **2019**, arXiv:1811.03742. [CrossRef]
52. Yao, M.; Yin, Q.; Yang, J.; Yu, T.; Shen, S.; Zhang, J.; Liang, B.; Huang, K. The partially observable asynchronous multi-agent cooperation challenge. *arXiv* **2021**, arXiv:2112.03809.
53. Lei, X.; Tao, X. Research on UAV swarm confrontation task based on MADDPG algorithm. In Proceedings of the Default Cover Image 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 25–27 December 2020; pp. 1513–1518.
54. Wang, Z.; Liu, F.; Guo, J.; Hong, C.; Chen, M.; Wang, E.; Zhao, Y. UAV swarm confrontation based on multi-agent deep reinforcement learning. In Proceedings of the Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; pp. 4996–5001.
55. Liu, F.; Dong, X.; Yu, J.; Hua, Y.; Li, Q.; Ren, Z. Distributed Nash equilibrium seeking of N-coalition noncooperative games with application to UAV swarms. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 2392–2405.
56. Hagelback, J.; Johansson, S.J. Dealing with fog of war in a real time strategy game environment. In Proceedings of the IEEE Symposium Computational Intelligence and Games, Perth, WA, Australia, 15–18 December 2008; pp. 55–62.
57. Wang, H.; Tang, H.; Hao, J.; Hao, X.; Fu, Y.; Ma, Y. Large scale deep reinforcement learning in war-games. In Proceedings of the IEEE International Conference Bioinformatics Biomedicine (BIBM), Seoul, Korea, 16–19 December 2020; pp. 1693–1699. [CrossRef]
58. McLemore, C.; Gaver, D.; Jacobs, P. A model for geographically distributed combat interactions of swarming naval and air forces. *Nav. Res. Logist.* **2016**, *63*, 562–576. [CrossRef]
59. Seagren, C.W.; Gaver, D.P.; Jacobs, P.A. A stochastic air combat logistics decision model for Blue versus Red opposition. *Nav. Res. Logist.* **2019**, *66*, 663–674. [CrossRef]
60. De Lima Filho, G.M.; Kuroswiski, A.R.; Medeiros, F.L.L.; Voskuijl, M.; Monsuur, H.; Passaro, A. Optimization of unmanned air vehicle tactical formation in war games. *IEEE Access* **2022**, *10*, 21727–21741.
61. Alkebsi, K.; Du, W. A fast multi-objective particle swarm optimization algorithm based on a new archive updating mechanism. *IEEE Access* **2020**, *8*, 124734–124754.
62. Mahmoud, A.; Yuan, X.; Kheimi, M.; Almadani, M.A.; Hajilounezhad, T.; Yuan, Y. An improved multi-objective particle swarm optimization with TOPSIS and fuzzy logic for optimizing trapezoidal labyrinth weir. *IEEE Access* **2021**, *9*, 25458–25472.
63. Xu, Z.; Zhang, E.; Chen, Q. Rotary unmanned aerial vehicles path planning in rough terrain based on multi-objective particle swarm optimization. *J. Syst. Eng. Electron.* **2020**, *31*, 130–141.