Autonomous Strategic Defense: An Adaptive Clustering Approach to Capture Order Optimization

Noah D. Zepp*, Han Fu† and Hugh H.-T. Liu‡
Institute for Aerospace Studies, University of Toronto, 4925 Dufferin Street, Toronto, Ontario M3H5T6, Canada

The proliferation of UAV technology has introduced a new risk to the security of high-value assets. Emerging advancement in cooperative multi-agent control of UAVs presents a means of automating a defensive response to these new threats. A practical realization of an automated defense strategy is limited by the computational constraints of onboard computers. The UAV's onboard computer must solve multiple non-convex NP-hard navigation optimization problems to maximize the effectiveness of its defensive strategy. One such problem is the challenge of finding the optimal flight path for a single defender that must capture multiple slower invaders. This problem has been labeled as the n-Invader Capture Order Problem, abbreviated as n-ICOP. This research proposes an approximation method for reducing the solution space of n-ICOP. Given a specific constraint on computational resources, the method can adaptively reduce the computational load while optimizing the accuracy of the approximation. The new method splits the n-ICOP into a grouping problem and an ordered set problem, like the clustered variant of the Traveling Salesman Problem. The optimal grouping of invaders is estimated efficiently through the k-means clustering algorithm. The estimated grouping scheme reduces the complexity of an approximated n-ICOP solution because all strategies that separate members of a group are excluded from the search space. Simulations of the approximated n-ICOP solution were performed on a large data set of randomized defender-invader scenarios. Analysis suggests that this novel algorithm can reliably generate near-optimal strategies at a small fraction of the computational cost of a full exact solution. The results of the simulated trials demonstrate that the reduction in search space is substantial for the vast majority of randomized scenarios. This significant improvement in computational efficiency, with a sufficient degree of reliability, provides a practical means of solving for feasible n-ICOP solutions in a computationally limited environment.

I. Nomenclature

$\mathcal{D}$	=	dominant region of invader
D	=	position of the defender
I_i	=	position of an Invader
v_D, v_I	=	velocity of the defender and invader
α	=	average ratio of the defender velocity to the invader velocity (v_D/v_I)
r	=	capture range of the defender
P	=	arbitrary position

* M.Eng Student, Institute for Aerospace Studies, University of Toronto, 4925 Dufferin Street, Toronto, Ontario M3H5T6, Canada, AIAA Student Member
† Ph.D. Candidate, Institute for Aerospace Studies, University of Toronto, 4925 Dufferin Street, Toronto, Ontario M3H5T6, Canada
‡ Professor, Institute for Aerospace Studies, University of Toronto, 4925 Dufferin Street, Toronto, Ontario M3H5T6, Canada, AIAA Associate Fellow

P^*	=	optimal capture position
$\mathbb{P}$	=	set of optimal capture points
ℓ	=	capture order of Invaders
ℓ^*	=	optimal capture order of Invaders
$\mathcal{L}$	=	set of all possible capture orders
$\mathbb{m}$	=	number of invaders captured
n_{inv}	=	number of invaders in a single-defender multi-invader (SDMI) problem
K_{SDSI}	=	number of single-defender single-invader (SDSI) sub-problems that are solved
$e^{i,j}$	=	performance index of a SDSI subproblem
$\mathcal{g}$	=	valid grouping of invaders
G	=	ordered set of valid groups
$\mathbb{G}$	=	set of all possible ordered sets of valid groups
$g, \mathcal{g}$	=	convex function
$\mathcal{A}$	=	target region
S	=	starting location
δt	=	total time that the defender has not followed an optimal pursuit strategy

II. Introduction

This research paper investigates an exact and approximate solution to a path planning optimization problem that arises in the development of an autonomous target defense system. One of the primary objectives of such a system is to maximize the number of invaders that can be neutralized before reaching a prohibited region. A previous investigation into the optimal strategy for a single faster defender against multiple slower invaders has demonstrated that, given a prescribed capture order, the optimal control problem of the defender is convex [1]. This optimal control problem can be solved quickly, efficiently, and is well suited for the limited computational resources of a low-power onboard computer. However, the best capture order strategy of the defender to maximize the number of invaders captured must be solved before solving for the optimal capture points. The problem of finding the best capture order is called the n-Invader Capture Order Problem, abbreviated as n-ICOP. The number of possible capture orders is computed by the factorial of the number of invaders; the solution space grows rapidly as the problem scales. This rapid increase in complexity presents a problem for solving n-ICOP in computationally limited environments.

This work reformulates n-ICOP as a variation of the Traveling Salesman Problem, abbreviated as TSP. This reformulation is done to demonstrate that existing methods for solving NP-hard problems can be modified to solve n-ICOP exactly. An exhaustive search method is then adapted to solve the non-convex capture order problem by evaluating the solution to the convex navigation problems for all capture order options. As part of this exact solution to n-ICOP, the concept of adding auxiliary objectives to the problem to optimize additional performance characteristics of the defender. This exact solution can guarantee optimality but suffers from a large computational overhead.

Sacrificing guaranteed optimality for significant improvements in computational efficiency is often favourable for a practical realization of an autonomous robotic system. In addition to the exact n-ICOP solution, this research paper proposes a virtual invader approximation that allows for a significant reduction of the solution space. Furthermore, this approximation allows for the efficient heuristic algorithm known as k-means clustering to estimate an optimal solution for a sub-problem that arises within n-ICOP.

This novel algorithm for approximating the exact n-ICOP solution can adapt to optimize the effectiveness of the virtual invader approximation. The algorithm can estimate if a given scenario represents a ‘worst-case’ that cannot be approximated and will deploy the full exact solution if it is required.

A simulation framework is developed for evaluating and testing the performance of both the exact and approximate algorithms. The simulation propagates the optimal strategy of the invaders and defender through time. The software tracks the computational effort and optimality of the solution from the two algorithms. The data collected from various large data sets of randomized defender-invader games shows that the approximate method can reliably find feasible near-optimal solutions. The approximate solution is frequently computed at a small fraction of the cost of the exact n-ICOP solution.

The rest of this paper is organized as follows: First, the n-Invader Capture Order Problem is formulated. Second, exact and approximate solutions are formulated. Finally, the performance and efficiency of the solutions are investigated through simulation results.

III. Problem Formulation

The objective of finding the capture order that maximizes the number of invaders captured by a single defender can be formally represented as the maximization problem:

$$\ell^* = \underset{\ell \in \mathcal{L}}{argmax}\, \mathbb{m}(\ell) \tag{1}$$

where ℓ is an ordered set of invaders, $\mathbb{m}$ is the number of invaders captured, and $\mathcal{L}$ is the superset of all possible capture orders. The defended target region is defined as any convex region described by $\mathcal{A} = \{P \mid g(P) \leq 0\}$ [1]. The value of $\mathbb{m}$ is defined as $\mathbb{m} = |\mathbb{P}|$, where $\mathbb{P} = \{P^* \mid g(P^*) > 0\}$ for a given capture order. The set $\mathbb{P}$ has a unique solution for all capture orders. This solution can be found by decomposing a single defender multi-invader game into a set of n_{inv} single defender single invader games, all with a unique P^*. These single defender single invader games, abbreviated as an SDSI game, were found to be trivial to solve efficiently in a previous investigation [1]. A dominant region is a region of space where the invader can arrive at any point before the defender arrives. The dominant region is defined formally as:

$$\mathcal{D} = \left\{ P \middle| \frac{||PD|| - r}{v_D} \geq \frac{||PI||}{v_I} \right\} \tag{2}$$

In a single defender multi-invader game, abbreviated as an SDMI game, the definition of the dominant regions must be augmented to account for the time spent by the defender capturing other invaders. This augmented dominant region is defined as:

$$\mathcal{D}' = \left\{ P \middle| \frac{||PD|| - r}{v_D} + \delta t \geq \frac{||PI||}{v_I} \right\} \tag{3}$$

The optimal capture point for a specific invader, P_i^*, is then found as the solution to the convex optimization problem,

$$P_i^* = \min_{P \in \mathcal{D}'_i} g(P) \tag{4}$$

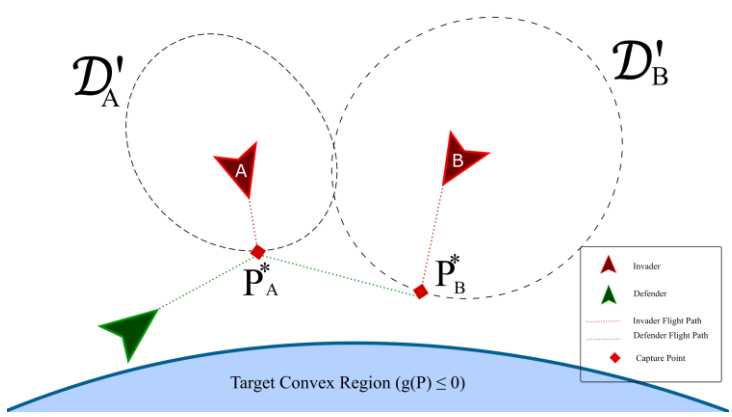


Fig. 1 Illustration of an SDMI game with 2 invaders

As an example of how this method is deployed in practice, consider the SD2I game depicted in Fig. 1, when $\ell = (A, B)$, the two dominant regions are defined as:

$$\mathcal{D}'_A = \left\{ P \middle| \frac{||PD|| - r}{v_D} + \underbrace{0}_{\delta t_A} \geq \frac{||PI_A||}{v_{I_A}} \right\}, \quad \mathcal{D}'_B = \left\{ P \middle| \frac{||PP_A^*|| - r}{v_D} + \underbrace{\frac{||D\,P_A^*||}{v_D}}_{\delta t_b} \geq \frac{||PI_B||}{v_{I_B}} \right\} \tag{5}$$

Subsequently, the two optimal capture points are computed by:

$$P_A^* = \underset{P \in \mathcal{D}'_A}{argmin}\, g(P), \qquad P_B^* = \underset{P \in \mathcal{D}'_B}{\text{argmin}}\, g(P), \tag{6}$$

For the game shown in Fig. 1, $\ell = (A, B) \rightarrow \mathbb{P} = \{P_A^*, P_B^*\}$. It can be determinined from the computed positions of P_A^* and P_B^* in Fig. 1 that $\mathbb{m} = |\mathbb{P}| = 2$. Note that knowledge of $\mathbb{m}$ for $\ell = (A, B)$ provides no insight into the value of $\mathbb{m}$ for $\ell = (B, A)$. This means that to decide which capture order is best, $\mathbb{P}$ must be solved for all permutations of ℓ so that the maximum $|\mathbb{P}|$ of all the options can be selected.

This formulation of n-ICOP would suggest that the problem falls into a category of problem known as NP-hard. To verify NP-hardness, consider that this problem can be framed as a weighted graph.

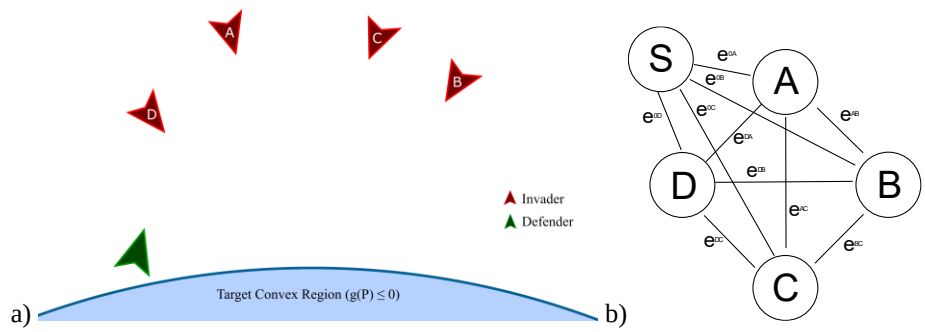


Fig. 2 a) Spatial representation of an SDMI problem. b) Weighted graph representation of an ASMI problem.

In this case, the value of the edge weights can be defined by an arbitrary performance metric. An effective performance metric should be a non-linear function of initial conditions and the flight properties of the defender and invaders. One example of such a performance metric is the capture efficiency objective function described in [1]. In a graphical form, such as that seen in Fig. 2, a simple case of the capture order problem represents a variation of the generalized traveling salesman problem (TSP), a classic NP-hard problem typically formulated as a weighted graph problem and solved using integer linear programming [2]. The capture order problem is further complicated by the fact that the edge weights, e^{ij}, of the graph are non-linear functions of ℓ and the component SDSI problems that make up the SDMI problem. Therefore, the capture order problem is at least as hard as TSP and thus itself must be NP-hard.

The reason for establishing hardness is that it provides insight into the nature of the solution implementation. As an NP-hard problem, the optimal solution must be found through dynamic programming and brute-force search methods [2]. The number of SDSI subproblems, K_{SDSI}, that must be solved to complete an exhaustive search of the solution space is a function of the number of non-repetitive permutations of the capture order and the number of SDSI subproblems solved for each permutation.

$$K_{SDSI} = n_{Inv}(n_{Inv})! \tag{7}$$

If the number of invaders is small, this problem is tractable with limited computational resources. However, as the number of invaders grows, the exact optimal solution quickly becomes dramatically more intensive and expensive to compute. For many real-world realizations of NP-hard problems, it is often more favourable to seek practical and efficient solutions, at the expense of guaranteed optimality. This sub-optimal approach employs assumptions and application-specific information to make an initial guess to reduce the search space and approximate a feasible solution to the problem.

As defined previously, the goal is to develop a solution to the capture order problem that can be scalable to maximize the optimality of the solution within the limits of available computational resources. To achieve this, first, a method of solving the exact solution to the n-invader capture order problem must be devised, then an approximation of the exact solution must be developed and validated against the exact solution.

IV. Proposed Solution Formulation

Two algorithms have been developed to solve n-ICOP; the exact solution and the approximate solution. The exact solution guarantees that the maximum number of invaders are captured but requires an exhaustive search approach that quickly becomes intractable for low-power computer hardware. The approximate solution utilizes key spatial information about the invaders and heuristic methods to reduce the n-ICOP solution space.

A. Formulation of The Exact Solution Algorithm

The exact method is a brute force approach to solving n-ICOP. The algorithm simply solves for $\mathbb{m}$ using a generalized form of Eq. 5 and Eq. 6 for all permutations of ℓ, the capture order that maximizes $\mathbb{m}$ is selected as ℓ^*. An important property of the exact solution is that ℓ^* does not need to be unique, as a result, there can often be multiple capture orders that maximize $\mathbb{m}$. If the only objective of the automated defense system is to optimize $\mathbb{m}$, then any ℓ^* can be selected at random. This random selection approach spares any additional computational effort.

However, the fact that multiple ℓ^* exists means that an auxiliary objective can be applied to further improve the performance of the system. Two examples of these auxiliary objectives were investigated as part of this research. The first objective function that was explored is designed to select a ℓ^* that maximizes the distance invaders are captured from the defended target region. The second auxiliary objective function that was investigated, aims to improve the traversability of the flight path by selecting a ℓ^* that minimizes extreme maneuvering throughout the capture order.

1. *Auxiliary Objective I*

The objective of maximizing the distance away from the target region that invaders are captured can be formally expressed as:

$$(\ell^*)^* = \max_{\ell^* \in \mathcal{L}^*} \sum_{n \in \ell^*} g(P_n^*) \tag{8}$$

The benefit of applying this objective function on top of the objective function found in Eq. 1 is that it improves the robustness of the solution if model uncertainty increases. In practice, this additional robustness stems from the fact that if an error occurs in the defender's strategy while the invaders are further from the target, there is more time to re-compute a new strategy.

2. *Auxiliary Objective II*

This auxiliary objective function is intended to promote the selection of a capture order that results in a more traversable flight path. Consider the two scenarios depicted in Fig. 3, both trajectories are assumed to result in an optimal solution, but a method is needed to quantify which one is favourable.

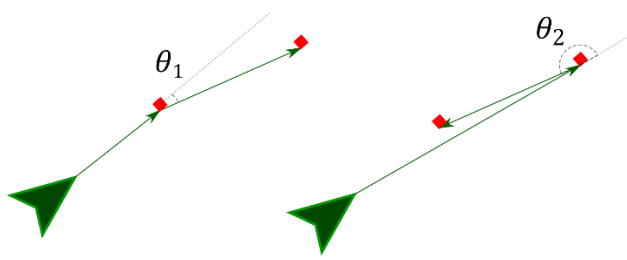


Fig. 3 A pair of possible permutations of a capture sequence

A multi-rotor aircraft can perform extreme turning maneuvers, such as the one in Fig. 3 that is defined by θ_2. However, the most efficient and reliable mode of flight is a steady level forward flight path with gradual changes in direction. This mode of flight allows properties such as disk loading and airspeed to be optimized effectively. One way to select a capture order that maximizes the aircraft performance is to simulate the dynamic motion model of the aircraft and track quantities such as energy expenditure. A simpler approach is to pick a cost function that reaches a minimum when the angle between one segment of the flight path and the next is at 0 radians or 2π radians but is maximized when this angle is π radians. An example of a simple cost function that achieves this quality is expressed as:

$$J(\theta) = \frac{1}{2}(1 - cos\theta) \tag{9}$$

This cost function is evaluated along the flight path for a given trajectory. The ℓ^* with the lowest sum of J along the trajectory can be selected as the option with the least extreme maneuvers.

B. Formulation of The Approximate Solution Algorithm

The approximate solution reduces the n-ICOP solution space by grouping sets of invaders together to form a set of smaller SDMI problems to be solved. The exact optimal solution for the superset of smaller subproblems is again

only found through exhaustive search. However, making assumptions about the collective behaviour of groups of invaders, while still solving the optimal solutions of the subproblems will provide a considerable decrease in the computational complexity of the problem while still providing near-optimal feasible solutions. To support this claim, consider an acyclic directed graphical representation of an optimal solution (ℓ^*),

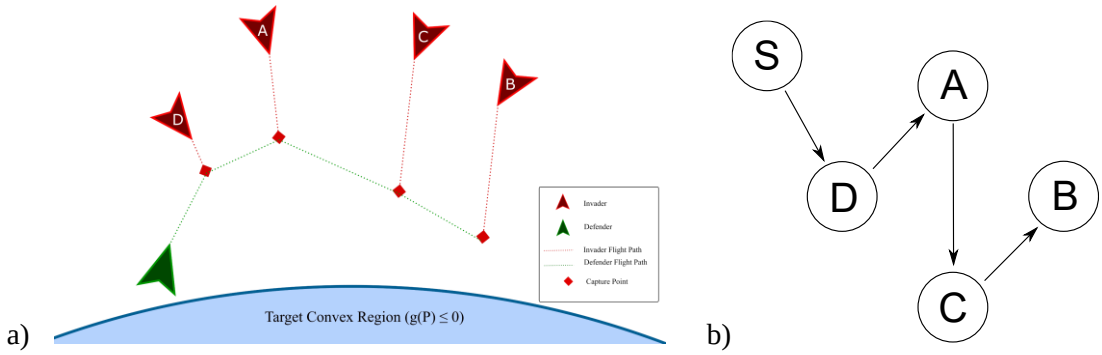


Fig. 4 a) Spatial representation of an optimal solution, b) Direct-graph representation of an optimal solution.

Given that an optimal solution exists, the adjacent nodes in the directed graph can be grouped in several ways according to three rules:

i. Groups must contain at least one invader
ii. All members of a group must form a valid directed graph
iii. No single group can contain more then $(n_{Inv} - 1)$ invaders

A non-exhaustive list of a few examples of valid groupings is displayed in Fig. 5.

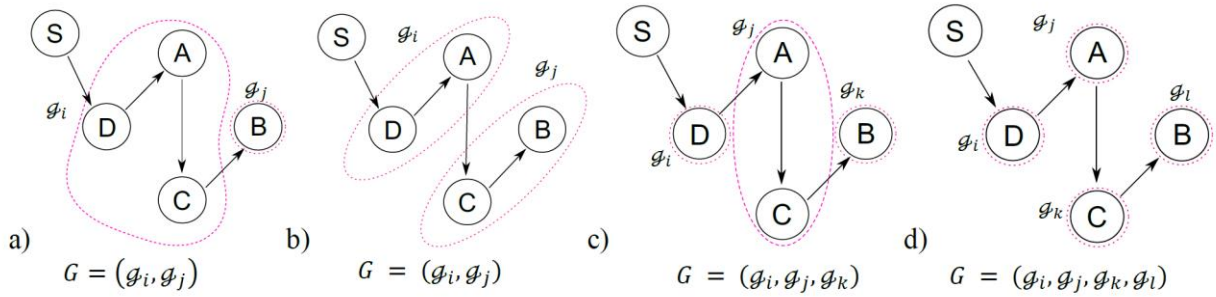


Fig. 5 Examples of valid grouping schemes.

The set of all valid groupings is denoted $\mathbb{G}$. The ordered set G represents an ordered set of valid groups $(\mathcal{g}_1 \dots \mathcal{g}_n)$.

$$\mathbb{G} = \{G \mid \forall G = (\mathcal{g}_i, \mathcal{g}_j, \dots \mathcal{g}_n)\} \tag{10}$$

In the optimal solution, the order of G corresponds to the order with which the groups are visited by the defender to maximize $\mathbb{m}$. Therefore, the following optimization problem can be stated:

$$G^* = \max_{G \in \mathbb{G}} \mathbb{m}(G) \tag{11}$$

Consider the case when ℓ^* is unknown but the optimal grouping scheme is given. The process of solving for the G^* to maximize $\mathbb{m}$ for any G is the same as solving for the full n-ICOP solution. This equivalence between the ordered set problem and n-ICOP can be validated by the fact that the non-grouped formulation of n-ICOP is itself a valid grouping, as seen in Fig. 5.d and thus an element of $\mathbb{G}$.

If the G and ℓ^* are unknown, then this new ordered set problem is harder to exactly solve than n-ICOP. However, this expansion of the problem presents a means of efficient approximation. Given that n-ICOP itself is a subset of the ordered set problem, these approximations prove useful in developing an approximate solution to n-ICOP.

To demonstrate the value of this approach, consider a case where the elements of G are known but the order of the set is unsolved. Additionally, the capture order of the invaders within the groups remains unsolved. An approximation is applied which allows the order of G to be estimated reasonably well using a subset of n-ICOP. This method has been named the virtual invader approximation. First, apply one of the valid groupings from Fig. 5 to an SDMI game similar to the case seen in Fig. 6. Then for each group, generate a virtual invader at the mean spatial position of the group. Propagate forward the exact n-ICOP solution but only for the virtual invaders.

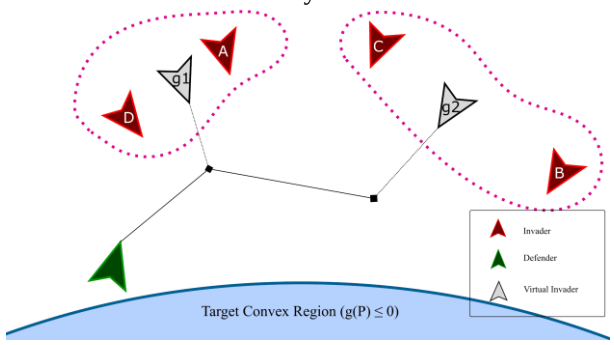
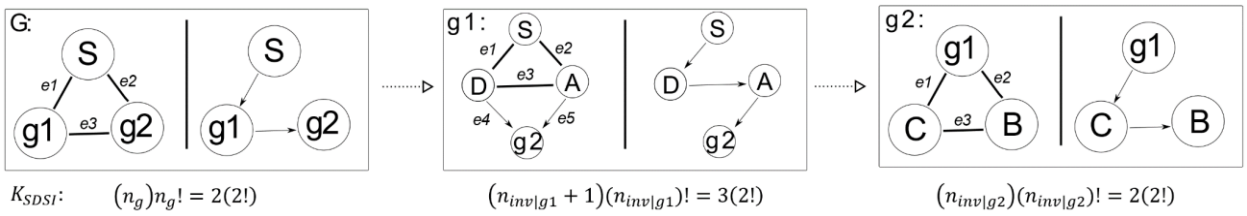


Fig. 6 Demonstration of the virtual invader approximation

The solution for the optimal capture order of the virtual invaders can then be used as an approximate solution to the general ordered set problem for this proposed grouping scheme. The intuition for why this approximation can be effective derives from the optimal strategy of the invader [1]. The invader's strategy is largely a function of the spatial relationship between the invader, the defender, and the target region. Consider, two invaders that are spatially close to one another and are adjacent to each other in the optimal capture order, both invaders will have optimal capture points close together. As the two invaders get arbitrarily closer to one another, the approximation of the group behavior by a single virtual invader becomes increasingly more accurate and precise.

The exact order of capture for the individual invaders is still unknown, but the search space is reduced because solutions that separate members of a group are rejected. Given this approximation, the calculation of the exact reduction in complexity is trivial. As a measure of complexity, the number of SDSI subproblems that must be solved is used. Consider the graphical representation of an algorithm that would implement this approximation in Fig. 7.



K_{SDSI}: $(n_g)n_g! = 2(2!)$ $(n_{inv|g1} + 1)(n_{inv|g1})! = 3(2!)$ $(n_{inv|g2})(n_{inv|g2})! = 2(2!)$

Fig. 7 Graphical representation of the search space reduced algorithm

The total number of SDSI subproblems that are solved by this algorithm is 14, this is in contrast with the exact solution for a 4-invader problem which must solve 4(4)! = 96 SDSI subproblems.

There are still two barriers to a full realization of this approximate solution. Thus far, it has been assumed that the grouping information was known ahead of time. Additionally, it has been assumed that the virtual invader solution provides a reasonable approximation of the optimal ordered set of groups. Rigorously proving the accuracy of the approximation from a theoretical standpoint is beyond the scope of this research paper, but similar methods have previously been applied to other NP-hard problems [3] [4].

Some intuitive observations can be made, first and foremost the choice of grouping scheme has a large impact on the accuracy of the approximation and the complexity of the solution. To demonstrate the impact specific grouping options can have, consider the selection of the grouping scheme in Fig. 8.b over Fig. 8.a. The arrangement with fewer invaders in a group results in less information being ignored by an approximation. The extreme case of this is when all groups have exactly one invader, in this case, no information is lost but the exact n-ICOP solution must be solved, and subsequently no computational effort is spared. A second observation that can be made is that low spatial variance

amongst members of a group increases the accuracy of the approximation. Similarly, the larger the distances between groups are, relative to the distance between invader within groups, the better the virtual invader approximation will perform.

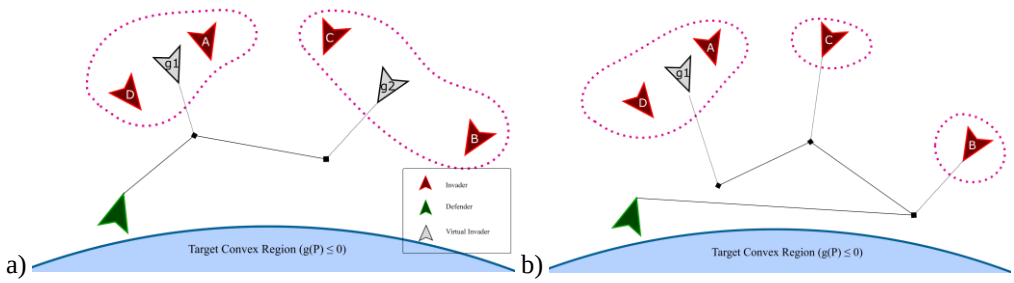


Fig. 8 Two different grouping schemes for an SDMI game

From these observations, the following conjecture can be asserted:

Across all possible grouping schemes, there exists at least one choice that maximizes the effectiveness of the approximation by minimizing spatial variance within groups and optimizing the number of groups used. This grouping scheme will reduce the search space while ensuring the near-optimality of the solution.

Similar to n-ICOP, the exact solution to which grouping scheme is the best must be found through exhaustive search algorithms. However, the objective of minimizing spatial variance within groups can be approximately solved by several heuristic data clustering algorithms used for unsupervised learning applications. Though these algorithms are iterative, the number of invaders that could realistically be captured by a single defender is low enough that the algorithms will converge on a local minimum very quickly. An example of a fast and efficient algorithm that can achieve acceptable results is the k-means algorithm. Additionally, including silhouette analysis in the k-means clustering algorithm has the benefit of automating the process of finding the optimal number of clusters to fit the data and meet computational constraints. This silhouette analysis of clusters will introduce an adaptive element to the approximate method.

There is no guarantee that the clustered invaders that are returned by a heuristic method will constitute a valid grouping, according to the defined grouping rules. This is because no information about the optimal solution is known. However, it is assumed that members of an optimal valid group are typically spatially close together, thus, groupings that minimize the k-means objective function are thought to likely also constitute a valid grouping. Experimentation demonstrates that by fine-tuning the hyperparameters of the k-means algorithm, the estimated grouping scheme and the virtual invader approximation produced feasible n-ICOP solutions reliably at a fraction of the cost of the full optimal n-ICOP solution.

V.Simulation Results

A python API was developed to facilitate the simulation of the SDMI games that fall within the scope of this research. The software package includes functions to implement both the exact solution and approximate solution to n-ICOP. Additionally, the performance characteristics of the solutions are tracked such that the reliability and efficiency of the approximate solution can be evaluated from the aggregated performance of the algorithm across a data set of randomized SDMI games.

A. Exact Solution Algorithm Simulation

To demonstrate the functionality and operation of the algorithm that implements the exact n-ICOP solution a randomized 4 invader SDMI game ($r = 0,\ \alpha = 4$) was generated in Fig. 9.

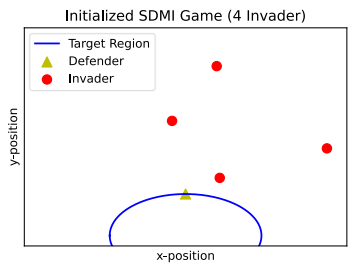


Fig. 9 Randomized SDMI game.

The exact solution iterates all possible permutations of the capture order and solves the SDMI and SDSI sub-problems for each capture order. The value of $\mathbb{m}$ for all ℓ is computed, this data is compiled in Fig. 10 for the example game in Fig. 9.

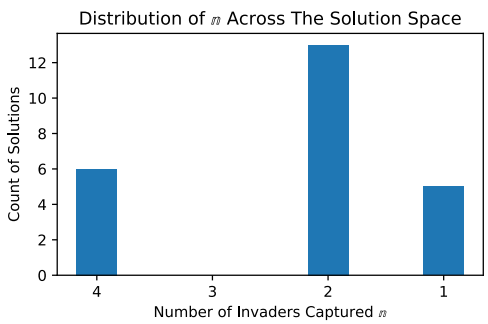


Fig. 10 Bar graph of the number of ℓ for each value of $\mathbb{m}$ from the randomized SDMI example game.

For this example, Fig. 10 shows that $|\mathcal{L}^*| = 6$, confirming that ℓ^* is not unique. All $\ell \notin \mathcal{L}^*$ are discarded from the exact solution. In section IV.A, a few methods for selecting one of the optimal capture orders were discussed. The simplest method is to select an arbitrary ℓ^* from $\mathcal{L}^*$, seen in Fig. 11.A. The more sophisticated approach is to implement an auxiliary objective that utilizes the data from the individual SDMI solutions to select one ℓ^* from $\mathcal{L}^*$. Two examples of these auxiliary objectives were described, the results of applying auxiliary objectives I and II are shown in Fig. 11.B and Fig. 11.C, respectively.

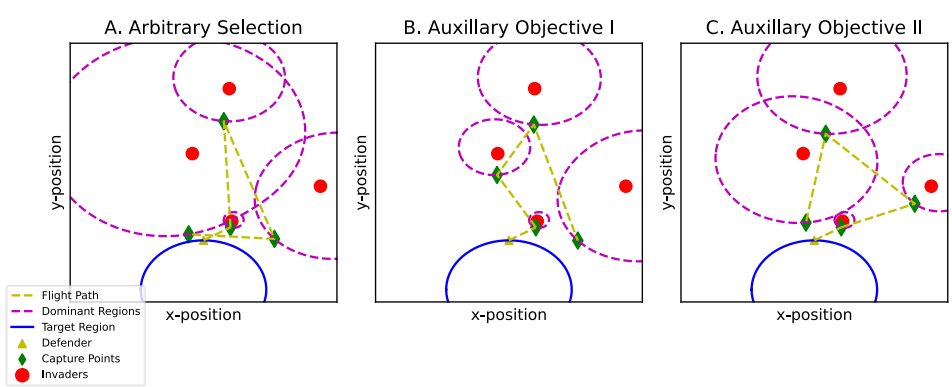


Fig. 11 Three optimal n-ICOP solutions. A) Arbitrarily select ℓ^* from $\mathcal{L}^*$. B) ℓ^* is selected to maximize distance from $\mathcal{A}$. C) ℓ^* is selected to optimize aircraft performance.

The solutions plotted in Fig. 11 demonstrate the benefit of utilizing an auxiliary objective. The first auxiliary objective successfully promotes a solution with more distance between the capture points and the target region. This will maximize the success of the defender if model and state uncertainty exist. The simulation also confirms that the

second auxiliary objective is successful in promoting a solution that minimizes extreme maneuvers. The example in Fig. 11.C illustrates the selected ℓ^* that represents the most easily traversed route, as defined by auxiliary objective II.

Approximated Solution Simulation

The approximate method of solving n-ICOP described in Section VI.B can be demonstrated in a similar manner to the exact method. First, a randomized SDMI game ($r = 0,\ \alpha = 5$) is generated, the example game is found in Fig. 12.A. In this example, the SDMI game is scaled up to 6 invaders to highlight the efficiency of the approximate method.

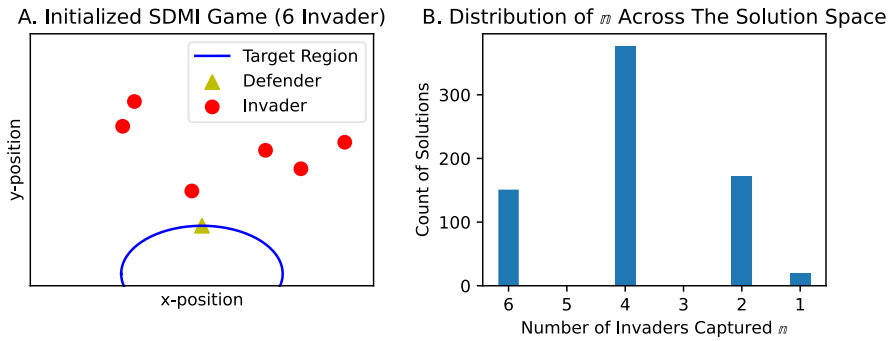


Fig. 12 A) Unsolved 6 Invader SDMI game. B) Bar graph of the number of ℓ for each value of $\mathbb{m}$ from the SDMI example game.

The distribution of $\mathbb{m}$ across the solution space can be seen in Fig. 12.B. This data is generated from the exact method and is not required for the approximate solution. It is presented here to visualize the initial volume of the solution space. The approximate and exact method was then applied to the SDMI game to find an optimal capture order, the simulation results are shown in Fig. 13.

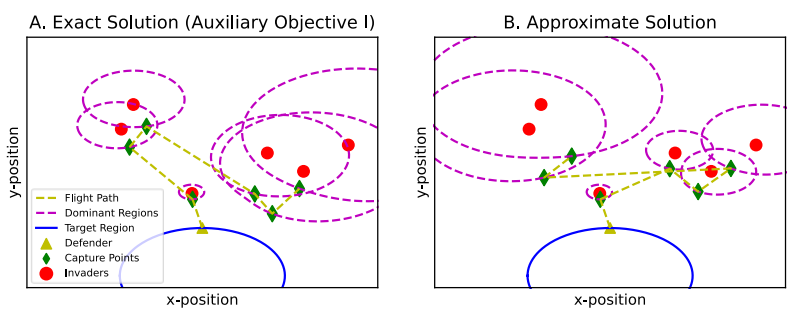


Fig. 13 Solution comparison for a 6 invader SDMI game. A) Exact Method Solution ($K_{SDSI}^{exact} = 4320$) B) Approximate Method Solution ($K_{SDSI}^{approx} = 620$)

In Fig. 13, the approximate method does not produce the same solution as the exact method, however, the solution found through approximation still maximizes $\mathbb{m}$. The optimal capture order found through the approximate method required only 620 SDSI subproblems to be solved, this is in contrast to the 4320 SDSI subproblems solved by the exact method. For this example, the virtual invader approximation and clustering reduce the search space by over 85%. This represents a significant reduction in complexity of an n-ICOP solution.

To verify that the approximate method can find an optimal capture order with a significant degree of reliability, a data set of 150 randomized 6 invader SDMI games was generated so that the aggregated performance data could be analyzed. The approximate algorithm was first deployed on the data set with no re-initialization allowed, this means that the algorithm is only run once. Disabling re-initialization ensures that $K_{SDSI}^{approx} \leq K_{SDSI}^{exact}$, however, the solution is vulnerable to local minima in the approximation and thus does not guarantee optimality. This notion is supported by the data in Fig. 14.

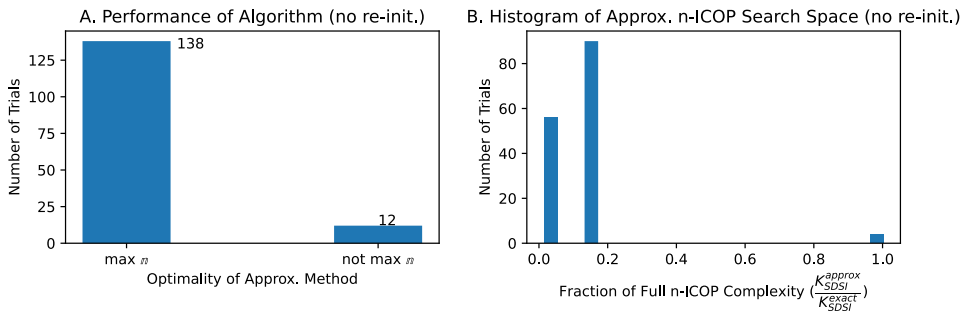


Fig. 14 Aggregated performance data of the approximate method with re-initialization disabled.

The approximate method with no re-initialization was able to capture no fewer invaders than the exact method in 92% of the randomized games. This high degree of reliability was achieved for most trials at a small fraction of the cost of the exact solution. It can also be noted from Fig. 14.B, that the few trials where $K_{SDSI}^{approx} = K_{SDSI}^{exact}$ are games that the adaptive clustering algorithm identifies as worst-case problems and allots more resources to solve them.

The technique of re-initialization can be applied such that optimality is guaranteed but K_{SDSI}^{approx} is not guaranteed to be lower then K_{SDSI}^{exact}. In this next set of simulations, the solution returned by the approximate method was checked for optimality before deciding on a capture order. If the incumbent solution is not optimal, the clustering algorithm is re-initialized to include an additional cluster, the solution is then recomputed. This process is repeated until either an optimal solution is found, or the number of clusters equals the number of invaders. The result of this technique is that the solution will be optimal but as a consequence some cases $K_{SDSI}^{approx} \geq K_{SDSI}^{exact}$.

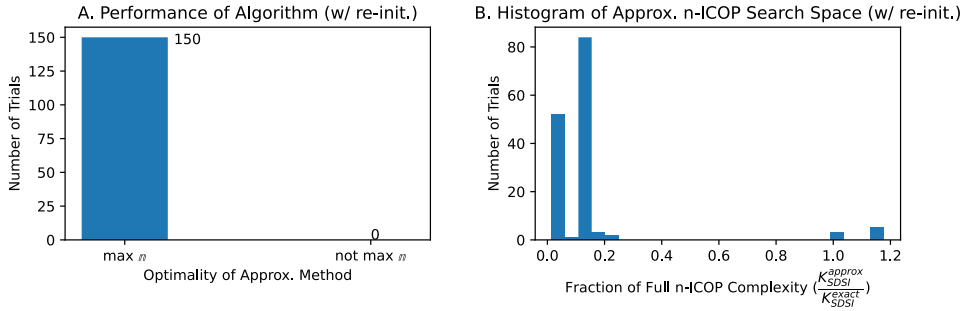


Fig. 15 Aggregated performance data of the approximate method with re-initialization enabled.

The results of the games that utilized re-initialization, seen in Fig. 15, are consistent with the expected behavior of the re-initialization technique. Despite the additional overhead of checking the solution and iterating until an optimal solution is found, a substantial majority of the games are still solved much faster than with the exact n-ICOP method.

VI. Conclusion

This paper details an investigation of the n-Invader Capture Order Problem that arises in the development of optimal control strategies for an autonomous UAV defense system. Two solutions to n-ICOP have been developed, an exact brute force solution and an approximate solution. Both methods provide a practical framework to solve the optimal capture order problem on existing flight computer hardware.

The introduction of auxiliary objectives to the exact n-ICOP solutions presents the opportunity to further refine the solution to meet application-specific goals. The simulation of two examples of auxiliary objectives demonstrates the success of this solution refinement method for a set of simple use cases.

The proposed method of reducing the n-ICOP solution space through a virtual invader approximation and k-means clustering was found to be effective at reducing computational complexity at minimal cost to optimality. This novel heuristic approach to n-ICOP offers a viable alternative to the exact solution for computationally limited environments. Combined with the re-initialization technique, it is possible to benefit from a significant improvement in computational efficiency for most SDMI games while not sacrificing the optimality in the event of a worst-case scenario. Furthermore, a restriction on the number of re-initialization events can be placed on the algorithm to limit the upper bound of K_{SDSI}^{approx} to meet the reliability and efficiency needs of a specific application.

Further investigation into the nature of worst-case games is warranted to improve the performance of the adaptive clustering algorithm. Improved detection of worst-case conditions will limit unnecessary calculation and reduce the excessive computational overhead of the re-initialization strategy.

References

[1] H. Fu and H. H.-T. Liu, "On the Strategies of Defending a Target against Multiple," *scitech AIAA*, 2021.

[2] G. Dantzig, Linear Programming and Extensions, Princeton, NJ: Princeton University Press, 1963.

[3] Z. H. Ahmed, "The Ordered Clustered Travelling Salesman Problem: A Hybrid Genetic Algorithm," *The Scientific World Journal* , vol. 2014, 2014.

[4] A. Jaradat, B. Matalkeh and W. Diabat, "Solving Traveling Salesman Problem using Firefly algorithm and K-means Clustering," in *IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology*, Amman. Jordan, 2019.