中国科学院大学
University of Chinese Academy of Sciences

博士学位论文

复杂收益安全博弈研究

作者姓名：殷　越

指导教师：安波　副研究员，山世光　研究员

中国科学院计算技术研究所

学位类别：工学博士

学科专业：计算机科学与技术

研 究 所：中国科学院计算技术研究所

二〇一六年一二月

Security Games with Complex Payoff Structures

A Dissertation Submitted to

The University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Doctor of Philosophy

in

Computer Science and Technology

by

Yue Yin

Dissertation Supervisor: Professor Bo An, Shiguang Shan

Institute of Computing Technology

Chinese Academy of Sciences

December, 2016

声　　明

我声明本论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，本论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

作者签名：　　日期：

论文版权使用授权书

本人授权中国科学院计算技术研究所可以保留并向国家有关部门或机构送交本论文的复印件和电子文档，允许本论文被查阅和借阅，可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编本论文。

（保密论文在解密后适用本授权书。）

作者签名：　　导师签名：　　日期：

摘　要

关于安全博弈的研究近年来广受重视，许多基于安全博弈论的系统已在现实世界中得到了成功应用。在该研究的理论框架中，博弈双方分别为安保部门和不法分子。其中安保部门首先确定一种策略并根据该策略执行安保任务，而不法分子可通过观测等方式完全知晓安保部门的策略，并根据该策略采取对自己最有利的行动。虽然第一代安全博弈研究成功的对于多种安全问题进行了基于斯塔克伯格博弈的建模，并提出了高效的算法进行相关模型的求解，但其存在三项局限，因而制约了该研究在更广泛领域中的应用。第一，大部分第一代安全博弈研究都假设可能被攻击的目标，也即需要被保护的目标，其价值是相互独立的，且不法分子最终只能攻击一个目标。这个假设在保护选举进行等领域并不成立。第二，此前研究大都假设目标的重要程度是固定不变的。这个假设在重大赛事安保、城市安保等领域不能成立。第三，第一代研究通常认为博弈参与方仅需选择目标进行保护或攻击，而不考虑他们如何到达这些目标，也不考虑安保部门可在不法分子到达目标的路途中对其进行拦截。该假设使得在研究海洋保护区巡逻等基于图的安全博弈时不能计算出最优的安保策略。针对这三项局限，本文研究了目标非独立、攻击目标不唯一的安全博弈，动态收益安全博弈以及图安全博弈，并对每种博弈下的代表性问题进行建模和均衡求解算法设计。本文的主要创新点如下。

1. 本文基于保障选举进行的实际问题研究了目标价值不独立、攻击目标不唯一的安全博弈。本文考虑了选举中可能面临的破坏性攻击（不法分子的目标是阻止最初的赢家获胜）和建设性攻击（不法分子的目标是使得某个原本不能获胜的候选人获胜）两种攻击方式。本文首先分别分析了这两种攻击方式的复杂度和抵制这两种攻击以保障选举进行的复杂度。随后，本文利用斯塔克伯格博弈模型，对保障选举进行以及抵制两种攻击的问题进行建模。针对抵制破坏性攻击，本文提出了一种线性规划算法，求解抵制攻击并保护选民的最优策略。本文又提出了一种基于 Double Oracle 框架的更高效的算法，在大多数情况下能够在避免遍历策略空间的情况下即求得最优解。针对抵制建设性攻击，本文提出了一种混合整数线性规划算法以求解抵制攻击的最优策略。本文还提出了一种启发式算法，其能够在一部分情况下在多项式时间内求出最优策略。此外，本文还提出了一种近似算法以求解一般问题中“足够好”的策略，该算法能够在绝大多数情况下返回最优解。本文在真实数据集和模拟数据集上进行实验，验证了模型和算法的有效性。

2. 本文研究了动态收益安全博弈，考虑了重大赛事安保与城市安保两个场景，以分别探索动态收益安全博弈中纯策略均衡和混合策略均衡的计算。在两种场景下本文均考虑了博弈参与双方具有连续策略空间的情形。针对重大赛事安保问题，本文首先

关注了资源的转移时间远小于赛事的进行时间的情形，并提出算法 SCOUT-A 来求解此情形下安保方的最优策略。然后，本文又研究了更具一般性的资源转移时间不可忽略的情况。针对这种情形，本文从离散时间假设着手，提出算法 SCOUT-D 来求解在离散时间假设下的安保方最优策略，并在此基础上设计出算法 SCOUT-C 以求解连续策略空间、资源转移时间不可忽略的情形下的安保方最优策略。针对城市安保问题，本文首先设计算法 COCO 来求解安保资源转移时间可忽略情形下的紧凑表示的最优混合策略。基于 COCO 计算出的紧凑表示的最优混合策略，本文提出一种采样方法以确定每次安保任务的具体执行情况。对于混合策略均衡模型下资源转移时间不能忽略的情况，本文提出能够在多项式时间内求出 ϵ 近似解的算法 ADEN。实验证明 COCO 和 ADEN 均取得了比现存算法更好的效果。

3. 本文基于海洋保护区巡逻问题进行了基于图的安全博弈的研究。图中的每个节点表示一个时间和地点的二元组，图中的任一条路线均为安保部门或不法分子的可行策略。本文利用网络流的方式来紧凑地表示安保部门的混合策略，并提出一种线性规划算法 CLP 求解安保部门的最优巡航策略。此外，本文还将网络流思想和 Double Oracle 框架相结合并提出 CDOG 算法以高效求解最优策略。实验结果表明 CLP 和 CDOG 均取得了明显好于现存算法的效果，CDOG 的可扩展性也明显好于已知的最好算法。

综上所述，本文研究了三种复杂收益安全博弈以及这三种博弈下的四个代表性问题。针对每一个问题，本文均采用了新的建模方式以解决此前研究工作中模型的局限性，并提出了高效的算法以求解各模型中的最优安保策略。实验证明本文所提出的模型和算法取得了比现有研究成果更好的效果。

关键词：安全博弈，目标非独立，动态收益，图论

Abstract

Security games have been a hot research topic, aiming at handling complex resource allocation and patrolling problems in security domains. Many real-world applications have been deployed for different domains based on the framework of security games, where the defender (e.g., security agency) has a limited number of resources to protect a set of targets from an adversary (e.g., terrorist), who can know the defender's strategies and respond the best. Whereas the first generation of security games research has proposed significant models and algorithms for many problems, they mainly subject to three constraints. First, most previous works assume that the values of targets are independent, and that the attacker can only attack one target at last. This assumption does not hold in domains like protecting elections. Second, most previous works are based on the assumption that the values of targets are static, and will not change over time. In domains like protecting large public events or urban patrol, this assumption is usually violated. Finally, it is widely assumed that the defender and the attacker just 'choose' targets to protect or attack, without considering that both the defender and the attacker can take paths to get to the targets, and the attack can be deterred when the attacker is on his way to the targets. To extend the first generation research on security games, this paper investigates three classes of security games with complicated payoff structures: security games with dependent targets, security games with dynamic payoffs and security games on graphs. For each class of games, this paper models the game in a new way, and proposes algorithms to compute the equilibrium strategies. The main contributions of this paper are listed as follows.

1. This paper studies security games with dependent targets based on the scenario of protecting elections against group-level election control, where control is allowed at the granularity of groups of voters (e.g., voting locations), through a denial-of-service attack, and the defender allocates limited protection resources to prevent control. This paper considers plurality voting, and shows that it is NP-Hard to prevent both constructive and destructive control, though destructive control itself can be performed in P. For defense against destructive control, this paper presents a double-oracle framework for computing an optimal prevention strategy, developing exact mixed-integer linear programming formulations for both the defender and attacker oracles (both of these subproblems we show to be NP-Hard), as well as heuristic oracles. Experiments demonstrate that the proposed computational framework can scale to large problem instances. For defense against constructive control, this paper develops a heuristic algorithm and an approximate algorithm to compute defender strategies. Experiments conducted on both synthetic

and real data show that the resulting strategies are optimal with high probability.

2. This paper studies security games with dynamic payoffs and considers two scenarios —protecting large public events and urban patrol —to analyze pure strategy equilibrium and mixed strategy equilibrium in such games respectively. For both scenarios, this paper builds models considering the continuous strategy space of the defender and the attacker. For protecting large public events, this paper first proposes SCOUT-A, which makes assumptions on relocation cost, exploits payoff representation and computes optimal solutions efficiently. Then proposes SCOUT-C to compute the exact optimal defender strategy for general cases despite the continuous strategy spaces. For urban control, this paper proposes an optimal algorithm and an arbitrarily near-optimal algorithm to compute security strategies under different conditions. Experimental results show that both algorithms significantly outperform existing approaches.

3. For security games on graphs, this paper mainly focuses on the scenario of protecting marine protected areas (MPAs), and proposes a new Stackelberg game model to formulate the problem of protecting MPAs. This paper also proposes two algorithms to compute the efficient protection agency's strategies: CLP in which the protection agency's strategies are compactly represented as fractional flows in a network, and CDOG which combines the techniques of compactly representing defender strategies and incrementally generating strategies.

To summarize, this paper addresses three significant types of security games, proposes new models to get rid of the restrictions caused by previous assumptions, and proposes algorithms to solve the games. Experimental results show that the models and algorithms proposed by this paper significantly outperform previous work, and expand the frontiers of security games.

Keywords: Security games; Dependent targets; Dynamic payoffs; Games on graphs

目　录

图目录

表目录

第一章　绪论

1.1　研究的背景和意义

安全问题是现代社会各国都必须面对的重大问题，包括保护城市基础设施免受恐怖分子的攻击，保护重大赛事、选举活动等公共活动免遭蓄意破坏，以及保护自然资源免遭不法分子破坏等。解决上述安全问题具有两项基本挑战：其一，安保资源数量有限，却需要被用于保护众多的目标。例如，公安部门往往需要用少量的巡逻车监管整个城市众多设施、景点的安全情况。这意味着安保部门不可能随时随地保护所有需要被保护的目标，因此高效地分配安保资源至关重要。其二，现代社会的不法分子往往能够进行策略性行为，例如在进行破坏活动之前观测学习警方的安保策略，并根据已学到的安保策略选择成功率最高、破坏性最大的破坏活动。这就要求安保部门在进行安保策略设计时即考虑不法分子可能的观测与回应。

为应对这两项挑战，博弈论被广泛应用于安保问题的建模和安保策略的规划设计 [104]。其中，斯塔克伯格博弈模型以及其对应的强斯塔克伯格均衡解（称为 SSE 均衡）的概念起到了关键作用 [8,22,99]。在该模型中，博弈双方为安保部门和不法分子。安保部门首先确定一种策略（通常是一个随机化的策略分布，也称混合策略），并根据该策略执行每次的安保任务。不法分子能够完全知晓安保部门的策略，并根据该策略采取对自己最有利的行动。在 SSE 均衡解下，安保部门所确定的策略应保证，即使不法分子完全知晓该策略并作出最优反应，该策略仍能使安保部门取得在此情况下的最优收益。由于安全问题的多样性，斯塔克伯格博弈模型在应用时需针对具体安保问题进行改进与扩充。第一代安全博弈研究成功地对于多种安全问题进行了基于斯塔克伯格博弈的建模，并提出了高效的算法进行相关模型地求解。基于斯塔克伯格博弈模型的系统也已在多个安全领域被成功部署应用并发挥了巨大作用。例如用于洛杉矶机场安保的 ARMOR 系统 [90]，用于美国港口巡逻的 PROTECT 系统等 [98]。

然而，第一代安全博弈研究具有几项局限，制约了其在更广泛领域中的应用。第一，大部分第一代安全博弈研究都假设可能被攻击的目标（也即需要被保护的目标），其价值是相互独立的 [34]。不法分子的最终目标是攻击使其自身获利最高的一个目标 [37]。例如，在针对洛杉矶机场安保的研究中 [90]，攻击目标和保护目标均为机场的 7 个航站楼。由于航班数、人流量不同，各个航站楼如果被成功攻击，不法分子能够得到的收益不同。但是，攻击一个航站楼的收益并不会受其他航站楼是否被成功攻击的影响。因此不法分子会综合考虑成功的概率与航站楼本身的价值，选择攻击一个对自身最有利的航站楼。然而，在很多其他问题中，如保障选举免受操纵的问题中，攻击目

标的价值都不是相互独立的，犯罪分子也会选择一组而非一个目标进行攻击。第二，大部分第一代安全博弈研究都假设目标的重要程度是固定不变的，因此安保部门与不法分子的收益仅取决于是否有目标被成功攻击以及被成功攻击的目标的价值，而与攻击发生的时间并无关系 [114]。然而，在重大赛事安保，城市巡逻等问题中，需要被保护的目标，如城市景点，其重要程度随时间变化，攻击收益不仅取决于目标，也取决于攻击时间。第三，大多数此前研究工作均假设安保部门与不法分子的策略简单，例如安保部门只需在各攻击目标间分配安保资源，不法分子只需选择要攻击的目标，而不考虑双方如何到达目标，在目标间转移的路线，也不考虑攻击可能会持续一定时长等因素。但在海洋巡逻等问题中，安保部门实际上可以通过考虑不法分子的行动路线，在不法分子到达目标之前就对其进行拦截。这几项局限使得第一代安全博弈研究提出的理论与算法在很多现实问题中不能应用。本文的目的即在于解决这三项局限，扩展对于安全博弈的研究，使其能够更广泛地应用于现实问题。

1.2 研究的问题和挑战

本文主要关注三种复杂收益安全博弈。第一是目标价值不独立、攻击目标不唯一的安全博弈。第二是动态收益安全博弈，也即参与双方的收益不仅取决于被攻击的目标，还取决于攻击发生的时间的安全博弈。第三种是图安全博弈，参与双方的策略均为图上的路径，双方的收益不仅与被攻击的目标有关，还与双方路径重叠处的长度，以及攻击持续的时间有关。下面简单介绍这三种安全博弈的研究动机和相应的问题与挑战。

1. 非独立，不唯一目标安全博弈。第一种安全博弈的研究动机主要是为了保障选举依法进行，选票真实可信。选举过程的公正公平是维护社会稳定公平的重要条件。然而，存在一些不法分子，希望通过阻止部分选民投票的方式以影响选票数量，操纵选举结果。例如，在 2013 年的巴基斯坦大选中，就有不法分子在两个投票站附近引爆炸弹，使得需要在此投票站进行投票的选民不能正常投票 [96]。虽然选举制度的可靠性广受科研人员关注，但此前的研究多聚焦于选举制度本身，也即研究给定选举制度，采取某种方法控制选举结果是否为 NP 难的。如果是，那么这个选举制度即被认为是可靠的，否则则被认为是不可靠的。但关于选举制度本身的研究并不能解决实际中的问题。第一，即使控制选举的方式为 NP 难的，随着近年来对于 NP 难问题算法研究的进展，不法分子仍有可能在短时间内得到高效的近似解。第二，即使选举制度是不可靠的，由于历史背景等原因，骤然改变选举制度也并非朝夕可行。本文首次从防御的角度来考虑保障选举进行的问题，也即选举的组织机构可以通过分配安全资源的方式保护选民，以保障选举进行。

意欲控制选举结果的不法分子，其攻击目标为有投票权的选民。事实上，相对于攻击单个选民，更值得关注的问题，也是更具有普遍性的问题，是不法分子的攻击目标为某个选民群体（攻击单个选民的情况是其中的一个特例，即选民群体中只包含一位

选民）。如上文提及的巴基斯坦选举中的攻击案例，攻击某个投票站，其目标在于攻击依赖该站点进行投票的选民群体。同时，攻击单个选民群体并不一定能够达到控制选举的目的，因此不法分子攻击多个选民群体的情况需要被考虑，这也就意味着不法分子可能进行攻击的方式随着选民群体总数和可攻击群体总数的增长而组合增长。另外，选举结果是否被控制并不取决于某一个选民群体是否被成功攻击，而取决于当不法分子发动全部攻击后，减少的总票数是否会影响最终结果。同一个选民群体，在不同的攻击组合里，造成的影响是不一样的。因此基于目标价值独立和单一攻击目标假设的安全博弈模型以及相应的求解算法都不能用于解决这个问题。综上，建立考虑目标非独立、攻击目标不唯一等问题的安全博弈模型，并提出高效的算法以求出安保资源的最优分配，既至关重要，又面临着技术上的众多挑战。

2. 动态收益安全博弈。针对第二种安全博弈，本文主要研究了两个场景所对应的两种重要模型。第一个场景是重大赛事的安保问题。2013 年的波士顿马拉松爆炸案使得全世界都对重大赛事的安保问题愈发重视。在类似于马拉松比赛的赛事中，不法分子可能的攻击目标为比赛的各个赛段（或者大型体育馆的各个区域等）。随着比赛的进行，选手的移动，马拉松各个赛段聚集的选手、观众数目均在变化，因此不法分子与安保部门的收益不仅取决于一个赛段是否被攻击，还取决于攻击发生的时间。同时，重大赛事往往信息公开（如时间、观众数量等），但频次较低（如一年一次等），因此不法分子可以根据公开的信息推测安保策略，但难以通过观察以往的安保策略获取更多信息。因此，解决重大赛事的安保问题，更需要的是一次最优的单次策略，而非期望收益最优的随机化混合策略。

第二个场景是城市安保问题。随着恐怖活动日益猖獗，城市的公共设施、景点商场等场所都成为了不法分子可能攻击的目标。与重大赛事的安保问题类似，此场景下各个目标的重要程度依然随时间变化。例如，博物馆等场所在上午重要性较高，而酒吧街等场所则在晚上更需要被保护。不同于重大赛事安保问题的是，城市的安保每天都需要进行，不法分子在活动前极容易通过观察过去一段时间内的安保策略来估算攻击各个目标的成功概率。因此，将策略随机化，计算最优的混合策略，能够使安保部门获得更高的收益。

对于重大赛事安保问题和城市安保问题，虽然由于其解空间的差异，需要用不同的模型来描述，但这两个问题都面临一项技术上的挑战，即，攻击可能发生在任何时间，而安保资源也可以在任何时间进行转移。这意味着安保部门和不法分子都有无限多个可能实施的策略。那么，如何利用合理的模型描述策略空间，如何设计高效的算法以求解安保部门的最优策略，就是亟待解决的问题。

3. 图安全博弈。第三种安全博弈主要是为解决海洋保护区的巡逻问题。海洋中的珊瑚礁、珍稀鱼类广受人类喜爱，但也面临着遭受恶意盗取、不法垂钓的威胁。许多国家都建立了海洋保护区，通过在保护区内巡逻，使海洋生态环境免遭破坏。然而，海洋

保护区的巡逻往往面临着保护区面积广阔而巡逻资源稀缺的问题。例如我国三亚的亚龙湾自然保护区，面积达 85 平方公里，但仅有三条巡逻船用于巡逻。因此设计高效的巡逻路线至关重要。海洋保护区内各个区域的重要程度也会随着海洋生物游动、破坏难度随时间变化等情况而发生变化。同时，破坏海洋环境的不法分子需要选择一条路线到达进行破坏的地点，因此环保部门不仅可以在“目标”处，也即不法分子进行破坏活动的地方抓获他，也可以在不法分子到达目标的路途中拦截他。这使得最优巡逻路线的计算需考虑的问题规模庞大，因此最优巡逻路线的求解也十分困难。

1.3 研究现状

安全博弈的相关研究主要关注当安保资源有限，而潜在的破坏分子又能够进行策略性攻击时，如何基于博弈理论进行安保资源的配置，以达到最优的安保效果[11,19,40,65,72,100,107,108,113,118,122]。其中基于斯塔克伯格博弈模型的研究受到了最广泛的关注 [1,22,38,53,83,92,93,104,126–128]。在这一系列研究中，研究人员关注了零和博弈 [52,130]、非零和博弈 [97,125]、贝叶斯博弈 [88,131] 等博弈类型，研究了纳什均衡、斯塔克伯格均衡、最大最小均衡 [23,32,46,60] 等解空间在模型中的应用，以及相应的均衡精炼问题，提出了 DOBSS[88]，ORIGAMI[55]，Double Oracle 框架 [46] 等用于求解相应模型均衡的算法。基于这些模型和算法的系统也已经被成功的部署在现实世界中，用于解决机场安保（ARMOR 系统等 [90]），港口安保（PROTECT 系统等 [98]），公共交通安保（TRUSTS 系统等 [130]）等重大问题，并取得了令人瞩目的成果。

第一代关于安全博弈的研究大多严格基于斯塔克伯格博弈模型 [14,70]，其基础假设包括攻击方（如不法分子）能够完全获知防守方（如安保部门）的策略 [20]，攻击方能够根据已知信息进行最优回应 [56,71]，以及防守方在制定最优策略后能够精确执行 [9,67,105,106,111]。此外，这些工作通常也假设攻守双方的策略简单（防守方选取目标保护，攻击方选取一个目标攻击，不考虑双方到达目标的路线等因素，不考虑在攻击方到达目标的路途中进行拦截等可能性）[49,84]，收益形式简单（攻守目标确定即确定收益，不考虑目标重要性的动态变化，不考虑目标之间重要性的相关性）[19,55,88]。虽然基于这些假设的模型具有一定的局限性，但相关研究开拓了将博弈理论应用于实际的安全问题的新方向，为之后的研究打下了基础 [104]。

其后，研究人员在第一代安全博弈研究的基础上，尝试了放松模型中与现实不符的假设，考虑了攻击方难以获得关于防守方策略的完全信息 [59]，攻击方有限理性或没有能力做出最优回应 [50]，防守方难以精准的执行最优策略等具有不确定性的情形 [82,91]；也关注了具有多个防守方 [51,75]、多个攻击方 [57] 的更加复杂的博弈形式 [41,109]；还研究了关于保护移动的目标（如渡轮），攻守双方进行序列博弈（如防止破坏森林）等问题，设计并部署了应用于保护渡轮，保护环境的新系统。此外，更多的技术也被整合用于解决安全问题 [132]，包括通过机器学习的方法预测攻击方的行为模式

等 [42,69]。

近年来，安全博弈的应用范围被进一步拓展，相关模型框架被应用于解决更多的社会问题，包括野生动物的保护 [35,124]，出租车定价 [36,39]，城市公共资源的配置与管控 [62,76,120]，网络攻防的研究 [5,64,66,81]，乃至病毒与疫苗的研究等 [85]。然而，如前文所述，此前关于安全博弈的研究多数着眼于攻击目标价值独立且恒定 [4,47,89]，且博弈中参与者策略简单的问题，因此其模型和算法不能直接用于解决本文研究的问题。

1.3.1 安全博弈与选举

关于攻击选举的复杂度问题起始于 [7] 中的研究，然而其研究仅考虑建设性攻击的问题，即攻击方希望某个特定的候选人获胜，本文同时考虑了破坏性攻击的问题，即攻击方的目的在于使初始的获胜方不能赢得选举。破坏性攻击的问题最初由 [44] 提出，而关于选举攻击的问题其后又被多名研究者扩展到更多的模型和选举规则下 [12,29,29–31,73,74,79,80,87,112,119]。然而，这些关于选举攻击的研究均从单个选民的角度来考虑攻击类型 [21,24,25,117,133]。[17] 首度从选民群组的角度研究攻击，但该工作认为选民的群组归属由其对候选人的偏好决定，且群组之间可交叉。本文考虑更具一般性的由选民本身属性决定的，不可交叉的分组形式。[27] 随后研究了基于群组的选举攻击，但其仅考虑建设性攻击。[18] 虽考虑了建设性攻击和破坏性攻击的不同影响，但其研究的攻击类型为针对候选人的攻击，而非本文研究的更常见的针对选民的攻击。[26] 中研究的问题与本文最为相似，但其仅从复杂度的角度分析攻击的难度，未从防御的角度考虑配置安保资源以保障选举进行的方法。

安全博弈已被应用于许多现实世界中的安保问题 [104]。但大部分此前工作均假设可能被攻击的目标价值相互独立 [38]，且攻击者只能攻击一个目标 [1]。此外的工作考虑的则多是攻守双方同时行动的情形 [58]，或未能给出精确的求解方式 [110]。与此不同，本文考虑目标非独立且攻击方可攻击多个目标的情形，以应对保障选举安全进行所面临的挑战。该情形下攻守双方的收益形式均比此前模型更加复杂。因此，基于此前模型中收益形式的算法 [46,78,114] 也无法直接用于求解本文中的问题。

1.3.2 动态收益安全博弈

关于安全博弈的研究工作大多假设可被攻击的目标的重要程度是固定不变的 [54,61,70]，但这个假设在很多现实问题中并不成立，因此近年来也有一些研究人员开始尝试放松这个假设。[129] 在研究地铁系统安保问题时考虑到了地铁车厢的重要程度在一天之内会有所变化的问题。但是，由于地铁的运行遵循固定的时刻表，因此该工作在考虑地铁安保策略时只考虑安保资源在预先固定的时刻发生转移的情形。与之类似的，[121] 研究了该情况下的安保策略设计的复杂度。[32,33] 研究了保护游轮等移动目标的问题，这些目标随着自身的移动，重要性也可能发生变化。该工作虽然考虑到了攻击可能发生在任何时刻的情形，但依然假设安保方只能在事前确定的固定时间点移动

资源，并且没有分析该假设对安保方的收益可能造成的损失。另外，该工作研究安保资源只能在一维直线上移动的情形。与之前的工作不同，本文在研究动态收益安全博弈时认为安保资源可在各个目标之间，在任何时间点进行移动，并理论证明了离散化安保策略可能带来的安保收益的损失界限。

1.3.3 图安全博弈

图安全博弈的问题被许多研究者所重视。[68] 从理论的角度研究了以紧凑表示策略的方式求解图安全博弈的复杂度。[46,48] 研究了城市道路中的安保问题。但在这些工作的模型中，可能受攻击的目标是道路，也就是图中的边，而非图中的节点。因此其模型和结论并不适用于类似于海洋保护区巡逻的，攻击目标是图中节点（保护区内某片区域）的情形。其在文中提出的基于 Double Oracle 框架的算法也并不能直接被用于解决文本提出的问题。[32] 和 [121] 研究对渡轮等移动目标的保护，并考虑了此情境下，巡逻艇等安保资源的策略可看做图上的路径的问题。但其模型依然假设攻击方可“空降”至攻击目标，不考虑在攻击方到达目标的图中拦截攻击方的可能性。[131] 和 [52] 考虑对于地铁系统的安保。虽然他们考虑了资源在地铁路线图中的移动，但该工作的目的在于打击逃票以提高地铁系统收益，因此在该模型中，攻击方（也就是可能逃票的乘客）选择的路径是固定的，即该乘客每日通勤的路径，于是攻击方的策略仅限于买票或不买票，而无需选择最优路径。[15] 研究博弈参与方的策略是序列进行的，双方逐次选取图中的一条边，最终形成两条图中的路径。博弈最终双方的收益不取决于攻击方的“目标”，而取决于双方每步对于边的选择。该工作首次考虑在求解过程中综合应用策略的紧凑表示法和 Double Oracle 框架，但由于模型设置与本文研究的问题不同，因此其算法也不能被用于解决本文提出的问题。

1.4 本文主要贡献

本文针对三种复杂收益的安全博弈（目标非独立，攻击目标不唯一的安全博弈，动态收益安全博弈，和图安全博弈）包含的四个主要问题（选举保障问题，重大赛事安保问题，城市安保问题，海洋保护区巡逻问题），做出的主要贡献如下。

- 针对选举保障问题，本文考虑了选举中可能面临的破坏性攻击（不法分子的目标是阻止最初的赢家获胜）和建设性攻击（不法分子的目标是使得某个原本不能获胜的候选人获胜）这两种攻击方式。本文分别分析了这两种攻击方式的复杂度以及了抵制这两种攻击的复杂度，并利用斯塔克伯格博弈模型对保障选举免受这两种攻击影响的问题进行建模。针对抵制破坏性攻击，本文提出了一种线性规划算法，求解抵制攻击、保护选民的最优策略，又提出了一种基于 Double Oracle 框架的更高效的算法，其在大多数情况下能够在避免遍历策略空间的情况下即求得最优解。针对抵制建设性攻击，本文提出了一种混合整数线性规划

算法以求解抵制攻击的最优策略。本文还提出了一种启发式算法，其能够在一部分情况下在多项式时间内求出最优策略。此外，本文还提出了一种近似算法以求解一般问题中“足够好”的策略，该算法能够在绝大多数情况下返回最优解。本文在真实数据集和模拟数据集上进行实验，验证了模型和算法的有效性。

- 针对重大赛事安保问题，本文设计了一种博弈模型来描述该问题。在此模型中，安保部门与不法分子均具有连续的策略空间。在此博弈模型的均衡策略下，即安保方的最优策略下，安保方可能遭受的最大的损失最小。本文首先关注了资源的转移时间远小于赛事的进行时间的情形，并提出算法 SCOUT-A 来求解此情形下安保方的最优策略。然后，本文又研究了更具一般性的资源转移时间不可忽略的情况。针对这种情形，本文从离散时间假设着手，提出算法 SCOUT-D 来求解在离散时间假设下的安保方最优策略，又在此基础上设计出算法 SCOUT-C，以求解连续策略空间、资源转移时间不可忽略的情形下的安保方最优策略。最后，本文通过实验验证了算法的有效性。
- 针对城市安保问题，本文首先设计了一种博弈模型，不仅考虑到安保部门与不法分子的连续策略空间问题，还考虑到在均衡解下安保部门的策略可能是最优的混合策略而非具体的某一个纯策略的情形。在此均衡解概念下，本文首先考虑资源转移时间可忽略不计的情形，采用紧凑的方法表示安保部门的混合策略，设计算法 COCO 来求解此情形下最优混合策略。基于 COCO 的结果（紧凑表示的混合策略），本文提出一种采样方法，以确定每次进行城市安保所使用的策略。本文随后考虑了资源转移时间不能忽略的情况，并提出算法 ADEN，其能够在多项式时间内求出 ϵ 近似解。实验证明 COCO 和 ADEN 均取得了比现存算法更好的效果。
- 针对海洋保护区巡逻问题，本文提出了基于图的安全博弈模型，图中的每个节点表示一个时间和地点的二元组，图中的任一条路线均为安保部门或不法分子的可行策略。本文利用网络流的方式来紧凑的表示安保部门的混合策略，并提出一种线性规划算法 CLP 求解安保部门的最优巡航路线。此外，本文还将网络流思想和 Double Oracle 框架相结合，提出 CDOG 算法，以高效求解最优策略。实验结果表明 CLP 和 CDOG 均取得了明显好于现存算法的效果，CDOG 的可扩展性也明显好于已知的最好算法。

1.5 本文组织结构

本文第二章首先介绍安全博弈的理论基础和现有应用案例。第三章研究目标非独立、攻击目标不唯一的安全博弈，重点关注保障选举进行，避免选民被第三方不法分子攻击的问题。第四章从重大赛事安保问题入手，研究纯策略均衡下的动态收益安全博弈。第五章研究混合策略均衡下的动态收益安全博弈，用以为城市安保设计最优策略。

第六章研究图安全博弈，并设计海洋保护区的最优巡逻方案。第七章对全文进行总结，并介绍下一步研究的可行方向。

第二章　安全博弈基础理论与应用简介

2.1　安全博弈基础模型

2.1.1　攻守策略

安全博弈是一种双方博弈，双方通常被称为“防守方”（如安保部门）与“攻击方”（如不法分子）。安全博弈模型假定有一组 n 个“目标”。防守方的目的是保护这些目标，攻击方的目的是成功地攻击这些目标。大部分关于安全博弈的工作均基于斯塔克伯格模型。该模型有一个领导者和一个跟随者，领导者先行动，跟随者在领导者行动后能够完全知晓领导者的行动，并根据领导者的行动确定最有利于自己的行动。在安全博弈问题中，防守方即为领导者，攻击方即为跟随者。假设防守方共有 m 个安保资源，那么他一次最多可以保护保护 m 个目标。防守方的一个“纯策略”就是这 m 个资源的一种分布情况，通常用一个向量表示：

$$s = \langle s_i \in \{0,1\} : \sum_{i \in \{1,\cdots,n\}} s_i = m \rangle. \tag{2-1}$$

其中 s_i 用于表示目标 i 是否被一个安全资源保护。由于安保资源的有限性，通常有 $m < n$，即安保方并不能保护所有的资源。

安保方也可使用随机化的混合策略以提升自身收益，混合策略是纯策略的一个分布。令安保方的所有纯策略构成的策略空间计作 $\mathcal{S}$，那么一个混合策略可计作：

$$\mathbf{x} = \langle x_s \in [0,1] : \sum_{s \in \mathcal{S}} x_s = 1 \rangle. \tag{2-2}$$

在基础的安全博弈模型中，攻击方的纯策略通常被认为是选择一个目标进行攻击。由于攻击方后行动并根据已知的防守策略做出最优回应，因此攻击方总是存在一个纯策略最优回应。令 a 表示一个攻击方的纯策略，可记作

$$a = \langle a_i \in \{0,1\} : \sum_{i \in \{1,\cdots,n\}} a_i = 1 \rangle. \tag{2-3}$$

其中 a_i 用于指示目标 i 是否为攻击方选择的攻击目标。令 $\mathcal{A}$ 表示攻击方的纯策略空间，即所有纯策略的集合。

2.1.2　收益与均衡

每个目标 i 都对应于一组收益值 $\{P_i^A, R_i^A, P_i^D, R_i^D\}$。如果攻击方攻击目标 i，而 i 被安保资源保护了，那么攻击方将得到收益 P_i^A，安保方将得到收益 R_i^D。反之，如果安保方

没有保护目标 i，而攻击方攻击了该目标，那么安保方的收益为 P_i^D，而攻击方的收益为 R_i^A。（P 为惩罚"Penalty" 的简写，R 为奖赏"Reward" 的简写，D 为安保方 "Defender" 的简写，A 为攻击方"Attacker" 的简写。）为使博弈合理，需要 $R_i^A > P_i^A$，$R_i^D > P_i^D$，即，安保方保护某目标，会使得自身收益升高，攻击方攻击该目标的收益减少。表2.1显示了一个在单目标情形下攻守双方的收益示例。在单目标情形下，攻击方总是会攻击这个目标，那么，如果防守方保护这个目标，那么双方收益分别为防守方 $R^D = 5$，攻击方 $P^A = -10$。如果防守方不保护这个目标，那么双方受益则为防守方 $P^d = -20$，攻击方 $R^A = 30$。

	保护	不保护
防守方	$R^D = 5$	$P^d = -20$
攻击方	$P^A = -10$	$R^A = 30$

表 2.1　单目标攻守双方收益示例

那么，给定一组博弈双方的纯策略 (s, a)，根据 a 中攻击的目标在 s 中是否被保护，以及这个目标所对应的收益值，即可得到在这组策略下双方的收益，记作攻击方收益 $u^A(s, a)$ 和防守方收益 $u^D(s, a)$。那么，给定一个防守方混合策略和攻击方纯策略的组合 $(\mathbf{x}, a)$，也可得到在该策略组合下攻守双方的收益，分别为

$$\begin{aligned} u^D(\mathbf{x}, a) &= \sum_s x_s u^D(s, a), \\ u^A(\mathbf{x}, a) &= \sum_s x_s u^A(s, a). \end{aligned} \tag{2-4}$$

强斯塔克伯格均衡通常被用作安全博弈的均衡解。在这个模型下，安保方在基于领导者会得知自己的行动，并根据自己的行动作出最优反应的假设下，寻求最优解。强斯塔克伯格均衡同时要求，如果攻击方攻击多个目标都会得到相同的收益，那么他选择的是对安保方较为有利的那个目标。强斯塔克伯格均衡的正式定义如下：

定义 1 令 $g(\mathbf{x}) : \mathbf{x} \to a$ 代表攻击方的反应函数。如果一个策略组合 $(\mathbf{x}^*, g(\mathbf{x}^*))$ 满足以下条件，那么这组策略构成强斯塔克伯格均衡。

1. 防守方使用最优策略：对于任一 $\mathbf{x}$，有 $u^D(\mathbf{x}^*, g(\mathbf{x}^*)) \geq u^D(\mathbf{x}, g(\mathbf{x}))$。
2. 攻击方根据防守方策略做出最优回应：$g(\mathbf{x}^*) \in f(\mathbf{x}^*)$，其中 $f(\mathbf{x}^*) = \text{argmax}_{a \in \mathcal{A}} u^A(\mathbf{x}^*, a)$ 是攻击方最优回应的集合。
3. 在收益相同情况下，攻击方选择最有利于防守方的策略：$u^D(\mathbf{x}^*, g(\mathbf{x}^*)) \geq u^D(\mathbf{x}^*, a), \forall a \in f(\mathbf{x}^*)$。

基于基础安全博弈模型，很多之前的工作都探索了模型均衡的求解算法，包括基于混合整数线性规划的 DOBSS，以及基于紧凑表示混合策略的 ERASER，ORIGAMI 算

法等。接下来简单介绍这三个算法。值得注意的是，如引言中介绍，该模型与相应算法均假设各个目标都具有独立的收益值，攻击方的最终目的是攻击一个目标。同时，该模型也假设目标的收益值是不随时间变化的。因此，在面对许多现实问题时，该模型都不能直接发挥作用，用于求解该模型的算法也不再适用。本文第四至七章均首先基于该模型，针对具体问题，进行新模型的建设，再根据新的安全博弈模型，设计求解算法。

2.2 安全博弈基础算法

2.2.1 DOBSS

DOBSS 的提出主要目的在于求解贝叶斯纳什博弈。在这类博弈中，攻击方可能存在多种“类型”，防守方只知晓各个类型的攻击方的分布情况，但不能确定在一次具体的攻防交手中会遇到哪个类型的攻击方。令 L 表示防守方类型的集合，令 $l \in L$ 表示其中的一个类型，令 p_l 表示攻击方属于该类型的概率。各个类型的攻击方策略空间均相同，即，可攻击各个目标中的一个，不同点在于各个类型对应的收益情况。给定一个防守方的混合策略和攻击方纯策略的策略组合 $(\mathbf{x}, a)$，令 $u_l^D(\mathbf{x}, a)$ 表示给定此策略集合，当遭遇的攻击方为类型 l 时，防守方的收益。类似的，令 $u_l^A(s, a)$ 表示在此情况下类型为 l 的攻击方收益。解决此类问题的一个常规思路即使用 Harsanyi 变换 [43]，将贝叶斯博弈转化为不完全信息博弈，再求解该博弈。仍然假设只有一个目标，假设有两种攻击类型，每种攻击类型出现的概率均为 0.5，每种攻击类型都会选择攻击这个目标。又假设两种攻击类型所对应的收益表如表2.2所示，那么，在经过 Harsanyi 变换后，即可得到如表2.3所示的新收益表。当防守方选择保护目标时，假如遇到第一类攻击方，那么防守方收益为 $R_1^D = 5$，假如遇到第二类攻击方，那么防守方收益为 $R_2^D = 20$，由于两种类型出现概率相同，那么防守方的期望收益即为 $u^D = 0.5 * 5 + 0.5 * 20 = 12.5$。类似的，可以得出当防守方不保护这个目标时的期望收益。

类型 1	保护	不保护
防守方	$R_1^D = 5$	$P_1^d = -20$
攻击方	$P_1^A = -10$	$R_1^A = 30$
类型 2	保护	不保护
防守方	$R_2^D = 20$	$P_2^d = 5$
攻击方	$P_2^A = 0$	$R_2^A = 10$

表 2.2 两种攻击方类型收益示例

直观上讲，在求解贝叶斯博弈时，与非贝叶斯博弈相同，每一个类型的攻击均会根据自身收益情况与防守方的策略做出对自身最有利的回应。不同点在于，防守方的最

	保护	不保护
防守方	$u^D = 0.5 * 5 + 0.5 * 20 = 12.5$	$P^d = 0.5 * (-20) + 0.5 * 5 = -7.5$
攻击方 1	$P^A = -10$	$R^A = 30$
攻击方 2	$P^A = 0$	$R^A = 10$

表 2.3　Harsanyi 变换后收益表

优回应此时在于最优化自身在面对各个类型的攻击方时的期望收益，即解决

$$\max_{\mathbf{x}} \sum_{s \in \mathcal{S}} \sum_{l \in L} p_l u_l^D(\mathbf{x}, g^l(\mathbf{x})) x_s. \tag{2-5}$$

其中 $g^l(\mathbf{x})$ 是类型 l 的攻击方在防守策略 $\mathbf{x}$ 时的最优回应策略。

[88] 中研究了此类安全博弈问题。为求解此类问题的均衡解，该工作首先提出了一种非线性规划方法如下。其中 q_{la} 是一个二值指示变量，用于表示对于类型为 l 的攻击方，策略 a 是不是一个最优回应。M 是一个极大的数。

$$\max_{\mathbf{x},q,c} \sum_{s \in \mathcal{S}} \sum_{l \in L} \sum_{a \in \mathcal{A}} p_l u_l^D(\mathbf{x}, a) x_s q_{la} \tag{2-6}$$

$$\sum_{s} x_s = 1 \tag{2-7}$$

$$\sum_{a} q_{la} = 1 \tag{2-8}$$

$$0 \le (c^l - \sum_{s \in \mathcal{S}} x_s u_l^A(s, a)) \le (1 - q_{la}) M \tag{2-9}$$

$$x_i \in [0, 1], q_{la} \in \{0, 1\}, c^l \in \mathcal{R} \tag{2-10}$$

方程 (2-7) 用于约束防守方混合策略的合理性。方程 (2-8) 用于约束每一种类型的攻击方均选择一个策略作为回应。方程 (2-9) 迫使各个类型的攻击方均选择对自己最为有利的策略，其中 c^l 用于记录选取此策略时类型为 l 的攻击方的收益值。这个规划算法的约束函数是非线性非凸的，因此难以求解。为解决此问题，令 $z_{la}^s = x_s q_{la}$，[88] 证明了上述非线性规划等价于以下的混合整数线性规划算法，称作 DOBSS。

$$\max_{q,z,c} \sum_{s \in \mathcal{S}} \sum_{l \in L} \sum_{a \in \mathcal{A}} p_l u_l^D(\mathbf{x}, a) z_{la}^s \tag{2-11}$$

$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} z_{la}^s = 1 \tag{2-12}$$

$$\sum_{a \in \mathcal{A}} z_{la}^s < 1 \tag{2-13}$$

$$q_{la} \le \sum_{s \in \mathcal{S}} z_{la}^s \le 1 \tag{2-14}$$

$$\sum_{a \in \mathcal{A}} q_{la} = 1 \tag{2-15}$$

$$0 \leq (c^l - \sum_{s \in \mathcal{S}} u_l^A(s,a) \sum_{a \in \mathcal{S}} z_{la}^s) \leq (1 - q_{la})M \tag{2-16}$$

$$\sum_{a \in \mathcal{A}} z_{la}^s = \sum_{a \in \mathcal{A}} z_{1a}^s \tag{2-17}$$

$$z_{la}^s \in [0,1], q_{la} \in \{0,1\}, c^l \in \mathcal{R} \tag{2-18}$$

DOBSS 在求解贝叶斯博弈均衡时，效率远高于基于 Harsanyi 变换的求解法。而对于非贝叶斯博弈的情形，也可看做是贝叶斯博弈中只有一种攻击方类型的特殊情形而使用 DOBSS 求解。因此，DOBSS 在被提出以后广受关注，被此后的多项工作作为基线算法。

2.2.2 ERASER

随着需要求解的安全博弈规模越来越大，DOBSS 已无法解决大规模的问题。因此，[55] 中研究了通过紧凑表示防守方策略，减少问题变量数量，从而有效求解大规模安全博弈均衡解。任何一个防守方的混合策略 $\mathbf{x}$，都可由一个等价的覆盖向量 $\mathbf{c}$ 表示。在有 n 个目标的博弈中，覆盖向量包含 n 个元素，其中每个元素 c_i 表示一个目标 i 被保护的概率。具体来讲，

$$c_i = \sum_{s \in \mathcal{S}} x_s \Phi_i(s). \tag{2-19}$$

其中 $\Phi_i(s)$ 是一个二值指示函数，用于指示在防守方的纯策略 s 中目标 i 是否被保护。那么，给定一个防守方的覆盖向量 $\mathbf{c}$，给定任一目标 i，如果攻击方攻击该目标，那么他的期望收益即为

$$u^A(\mathbf{c}, i) = c_i P_i^A + (1 - c_i) R_i^A. \tag{2-20}$$

类似的，防守方的收益则为

$$u^D(\mathbf{c}, i) = c_i R_i^D + (1 - c_i) P_i^D. \tag{2-21}$$

于是，求解最优混合策略的问题即被转化为了求解最优覆盖向量的问题，可通过如下混合整数线性规划 ERASER 得到。

$$\max \quad d \tag{2-22}$$

$$\sum_i a_i = 1 \tag{2-23}$$

$$\sum_i c_i \leq m \tag{2-24}$$

$$d - u^d(\mathbf{c}, i) \leq (1 - a_i)M \tag{2-25}$$

$$0 \leq k - u^a(\mathbf{c}, i) \leq (1 - a_i)M \tag{2-26}$$

$$a_i \in \{0,1\}, c_i \in [0,1] \tag{2-27}$$

方程 (2-23) 要求攻击方攻击一个目标。方程 (2-24) 确保覆盖向量对应的混合策略的可行性。当防守方最多可使用 m 个防守资源时，所有资源被保护的概率之和不大于 m。方程 (2-25) 与规划的目标函数共同保证了防守方选择最优的策略，d 记录使用最优策略时防守方的收益。方程 (2-26) 则迫使攻击方选择最优的目标进行攻击，k 记录在攻击最优目标时攻击方能够获得的收益。[55] 中证明了可在多项式时间内将覆盖向量 $\mathbf{c}$ 转化为与之相应的混合策略 $\mathbf{x}$，该混合策略即为博弈中的强斯塔克伯格均衡策略。由于覆盖向量仅包含 n 个变量，当博弈规模较大时，远小于防守方所有纯策略的数量。因此，在求解大规模非贝叶斯博弈时，ERASER 的效率远高于 DOBSS。

2.2.3 ORIGAMI

基于紧凑表示防守方策略的思路，[55] 中进一步提出了比 ERASER 更为高效的 ORIGAMI 算法。首先，令 $\Gamma(\mathbf{c})$ 表示在给定防守方的覆盖向量 $\mathbf{c}$ 时，所有能够使得攻击方获得最优收益的目标的集合，将该集合成为一个攻击集合。令 i^* 表示在强斯塔克伯格均衡下攻击方最终选择攻击的目标。令 $\hat{u}^D(\mathbf{c})$ 和 $\hat{u}^A(\mathbf{c})$ 分别表示但 $\mathbf{c}$ 对应强斯塔克伯格均衡解中防守方的混合策略时，防守方与攻击方在均衡策略下的收益情况。ORIGAMI 算法基于以下三个命题。

命题 1 *其他条件不变，增加 $\Gamma(\mathbf{c})$ 之外的目标被保护的概率，不会影响 $\hat{u}^D(\mathbf{c})$ 和 $\hat{u}^A(\mathbf{c})$ 的值。*

这是因为只要 $\Gamma(\mathbf{c})$ 不改变，攻击方在均衡状态下的攻击目标即不改变，仍为 $\Gamma(\mathbf{c})$ 内的某个目标。而该目标被保护的概率也没有改变，因此双方的均衡收益都不变。

命题 2 *如果 $\Gamma(c) \subset \Gamma(\mathbf{c}')$，并且有 $\forall i \in \Gamma(c), c_i = c'_i$，那么 $\hat{u}^D(\mathbf{c}) \leq \hat{u}^D(\mathbf{c}')$。*

也就是说，增加攻击集合内目标的个数不会使得防守方的收益变差。这主要由于强斯塔克伯格均衡下，攻击方在面对效应相同的多个目标时选择对防守方最有利的目标，而攻击集合内的所有目标对于攻击方来说都是效用相同的。

命题 3 *如果 $\hat{u}^D(\mathbf{c}) = k$，那么对于任一满足 $P_i^D > k$ 的目标 i，有 $c_i \geq \frac{k - P_i^D}{R_i^D - P_i^D}$。*

这是由于攻击集合内的目标均有 $k = c_i R_i^D + (1 - c_i) P_i^D$。所以如果 $P_i^D < k$，那么目标 i 不在攻击集合内。否则目标应在攻击集合内，有 $c_i \geq \frac{k - P_i^D}{R_i^D - P_i^D}$。

[55] 中证明了算法1所示了 ORIGAMI 算法所求出的覆盖向量对应一个强斯塔克伯格均衡解中的防守方混合策略。该算法的基本思想基于前文所述的三个定理，意图通过直接求出攻击集合的方式得到最优的防守策略。算法首先认为攻击集合仅包含具有最大的 P_i^D 的目标，然后逐步扩张集合，每次扩张时都增加集合内所有目标被保护的概

Algorithm 1: ORIGAMI

targets ← 按 P_i^D 将目标 i 降序排列;
payoff[i] ← P_i^D, coverage[i] ← 0, left ← m, next ← 2, covBound ← $-\infty$;
while $next \le n$ **do**
    addedCov[i] ← $\frac{\text{payoff[i]}-P_i^D}{R_i^D-P_i^D}$ - coverage[i];
    if $coverage[i] + addedCov[i] \ge 1$ **then**
        covBound ← Max(covBound, R_i^D)
    if $covBound \ge -\infty$ OR $\sum_i addedCov[i] \le left$ **then**
        break
    coverage[i] += addedCov[i], left -= $\sum_i$addedCov[i], next++ ;
ratio[i] ← $\frac{1}{R_i^D-P_i^D}$, coverage[i] += $\frac{\text{ratio[i]}\cdot\text{left}}{\sum_i \text{ratio[i]}}$;
if $coverage[i] \ge 1$ **then**
    covBound ← Max(covBound, R_i^D)
if $covBound \ge -\infty$ **then**
    coverage[i] ← $\frac{\text{covBound}-P_i^D}{R_i^D-P_i^D}$

率。当防守方的资源已被用完，或者攻击集合内的某个资源被保护的概率达到 1 时，算法停止。攻击集合之外的所有目标均不需要被保护。

2.3 安全博弈应用案例

2.3.1 机场检查点的设置及巡逻调度

洛杉矶国际机场（LAX）是美国最大的机场之一，每年的旅客流量在 7000 万左右。洛杉矶警方采取不同的措施来保护机场，包括在进出机场的要道设置车辆检查点，派遣警察部队和警犬在航站楼巡逻和检查乘客行李。安全博弈论应用主要考虑以下两个方面的问题：（1）如何确定车辆检查点的设立地点和工作时间？由于警方所拥有的资源不足以在所有的道路上全天候进行检查，而每天的检查地点、检查时间均能被潜在的犯罪分子观察到，因此有效地配置检查点至关重要。（2）如何制定警犬在洛杉矶国际机场 8 个航站楼之间的巡逻路线？这 8 个航站楼有不同的特性，如大小、载客量、客流量、国际与国内航班数量。这些因素导致 8 个航站楼有不同的风险评估结果。同时，警犬队伍的数量也不足以同时有效覆盖所有的航站楼。因此，采取最佳方案分配资源提高效率才能降低因固定的部署模式带来的风险。

基于贝叶斯-斯塔克伯格博弈论的 ARMOR 系统 [90] 被用于规划洛杉矶国际机场检查点的设置以及警犬的巡逻路线。以设置检查点为例，假设洛杉矶国际机场有 n 条进入

机场的道路，警方在这 n 条道路上设置 m（m<n）个检查站，恐怖分子可以从任意一个入口进行攻击。ARMOR 系统考虑不同类型的攻击者具有不同的收益函数（不同类型代表各种具有不同能力和偏好的恐怖分子），采用 DOBSS 算法计算安全部门的最佳资源分配战略 [88]。该系统于 2007 年 8 月成功地部署在洛杉矶国际机场，并一直使用至今。

(a) (b)

图 2.1 ARMOR：洛杉矶机场安保

2.3.2 空中警察调度

美国联邦空中警察署（FAMS）负责分配空中警察到从美国发出的航班，以阻止潜在的攻击。空中警察的分配问题比 ARMOR 系统所面对的保护机场问题更具挑战：FAMS 每天需要将有限数量的空中警察分配到成千上万的商业航班，可行的分配策略的数目要远远多于分配检查点或巡逻警犬。同时，空中警察的分配必须遵守各种类型的限制条件，如每一名空中警察需要飞回其基地，并满足起飞、降落、休息等很多时间上的约束。因此，找出满足所有限制条件的最优随机调度策略是一项非常困难的任务。

(a) (b)

图 2.2 IRIS：空中警察调度

在此背景下，TEAMCORE 研究小组开发了 IRIS 系统 [55]，并于 2009 年 10 月开始为所有国际航班的空中警察进行调度。由于纯策略的资源分配数量随航班数量以及空中警察数量呈指数增长，DOBSS 算法无法求解最优空中警察调度策略。IRIS 系统使用更快的算法 ERASER-C 产生每天数千架商业航班的空中警察调度方案。IRIS 系统同时使用基于属性的偏好启发方法来确定斯塔克伯格博弈模型的收益函数。

2.3.3　海岸警卫队的巡逻

(a)

(b)

图 2.3　PROTECT：海岸警卫队巡逻

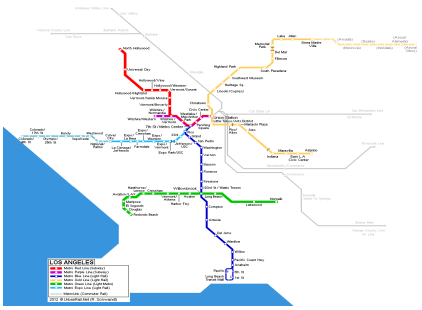
美国海岸警卫队（USCG）的任务包括维持海上安全、港口安全以及内河航道的安全。由于恐怖主义和毒品走私的威胁，这些地方面临的风险日益增加。美国海岸警卫队通过巡逻的方式来保护港口的基础设施。然而，有限的安全资源使海岸警卫队无法随时随地保护所有重要设施，也使得攻击者有了可乘之机。为了协助美国海岸警卫队的资源分配，TEAMCORE 研究小组设计了基于斯塔克伯格博弈模型的 PROTECT 系统[98]。开发 PROTECT 系统的目的是帮助美国海岸警卫队在执行保护港口、水路、和海岸安全（合称 PWCS）时提高效率。对 PWCS 的巡逻着眼于保护重点设施，由于资源所限，任何设施都无法获得全天候的保护，因此对资源配置的优化就变得至关重要。

PROTECT 系统不仅考虑了攻击者的观测能力，还考虑了不同的安保资源或巡逻活动可能对目标产生不同的保护程度。PROTECT 输出美国海岸警卫队巡逻的日程表，包括什么时候开始巡逻，每次巡逻经过哪些目标区域，以及在每个目标区域里执行的巡逻活动。这个系统有很多创新点。首先，它不像以前的系统那样假设攻击者是完全理性的；第二，为了提高效率，系统在寻找均衡和最优解时采取了更加紧凑的方式来表示攻击者的策略空间；第三，PROTECT 系统通过真实的数据来评价其性能。PROTECT 模型正被拓展到纽约的港口，并且可能被更多的美国港口采用。

2.3.4　城市交通系统安全

美国一些城市的交通系统要求乘客购票乘车，却没有采取强制措施。以洛杉矶地铁为例，它每天运送约 30 万乘客，逃票带来的损失预计每年为 560 万美元。洛杉矶警察局（LASD）雇佣一些工作人员在列车上或者站台上检票。由于巡逻检票的工作人员数量较少，不可能覆盖所有的列车和站台，因此洛杉矶警察局需要一些机制来设计检票人员的巡逻路线。如果巡逻检票的调度策略有比较固定的模式，那么逃票者可能会观察到这个模式并且利用它来逃票。目前洛杉矶警察局依赖人工制定巡逻日程。但是由于人工制定的调度策略通常有固定模式，而且日程的制定需要考虑很多复杂的因素，比如列车运行时间、发车间隔、日程长度等，制定调度策略对人来讲负担很重。

(a)



(b)

图 2.4　TRUSTS：洛杉矶地铁逃票检测

TRUSTS 系统 [130] 将地铁系统巡逻问题抽象成一个领导者和多个跟随者的斯塔克伯格博弈。领导者（洛杉矶警察局）采用混合策略，跟随者（可能逃票的乘客）观察到这个策略并决定是否买票。由于运输系统的复杂性使得可能的巡逻策略数量呈指数级增长，这给计算最优巡逻策略提出了很大挑战。为了解决此问题，TRUST 使用了紧凑的表达方式同时考虑到了时间和空间结构。洛杉矶警察局目前正在洛杉矶地铁上测试 TRUST 系统，计划根据系统产生的日程来安排巡逻，并检测收入是否增长以确定逃票者是否减少。

第三章　非独立目标安全博弈

3.1　引言

投票过程的公平公正对社会公平稳定至关重要。对于投票过程公正性的一个威胁在于，一些不法分子可能会通过一些手段操纵选举结果。例如，在 2013 年巴基斯坦大选的选举日发生了一系统爆炸事件，造成超过 30 人死亡，超过 200 人受伤的恶劣影响 [96]。在 2010 年的斯里兰卡选举中，共发生了 84 次严重的干扰投票站事件以及 202 次其他干扰事件 [13]。随着电子投票兴起，通过发动对服务器的攻击操纵选举结果的事件也时有发生 [6,116]。

为使得选举免受操纵，一些科学家开始研究一些投票机制下操纵选举的难度 [7]。在这一系列研究中，给定一个投票机制，如果使用某种手段操纵选举是 NP 难的，那么这个投票机制对于这种操纵手段来说就是可靠的，否则这个投票机制在这种操纵手段下就是不可靠的。一些投票机制已被证明对于一些操纵手段是可靠的，但是简单常见的多数制却在很多操纵手段下都不可靠 [7,44]。另外，对于操纵手段的研究大多聚焦在操纵的对象为单个选民，而对于保障选举的研究仅在于设计可靠的投票机制 [28,45]。虽然这样的研究对于深入理解投票机制大有裨益，但其也存在一些局限：首先，操纵选举不仅仅能够在单个选民的粒度上发生。更有可能的的情况是，不法分子的攻击目标是选民群体。极端的例子包括攻击投票站，使得依赖该站点进行投票的选民都不能投票。其他例子还包括通过改变投票规则，如投票的时间、地点等，使某些群体投票困难。第二，NP 难并不能够保证选举制度的可靠性。事实上，很多 NP 难的问题在现实中都可以通过高效的算法求得较好的解 [123]。从防御的角度考虑保障选举，例如配置安保资源来保护选民群体，尚未被以往的工作考虑过。

为解决这些局限，本章考虑通过配置安全资源保护选民群体，以防止通过攻击选民群体来操纵选举结果的行为。本章考虑破坏性攻击和建设性攻击者两种攻击方式。破坏性攻击的目标是使得原本的赢家不能赢得选举。建设性攻击的目标是使得原本不能获胜的某个候选人赢得选举。虽然此前工作 [27] 证明了在多数制下，基于选民群体的建设性攻击是 NP 难的，本章证明了破坏性攻击可以在多项式时间内完成。本章利用斯塔克伯格博弈模型对抵制这两种攻击的问题进行建模——用零和博弈描述抵制破坏性攻击，用非零和博弈描述抵制建设性攻击。此模型下防守方（如选举的组织方，纪检部门）的目的是保证原本能够赢得选举的候选人依然能够赢得选举。本章证明了不管对于破坏性攻击还是建设性攻击，防守方欲通过保护若干个选民群体，使得攻击完全不可能成功，都是 NP 难的。对于抵制破坏性攻击的一般性情形，本章提出了一种基于

Double Oracle 框架的算法来求解防守方的最优策略。本章证明了对于防守方和攻击方双方的 Oracle 问题都是 NP 难的，为双 Oracle 问题提出了混合线性规划算法求出 Oracle 的最优解，并提出启发式算法以提高整个算法的可扩展性。对于抵制建设性攻击的情形，本章首先提出了一种启发式算法，能够在某些情形下，在多项式时间内求出防守方的最优策略。随后指出，虽然该问题被建模为非零和博弈，但却能够通过求解零和博弈来获得该问题的一个近似最优解。事实上，实验证明该近似最优解在极多情况下都是最优解。综上，本章的主要贡献如下：

- 保障选举不受基于选民群组的操纵的新模型。
- 基于选民群组的破坏性操纵的多项式时间算法。
- 保障选举进行，抵制破坏性操纵和建设性操纵的复杂度分析。
- 一个具有高可扩张性的基于 Double Oracle 框架的算法用以求解防守方抵制破坏性攻击的最优策略。
- 求解防守方抵制建设性攻击的启发式算法和近似算法。

3.2 模型

本章的模型考虑了基于选民群组的破坏性攻击和建设性攻击。攻击方若进行破坏性攻击，其目标在于使得原本能够获胜的候选人不能获胜。攻击方若进行建设性攻击，其目标则为使得原本不能获胜的某候选人获胜。本章聚焦在多数制的选举制度，即得票最多的一个候选人获得胜利。

在本章的模型中，攻击都是基于选民群组的。本章首先假设攻击方和防守方都知晓各个选民群组的投票情况（如，一个 10 人组成的群组，有 6 人会投票给候选人 1，4 人会投票给候选人 2。该假设并不需要知晓群组中每个个体的投票情况。）这个假设在随后的章节中得以放松。更加正式的定义为，假设一个集合 I 包含 n 个互不覆盖的选民群组。一个候选人的集合 C，所有选票均投向该集合内的某个候选人。对于每个选民群组 $i \in I$，令 t_{ic} 表示该群体内选民投给候选人 c 的总票数。令 $t_c = \sum_i t_{ic}$ 表示某候选人 $c \in C$ 在所有群组中的得票总数。令 $\omega \in C$ 表示原本应该获胜的候选人，即 $\omega = \arg\max_c t_c$。攻击方可以攻击 k 个选民群组（I 的一个大小为 k 的子集），使得这些选民群组内的选民都不能投票。一旦一个群组被成功攻击，来自该群组的选票即不复存在，所有候选人的得票总数都有可能发生变化。在破坏性攻击下，攻击方的目标是使得 ω 不能获胜。在建设性攻击下，攻击方的目标是使得某个候选人 $v \in C(v \neq \omega)$ 最终获胜。

本章利用斯塔克伯格博弈模型对抵制攻击的问题进行建模。在该模型下，防守方有 m 个安保资源，利用这些资源，他可以保护 m 个选民群体。攻击方可以攻击 k 个选民群体。正式的定义可表示为，令向量 s 表示防守方的一个纯策略，其中 $s_i \in \{0,1\}$ 用于指示选民群体 i 是否被保护。另向量 a 表示攻击方的一个纯策略，其中二值元素 a_i 用来表示选民群体 i 是否被攻击。只有当 $s_i = 0$ 并且 $a_i = 1$ 时，一个选民群组才能够被成

功攻击。否则该群组安全，群组内的选票维持不变。另 $\mathcal{S}$ 和 $\mathcal{A}$ 分别表示防守方和攻击方的策略空间。给定一组纯策略 (s,a)，另 $W^c(s,a)\in\{0,1\}$ 表示候选人 c 在这组策略下是否能够赢得选举。值得注意的是，对于任何给定的策略二元组，$W^c(s,a)$ 都可以通过数未被攻击的选民群体中的总票数的方式被快速计算出。那么，防守方的收益即可表示为 $W^\omega(s,a)$。对于破坏性攻击，攻击方的收益为 $-W^\omega(s,a)$。对于建设性攻击，假设攻击方希望候选人 υ 获得胜利，那么他的收益即可表示为 $W^\upsilon(s,a)$。

该模型允许防守方采用随机化的策略，那么攻击方只能观测到每个纯策略被使用的概率，而无法得知当时正在被使用的策略是哪一个。这是由于攻击方在攻击之前往往会花费大量时间进行观测学习，研究各个纯策略被使用的概率，再据此确定攻击的目标。而由于实施攻击也需要较多的准备工作，因此攻击目标确定后，也难以因为某次具体的安保情况再进行快速改动 [104]。由于攻击方选择最优的策略，因此只需考虑攻击方的纯策略。令向量 $\mathbf{x}$ 表示防守方的混合策略，其中 x_s 表示纯策略 $s\in\mathcal{S}$ 被使用的概率。给定一个防守方的混合策略 $\mathbf{x}$，给定一个攻击方的纯策略 a，一个候选人 c 能够赢得选举的概率即可表示为

$$P^c(\mathbf{x},a)=\sum_{s\in\mathcal{S}}x_sW^c(s,a). \tag{3-1}$$

那么，防守方的期望收益即为 $P^\omega(\mathbf{x},a)$。对于破坏性攻击，攻击方的期望收益即为 $-P^\omega(\mathbf{x},a)$。对于建设性攻击，攻击方的期望收益则为 $P^\upsilon(\mathbf{x},a)$。对于破坏性攻击，由于博弈为零和，那么其斯塔克伯格均衡集即与纳什均衡解等价 [60]。对于建设性攻击，博弈不再为零和。因此两种攻击对应的均衡解不同，也不能用同样的方式求解。

3.3 基于群组的攻击复杂度

本部分研究研究群组的攻击的复杂度。前人的工作揭示出，在多数制的选举制度下，对于基于单个选民的攻击，不管是破坏性攻击还是建设性攻击，都可以在多项式时间内达到目的 [7,44]。然而，基于群组的攻击会使得攻击问题变得复杂。事实上，之前的工作指出 [27]，即使在多数制的选举制度下，基于群组的建设性攻击也是 NP 难的。有趣的是，本部分证明了基于群组的破坏性攻击在多数制的选举制度下依然能够在多项式时间内完成。这个结果使得关于攻击行为的研究更具有普遍适用性 [44]。

对于破坏性攻击，只要攻击方能够保证在 k 个选民群组被攻击后，存在任何一个候选人 $c\in C$ 得到至少和 ω 相同的票数，那么他的目标就可以达到。所以，攻击方可以每次考虑一个候选人 c，检查是否可以通过攻击 k 个群组以使得这个候选人胜过 ω。重要的是，一旦给定 $c\in C$，是否能够使得 c 战胜 ω 可以被快速的确定：攻击方只需尝试攻击 ω 的得票数超过 c 最多的 k 个选民群体。

令 $\mathbf{t}^{c-\omega}=\langle t_i^{c-\omega}:i\in I\rangle$ 表示一个向量，其中 $t_i^{c-\omega}=t_{ic}-t_{i\omega}$，也即表示在选民群组 $i\in I$ 内，候选人 c 比 ω 多出的票数。（若为负值，则说明该组内 c 得票少于 ω。）对于向

量 $\mathbf{t}^{c-\omega}$，定义 $sum(\mathbf{t}^{c-\omega}) = \sum_i t_i^{c-\omega}$。那么，$sum(\mathbf{t}^{c-\omega})$ 就是 c 和 ω 的得票数总差值。举例来讲，令 $\mathbf{t}^{c-\omega}$ 为 $\langle -3, -2, 1\rangle$。这意味着在前两个选民群组中，ω 得票数超过 c，但在第三个群组中，c 的得票数比 ω 多 1。如果攻击方能够攻击两个群组，那么只需攻击前两个，他就能使得 c 最终以一票领先于 ω，从而阻止 ω 获得胜利。

下述命题显示，事实上，检查攻击具有最小 $t_i^{c-\omega}$ 值的 k 个群组就足以得知 c 是否有希望赢过 ω。为表述的简洁性，令 $\mathbf{t}^{c-\omega} \setminus k$ 表示在具有最小 $t_i^{c-\omega}$ 值的 k 个群组被攻击后，向量 $\mathbf{t}^{c-\omega}$ 剩余的部分。

命题 4 给定一个候选人 $c \in C$，攻击 k 个选民群组即能使得 $t_c > t_\omega$ 当且仅当 $sum(\mathbf{t}^{c-\omega} \setminus k) > 0$。

证明 $\Leftarrow$：如果 $sum(\mathbf{t}^{c-\omega} \setminus k) > 0$，那么根据定义 $\mathbf{t}^{c-\omega} \setminus k$，可知有 $t_c > t_\omega$。

$\Rightarrow$：如果攻击具有最小 $t_i^{c-\omega}$ 值的 k 个群组仍然使得 $sum(\mathbf{t}^{c-\omega} \setminus k) < 0$，那么不可能通过攻击其他选民群组集合 $G \subseteq I$ 而使得 $t_c > t_\omega$。这是由于一旦选择了 k 个具有最大 $t_{i\omega} - t_{ic}$ 值的群组，即向 $sum(\mathbf{t}^{c-\omega})$ 增加了最大的 $\sum_i t_{i\omega} - t_{ic}$。如果剩余的选票仍然为负，$c$ 就不可能得到比 ω 更多的选票。 □

破坏性攻击的具体方法在算法 2中描述。

Algorithm 2: 基于群组的破坏性攻击

```
1 for c ∈ C^{-ω} do
2     t^{c-ω} \ k ← 将 t^{c-ω} 升序排列，攻击前 k 个元素;
3     if sum(t^{c-ω} \ k) > 0 then
4         返回 攻击与删除元素相对应的选民群组;
5 return 无可行攻击方案;
```

对于候选人 $c \in C \setminus \{\omega\}$（定义为 $C^{-\omega}$），第 1 行检查是否存在一种攻击方法能够使得 c 赢过 ω (基于命题 4)。如果对于任何 $C^{-\omega}$ 中的候选人都不存在这样的攻击方法，那么破坏性攻击就不可能成功。算法 2的复杂度为 $O(|C|n \log n)$，这使得本节得出以下定理：

定理 1 通过攻击 k 个选民群组进行破坏性攻击的策略可在 $O(|C|n \log n)$ 时间复杂度内实现。

3.4 抵制基于群组的攻击复杂度

由于前人的工作尚未从防御角度考虑过保障选举安全的问题，抵制基于群组的攻击的复杂度也并未被研究过。本节指出，不管是对于破坏性攻击还是建设性攻击，要完全抵制攻击都是 NP 难的。

3.4.1 抵制破坏性攻击复杂度

定义 2 (Hitting Set 问题) 集合 G, 集合 U 由 G 的子集 $\hat{G}$ 组成。**问题：** 是否存在一个“hitting set” $G' \subseteq G$，$|G'| = m$，使得 $\forall \hat{G} \in U, G' \cap \hat{G} \neq \emptyset$。

定理 2 检测 m 个安保资源是否足够抵制破坏性攻击是 NP 完全的。

证明 易知该问题属于 NP，接下来通过把问题规约为 Hitting Set 问题来证明该问题是 NP 难的。具体来讲，本证明过程指出，对于任意一个 Hitting Set 问题，可构建一个包括 n 个选民群组的选举，使得存在一个 hitting set G' 当且仅当 m 个安保资源足够在这个选举中抵制破坏性攻击。也就是说，攻击方不可能通过攻击未被保护的 $n-m$ 个选民群组而使得初始的赢家不能获胜。

给定 Hitting Set 问题，选举构建方式如下。有 $n = |G|$ 个选民群组，有 $|U|+1$ 个候选人。一个 $i \in G$ 对应一个选民群组。一个 $\hat{G} \in U$ 可被认为是除 ω 之外的一个候选人的标签。对于标签是 $\hat{G}$ 的候选人 c，给定任一选民群组 i，如果 $i \in \hat{G}$，那么即设定 $t_i^{c-\omega} = -1$，也就是说，候选人 c 比候选人 ω 在群组 i 中的得票少一票。否则设定 $t_i^{c-\omega} = 0$，也就是说候选人 c 和 ω 在群组 i 中得票数相同。假定攻击方最多可攻击 $k = n-m$ 个选民群组。

$\Leftarrow$ 方向：如果存在一个安保策略保护了 m 个选民群组，也就是说，$G' \subset G$，$|G'| = m$，以使得攻击方不能达成破坏性攻击的目标，那就说明对于任一候选人 c，其标签可能是 $\hat{G} \in U$，至少存在一个满足 $t_i^{c-\omega} = -1$ 的选民群组 i 被保护了，也就是说，$G' \cap \hat{G} \neq \emptyset$。这是由于，如果存在一个候选人 c，所有满足 $t_i^{c-\omega} = -1$ 的选民群组都未被保护，那么攻击方就可以通过攻击这些群组使得 c 最终的得票与 ω 持平。因此，被保护的群组应满足 $\forall \hat{G} \in U, G' \cap \hat{G} \neq \emptyset$，恰构成一个 Hitting Set。

$\Rightarrow$ 方向：给定一个 Hitting Set$G' \subset G$，防守方的一种策略是保护所有 $i \in G'$ 的选民群组。这样，即使攻击方可以攻击所有未被保护的选民群组，任一候选人 $c \in C^{-\omega}$ 的得票仍至少比 ω 少一票。因此，不存在攻击方法能够达成破坏性攻击的目的。 □

3.4.2 抵制建设性攻击复杂度

定义 3 (Set Cover 问题) 集合 G，集合 U 由 G 的子集 $\hat{G}$ 组成。**问题：** 是否存在一个“set cover” $U' \subseteq U$，$|U'| = r$, 使得 U' 的合集为 G。

定理 3 检测 m 个安保资源是否足够抵制建设性攻击是 NP 难的。

证明 本证明通过将原问题归约为 Set Cover 问题来证明 NP 难。具体的，给定一个 Set Cover 问题，可以构造一个选举问题，使得 m 个资源足以在该选举中抵制建设性攻击当且仅当存在 set cover。

令 $|G| = l$，$|U| = n$，构造一个选举包含 $l+1$ 个候选人，$n+r+2m$ 个选民群组。除第 $l+1$ 个候选人，每一个候选人均对应一个 $c \in G$。U 中的元素对应前 n 个选民群组，

也就是说，每个 $\hat{G} \in U$ 均可看做一个选民群组的标签。防守方有 m 个安保资源。对于前 n 个选民群组，如果该群组的标签为 $\hat{G}$，那么该群组内的选民对每个 $c \in G \setminus \hat{G}$ 投一票，对其他候选人不投票。对于第 $n+1$ 个群组到第 $n+m$ 个群组，每个群组对每个 G 内的候选人投一票。对于余下的 $r+m$ 群组，每个群组仅包含一个选民，他将票投向候选人 υ。攻击者的目标是使得第 $l+1$ 个候选人获胜，也就是说，第 $l+1$ 个候选人即为 υ。如果 υ 不是最初的赢家，那么直接假设攻击方可攻击最多 $k = n - r$ 个选民群体。否则，首先选取一个候选人 $c^* \in G$，该候选人在所有 G 内的候选人中得票最多，然后增加一个新组，共向该候选人投 $t_\upsilon - t_{c^*} + 1$ 票，不向其他候选人投票[①]。然后假设攻击方最多可攻击 $k = n - r + 1$ 个群组。例如，假设 $G = \{1,2,3\}, U = \{\{1,2\},\{2\},\{3\}\}$，$r = 2$，以及 $m = 2$。那么共有 4 个候选人和 10 个选民群组。由于在前九组中 υ 即为赢家，因此第十组被加入。各群组内对各候选人的投票情况如表 3.1所示。

	1	2	3	υ	标签
群组 1	0	0	1	0	{1, 2}
群组 2	1	0	1	0	{2}
群组 3	1	1	0	0	{3}
群组 4, 5	1	1	1	0	
群组 6 - 9	0	0	0	1	
群组 10	1	0	1	0	

表 3.1　构建选举中的票数示例

接下来，本证明通过证明以下两个子问题，来说明一旦能够确定是否存在一种方法，能够利用 m 个安保资源抵制建设性攻击，就能够确定在 Set Cover 问题中是否存在一个 r set cover。

1. 如果存在一种方法能够利用 m 个安保资源抵制建设性攻击，那么就不存在 r set cover。
2. 如果 m 个安保资源不足以抵制建设性攻击，那么一定存在 r set cover。

子问题 **1**：接下来通过证明，如果存在一个 r set cover U'，那么 m 个资源就足以抵制建设性攻击来证明第一个子问题。首先，如果存在一个 r set cover U'，那么一个使得 υ 最终获胜的攻击策略就是攻击标签为 $i \in U \setminus U'$ 的 $n - r$ 个群组（如果有新增组，也攻击这个新增组）。在余下的 $2k + 2m$ 个群组中，υ 的得票数高过其他任何候选人。这意味着如果防守方想要抵制攻击，至少需要保护一个标签为 $i \in U \setminus U'$ 的群组或者新增组。那么，对于 m 个向每个 G 中的候选人投了一票的群组，他们中至少有一个未被保护。于是，攻击方可以转而攻击这个没有被保护的群组，仍然使得 υ 获胜。以表格中显示的选票数为例。若防守方希望抵制攻击，那么需要保护群组 2。这意味着群组 4 和 5

① 为避免平局，可向得票最多的多个候选人 c^* 中的一个增加一票。增加该票后不影响随后的证明过程。

至少有一个未被保护。于是，攻击方可转而攻击未被保护的群组而达到目的。所以，如果存在一个 r set cover U'，那么 m 个资源就足以抵制建设性攻击来证明第一个子问题。与之等价的，如果存在一种方法能够利用 m 个安保资源抵制建设性攻击，那么就不存在 r set cover。

子问题 **2**：接下来证明，如果不存在 r set cover，那么 m 个安保资源就足够抵制建设性攻击。这是由于，一种能够完全抵制攻击的安保策略是，保护那些每个 G 中的候选人都得到一票的 m 个群组。（在前文的示例中，群组 4 和 5。）攻击方显然不会攻击那些只投票给了 υ 的群组，于是他会攻击那些以 $i \in U$ 为标签的 $n-r$ 个群组（如果有新增组，那么同样攻击新增组。）。由于前提假设是不存在 r set cover，那么在以 $i \in U$ 为标签的 r 个群组中，至少有一个候选人共得到了 r 票，于是这名候选人至少得到了和 υ 相同的票数，υ 就不会获得胜利。于是，如果不存在 r set cover, m 个资源就足以抵制建设性攻击。与之等价的，如果 m 个安保资源不足以抵制建设性攻击，那么一定存在 r set cover。 □

虽然抵制建设性攻击的一般性问题是 NP 难的，但可以证明，在一部分情形下，依然可以在多项式时间内求出能够抵制攻击的安保策略。假设存在某个候选人 $c \in C$，存在某种安保资源配置方法，使得在此安保策略下，无论攻击方如何进行攻击，c 在未被攻击的群组中的得票数总是超过 υ，那么这样的策略被称作 c-占优策略。给定一个候选人 c，可以快速判定是否存在一个 c-占优策略：只需尝试保护 c 得票数超过 υ 最多的 m 个群组，然后查看如果未被保护的群组中 c 得票数超过 υ 最多的 k 个群组被攻击，c 最终的得票数能否超过 υ。

令 $\mathbf{t}^{c-\upsilon} = \langle t_i^{c-\upsilon} : i \in I \rangle$ 表示一个向量，其中 $t_i^{c-\upsilon} = t_{ic} - t_{i\upsilon}$，也就是 c 在选民群组 i 中比 υ 多的票数。对于向量 $\mathbf{t}^{c-\upsilon}$，定义 $sum(\mathbf{t}^{c-\upsilon}) = \sum_i t_i^{c-\upsilon}$。于是，$sum(\mathbf{t}^{c-\upsilon})$ 就是 c 和 υ 得票总数的差值。下述命题指出了确定 c-占有策略是否存在的条件。为表述的简洁性，令 e^c 表示具有第 $m+1$ 大到第 $m+k$ 大的 $t_i^{c-\upsilon}$ 值（并且 $t_i^{c-\upsilon} > 0$）的群组所构成的集合。如果具有第 $m+1$ 大 $t_i^{c-\upsilon}$ 的群组即有 $t_i^{c-\upsilon} < 0$，那么 e^c 是一个空集。令 $\mathbf{t}^{c-\upsilon} \setminus e^c$ 表示，当与 e^c 中元素的群组被攻击之后，向量 $\mathbf{t}^{c-\upsilon}$ 所剩余的部分。例如，假设存在一个候选人 c，$\mathbf{t}^{c-\upsilon} = \langle 10, 5, -5, -3 \rangle$，假设 $m = 1$，$k = 2$，那么 e^c 仅包含与 5 对应的群组 (由于具有第三大 $t_i^{c-\upsilon}$ 的群组有 $t_i^{c-\upsilon} = -3 < 0$)，于是 $\mathbf{t}^{c-\upsilon} \setminus e^c = \langle 10, -5, -3 \rangle$。

命题 5 对于候选人 $c \in C$，存在一个 c-占优策略当且仅当 $sum(\mathbf{t}^{c-\upsilon} \setminus e^c) > 0$。

证明 首先，给定防守方的安保策略是保护 m 个群组，设这 m 个群组构成集合 ψ，为令 υ 赢过 c，攻击方希望通过攻击未被保护的群组，以尽可能压缩 c 所占的优势，即攻击 $e^* = \text{argmax}_{e, e \subset I \setminus \psi, |e| \le k} sum(e)$（不失一般性，此处 $sum(e) = sum(\{t_i^{c-\upsilon} : i \in e\})$）。对于 $\Leftarrow$ 方向，如果 $sum(\mathbf{t}^{c-\upsilon} \setminus e^c) > 0$，一个 c-占有策略就是保护具有最大 $t_i^{c-\upsilon}$ 值的 m 个群组，

于是 $e^* = e^c$，没有攻击策略能够使得 υ 赢过 c。对于 $\Rightarrow$ 方向，如果存在一个 c-占优策略，保护 ψ 中的群组，那么 $sum(e^*) \geq sum(e^c)$，保护具有最大 $t_i^{c-\upsilon}$ 值的 m 个群组一样是一个 c-占优策略，于是 $sum(\mathbf{t}^{c-\upsilon} \setminus e^c) > 0$。 □

基于命题 5，本节提出算法 3，当 c-占优策略存在时，该算法能够在多项式时间内找到一个这样的策略。

Algorithm 3: 贪心启发式算法

```
for c ∈ C do
    t^{c-υ} \ e^c ← 删除从第 m+1 大到第 m+k 大，并且值大于零的 t_i^{c-υ} 所对应的群组;
    if sum(t^{c-υ} \ e^c) > 0 then
        return 保护最大的 m 个 t^{c-υ} 所对应的群组
```

3.5 抵制基于群组的攻击算法

虽然抵制攻击在复杂度层面上是 NP 难的，但这主要是对最坏情形而言，并非所有的现实问题都不可解。本节接下来研究计算最优安保策略的一般性方法。本节将保护选举的问题建模成一个斯塔克伯格博弈。博弈中防守方首先确定一个混合策略，攻击方在完全得知该混合策略后，选取对自己最有利的纯策略。本节采用强斯塔克伯格均衡作为解空间，接下来研究在面对两种攻击时均衡策略的计算方式。

3.5.1 抵制破坏性攻击算法

在破坏性攻击的情况下，防守方与攻击方的目标恰恰相反——防守方希望原本能够赢得选举的候选人仍然获胜，而攻击方希望原本的赢家不再获胜，因此，博弈为零和博弈。由于 SSE 和纳什均衡在零和博弈下等价 [60]，博弈均衡解可用下述线性规划算法求得 [22]。

$$\text{Core-LP}(\mathcal{S}, \mathcal{A}): \quad \max_{\mathbf{x},p} \quad p \tag{3-2a}$$

$$p \leq \sum_{s \in \mathcal{S}} x_s P^{\omega}(s,a), \quad \forall a \in \mathcal{A}. \tag{3-2b}$$

然而，线性规划需要显示列出防守方和攻击方的所有策略。在该问题中，防守方和攻击方的策略数量随着选民群体数量和可用资源数量而呈几何增长，因此线性规划在求解大规模问题时效率较低。基于此，本节提出基于 Double Oracle 框架的算法应对大规模问题。

Double Oracle 框架是在求解大规模零和博弈时被广泛使用的框架 [46,78]。其基本思想是，首先限定博弈双方都只能使用一部分策略，得到受限制的博弈。利用 Core-LP 求得受限制的博弈的均衡解，可得到双方在受限制博弈中的最优混合策略。然后，给定一方的混合策略，查找另一方在全策略空间中是否能够找到一个纯策略，以得到比当前更好的收益。如果存在这样的纯策略，就把该策略加入这一方可使用的策略集中，即扩大受限制博弈中这一方的策略集，并重复上述过程。如果双方均不存在更好的策略，那么当前受限制博弈的均衡解也是原始问题的纳什均衡解。

Double Oracle 框架本身并不是一个完整的算法，因为这个框架本身并未说明如何查找博弈参与方在全策略空间内的最优纯策略回应。事实上，这样的查找仍然有可能需要显示列出全策略空间。因此，基于 Double Oracle 框架的算法的设计重心即在于设计寻找参与方在全策略空间内的最优回应的算法（称之为 Oracle，分为攻击方 Oracle 与防守方 Oracle），而这样的算法依赖于具体的问题。因此，此前基于 Double Oracle 框架的算法并不能直接适用于本文研究的问题 [46]。本部分的贡献在于，对于两个 Oracle，均提出新颖的混合线性规划算法，并提出可扩展性更好的启发式算法。

Double Oracle 的实现过程在算法 4展示。第 3行计算受限制博弈的纳什均衡 $(\mathbf{x}, \mathbf{y})$，其中 $\mathbf{y}$ 是 Core-LP 对偶问题的解，代表攻击方的混合策略。该算法应用了两种 Oracle：启发式 Oracle，用于快速查找一个“较好”的解（对于攻击方和安保方，分别被称为“AO-Better”和“DO-Better”）；精准 Oracle，用于求出一个最优解（AO-MILP 和 DO-MILP）。

Algorithm 4: Double Oracle 算法

```
1  Input: 𝒮′ ⊂ 𝒮; 𝒜′ ⊂ 𝒜;
2  while do
3      (x, y) ← Core-LP(𝒮′, 𝒜′);
4      a ← AO-Better(x);
5      if a = ∅ then a ← AO-MILP(x);
6      s ← DO-Better(y);
7      if s = ∅ then s ← DO-MILP(y);
8      if a ∈ A and s ∈ S then
9          return x;
10     else
11         𝒜′ ← 𝒜′ ∪ {a}, 𝒮′ ← 𝒮′ ∪ {s};
```

本节接下来介绍博弈双方 Oracle 问题的设计和计算。

3.5.1.1 攻击方 Oracle

复杂度 定理 1 已指出，在无安保情况下，破坏性攻击可在多项式时间内完成。然而，当防守方可以进行随机性安保时，攻击方要找出最优的攻击策略（也就是攻击方的 Oracle 问题）是 NP 难的。在下述定理中，$\mathcal{S}'$ 代表受限制博弈中防守方被允许使用的策

略空间。

定理 4 令 $\mathcal{S}'$ 表示一组防守方策略的集合。检测是否能够攻击 k 个选民群组，使得不论任何安保策略 $s \in \mathcal{S}'$ 被使用时破坏性攻击都能成功是 NP 难的，即使在只有两个候选人的情况下。

证明 接下来通过将该问题归约为难定义 1 中的 Hitting Set 问题来证明其 NP 难。具体的，本证明指出，对于任一个 Hitting Set 问题，可构造一个有 n 个选民群组和 2 个候选人的选举，一组防守方的纯策略 $\mathcal{S}'$，使得存在 hitting set G' 当且仅当攻击方存在一种方法能够在无论任何 $s \in \mathcal{S}'$ 被使用时都能成功进行破坏性攻击。

给定定义 1 中的 Hitting Set 问题，选举构建如下：有 $|G|+1$ 个选民群组，两个候选人，ω 和另一个候选人 c。任一 $i \in G$ 对应一个选民群体，其对应 $t_i^{c-\omega} = -1$，也就是说，该群体投给 ω 的票数比 c 多 1。在多加的那一组中（称该组为 j），令 $t_j^{c-\omega} = |G| - 1$。于是 c 总共比 ω 少一票。任一 $\hat{G} \in U$ 可被看做是防守方的一个纯策略的标签，在相应纯策略中，群组 j 和群组 $i \in G \setminus \hat{G}$ 被保护。例如，假设 $G = \{1,2,3\}$，$U = \{\{1,2\}\}$，那么共有四个群组。在前三个群组中，c 比 ω 各少 1 票，在最后一个群组中，c 比 ω 多两票。在标签为 $\{1,2\} \in U$ 的安保策略中，群组 3 和 4 被保护。

$\Leftarrow$ 方向：如果攻击方能够攻击 k 个选民群组，使得不论任何安保策略 $s \in \mathcal{S}'$ 被使用时破坏性攻击都能成功，那么给定一个标签为 $\hat{G} \in U$ 的安保策略 s，至少一个群组 $i \in \hat{G}$ 对应 $t_i^{c-\omega} = -1$ 被攻击了。于是 $\forall \hat{G} \in U, G' \cap \hat{G} \neq \emptyset$。否则当 s 被使用时，破坏性攻击就不能成功。所以，G' 是一个 hitting set。

$\Rightarrow$ 方向：给定一个 hitting set $G' \subset G$，$|G'| = k$，攻击方可以攻击所有群组 $i \in G'$。于是无论哪一个安保策略 $s \in \mathcal{S}'$ 被使用，至少有一个未被保护的群组，其 $t_i^{c-\omega} = -1$，受到攻击。由于 ω 在最初的选举中得票总数仅比 c 多一票，无论任何 $s \in \mathcal{S}'$ 被使用，这个攻击方式都能使得 ω 不再获得胜利。 □

随机性防守使得原本易于受到攻击的多数制选举也变得“可靠”了，这对于传统的仅关注选举制度是否可靠的研究提出了一个有趣的扩展方向。本节接下来研究攻击方 Oracle 问题的计算。

精准 Oracle 虽然攻击方最优回应的计算是 NP 难的，其依然可以用一个紧凑的混合线性规划表述，称作 **AO-MILP**。攻击方的最优回应也就是最小化 ω 获胜的概率，也就是说，给定一个防守方的混合策略 $\mathbf{x}$，求解

$$\min_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}'} x_s W^{\omega}(\mathbf{x}, a). \tag{3-3}$$

本节首先将该问题描述成一个非线性问题，再将其线性化。描述该问题的难点在于表达 $W^{\omega}(s,a)$，为此，本节引入一个辅助变量 z_s。

$$\text{AO-MILP}: \quad \min_{a_i,z_s,e_s^c\in\{0,1\}} \quad \sum_{s\in\mathcal{S}'} \quad (1-z_s)\cdot x_s \tag{3-4a}$$

$$\sum\nolimits_i a_i \le k \tag{3-4b}$$

$$\sum\nolimits_{c\in C^{-\omega}} e_s^c = 1, \quad \forall s\in\mathcal{S}' \tag{3-4c}$$

$$z_s\sum\nolimits_{c\in C^{-\omega}} e_s^c\left(\sum\nolimits_i t_i^{c-\omega}(1-(1-s_i)a_i\right) \ge 0, \forall s\in\mathcal{S}'. \tag{3-4d}$$

约束 (3-4b) 要求攻击策略 a 是可行的。接下来，关于约束 (3-4c)-(3-4d)，给定一个策略组合 (s,a)，当且仅当 $s_i=0$ 以及 $a_i=1$ 时，群组 i 才能被成功攻击。于是对于任一 $c\in C^{-\omega}$，c 与 ω 得票数的差值为

$$sum(\mathbf{t}^{c-\omega}\setminus k) = \sum\nolimits_i t_i^{c-\omega} - \sum\nolimits_i (1-s_i)a_i t_i^{c-\omega}. \tag{3-5}$$

给定一个策略组合 (s,a)，只要存在任一候选人得票数不少于 ω，即有 $sum(\mathbf{t}^{c-\omega}\setminus k)\ge 0$，可得 $z_s=1$。变量 e_s^c 被引入以检查是否存在这样的候选人。约束 (3-4c)， (3-4d)，以及目标函数共同保证，如果对于策略 s 存在这样的候选人 c^*，那么 $e_s^{c^*}$ 将被设置为 1，对其他候选人的 e_s^c 将被设置为 0。于是有 $\sum_{c\in C^{-\omega}} e_s^c\left(\sum_i t_i^{c-\omega}(1-(1-s_i)a_i)\right)\ge 0$，相应的 $z_s=1$，与约束 (3-4b) 组合后可得使得 ω 获胜概率最小的攻击策略。虽然约束 (3-4d) 是非线性的，但由于所有变量都是二值的，该问题可经由 McCormick 不等式线性化 [77]，最终得到一个混合整数线性规划。

启发式 Oracle AO-MILP 的问题在于其可扩展性不够强。但事实上，在每次 Oracle 运行时，攻击方只要能够找到比当前策略更好的策略，就能够使算法继续运行下去并收敛至均衡解。本节提出寻找更好的策略的多项式时间启发式算法。该算法在一些情况下不能返回结果，在这些情况下，仍可调用 AO-MILP，因此最终能会收敛至均衡。

该启发式算法共分两步。首先，寻找一个集合 $\mathcal{S}''\subset\mathcal{S}'$ 满足 $\sum_{s\in\mathcal{S}''} x_s > 1-p$，其中 p 是 Core-LP 求得的受限制博弈的目标解，也就是在受限制博弈下 ω 在均衡解时能够获胜的概率。第二部是寻找一个攻击策略，使得无论任何 $s\in\mathcal{S}''$ 被使用时，ω 都不能获胜，也就是 $W^{\omega}(s,a)=0\ \forall s\in\mathcal{S}''$。如果这样的集合和策略均能够被成功找到，那么攻击方能够使 ω 以至少 $\sum_{s\in\mathcal{S}''} x_s$ 的概率失败，也即这个策略是比当前受限制博弈中的均衡解更好的策略。

具体过程在算法 5中描述。首先，将 $\mathcal{S}'$ 中策略按照 x_s 降序排列，得到向量 $\bar{S}$，其中 s^{ρ} 表示该向量中第 ρ 大的元素（第 3行）。然后关注集合 $\mathcal{S}''$，它包括向量 $\bar{S}$ 中相邻的元素 (第 5 - 6 行)。对任一 $\mathcal{S}''$，检测是否存在候选人 c，能够使得如果攻击方攻击了 k 个未被任何 $s\in\mathcal{S}''$ 保护的群组，c 的剩余票数不少于 ω。如果存在这样的候选人，那

么这个攻击策略使得 ω 不能获胜，无论任何 $s \in \mathcal{S}''$ 被使用。也就是一个比任一受限博弈中允许使用的策略更好的攻击策略。（第 8 - 11 行）。否则算法返回空集。

Algorithm 5: AO-Better

```
1  输入：S', x, p;
2  S̄ = ⟨s^ρ, ρ ∈ 1,2,3,···⟩ ← 将 s ∈ S' 按照 x_s 降序排列;
3  for ρ in 1..|S̄| do
4      p' ← x_{s^ρ}, S'' ← {s^ρ}, ρ' ← ρ + 1;
5      while p' ≤ 1 − p and ρ' ≤ |S̄| do
6          p' ← p' + x_{s^ρ'}, S'' ← S'' ∪ {s^ρ'}, ρ' ← ρ' + 1;
7      if p' > p then
8          for c ∈ C^{−ω} do
9              t^{c−ω} \ k ← 删除具有最小 t_i^{c−ω} 值且 s_i = 0 的 k 个元素;
10             if sum(t^{c−ω} \ k) ≥ 0 then
11                 return 攻击与被删除元素相应的群组;
12 return ∅;
```

3.5.1.2 防守方 Oracle

定理 2揭示了防守方 Oracle 的难度。现在本节提出数学方法以求解现实中的问题。

精准 Oracle 给定攻击方的混合策略 $\mathbf{y} = \langle y_a : a \in \mathcal{A}' \rangle$，防守方的 Oracle，也就是最优回应，意图是最大化 ω 获胜的概率。即求解

$$\max_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}'} W^{\omega}(s,a) y_a. \tag{3-6}$$

与对攻击方 Oracle 的处理类似，本节由将该问题描述成一个非线性问题开始，再将其线性化。

$$\text{DO-MILP:} \qquad \max_{s_i, z_a \in \{0,1\}} \sum_{a \in \mathcal{A}'} z_a \cdot y_a \tag{3-7a}$$

$$\sum\nolimits_i s_i \le m \tag{3-7b}$$

$$z_a \sum\nolimits_i (t_i^{c-\omega}(1-(1-s_i)a_i)+1) \le 0, \forall c \in C^{-\omega}, a \in \mathcal{A}'. \tag{3-7c}$$

与攻击方 Oracle 的处理不同在于，仅当所有的候选人 $c \in C^{-\omega}$ 的得票数都少于 ω，攻击方能够成功抵制一个攻击策略 $a \in \mathcal{A}'$，也就是 $W^{\omega}(s,a) = 1$ 时，$z_a = 1$。其中 $\mathcal{A}'$ 是在受限制博弈中攻击方被允许使用的策略的集合。约束 (3-7c) 确保只有当 $\forall c \in C^{-\omega}$, $\sum_i t_i^{c-\omega} - \sum_i (1-s_i) a_i t_i^{c-\omega} < 0$ 时有 $z_a = 1$。约束 (3-7b) 确定防守策略是可行的。于是，DO-MILP 的结果就是在给定攻击方混合策略 $\mathbf{y}$ 后，选择能够最大化 ω 获胜概率的防守策略。利用 McCormick 不等式 [77]，可将约束 (3-7c) 线性化，从而得到一个混合整数线性规划。

启发式 Oracle 首先关注集合 $\mathcal{A}'' \subset \mathcal{A}'$ with $\sum_{a\in\mathcal{A}''} y_a > p$（这与在攻击方的启发式 Oracle 中首先寻找 $\mathcal{S}''$ 类似）。然后寻找一个防守方的纯策略 s，使其能够抵御所有攻击策略 $a \in \mathcal{A}''$，以使得 ω 能够以大于 p 的概率获得胜利。如果能够找到这样的安保策略，那么该策略就是比受限制博弈中安保方被允许使用的任何策略都更好的策略。具体过程在算法 6中描述。

Algorithm 6: DO-Better

$s = \langle s_i = 0 : \forall i \in \{1, \cdots, n\}\rangle$;
for $\mathcal{A}''$ *with* $\sum_{a\in\mathcal{A}''} y_a > p$ **do**
  res $\leftarrow 0$;
  for $c \in C^{-\omega}$ **do**
    $\mathbf{t}' \leftarrow \langle t_i^{c-\omega} : i$ with $a_i = 0, \forall a \in \mathcal{A}''$ or $s_i = 1\rangle$;
    while $sum(\mathbf{t}') \geq 0$ *and res* $< m$ **do**
      $\mathbf{t}'' \leftarrow \mathbf{t}^{c-\omega} \setminus \mathbf{t}', i^* \leftarrow \text{argmin}_i\{\mathbf{t}''\}$;
      $\mathbf{t}' \leftarrow \mathbf{t}' \cup \{t_{i^*}^{c-\omega}\}$, $s_{i^*} \leftarrow 1$, res $\leftarrow$ res + 1;
  if $\forall c \in C^{-\omega}, sum(\mathbf{t}') < 0$ **then**
    return: s;
return: $\emptyset$;

3.5.2 抵制建设性攻击算法

本节研究现实问题中抵制建设性攻击的算法。该问题的主要挑战在于，当攻击方意图进行建设性攻击时，博弈模型不再是零和，因此上节介绍的 Double Oracle 框架不再使用。强斯塔克伯格均衡策略可用下述基于混合线性规划的 SSE-MILP 算法求得。首先引入二值变量 $q_a \in \{0, 1\}(\forall a \in \mathcal{A})$，以指示攻击策略 a 是否被使用。令 M 表示一个大的实数。于是该问题可由方程 (3-8a)-(3-8c) 描述。方程 (3-8b) 将 p^D 的上界设置为 $u^D(\mathbf{x}, a)$，但对于攻击方采用的攻击策略有效。由于目标函数是最大化 p^D，这也就意味着安保方使用了最优策略。类似的，方程 (3-8c) 迫使攻击方在给定 $\mathbf{x}$ 的时候选择最优策略。第一部分 $p^A - u^A(\mathbf{x}, a) \geq 0$ 预示 p^A 必须至少和使用任一攻击策略所获得的收益一样大。第二部分破坏时对于选择的攻击策略，有 $p^A - u^A(\mathbf{x}, a) \leq 0$。如果攻击方不选择最优策略，那么该方程将被违反。于是约束和目标函数总体保证了 SSE-MILP 的结果就是 SSE 均衡策略。

$$\text{SSE-MILP}: \quad \max_{\mathbf{x},p} \quad p^D \tag{3-8a}$$

$$p^D - u^D(\mathbf{x}, a) \leq (1 - q_a)M, \quad \forall a \in \mathcal{A} \tag{3-8b}$$

$$0 \leq p^A - u^A(\mathbf{x}, a) \leq (1 - q_a)M, \quad \forall a \in \mathcal{A} \tag{3-8c}$$

3.5.2.1 零和近似算法

上一节提出的 MILP 算法依然要求显示列出双方的所有策略，这使得该算法在求解大规模博弈时效率较低。虽然 Double Oracle 框架在求解非零和博弈时不再适用，但本节指出，可通过求解一个零和博弈，以获得抵制建设性攻击的近似最优解。

首先考虑在抵制破坏性攻击的章节中所介绍的零和博弈，为表述的简洁性，称其为 ω-零和博弈。令 $(\mathbf{x}^\omega, a^\omega)$ 表示在该零和博弈下防守方和攻击方的均衡策略。令 p_0^ω 表示在给定策略组合 $(\mathbf{x}^\omega, a^\omega)$ 时 ω 赢得选举的概率。值得注意的是，当防守方采用策略 $\mathbf{x}$ 时，即使攻击方的目的在于进行建设性攻击（使得候选人 υ 获得胜利）并根据该目标选取给定 $\mathbf{x}$ 时的最优策略 $a_\upsilon^\omega = g(\mathbf{x}^\omega)$，其中 $a_\upsilon^\omega = \text{argmax}_a P^\upsilon(\mathbf{x}^\omega, a)$，$\omega$ 获胜的概率也不会小于 p_0^ω。这是由于，当攻击方使用策略 a_υ^ω 时，事实上等价于在零和博弈中攻击方偏离了自己的最优策略，而该偏离不会损伤防守方的收益。于是，该零和博弈的解就是在抵制建设性攻击时安保方收益的一个下界。

虽然 ω-零和博弈的解能够为抵制建设性攻击指明下界，但仍需探索该下界与最优解的差距。这可以通过求解另一种零和博弈，称之为 υ-零和博弈来实现。υ-零和博弈与 ω-零和博弈的设置基本相同，唯一的差异在于，在 υ-零和博弈下，防守方的目的是使得 υ 获胜的概率尽可能小。于是，υ-零和博弈的斯塔克伯格均衡（等价于纳什均衡）可以用求解 ω-博弈相同的方式求解。令 $(\mathbf{x}^\upsilon, a^\upsilon)$ 表示 υ-零和博弈中攻守双方的均衡策略，令 p_0^υ 表示在给定策略 $(\mathbf{x}^\upsilon, a^\upsilon)$ 时 υ 获胜的概率，下述命题指出，抵制建设性攻击时防守方收益的上界是 $1 - p_0^\upsilon$。

命题 6 *在抵制建设性攻击的博弈模型的强斯塔克伯格均衡策略 $(\mathbf{x}^*, a^*)$ 下，防守方的收益满足 $u^D(\mathbf{x}^*, a^*) \le 1 - p_0^\upsilon$。*

证明 *该命题可通过反证法证明。如果 $u^D(\mathbf{x}^*, a^*) > 1 - p_0^\upsilon$，也就意味着给定 $(\mathbf{x}^*, a^*)$，ω 以大于 $1 - p_0^\upsilon$ 的概率获得成功。于是有 υ 只可能以小于 p_0^υ 的概率获得成功。由于 a^* 是在给定 $\mathbf{x}^*$ 时，攻击方所作出的使得 υ 获胜概率最大的最优回应，这意味着策略组合 $(\mathbf{x}^*, a^*)$ 在 υ-零和博弈中会得到比 $(\mathbf{x}^\upsilon, a^\upsilon)$ 更高的收益，那么 $(\mathbf{x}^\upsilon, a^\upsilon)$ 就不可能是均衡策略组合。* □

给定命题 6，即可继续探究在何种情形下，υ-零和博弈的均衡策略也会是抵制建设性攻击的最优策略。直观上讲，如果 υ-零和博弈存在一组均衡策略 $(\mathbf{x}^\upsilon, a^\upsilon)$，满足该策略下只有 ω 和 υ 有可能赢得选举，也即，υ 以 p_0^υ 的概率获得胜利，ω 以 $1 - p_0^\upsilon$ 的概率获得胜利，那么 $\mathbf{x}^\upsilon$ 就是抵制建设性攻击的最优策略。正式的定理描述如下。

定理 5 *如果存在一个攻击策略 a 满足*

1. *a 是在 υ-零和博弈中对于 $\mathbf{x}^\upsilon$ 的最优回应，*
2. *$P^\omega(\mathbf{x}^\upsilon, a) = 1 - p_0^\upsilon$,*

那么 (x^{υ}, a) 就是在抵制建设性攻击的博弈中的一组强斯塔克伯格均衡策略。

证明 由于在 υ-零和博弈中，攻击方的目标同样是使得 υ 获得胜利，那么 a 也是攻击方在抵制建设性攻击的博弈中对于 $\mathbf{x}^{\upsilon}$ 的最优回应。于是，基于命题 6，可知 (x^{υ}, a) 是抵制建设性攻击的博弈中的强斯塔克伯格均衡集。 □

求解抵制建设性攻击的近似最优解的具体过程在算法 7。随后的实验将展示，该算法几乎总是返回最优解。

Algorithm 7: 零和近似算法

$(\mathbf{x}^{\upsilon}, \mathbf{a}^{\upsilon}), p_0^{\upsilon} \leftarrow$ 求解 υ-零和博弈，$u_1^D = \text{argmax}_{a \in \mathcal{A}^{\upsilon}} P^{\omega}(\mathbf{x}^{\upsilon}, a)$;

/* $\mathbf{a}^{\upsilon}$ 是攻击方的最优混合策略，$\mathcal{A}^{\upsilon}$ 是其支持集 */;

if $u_1^D = 1 - p_0^{\upsilon}$ **then return** $\mathbf{x}^{\upsilon}$;

$(\mathbf{x}^{\omega}, \mathbf{a}^{\omega}) \leftarrow$ 求解 ω-零和博弈;

$\mathcal{A}' = \text{argmax}_{a \in \mathcal{A}} P^{\upsilon}(\mathbf{x}^{\omega}, a)$;

/* 攻击方为进行建设性攻击，针对 $\mathbf{x}^{\omega}$ 的最优回应的集合 */;

$u_2^D = \text{argmax}_{a \in \mathcal{A}'} P^{\omega}(\mathbf{x}^{\omega}, a)$;

$u_1^D > u_2^D$? **return** $\mathbf{x}^{\upsilon}$: **return** $\mathbf{x}^{\omega}$

3.6 选票不确定性

至此，本节的研究与分析均基于一个基本假设：防守方和攻击方对于各个选民群组中各个候选人的得票数都完全明了。本节放松这个假设，考虑防守方对于选票情况的信息不完全的情形。具体来讲，仍假设攻击方知晓全部选票信息，而安保方只知晓选票信息的分布。令 V 表示某一个具体的选票状态。令 R_V 表示先验情况下，防守方认为选票情况是 V 的概率。防守方的目标仍然是使得 ω 获胜的概率最高。本节只考虑抵制破坏性攻击的情形。本节的结果可以被直观地扩展到抵制建设性攻击的问题中。由于攻击方知道真实的选票情况，因此该问题即为一个贝叶斯博弈，其中 V 可被看做攻击方的类型。

令 $W_V^{\omega}(s, a)$ 作为一个二值指示器，指示在选票情况为 V 时，给定策略组合 (s, a)，ω 是否能够赢得选举。于是防守方的最优策略就可通过解如下所示线性规划求得。该线性规划是对第 3.5.1章所提出的 Core-LP 的一个贝叶斯扩展。

$$\text{Bayesian-LP}(\mathcal{S}, \mathcal{A}): \qquad \max_{\mathbf{x}} \sum_{V} R_V P_V \tag{3-9a}$$

$$P_V \leq \sum_{s \in \mathcal{S}} x_s W_V^{\omega}(s, a), \quad \forall a \in \mathcal{A}, \forall V \tag{3-9b}$$

Byesian-LP 依然可以用基于 Double Oracle 框架的算法求解。与前述算法不同的是，对于每一种可能的投票情况 V 均需运行一次攻击方 Oracle，而安保方 Oracle 的目标函数也需改成贝叶斯形式。事实上，由于可能存在的投票情况非常多，在求解时可以通过只根据分布采样少数种投票情况，再利用这些样例运行 Bayesian-LP，以求出近似最优解。

3.7 实验与分析

本节通过算法性能和可扩展性两方面，检测提出的算法。线性规划和混合线性规划由 CPLEX 12.6.1 求解。算法性能通过其在模拟数据集和真实数据集上的表现来衡量。对于模拟数据集，实验在每个群组内为每个候选人随机生成 0 到 100 之内的票数。除特殊说明外，图中的每个点均为 101 次样本平均的结果，误差线展现了 95% 的置信区间的结果。算法性能的测试考虑以下变量变动情况：攻击资源数量，安保资源数量，选民群组数量以及候选人数量。真实数据集为 2002 年法国总统选举数据 [63]。该数据集包含来自 6 个选区，投向 16 位候选人的 2597 票。

3.7.1 算法性能

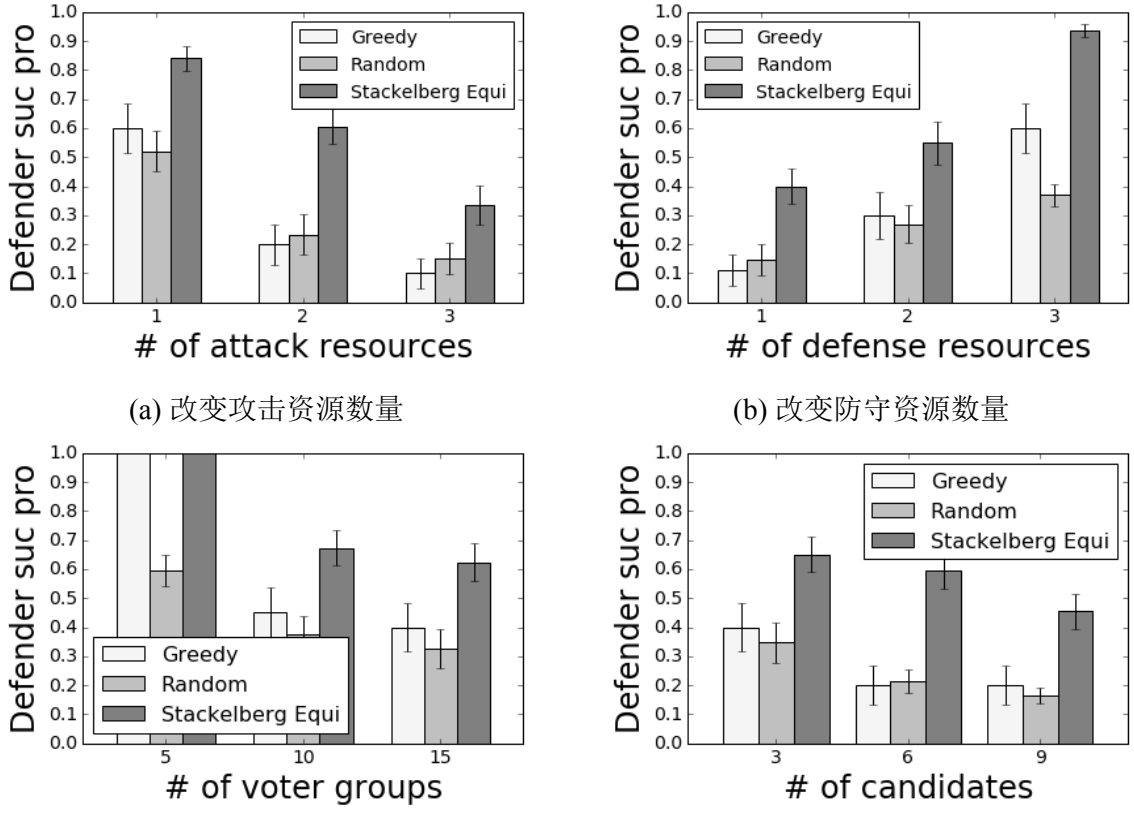
文中提出的算法与两种基线算法对比，以检测其性能。第一种，图中标注为 Random，假定安保方以等概率使用策略空间内的所有纯策略。第二种，图中标定为 Greedy，确定性的保护 ω 比排名其后的候选人领先票数最多的 m 个选民群组。算法性能通过防守方成功的概率体现，也就是 ω 最终赢得选举的概率。

第一组实验测试抵制破坏性攻击的情形。图3.1(a)和3.1(b)展示了当攻击资源与安保资源的数量发生变化时，本节提出的算法与基线算法的算法性能。在这组实验中，选民群体的数量被固定为 10，候选人数量被固定为 4。在图3.1(a)中，安保资源的数量被固定为 2；在图3.1(b)中，攻击资源的数量被固定为 2。图3.1(c)和3.1(d)展示了当群组数量变化和候选人数量变化时各算法的性能。其中攻守资源数都被固定在 2。在各图中，“Stackelberg Equi”指代斯塔克伯格博弈均衡解，也就是 Core-LP 的目标值。从图中结果可见，斯塔克伯格均衡解明显好于基线算法求出的解。

第二组实验测试抵制建设性攻击的情形。实验设置与第一组实验相似。其中斯塔克伯格均衡解也即 SSE-MILP 所求出的目标值。图3.2(a) -3.2(d)展现了在和图3.1(a) -3.1(d)相似的配置下，各算法在抵制建设性攻击时的性能。可见斯塔克伯格均衡解依然明显好于基线算法。

接下来的实验检测本节提出的算法与基线算法在真实数据集上的性能。首先考虑抵制破坏性攻击，图3.3(a)和3.3(b)展示了当攻击资源数量与安保资源数量分别变化时的算法性能。斯塔克伯格均衡解相对于基线算法的优势甚至比在模拟数据下更大。在真实数据集下，攻击结果是确定性的，因此此处图中没有误差线。

最后检测在真实数据集上各算法抵制建设性攻击的性能。由于真实数据集中共有



(a) 改变攻击资源数量 (b) 改变防守资源数量

(c) 改变群组数量 (d) 改变候选人数量

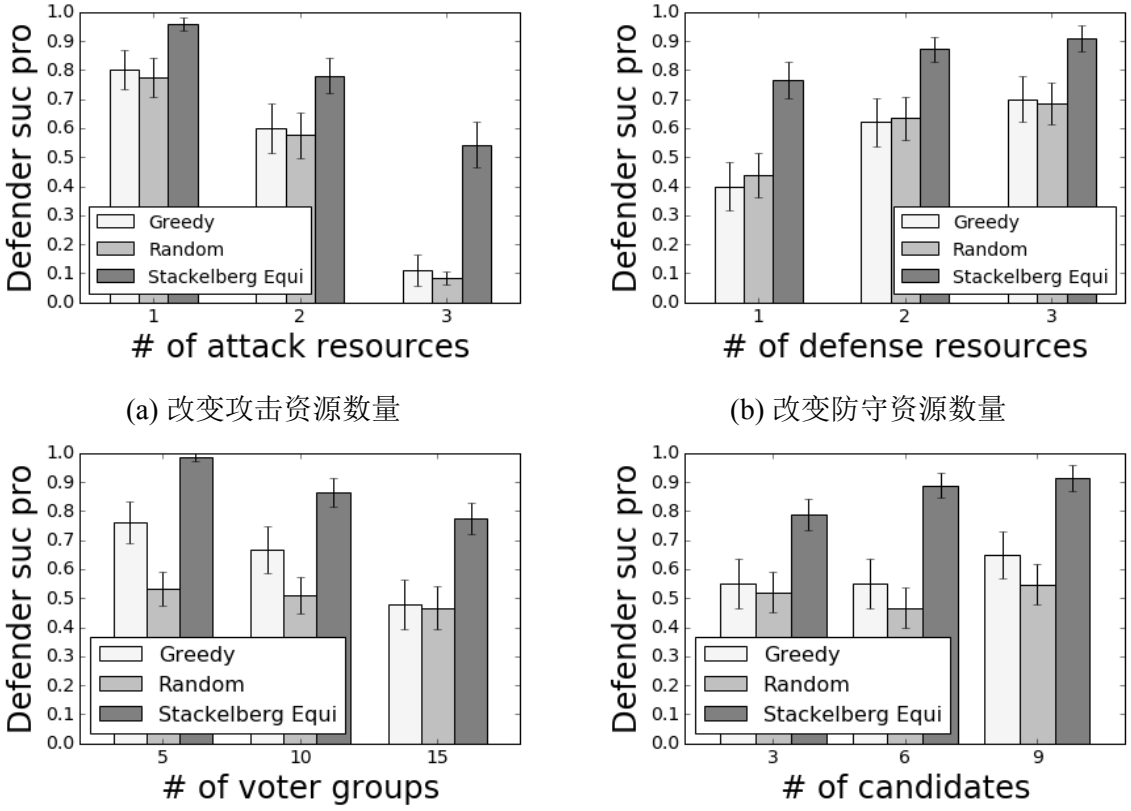
图 3.1 模拟数据集上的算法性能：抵制破坏性攻击

16 个候选人，因此在实验中假设攻击方分别意欲使得落败的 15 人获得胜利，每个点都是 15 次实验的平均值，误差线是其标准误。图3.4(a)和3.4(b)展示了当攻击资源与安保资源数量发生变化时的算法性能。斯塔克伯格博弈均衡依然明显优于基线算法。

3.7.2 可扩展性

本节测试算法的可扩展性。对于求解抵制破坏性攻击的最优策略，测试 Core-LP 和两种基于 Double Oracle 的方法：第一种仅使用精准 Oracle，在图中标记为 DORA。第二种使用启发式 Oracle，当其不能返回结果时调用精准 Oracle，在图中标记为 DORABE。为求解抵制建设性攻击的最优策略，检测 SSE-MILP 和启发式算法，在图中标记为 Heu-Apro-MILP。该启发式算法首先运行贪心启发式算法。如不能返回最优抵制策略，那么调用零和近似算法。如果零和近似算法的结果不是最优，那么再调用 SSE-MILP 求得最优解。

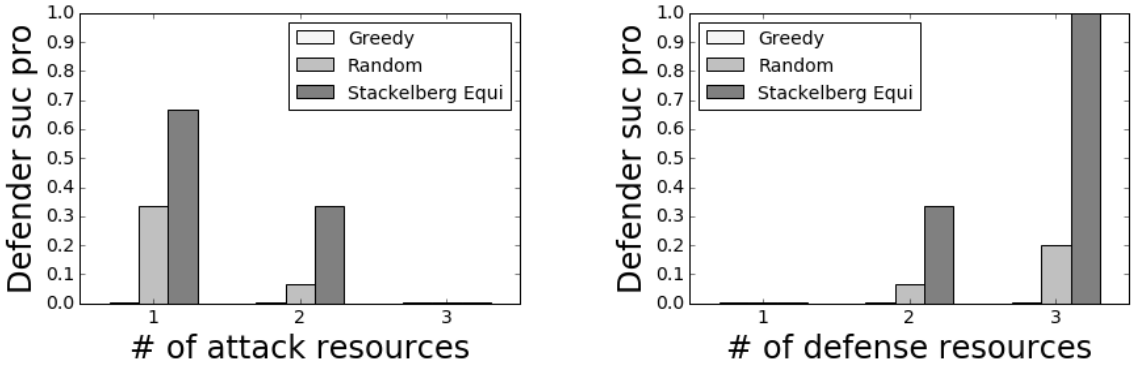
图3.5显示，随着问题规模的增加（无论群组数量还是防守资源数量），基于 Double Oracle 框架的算法都明显优于 Core-LP。DORABE 通常要经过比 DORA 更多的循环数才能够收敛，但每次循环的运行时间均小于 DORA，两者的总运行时间较为接近。SSE-MILP 是可扩展性最差的，但 Heu-Apro-MILP 可以相当高效地求解大规模问题。事实上，当防守资源数量增加时，Heu-Apro-MILP 的运行时间几乎没有增加。这主要是由



(a) 改变攻击资源数量　(b) 改变防守资源数量

(c) 改变群组数量　(d) 改变候选人数量

图 3.2　模拟数据集上的算法性能：抵制建设性攻击

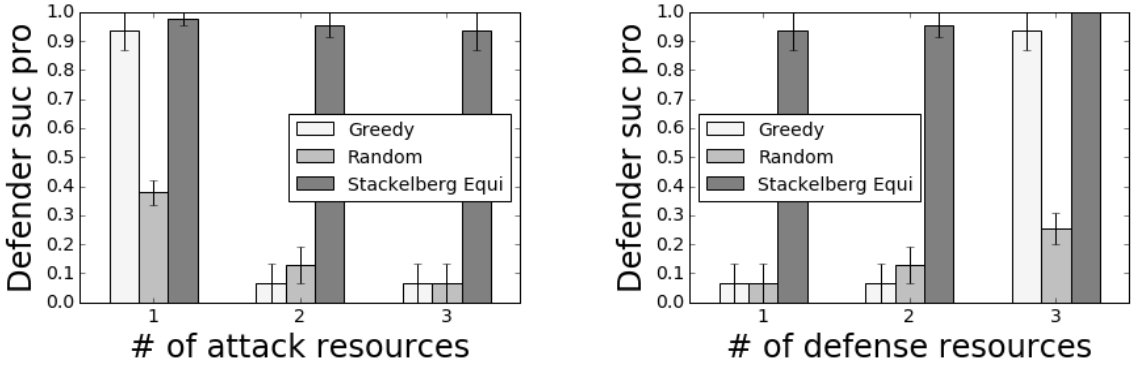


(a) 改变攻击资源数量　(b) 改变防守资源数量

图 3.3　真实数据集上的算法性能：抵制破坏性攻击

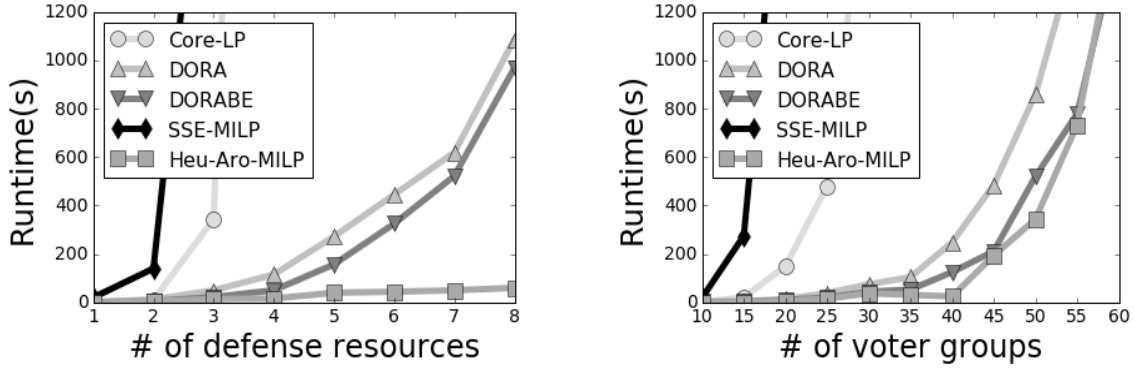
于贪心启发式算法在大多数情况下都能直接返回最优的抵制策略，尤其是在安保资源充足的情况下。当群组数量增加时，Heu-Apro-MILP 的运行时间与 DORABE 的运行时间接近。

接下来更细致地检测贪心启发式算法和零和近似算法在抵制建设性攻击中的效用。图3.6(a)显示了当安保资源数量增加时，贪心启发式算法成功找到一个能够抵制建设性攻击的安保策略的概率。直观来讲，更多的安保资源意味着更高的成功率，事实上当安保资源数量为 7 时（选民群组数为 20），贪心启发式算法几乎总能找到一个安保策略。



(a) 改变攻击资源数量　　(b) 改变防守资源数量

图 3.4　真实数据集上的算法性能：抵制建设性攻击



(a) 改变防守资源数量　　(b) 改变群组数量

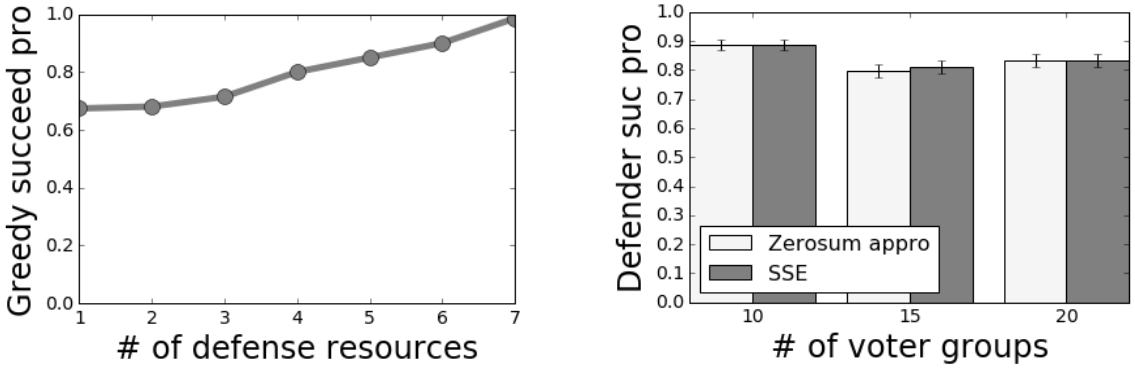
图 3.5　算法可扩展性

图3.6(b)显示零和近似算法和强斯塔克伯格均衡解的对比，该组实验中每个点均是 500 样本的平均值。结果显示零和近似算法的解和强斯塔克伯格均衡解极为接近。事实上，在所有的 1500 次实验中，仅有 1 次（群组数量为 15），零和近似算法返回了非斯塔克伯格均衡解。

3.7.3　不确定性

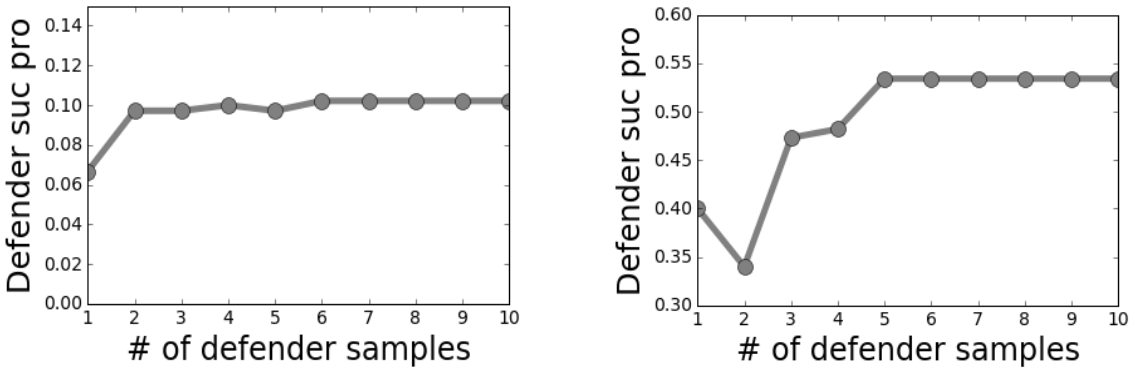
最后，检测考虑不确定性后的算法性能。总体来讲，贝叶斯斯塔克伯格均衡解依然显著优于基线算法。关于不确定性的策略主要考虑在所有可能的选票构成中的抽样数量对于结果的影响。这部分实验分两组：低不确定性，其中高斯噪音方差为 10；高不确定性，其中每组为每个候选人投票数在 0 到 400 间随机产生，高斯方差为 20。在两种情况下，均假设共有 60 种可能的选票分布情况。在图3.7(a)和3.7(b)中，横轴展示为求解贝叶斯线性规划而进行的抽样数量，纵轴显示防守方成功的概率。图中结果显示，在两种情况下，均有少数的抽样（≤ 6）即可得到近似最优的解。

接下来的实验检测算法在数据存在误差时的鲁棒性。首先考虑防守方关于选票的信息并不精确，而是在真实票数上存在一个基于标准高斯分布的随机误差。假设防守



(a) 贪心启发式算法成功概率　　(b) 零和近似算法性能

图 3.6　启发式算法性能



(a) 低不确定性　　(b) 高不确定性

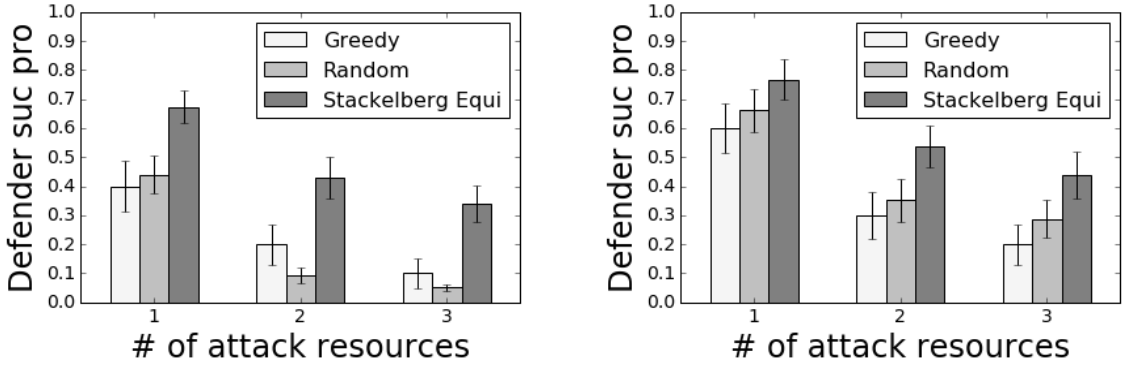
图 3.7　贝叶斯-LP：样本数量对算法性能的影响

方在基于该误差的信息下求解其最优策略，而攻击方依然根据真实情况进行攻击，然后计算防守策略成功的概率。图3.8(a)和3.8(b)分别显示了在存在误差的情况下，本文提出算法和基线算法在抵制破坏性攻击和建设性攻击时的效用。结果展示，即使在存在误差的情况下，斯塔克伯格均衡解仍显著优于基线算法的解。

接下来检测在贝叶斯博弈的情况下，防守方对于选票分布情况的信息有误差的情形。首先设定一种分布情况作为真实分布，再假设防守方关于分布的信息是该分布加上 $\mu = 0, \sigma^2 = 0.1$ 的高斯误差，将得到的新分布映射到总和为 1 的空间内。图3.9(a)显示了在抵制破坏性攻击的情况下各算法的效用，可见即使存在这样的误差，斯塔克伯格均衡解依然优于基线算法。图3.9(b)展示了无误差和有误差的斯塔克伯格均衡解的差异，结果显示误差并不显著影响均衡解。

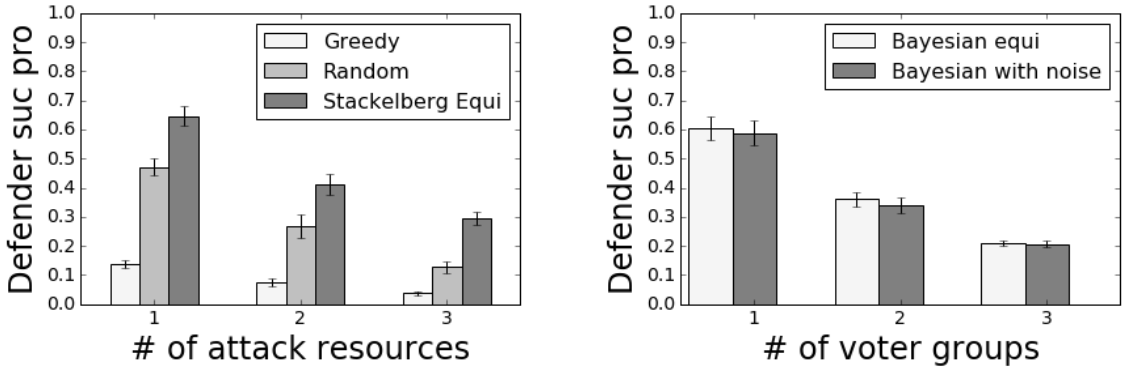
3.8　本章小结

本章研究利用安全博弈模型，设计安保策略，以抵制选举中的破坏性攻击和建设性攻击的问题。多数制选举制度面对基于群组的破坏性攻击时是不可靠的。但保护选举，抵制破坏性攻击和建设性攻击，都是 NP 难的。对于抵制破坏性攻击，本章提出了



(a) 破坏性攻击 (b) 建设性攻击

图 3.8 防守方关于票数信息误差的影响



(a) 算法性能 (b) 初始分布 v.s. 有误差分布

图 3.9 防守方关于分布信息误差的影响

基于 Double Oracle 框架的算法以求解最优的安保策略，并为每个 Oracle 设计了精准的最优求解算法和启发式算法。对于抵制建设性攻击，本章提出了一种混合线性规划算法来求解博弈的强斯塔克伯格均衡解，并提出了一种启发式算法和一种基于求解零和博弈的近似最优算法。实验表明，无论是对于抵制破坏性攻击还是建设性攻击，本章提出的模型和算法都显著优于基线算法。本章提出的启发式算法和近似最优算法都以相当高的概率返回最优解。

第四章　动态收益安全博弈纯策略均衡

4.1　引言

近年来，恐怖活动日益猖獗，袭击重大公共活动的事件时有发生。例如 2013 年的波士顿马拉松爆炸案，就造成了重大的损失和恶劣的影响。保障重大赛事的安全是各国的安全部门都亟需解决的问题。然而，利用有限的安保资源设计高效的重大赛事安保策略却面临着技术上的挑战，主要原因有如下三点。第一，重大赛事中可能被攻击的目标的重要程度是随时间变化的。例如，在马拉松比赛中，赛道的某一段可视为一个可能被攻击的目标。而随着参赛选手和观众随着比赛进行而沿着赛道转移，不同赛段的重要程度也随之变化。这就要求安保资源随时间动态分配。第二，潜在的恐怖分子可能在任何时间发动攻击，而安全部门也可以在任何时间转移资源，这也就意味着双方的策略空间都无限大。第三，由于重大赛事的相关信息通常会提前公布，恐怖分子在制定攻击计划时也会策略性地考虑这些信息。因此，安保部门在设计安保策略时也需要考虑恐怖分子的策略。本章致力于解决这三个问题，以博弈模型描述安全部门和恐怖分子的策略行为，并设计算法求解安全部门的最优策略。

博弈论已被成功应用于很多安全领域，例如机场安保 [90]，港口巡逻 [98]，以及其他多种形式的安保策略设计 [94,95,101–103]。然而，已有的模型与算法并不能被直接用于解决重大赛事的安保问题。首先，大多数的研究都假设可能被攻击的目标的重要程度是不随时间变化的 [1,38]，这与重大赛事场地复杂的情况不符。虽然有部分研究人员考虑了攻击目标的重要程度可能随时间变化的情况 [32]，他们的工作与本章有两点本质的区别。其一，他们假设安全部门和恐怖分子都只能在预先设定的某些时间点行动 [121]。第二，他们关注的是常规的、通常需要每天都执行的安保策略 [48]，因而恐怖分子在行动时会参考安保部门已经执行过的策略。而由于重大公共活动频率较低，恐怖分子并没有关于安保策略的前期资料可资参考，其行为取决于其对赛事情况的了解以及基于此对于安保策略的预测。因此，在重大赛事的安保领域，恐怖分子与安保部门直接的交手更应该被看做无历史信息的一次性行为。

本章首先设计了一种博弈模型来描述重大公共活动的安保问题。在此模型中，安全部门（以下简称安保方）与恐怖分子（以下简称攻击方）均具有连续的策略空间。在此博弈模型的均衡策略下，即安保方的最优策略下，安保方可能遭受的最大的损失最小。接着，本章关注了资源的转移时间远小于活动的进行时间的情形，并提出算法 SCOUT-A 来求解此情形下安保方的最优策略。然后，本章又研究了更具一般性的资源转移时间不可忽略的情况。针对这种情形，本章从离散时间假设着手，提出算

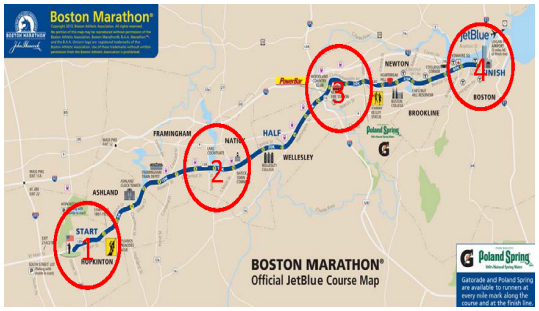


图 4.1　保护及攻击目标样例

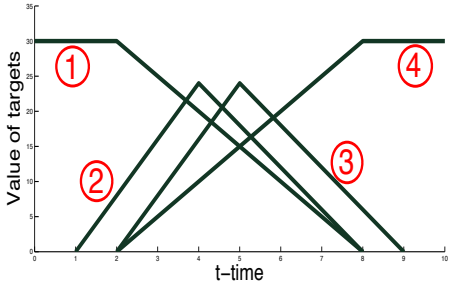


图 4.2　重要性函数样例

法 SCOUT-D 来求解在离散时间假设下的安保方最优策略，又在此基础上设计出算法 SCOUT-C，以求解连续策略空间、资源转移时间不可忽略的情形下的安保方最优策略。最后，本章通过实验验证了算法的有效性。

4.2　问题描述与博弈模型

假设在活动中共有 n 个可能被攻击的目标，例如马拉松比赛中的 n 个赛段，表示为 $\mathcal{T}=\{1,\ldots,n\}$。图4.1通过波士顿马拉松的赛道展示了一个将比赛场地分为四个可能被攻击的目标的样例。假设安保方共有 $m(m\le n)$ 个相同的安保资源，例如，每个资源可能是一个巡逻队。假设赛事在时刻 0 开始，在时刻 $t_e>0$ 结束。每个目标 $i\in\mathcal{T}$ 均对应一个重要性函数 $v_i(t)(t\in[0,t_e])$，其在时刻 t 的函数值即为目标 i 在时刻 t 的重要程度。图4.2基于图4.1的假设，展示了四个目标所对应的重要性函数的一个样例。在比赛刚开始时，人群聚集在赛道的起始路段，因此目标 1 的重要性函数值最高。随着比赛的进行，人群逐渐沿赛道转移，因此目标 1 的重要性函数值逐渐下降。相反的，目标 4，也即靠近终点的赛道，重要性函数的值随时间而增加。由于重要性函数实际上取决于赛事的地形、参赛者、观众数等基本信息，而这类信息通常都是公开的，因此假设重要性函数对于安保方与攻击方均为已知。为了计算的便捷，假设 $v_i(t)$ 为分段线性连续函数。此类函数被大量用来近似更为复杂的函数形式 [3]，而近似的精确性能够通过设置线段数量来控制。

安保方在时刻 0 将安保资源部署在一些目标上, 随着活动的进行，他在某些时刻将资源转移到其他的目标去。安保方的一个纯策略 S 即包含时刻 0 的资源部署情况以及在时间段 $[0,t_e]$ 内所做的所有转移。用向量 $C=\langle C_k\rangle$ 来表示所有转移，其中 $C_k=\langle c_{ij}^k:i,j\in\mathcal{T}\rangle$ 表示第 k 次转移，c_{ij}^k 代表第 k 次转移中从目标 i 到目标 j 转移的资源的个数。用一个向量 $D=\langle d_{ij}:i,j\in\mathcal{T}\rangle$ 来表示转移资源所需的时间，其中 d_{ij} 即为从目标 i 到目标 j 转移资源所需时间。另外，定义向量 $\tau=\langle\tau_k\rangle$，令 τ_k 表示第 k 次转移发生的时间。令 $Q^0=\langle q_i^0\rangle$ 表示在赛事开始时安保资源的部署情况，其中 q_i^0 表示赛事开始时目标 i 被分配的资源个数。相似的，令 $Q^t(S)=\langle q_i^t(S):i\in\mathcal{T}\rangle$ 表示任意时刻 t 安保资源的部署情况。那么，一个安保策略即可被完整的表示为 $S=(Q^0,C,\tau)$。而 $q_i^t(S)$ 可

以通过如下公式得到。

$$q_i^t(S) = q_i^0 + \sum_{C_k \in C, \tau_k \le t - d_{ji}, j \in \mathcal{T}} c_{ji}^k - \sum_{C_k \in C, \tau_k \le t, j \in \mathcal{T}} c_{ij}^k \tag{4-1}$$

攻击方的一个纯策略是在时间段 $[0, t_e]$ 内选择一个时间点 t，并选择一个目标 i，在这个时间点对这个目标进行攻击，可表示为一个元组 (i, t)。

值得注意的是，某个目标在某个时刻可能被多个安保资源保护，而不同数量的安保资源对于目标的保护程度是不同的。令 $p(r)$ 表示当有 r 个资源保护某个目标时，若攻击方选择攻击此目标，其攻击能够成功的概率。常规来讲，$p(r)$ 应满足以下两个条件：

1. $p(r) \in [0, 1]$，且当 $r = 0$ 时，$p(r) = 1$（即无保护时攻击总能成功），当 $r = \infty$ 时 $p(r) = 0$（即当有无穷多资源保护某目标时攻击不可能成功）。
2. $\frac{\partial p(r)}{\partial r} \le 0$ 并且 $\frac{\partial^2 p(r)}{\partial r^2} \ge 0$，即经济学上的边际递减效应——随着资源数目的增加，再增加一个资源带来的边际收益减少。

为满足这两个条件，令 $p(r) = \frac{1}{e^{\lambda r}} (\lambda > 0)$，其中 λ 是用来衡量增加一个资源带来的收益的参数。那么，如果攻击方选择策略 (i, t)，其收益即可表示为

$$W_i^r(t) = p(r) v_i(t), \tag{4-2}$$

其中 r 为安保方在时刻 t 部署在目标 i 上的资源数。可想而知，安保方的目的是和攻击方相反的，因此考虑零和博弈，即安保方的收益为 $-W_i^r(t)$。

由于安保方与攻击方之间进行的是一次性零和博弈，在均衡条件下，安保方的策略应满足使其最坏收益最好，也即，即使攻击方能够对安保策略做出最优回应，安保方的策略也能为其带来最优收益。假设给定某个安保策略 S，攻击方能够选择的最优策略是 $f(S) = \{f_{tg}(S) : S \to i, f_{tm}(S) : S \to t\}$，其中 $f_{tg}(S)$ 是其选择攻击的目标，而 $f_{tm}(S)$ 是其选择攻击的时间。另 $U(i, t, S)$ 表示给定安保策略 S，攻击方选择策略 (i, t) 所能得到的收益。那么一组均衡策略 $(S, f(S))$ 应满足如下条件：

$$U^a(f_{tg}(S), f_{tm}(S), S) \ge U^a(i, t, S), \forall i \in \mathcal{T}, t \in [0, t_e], \tag{4-3}$$

$$U^d(f_{tg}(S), f_{tm}(S), S) \ge U^d(f_{tg}(S'), f_{tm}(S'), S'), \forall S' \in \mathcal{S}. \tag{4-4}$$

不等式 (6-3) 限定给定安保方的策略 S，攻击方做出最优的选择。不等式 (4-4) 则限定即使攻击方做出最优选择，安保方的均衡策略仍应使其获得最优的收益。接下来，介绍算法来求解这样的安保方策略。

4.3 转移时间可忽略的情形

首先从一个简单的情形开始研究这个问题，即，安保资源的转移时间可以忽略不计。这种情形对应的现实场景即为各个目标距离较近，因而资源的转移时间远小于赛事的进行时间。当转移时间为 0 时，一个直观的求解安保方最优策略的方法

即求出在 $[0,t_e]$ 内的每个时刻安保方的最优资源配置方案。将一个配置方案定义为 $A=\langle a_i:\sum_{i\in\mathcal{T}}a_i=m\rangle$，其中 a_i 表示目标 i 所配置的资源数量。令 $\mathcal{A}$ 表示所有资源配置方案的集合。对于任何一个安保策略 S，在零转移时间的假设下，任何一个时刻所有资源均在使用中，故 $Q^t(S)\in\mathcal{A}$。以下示例可直观展示此种情形。

示例 1: 两个目标的重要性函数 $v_i(t)$ 如图4.3所示。安保方有一个资源。故而 $\mathcal{A}=\{A=\langle 1,0\rangle,A'=\langle 0,1\rangle\}$。如果安保方在时刻 t 使用配置方案 $\langle 1,0\rangle$，那么攻击方在任一时刻攻击两个目标所能够获得的收益，即 $W_1^1(t)$ 和 $W_2^0(t)$，可被图4.4中以方块标示的线条标示。类似的，以三角标示的线条显示当安保方采用配置方案 $\langle 0,1\rangle$ 时攻击方可能获得的收益。的目标即计算何时使用方案 $\langle 1,0\rangle$，何时使用方案 $\langle 0,1\rangle$，以使得攻击方在 $[0,t_e]$ 时间段内可能获得的最大攻击收益最小化。

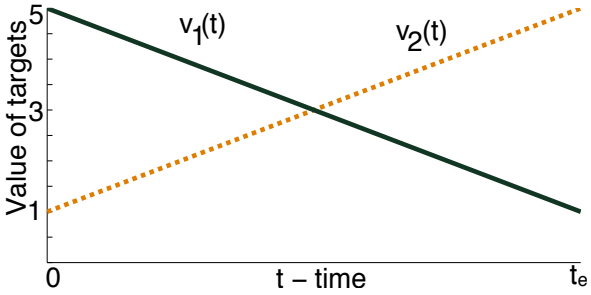


图 4.3　重要性函数

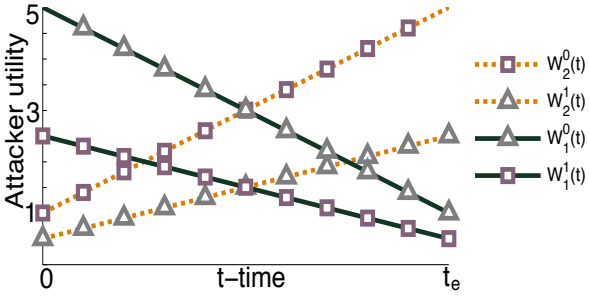


图 4.4　可能的攻击方收益

定义一个配置方案 $A=\langle a_i\rangle$ 是时刻 t 的最优配置，仅当对于任何 $A'=\langle a_i'\rangle\in\mathcal{A}$，有 $\max_{i\in\mathcal{T}}W_i^{a_i}(t)\le\max_{i\in\mathcal{T}}W_i^{a_i'}(t)$。考虑示例 1 中的情形，$\max_{i\in\mathcal{T}}W_i^{a_i}(t)(A=\langle 1,0\rangle)$ 由图4.5中黑色粗线条表示，$\max_{i\in\mathcal{T}}W_i^{a_i'}(t)(A'=\langle 0,1\rangle)$ 由灰色粗线条表示。不难发现，在时刻 t_0，$\max_{i\in\mathcal{T}}W_i^{a_i}(t_0)\le\max_{i\in\mathcal{T}}W_i^{a_i'}(t_0)$，所以 A 是时刻 t_0 的最优配置方案。下述命题描述了一组最优策略。

命题 7 *如果一个安保策略 S 满足 $\forall t\in[0,t_e]$，$Q^t(S)$ 是在时刻 t 的最优配置，那么 S 是攻击方的最优策略。*

证明 由于 $Q^t(S)$ 是时刻 t 的最优配置，那么对于 $S'\in\mathcal{S}$ 和 $\forall t\in[0,t_e]$，有 $\max_{i\in\mathcal{T}}W_i^{q_i^t(S)}(t)\le\max_{i\in\mathcal{T}}W_i^{q_i^t(S')}(t)$。于是，

$$\begin{aligned}U^a(f_{tg}(S),f_{tm}(S),S)&=\max_{i\in\mathcal{T}}W_i^{q_i^t(S)}(t)\quad(t=f_{tm}(S))\\&\le\max_{i\in\mathcal{T}}W_i^{q_i^t(S')}(t)\quad(t=f_{tm}(S))\\&\le U^a(f_{tg}(S'),f_{tm}(S'),S').\end{aligned}$$

也即 S 使得攻击方的最大收益最小。 □

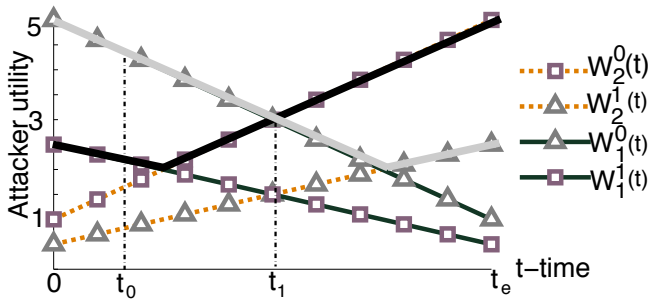


图 4.5 最优配置方案

命题 1 说明可以通过求出每个时刻的最优配置来求解安保方的最优策略。在本章中，关注最优配置的一个子集——完全最优配置方，这种配置方案总是存在且利于计算。其定义如下。

定义 4.1 配置方案 A 是一种在时刻 t 的完全最优配置（简称 PA），如果其满足 $\max_{j\in\mathcal{T}} W_j^{a_j}(t) \le W_i^{a_i-1}(t) = \frac{v_i(t)}{e^{\lambda(a_i-1)}}, \forall i \in \mathcal{T}, a_i > 0.$

直观上讲，如果一种资源的配置方法与 PA 相比，只有一个资源的分配情况不同，那么 PA 比这种资源配置的方法更优。

命题 8 一个时刻 t 的完全最优配置 A 是该时刻的最优配置。

证明 给定 $A = \langle a_i \rangle$，任何一种配置 $A' = \langle a_i' \rangle$ 可被表示为 $a_i' = a_i + k_i (\forall i \in \mathcal{T})$，其中 k_i 是一个整数。由于资源总数固定，有 $\sum_{i\in\mathcal{T}} k_i = 0$。如果 $A' \neq A$，那么必然存在至少一个目标 $j \in \mathcal{T}$，满足 $k_j < 0$。于是有 $W_j^{a_j'}(t) = \frac{v_j(t)}{e^{\lambda(a_j+k_j)}} \ge \frac{v_j(t)}{e^{\lambda(a_j-1)}} = W_j^{a_j-1}(t)$。如果 A 是时刻 t 的完全最优配置，那么 $W_j^{a_j-1}(t) \ge \max_{i\in\mathcal{T}} W_i^{a_i}(t)$，于是 $W_j^{a_j'}(t) \ge \max_{i\in\mathcal{T}} W_i^{a_i}(t)$。由于 $\max_{i\in\mathcal{T}} W_i^{a_i'}(t) \ge W_j^{a_j'}(t)$，则有 $\max_{i\in\mathcal{T}} W_i^{a_i'}(t) \ge \max_{i\in\mathcal{T}} W_i^{a_i}(t)$。 □

那么，就可以通过计算每个时刻 $t \in [0, t_e]$ 的 PA 来求解安保方的最优策略。然而，由于时间的连续性，列举每个时刻并求解其 PA 在计算上并不可行。但可以证明，虽然时间是连续的，每个时刻对应的 PA 并不会连续变化，那么就只需计算有限个 PA。假设配置 A 是时刻 t_0 的 PA。对于任何一对目标 i, j，如果它们满足 $\exists t \in [t_0, t_e]$，就有 $\frac{\partial W_i^{a_i-1}(t)}{\partial t} < \frac{\partial W_j^{a_j}(t)}{\partial t}$。令 $I_{ij}(A, t_0)$ 表示线 $W_i^{a_i-1}(t)$ 和线 $W_j^{a_j}(t)$ 的第一个交点。以下命题说明了 PA 有效的时间段。

命题 9 如果配置 A 在时刻 t_0 是 PA，那么 A 在任何一个时刻 $t \in [t_0, \min(\mathbf{I}(A, t_0))]$ 均为 PA。

证明 通过反证法来证明这个命题。令 $t_1 = \min \mathbf{I}(A, t_0)$。如果存在一个时刻 $t_2 \in (t_0, t_1)$ 使得在此时刻 A 不是一个完全最优配置，那么即存在一对目标 i, j 使得 $W_i^{a_i-1}(t_2) < W_j^{a_j}(t_2)$。由于 A 是 t_0 时刻的完全最优配置，有 $W_i^{a_i-1}(t_0) \ge W_j^{a_j}(t_0)$。那么，线 $W_i^{a_i-1}(t)$ 和线 $W_j^{a_j}(t)$

必然在某个时刻 $t \in [t_0, t_2]$ 处相交。那么 $t_1 > t_2 \geq \min(\mathbf{I}(A, t_0))$，这与在 t_1 时刻的假设不符。

由于 A 是 t_0 时刻的完全最优配置，那么有 $\max_{j \in \mathcal{T}} W_j^{a_j}(t_0) < W_i^{a_i-1}(t_0)(\forall i \in \mathcal{T}, a_i > 0)$。如果不存在目标对 i, j 在任一时刻 $t \in [t_0, t_e]$ 满足 $\frac{\partial W_i^{a_i-1}(t)}{\partial t} < \frac{\partial W_j^{a_j}(t)}{\partial t}$，那么 $\max_{j \in \mathcal{T}} W_j^{a_j}(t) \leq W_i^{a_i-1}(t)(\forall i \in \mathcal{T}, t \in [t_0, t_e])$。那么 A 直到 t_e 都是完全最优配置。类似的，如果不存在目标对 i, j 使得线 $W_i^{a_i-1}(t)$ 和线 $W_j^{a_j}(t)$ 在时间段 $[t_0, t_e]$ 内相交，A 直到 t_e 都是完全最优配置。 □

命题9说明，欲求解安保方的最优策略，只需考虑那些 PA 发生变化的时间点，并求解这些时间点上的新 PA。再次考虑示例 1 中的情形。图4.5显示，在时刻 0，$A = \langle 1, 0 \rangle$ 是最优配置（$W_1^0(0) > \max\{W_1^1(0), W_2^0(0)\}$）。$t_1$ 是线 $W_1^0(t)$ 和线 $W_2^0(0)$ 的交点，也是集合 $\mathbf{I}(A, 0)$ 中唯一的元素。所以 A 在 $[0, t_1]$ 内均为 PA。可以从图4.5中直接观测到，在时刻 t_1 之后，A 即不再是完全最优配置。接下来，探索如何在 PA 发生变化的时刻求解新的 PA。

命题 10 *如果配置 A 在时刻 t 是 PA，并且存在目标对 i, j 使得 $W_i^{a_i-1}(t) = W_j^{a_j}(t)$，那么如果一个配置 A' 满足 $a'_i = a_i - 1, a'_j = a_j + 1, a'_k = a_k(\forall k \in \mathcal{T}, k \neq i, j)$，$A'$ 也是时刻 t 的* PA。

证明 *由于只有分配给目标 i 的资源数目减少，分配给目标 j 的资源数目增加，而其他目标所享有的资源数目保持不变。为证明 A' 是完全最优配置，只需证明 $W_j^{a'_j-1}(t) \geq W_k^{a'_k}(t)$* and *$W_k^{a'_k-1}(t) \geq W_i^{a'_i}(t)(\forall k \in \mathcal{T}, a'_k > 0)$。首先，由于 $W_j^{a'_j-1}(t) = W_j^{a_j}(t) = W_i^{a_i-1}(t)$ 以及 $W_i^{a_i-1}(t) \geq W_k^{a'_k}(t)$，有 $W_j^{a'_j-1}(t) \geq W_k^{a'_k}(t)$。类似的，由于 $W_k^{a'_k-1}(t) \geq W_j^{a_j(t)}$ 和 $W_j^{a_j(t)} = W_i^{a_i-1}(t) = W_i^{a'_i}(t)$，有 $W_k^{a'_k-1}(t) \geq W_i^{a'_i}(t)$。* □

图4.5显示，在示例 1 中，$A = \langle 1, 0 \rangle$ 在时刻 t_1 是 PA。由于 $W_1^{a_1-1}(t_1) = W_1^0(t_1) = W_2^{a_2}(t_1) = W_2^0(t_1)$，$A' = \langle 0, 1 \rangle$ 在时刻 t_1 同样是 PA。对于 A'，在 t_1 之后，没有目标对 i, j 满足 $\frac{\partial W_i^{a_i-1}(t)}{\partial t} < \frac{\partial W_j^{a_j}(t)}{\partial t}$。所以直到 t_e，A' 都是 PA。

SCOUT-A 算法 (Scheduling seCurity resOurces in pUblic evenTs with no relocating delAy) 利用了上文中命题所讨论的性质，其主要思想如下：1) 在当前时间点计算 PA。2) 计算此 PA 失效的时间点。3) 将该时间点设置为当前时间点。重复这三步直到对于所有目标对 i, j，线 $W_i^{a_i-1}(t)$ 和线 $W_j^{a_j}(t)$ 在当前时间点到 t_e 内都不相交。

现在逐行解释算法 1 所示的 SCOUT-A 算法。行 1-7 计算时刻 0 的 PA。首先，所有的目标都没有被保护（第 1 行）。然后，资源被逐个分配给当前具有最高攻击方收益的目标（第 4-7 行）。第 8 行中的 t_m 用来记录当前 PA 失效的时间点。k 用来记录转移的次数。第 9-22 行计算下一次转移的时间点。如果在时刻 t_e 之前，任何 $W_i^{a_i-1}(t)$ 与 $W_j^{a_j}(t)$

Algorithm 8: SCOUT-A

1 **for** $i \in \mathcal{T}$ **do**
2 $\quad V_i \leftarrow v_i(0), a_i \leftarrow 0$
3 $left \leftarrow m$;
4 **while** $left > 0$ **do**
5 $\quad i \leftarrow \text{argmax}_{i \in \mathcal{T}} V_i$, a_i++, $left$- -, $V_i \leftarrow W_i^{a_i}(0)$;
6 $t_m \leftarrow 0, k \leftarrow 0$;
7 **while** $t_m < t_e$ **do**
8 $\quad \mathbf{I} \leftarrow \emptyset$;
9 $\quad$ **for** $\forall i, j \in \mathcal{T}$ **do**
10 $\quad\quad$ **if** $\exists t$ *such that* $\frac{\partial W_i^{a_i-1}(t)}{\partial t} < \frac{\partial W_j^{a_j}(t)}{\partial t}$ **then**
11 $\quad\quad\quad I_{ij} \leftarrow I_{ij}(A, t_m)$;
12 $\quad\quad\quad$ **if** $I_{ij} < t_e$ **then** $\mathbf{I} \leftarrow \mathbf{I} \cup I_{ij}$;
13 $\quad$ **if** $\mathbf{I} = \emptyset$ **then** break;
14 $\quad I_{ij} \leftarrow \min(\mathbf{I})$;
15 $\quad$ **if** $W_i^{a_i-1}(I_{ij} - \Delta t) \geq W_j^{a_j}(I_{ij} - \Delta t)$ **then**
16 $\quad\quad \tau_k \leftarrow I_{ij}, t_m \leftarrow I_{ij}, a_i - -, a_j + +, c_{ij}^k \leftarrow 1, k + +$;

均无交点，当前 PA 直到 t_e 都是 PA，算法结束（第 19-20 行），否则 I_{ij} 被用于记录其交点。第 23-25 行计算新的 PA。在第 23 行中，Δt 是一个比任何相邻的两个 $W_i^{a_i}(t)$ 和 $W_j^{a_j}(t)(\forall i, j \in \mathcal{T}, \forall a_i, a_j \in \{0, \ldots, m\})$ 的交点都要小的值。第 23 行检测是否该配置在 I_{ij} 之后仍为 PA。这是为了避免算法在某个时刻循环交换两种配置。由于 SCOUT-A 所求出的安保策略在任何时间点均为最优配置，因此该策略也即安保方的最优策略。

4.4 转移时间不可忽略的情形

本部分讨论更一般的转移时间不能忽略的情形。在此情形下，在某个时间点 t 可能有大于等于 1 个资源正在转移途中，因此每个时间点在使用中的资源数目不固定，SCOUT-A 就不能用来求解此情形。接下来，本部分从较简单的离散策略空间入手，进而推进到最具一般性的转移时间不可忽略且策略空间连续的情况。

4.4.1 离散策略空间

首先假设安保方只能在既定的、离散的时间点进行资源的转移，设时间点的集合为 $\Phi = \{t_k\}$。这样，一个资源到达任何一个目标的时间只可能在集合 $\phi = \{t_\delta : t_\delta = t_k + d_{ij}, \forall t_k \in \Phi, \forall i, j \in \mathcal{T}\}$ 中。于是，可以用一个有序的向量 $\Psi = \{t_\eta : t_\eta \in \Phi \quad \text{or} \quad t_\eta \in \phi\}$

来表示所有可能的有资源被转移或有资源到达某个新目标的时间点。令 η 表示 Ψ 中元素的索引值，如，$t_\eta \in \Psi$。令 $H = |\Psi|$。对于 $\eta \in \{1,\ldots,H\}$, 定义一个向量 $\sigma_\eta = \langle \sigma_{ij}^\eta : \sigma_{ij}^\eta \in \{1,\ldots,H\}\rangle$，其中 σ_{ij}^η 表示如果一个资源从目标 i 转移到目标 j, 而且预计到达 j 的时刻为 t_η，那么转移应该在时刻 $t_{\sigma_{ij}^\eta}$ 开始。引入两个新变量来表示某个时刻资源的分配情况：$a_i^{t_\eta}$，代表在时刻 t_η，尚没有资源被从目标 i 转移走时，目标 i 所拥有的资源数目；$b_i^{t_\eta}$，代表在时刻 t_η，所有预计在此刻开始从目标 i 转移的资源均已移出后，目标 i 所剩下的资源数目。那么，安保方的最优策略就可以通过如下的混合线性规划求得。将此混合线性规划命名为 SCOUT-D(Scheduling seCurity resOurces in pUblic evenTs with Discrete defender strategy space)。

$$\min\ U \tag{4-5}$$

$$\textbf{s.t.}\quad \sum\nolimits_{i\in\mathcal{T}} a_i^0 = m \tag{4-6}$$

$$a_i^{t_{k+1}} = b_i^{t_k} + \sum\nolimits_{j\in\mathcal{T}} c_{ji}^{\sigma_{ji}^{k+1}} \quad \forall k \in \{1,\ldots,H-1\} \tag{4-7}$$

$$b_i^{t_k} = a_i^{t_k} - \sum\nolimits_{j\in\mathcal{T}} c_{ij}^k \quad \forall k \in \{1,\ldots,H\} \tag{4-8}$$

$$c_{ij}^k \in \{0,1,\ldots\} \quad \forall i,j\in\mathcal{T}, \forall k\in\{1,\ldots,H\} \tag{4-9}$$

$$\sum c_{ij}^{t_\eta} = 0 \quad \forall t_\eta \in \phi \text{ and } t_\eta \notin \Phi \tag{4-10}$$

$$\sum c_{ij}^{\sigma_{ij}^\eta} = 0 \quad \forall t_\eta \in \Phi \text{ and } t_\eta \notin \phi \tag{4-11}$$

$$b_i^{t_k} \geq 0 \quad \forall i\in\mathcal{T}, \forall k\in\{1,\ldots,H\} \tag{4-12}$$

$$U \geq \max_{t\in[t_k,t_{k+1}]} W_i^{b_i^{t_k}}(t) \quad \forall i\in\mathcal{T}, \forall k\in\{1,\ldots,H-1\} \tag{4-13}$$

方程 (4-6) 保证了初始资源配置的有效性，即资源总数为 m 个。在方程 (4-7) 中，$\sum_{j\in\mathcal{T}} c_{ji}^{\sigma_{ji}^{k+1}}$ 代表在时刻 t_{k+1} 到达目标 i 的资源数目。类似的，在方程 (4-8) 中，$\sum_{j\in\mathcal{T}} c_{ij}^k$ 代表在时刻 t_k 由目标 i 转移到其他目标的资源数。方程 (4-9) 限定了转移资源的数量应该为整数。方程 (4-7)-(4-9) 共同限制了资源转移的可行性，类似于网络流问题中对于流可行性的限制。方程 (4-10) 处理那些属于 ϕ 但不属于 Φ 的时间点，在这些时间点安保方不能开始转移资源。类似的，方程 (4-11) 处理那些属于 Φ 但不属于 ϕ 的时间点，在这些时间点没有资源会到达新的目标。方程 (4-13) 用于限制攻击方做出最优反应，此处攻击方可以在任何时间进行攻击。由于 $v_i(t)$ 是连续函数，$\max_{t\in[t_k,t_{k+1}]} W_i^{b_i^{t_k}}(t)$ 总是存在，并可以被事先计算出作为此混合线性规划的输入。目标函数和方程 (4-13) 共同保障了即使攻击方做出最优的回应，安保方仍能取得最优的收益。

4.4.2 连续策略空间

安保方事实上可以在时间段 $[0,t_e]$ 内的任一时刻开始进行资源转移。值得注意的是，并不是任何时刻的转移都会带来好的收益，并且，如果把一些转移提前或延迟到另一

些时刻进行，安保方的收益可能并不会变化。也就是说，有可能可以通过合理的设置时间集 Φ，使得在上一章中提出的混合线性规划求出的解同样是拥有无限策略空间的安保方的最优解。在本章中，首先证明，对于任何一个安保方拥有无限策略空间的博弈，必然存在一个与之等价的博弈，其中安保方只能在某些既定的时间点进行资源转移。然后，提出一个算法来求解这些时间点，那么所求得的时间集即可作为混合线性规划的输入，以求出原博弈 (即安保方拥有无限空间的博弈) 的最优解。

首先，将重要性函数分割成一系列的单调区间。对于每个目标 i，令 $\xi_1^i, \xi_2^i, \ldots, \xi_{R_i}^i$ 按序的表示所有函数 $v_i(t)$ 的单调性发生变化的时刻。令 $\xi_0^i = 0$ 并且 $\xi_{R_i+1}^i = t_e$。定义一个集合 $\Xi^i = \{\xi_\rho^i : \rho \in \{0, \ldots R_i + 1\}\}$，其中 ρ 是用来表示某个特定时间点的索引，$\xi_\rho^i \in \Xi^i$。那么，在任意两个在 Ξ^i 中相邻的时间点所代表的时间段内，$v_i(t)$ 是单调的。定义 $\Xi = \{\Xi^i : \forall i \in \mathcal{T}\}$。

如果说，在一个安保方具有无限策略空间的博弈中，任何一个安保方的最优策略 S，都可以被转化成另一个同样是最优的策略，其中转移只开始在某些特定的时间点，那么就可以先求出这些特定的时间点，并将其作为混合线性规划的输入。显然，在此情况下，混合线性规划返回的最优解也即原始博弈的最优解。接下来，讨论如何将原博弈的最优策略 S 进行转化。首先，可以证明，将一些转移进行合并并不会影响安保方的收益。

定理 4.1 *如果在策略 S 中，一个资源在时刻 t_1 被从目标 i 转移到目标 j，随后在时刻 t_2 被从目标 j 转移到目标 l，如果 $t_1 + d_{ij} \in [\xi_\rho^j, \xi_{\rho+1}^j)$ 并且 $t_2 \in [t_1 + d_{ij}, \xi_{\rho+1}^j)$，那么如果合并这两次转移，即，直接在时刻 $t_3 \in [t_1, t_2 + d_{jl} - d_{il}]$ 将资源从目标 i 转移到目标 l，安保方的收益不会降低。*

证明 *假设在初始策略 S 中，在 t_1 时刻的转移发生前，分布在目标 i, j, l 上的资源数量分别是 a_i, a_j, a_l。那么在策略 S，这三个目标拥有的资源数量随时间变化的情况即如表4.1所示。*

时间段	$q_i^t(S)$	$q_j^t(S)$	$q_l^t(S)$
t_1 时刻前不久	a_i	a_j	a_l
$[t_1, t_1 + d_{ij})$	$a_i - 1$	a_j	a_l
$[t_1 + d_{ij}, t_2)$	$a_i - 1$	$a_j + 1$	a_l
$[t_2, t_2 + d_{jl})$	$a_i - 1$	a_j	a_l
$t_2 + d_{jl}$ 时刻后不久	$a_i - 1$	a_j	$a_l + 1$

表 4.1 策略 S 中目标 i, j, l 的资源数量

值得注意的是，应有 $d_{il} \le d_{ij} + d_{jl}$，否则安保方可将 l 作为中转站来从目标 i 向目标 l 转移资源。于是，总是存在时间点 $t_3 \in [t_1, t_2 + d_{jl} - d_{il}]$。现在构造一个安保策略 S^1。

在 S^1 中，除资源直接在 t_3 时刻从目标 i 转到目标 l 之外，其他均与 S 相同。那么，目标 i,j,l 在 S^1 中的资源数量随时间变化情况即如表4.2所示。

时间段	$q_i^t(S)$	$q_j^t(S)$	$q_l^t(S)$
t_3 时刻前不久	a_i	a_j	a_l
$[t_3, t_3+d_{il})$	a_i-1	a_j	a_l
t_3+d_{il} 时刻后不久	a_i-1	a_j	a_l+1

表 4.2　策略 S^1 中目标 i,j,l 的资源数量

由表4.1和表4.2所示可知，只有在时间段 $[t_1+d_{ij}, t_2)$ 内，S^1 中分配给目标 j 的资源数量少于 S 中分配给 j 的资源数量。其他的任何时刻，S^1 中任何目标所拥有的资源数均不少于其在 S 中拥有的资源数。也就是说，仅当 $\max_{t\in[t_1+d_{ij},t_2)} W_j^{q_t^j(S^1)}(t) > U^a(f_{tg}(S), f_{tm}(S), S)$ 时，S^1 带来的安保方收益才会小于 S。然而，由于 $v_j(t)$ 在时间段 $[\xi_\rho^j, \xi_{\rho+1}^j]$ 内单调，那么 $\max_{t\in[t_1+d_{ij},t_2)} W_j^{a_j}(t) \le \max\{W_j^{a_j}(t_1+d_{ij}), W_j^{a_j}(t_2)\} \le U^a(f_{tg}(S), f_{tm}(S), S)$。那么合并转移以后的策略 S^1 所带来的安保方收益不会少于 S。 □

基于定理4.1，可以将一个原博弈中的任一最优策略 S 转化为一个新的最优策略 S^1。接下来，证明可以通过替换部分 S^1 中的转移得到一个新的最优策略，其中转移只发生在特定的时间点。假设在 S^1 中，一个资源在时刻 $t_1 \in [\xi_\rho^i, \xi_{\rho+1}^i)$ 被从目标 i 转移出，并在时刻 $t_2 \in [\xi_{\rho'}^j, \xi_{\rho'+1}^j)$ 到达目标 j。假设在 t_1 时刻之前不久，目标 i 所拥有的资源数目为 a_i。假设在 t_2 时刻之前不久，目标 j 所拥有的资源数量为 a_j。那么 $Tr=(i,j,a_i,a_j,\rho,\rho')$ 即可表示一次转移。为了探索如何替换 Tr，首先定义一个和 Tr 相关的特定时间点 $\theta(Tr)$。

$$\theta(Tr) = \arg \min_{t\in[\xi_\rho^i, \min\{\xi_{\rho+1}^i, \xi_{\rho'+1}^j - d_{ij}\}]} \Big(\max_{t'\in[t,t+d_{ij}]} \{W_i^{a_i-1}(t'), W_j^{a_j}(t')\} \Big). \tag{4-14}$$

$\theta(Tr)$ 的含义是，已知目标 i 所拥有资源数量为 a_i，目标 j 所拥有资源数量为 a_j，如果一个资源在时间段 $[\xi_\rho^i, \xi_{\rho+1}^i]$ 被从目标 i 转出，并在时间段 $[\xi_{\rho'}^j, \xi_{\rho'+1}^j]$ 到达目标 j，那么如果将此转移改为在时刻 $\theta(Tr)$ 开始，如果攻击方选择在此资源转移途中攻击目标 i 或 j，那么攻击方所获得的最大收益将被最小化。基于 $\theta(Tr)$ 的定义，有如下关于 S^1 的转化的定理。

定理 4.2 如果将 S^1 中所有转移 Tr 的开始时间都移至 $\theta(Tr)$，安保方的收益不变。

证明 假设在策略 S^1 中，转移 Tr 的开始时间为 t_1。构造一个策略 S^2，其中转移 Tr 开始的时间均被移至 $\theta(Tr)$。接下来，在 $t_1 < \theta(Tr)$ 的假设下证明定理4.2。如果 $t_1 \ge \theta(Tr)$，证明过程相似。当 $t_1 < \theta(Tr)$，仅当 $t \in (t_1+d_{ij}, \theta(Tr)+d_{ij})$，$S^2$ 中分配给目标 j 的资

源数比 S^1 少一个。对于其他任何目标 k，在任何时间点 t，均有 $q_k^t(S^2) \geq q_k^t(S^1)$。那么 $\max_{t\in(t_1+d_{ij},\theta(Tr)+d_{ij})} W_j^{q_t^j(S^2)}(t) > U^a(f_{tg}(S^1), f_{tm}(S^1), S^1)$ 时，S^1 才会比 S^2 带来更高的安保方收益。

给定 $\theta(Tr)$ 的定义，可知 $\theta(Tr) + d_{ij} \in [\xi_{\rho'}^j, \xi_{\rho'+1}^j]$。$S^1$ 的性质保证了在时间段 $[t_1+d_{ij}, \xi_{\rho'+1}^j]$ 内，没有资源被从目标 j 转出。令 ω 代表时间段 $[t_1 + d_{ij}, \theta(Tr) + d_{ij}]$, 那么当 $t \in \omega$，有 $q_j^t(S^2) \geq a_j$。于是，$\max_{t\in\omega} W_j^{q_t^j(S^2)}(t) \leq \max_{t\in\omega} W_j^{a_j}(t)$。为证明 $\max_{t\in\omega} W_j^{a_j}(t) \leq U^a(f_{tg}(S^1), f_{tm}(S^1), S^1)$，将以下两种情况分开讨论。

情况 **1**：$(t_1 + d_{ij} \geq \theta(Tr))$。

在此情况下，有

$$\begin{aligned}\max_{t\in\omega} W_j^{a_j}(t) &\leq \max_{t\in[\theta(Tr),\theta(Tr)+d_{ij}]} W_j^{a_j}(t) \\ &\leq \max_{t\in[\theta(Tr),\theta(Tr)+d_{ij}]} \{W_i^{a_i-1}(t), W_j^{a_j}(t)\}.\end{aligned}$$

基于 $\theta(Tr)$ 的定义，可知

$$\begin{aligned}&\max_{t\in[\theta(Tr),\theta(Tr)+d_{ij}]} \{W_i^{a_i-1}(t), W_j^{a_j}(t)\} \\ &\leq \max_{t\in[t_1,t_1+d_{ij}]} \{W_i^{a_i-1}(t), W_j^{a_j}(t)\} \\ &\leq U^a(f_{tg}(S^1), f_{tm}(S^1), S^1).\end{aligned}$$

于是 $\max_{t\in\omega} W_j^{q_t^j(S^2)}(t) \leq U^a(f_{tg}(S^1), f_{tm}(S^1), S^1)$，也即策略 S^2 带来的安保方收益不少于策略 S^1。

情况 **2**：$(t_1 + d_{ij} < \theta(Tr))$。

如果 $\max_{t\in[t_1+d_{ij},\theta(Tr)]} W_j^{a_j}(t) \leq \max_{t\in[\theta(Tr),\theta(Tr)+d_{ij}]} W_j^{a_j}(t)$，那么不难推知

$$\max_{t\in\omega} W_j^{a_j}(t) \leq \max_{t\in[\theta(Tr),\theta(Tr)+d_{ij}]} W_j^{a_j}(t),$$

那么就可通过情况 1 的方法证明 $\max_{t\in\omega} W_j^{a_j}(t) \leq U^a(f_{tg}(S^1), f_{tm}(S^1), S^1)$。否则，由于函数 $v_j(t)$ 在时间段 $[\xi_{\rho'}^j, \xi_{\rho'+1}^j]$ 内单调，当 $t \in [\xi_{\rho'}^j, \xi_{\rho'+1}^j]$，有 $\frac{\mathrm{d}W_j^{a_j}(t)}{\mathrm{d}t} \leq 0$。那么，

$$\begin{aligned}\max_{t\in\omega} W_j^{a_j}(t) &\leq W_j^{a_j}(t_1 + d_{ij}) \\ &\leq U^a(f_{tg}(S^1), f_{tm}(S^1), S^1).\end{aligned}$$

证明完毕。 □

接下来，描述对于任意一个安保方有连续无限策略空间的博弈，求解其最优解的

算法。首先，定义一个集合

$$\Theta = \{\theta(Tr) : Tr = (i, j, a_i, a_j, \rho, \rho'), \forall i, j \in \mathcal{T}, \forall a_i, a_j \in \{0, \dots m\}, \\ \forall \rho \in \{0, \dots, R_i\}, \forall \rho' \in \{0, \dots, R_j\}\}.$$

定理4.2显示可将任一安保策略 S^1 转化为另一个策略 S^2 而不影响安保方的最优收益，在 S^2 中，所有的转移仅在时间点 $t \in \Theta$ 开始。也就是说，只要原始博弈，即安保方有连续无限策略空间的博弈，存在一个最优策略 S，那么它还必然存在一个最优策略，其中转移仅在时间点 $t \in \Theta$ 开始。提出算法 SCOUT-C(Scheduling seCurity resOurces in pUblic evenTs against Continuous strategy space) 来求解这一最优策略。SCOUT-C 首先计算出在一个最优策略中所有转移可能开始的时间点和所有转移可能结束的时间点（第 3 行）。这些时间点被记录在一个集合 Ψ 中。接下来，Ψ 被用作混合线性规划 SCOUT-D 的输入，而如前所述，SCOUT-D 所返回的最优策略，同样是原始博弈，即安保方拥有无限连续策略空间博弈的最优解。

Algorithm 9: SCOUT-C

```
1 Ψ ← ∅
2 for i, j ∈ T do
3     for a_i ∈ {0,...,m}, a_j ∈ {0,...,m} do
4         for ρ ∈ {0,...,R_i}, ρ' ∈ {0,...,R_j} do
5             Ψ ← Ψ ∪ {θ(Tr)} ∪ {θ(Tr) + d_ij}□ 其中 Tr = (i, j, a_i, a_j, ρ, ρ')
6 令 Ψ 作为时间点集，运行 SCOUT-D
```

4.5 实验与分析

分别在完全随机数据和基于真实场馆分布的模拟数据上测试了文章提出的算法。首先介绍随机数据下的实验结果。在随机数据中，生成重要性函数时，对于每个目标，首先在时间段 $[0, t_e]$ 随机选择一段，要求其中重要性函数的值不为零。然后随机产生该目标的重要性函数对应的线性阶段数，再生成每个阶段对应的线性函数。假设一个目标的重要性函数值在 $[0, 100]$ 以内。图4.6 展示了一个随机生成的 4 个目标的重要性函数示例，其中 t_e 被设为 10，x 轴表示时间，y 轴表示重要性函数的值。

首先检测在不同的 λ 值和不同的转移时间下，算法对于安保方收益的影响。为便于展示，用攻击方的收益来衡量安保方收益。由于安保方收益与攻击方收益相反，越低的攻击方收益对应越高的安保方收益。为了更好的衡量本章算法的有效性，同时展示了另外两个算法，SDS (Static Defender Strategy) 和 DDS (Dynamic Defender Strategy)，所带来的攻击方收益。在 SDS 中，安保方使用静态策略，即，每个目标被分配的资源

数目与其重要性函数的最大值成正比，在赛事进行的整个过程中，资源不发生转移。类似的静态策略被广泛应用于之前关于安全博弈的研究工作 [32,121]。在 DDS 中，假设时间被时间分割为离散的点 $\Phi = \{\frac{n \cdot t_e}{4} : n \in \{0, 1, \ldots 4\}\}$，并用 SCOUT-D 求解在此假设下的安保方最优策略。

图4.7展示了在不同的 λ 值下，SCOUT-A, SCOUT-C, DDS 和 SDS 所带来的攻击方收益。横轴对应三种 λ 值，纵轴对应攻击方收益。随着 λ 值的增大，四种算法带来的攻击方收益都减少，也即安保方收益都增加，这是由于越大的 λ 值说明安保资源的保护效力越大，安保方更容易获得好的收益。值得注意的是，不论 λ 取何值，SCOUT-C 和 SCOUT-A 所对应的攻击方收益都明显小于 DDS，而 SDS 带来的攻击方收益都是四种算法中最高的。这说明 SCOUT-C 和 SCOUT-A 所产生的安保策略要优于 DDS，而动态的安保策略又要优于静态的安保策略。

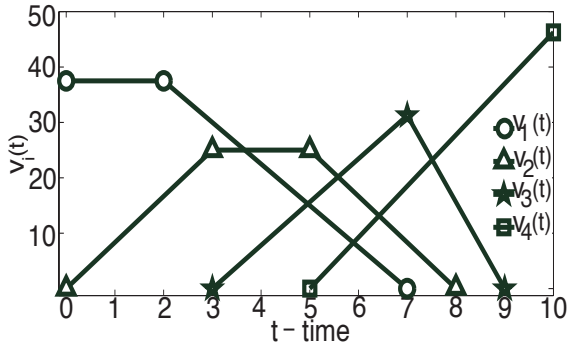


图 4.6 阶段线性重要性函数

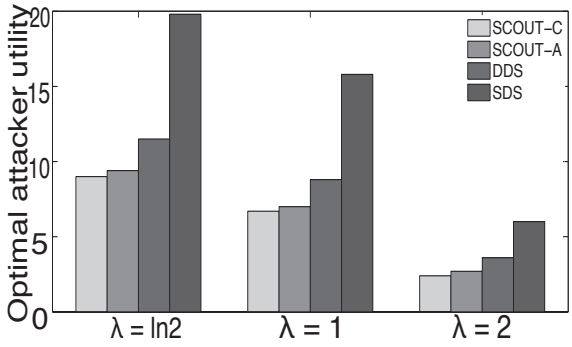


图 4.7 改变 λ 值

图4.8展示了在不同的转移时间级别下四种算法所带来的攻击方收益。横轴对应三种不同等级的转移时间：对于 Level 1，任何两个目标之间的转移时间，即 d_{ij}，被认为是在 $[0, 0.1]$ 内均匀分布；对于 Level 2，d_{ij} 被认为在 $[0, 1]$ 内均匀分布；对 Level 3，d_{ij} 则被认为在 $[0, 5]$ 内均匀分布。纵轴表示四种算法所带来的攻击方收益。无论转移时间在何种级别，SCOUT-C 都带来最低的攻击方收益，SDS 都带来最差的安保策略。图4.9展示了在不同的转移时间范围内，SCOUT-A 与 SCOUT-C 带来的攻击方收益的差异。如果4.8，横轴对应三种转移时间等级，而纵轴表示攻击方收益的差值。当转移时间很小时，两种算法的差别也很小。当转移时间增加，两种算法的差距也变大。

还检测了算法的运行时间随博弈规模增加而变化的情况。图4.10展示了两种算法的运行时间。其中 x-轴表示 4 组目标数/资源数的组合情况，如 8/10 表示该博弈中共有 8 个目标需要保护，共有 10 个安保资源可供分配。y-轴为运行时间。SCOUT-A 表现出了更好的可扩展性。图4.11展示了在更大规模的博弈上 SCOUT-A 的运行时间。横轴表示目标数目，三条曲线分别表示安保资源数为 10,20,30。即使目标数目达到 100 而资源数目达到 30，SCOUT-A 仍能在 30 分钟内求解出安保方的最优策略。

为了检测算法的扩展潜能，考虑了形式更一般的重要性函数，即，重要性函数可为

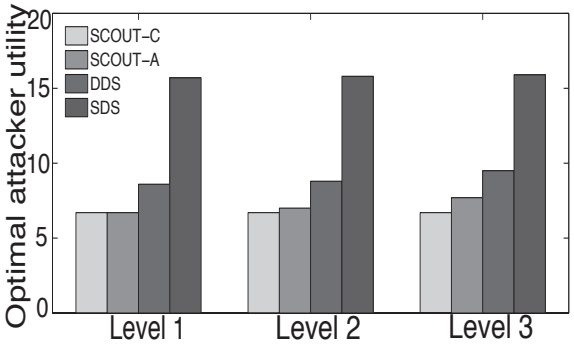


图 4.8　改变转移时间

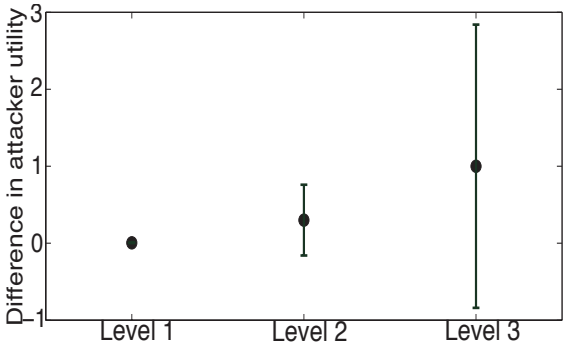


图 4.9　SCOUT-C 与 SCOUT-A 的差别

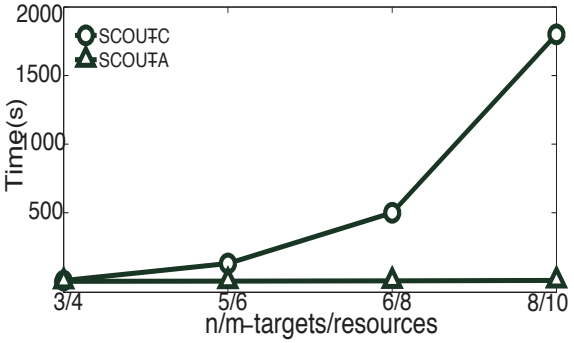


图 4.10　运行时间

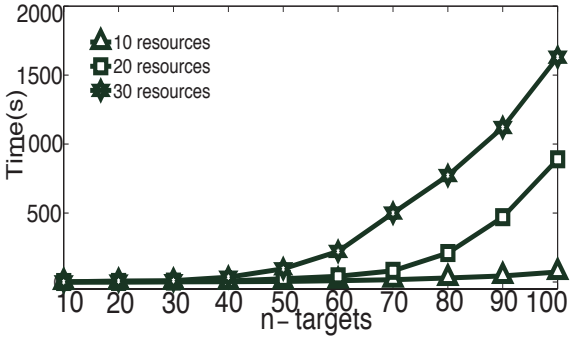


图 4.11　SCOUT-A 可扩展性

分段二次函数。图4.12展示了随机生成的 4 个目标的阶段二次函数形式的重要性函数。图4.13展示了在此类函数形式下，SCOUT-A，SCOUT-C, SDS 和 DDS 分别对应的攻击者收益情况。结果显示，即使在更一般的重要性函数形式下，SCOUT-C 和 SCOUT-A 依然带来比 SDS 和 DDS 更优的安保策略。

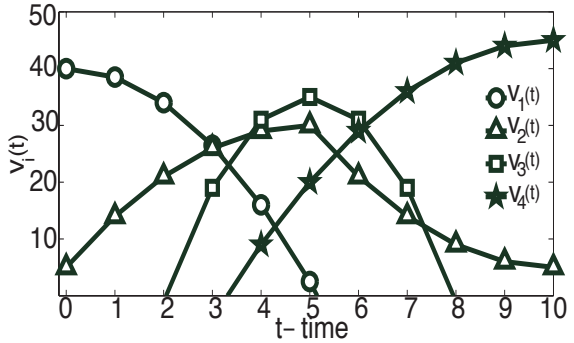


图 4.12　更普适的重要性函数

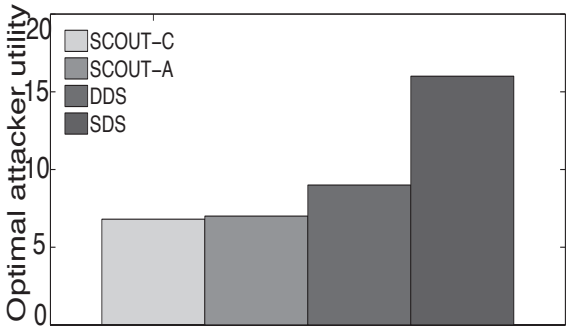


图 4.13　攻击方收益

最后，本部分还在真实场馆的地形分布上测试了算法的效果。图4.14展示了即将到来的 2022 年北京冬奥会中，北京城区的五个主要场馆的分布情况。表4.3展示了在这些场馆之间转移所需要的时间。由于当前比赛日程并未公布，仍使用随机生成的数据来模拟各场馆的重要性变化情况。假设所有场馆的赛事进行时间均为 10 小时，以分钟为时间单位，即 $t_e = 600$。同时，以随机实验中所采用的方法为各个场馆生成阶段线性的重要性函数。假设有 10 个巡逻队可在这 5 个场馆间机动转移。图4.15展示了四种

	鸟巢	奥林匹克公园	首体	工体	五棵松
鸟巢	-	8	14	17	29
奥林匹克公园	8	-	20	24	33
首体	14	20	-	24	17
工体	17	24	24	-	36
五棵松	29	33	17	36	-

表 4.3　场馆间转移时间（分钟，来自谷歌地图的估算）

算法所带来的收益情况。类似于图4.7，横轴表示 λ 的数值，纵轴表示攻击方的最优收益。SCOUT-C 的表现仍然明显优于 SDS 和 DDS。事实上，由于场馆间的转移时间远小于赛事的总进行时间，SCOUT-C 所求得的攻击方收益仅仅略差于不考虑转移时间的 SCOUT-A 的效果。

图 4.14　北京冬奥会主要场馆 (图片来自申奥委官网)

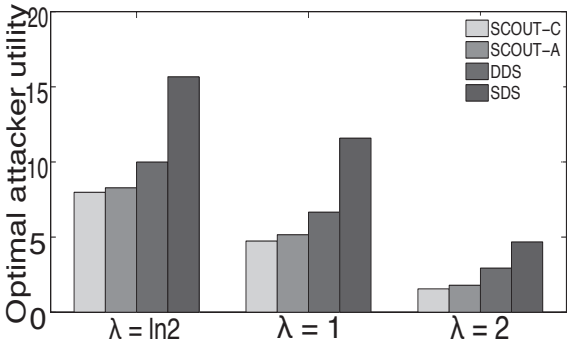


图 4.15　攻击方收益

4.6　本章小结

为重大赛事设计高效的安保策略是具有重大现实意义的议题。然而，由于在重大赛事中，可能被攻击的目标的重要程度是随时间变化的，而可行的安保策略以及安保方可能面对的攻击策略都数量巨大，因此设计最优的安保策略在技术上极具挑战性。本章用博弈模型来描述重大赛事的安保问题，在模型中既考虑了目标重要性的动态变化，也考虑了攻击方与安保方连续无限的策略空间。基于此模型，本章考虑了转移时间可忽略不计，转移时间不可忽略不计的两种情况。针对第一种情况，本章提出算法 SCOUT-A 求解安保方的最优收益。针对第二种情况，本章从离散时间点着手，提出了在此假设下求解最优安保策略的算法 SCOUT-D，进而考虑了更为复杂的连续时间的情形，提出了基于 SCOUT-D 的求解最优安保策略的算法 SCOUT-C。实验结果表明，SCOUT-A 和 SCOUT-C 所产生的安保策略均优于此前研究中不考虑目标重要性动态变化时所生成的安保策略。

第五章　动态收益安全博弈混合策略均衡

5.1　引言

近年来，随着恐怖活动日益猖獗，城市的公共设施、旅游景点、商场中心等场所都成为了不法分子可能攻击的目标。由于恐怖分子日趋专业化与智能化，他们往往能够利用安保策略中的漏洞，进行策略性地攻击。因此，在设计安保策略时，不仅需要考虑各潜在目标的重要程度，也需要考虑不法分子对于安保策略的观测与回应。

城市安保问题的一项挑战在于，可能被攻击的目标的重要程度是随时间变化的，所以有限的安保资源需要在安保过程中被动态配置，以达到最优的安保效果。例如，博物馆的入口处通常在上午较为拥挤，使得此处的重要程度较高；而酒吧街等娱乐场所则在入夜后迎来客流量高峰，需要安保部门提高警惕。直观上讲，为了使得有限的安保资源发挥出最大的效用，安保部门应在早上着重保护博物馆，在白天的某个时间将资源转移至酒吧街，以使其在晚上得到保护。而与重大赛事安保不同的是，城市的安保每天都在进行，如果安保策略存在固定形态，就很容易被不法分子利用。因此，本章着重研究为城市安保这类常规进行的、目标重要性随时间变化的问题，设计最优的随机化安保策略。

自博弈论被应用于研究安全问题以来，关于安全博弈的研究备受瞩目，许多基于安全博弈理论的系统也被成功应用。然而，多数之前的研究工作都假设目标的重要性程度是固定不变的。虽然有些工作考虑到了动态收益的情形，他们大多假设安保部门或不法分子的策略空间是离散化的，即，安保部门只能在事先确定的时间点上进行资源转移。关于重大赛事安保问题的研究虽然考虑了动态收益和连续策略空间，但由于重大赛事的稀缺性，在技术上仅考虑纯策略均衡的求解方法。因此，此前工作所提出的模型和算法都不能直接被用于解决城市安保问题。

本章的贡献主要有以下四点：

- 提出斯塔克伯格博弈模型对问题建模，在模型中考虑博弈参与双方的连续策略空间以及目标重要程度的连续变化，以强斯塔克伯格均衡解为均衡策略。
- 提出算法 COCO (Computing Optimal Coverage withOut transfer time)，以计算当安保资源可在目标之间无延迟转移时安保方的最优策略。COCO 以新颖的方式紧凑的表示安保方的策略空间，避免了求解过程中对无限策略空间的遍历。
- 针对 COCO 的结果，提出一种算法进行纯策略采样，以获取可用于每天执行的、每次执行过程中仅需进行有限次资源转移的安保策略。
- 提出算法 ADEN (computing Approximately optimal DEfender strategies in games

with Nonzero transfer time)，以计算当资源在目标之间的转移时间不能忽略时的最优安保策略。首次提出能够保证近似度（ϵ-近似）的连续策略离散方法。

5.2 模型

假设共有 n 个目标 $\mathcal{T}=\{1,\ldots,n\}$，安保部门有 m 个相同的安保资源（$n \geq m$）。每次安保策略的执行时间段是 $[0,t_e]$（例如，可映射到每天的早八点到晚十二点）。目标的重要性在这段时间内连续变化。以一个连续函数 $v_i(t)(i \in \mathcal{T}, t \in [0,t_e], v_i(t) > 0)$ 来描述目标 i 在时间 t 的重要性，并假设该函数为阶段线性函数[①]。同此前关于安全博弈的研究类似，以斯塔克伯格博弈对问题建模，建设安保方先行动并确定一个随机化策略，攻击方能够完全知晓此策略并作出最优回应。

攻击方的一个纯策略 S 包含在策略开始的时刻（时刻 0）时的资源分布情况以及在策略执行过程中（时间段 $[0,t_e]$ 内）的策略转移情况。令

$$S=\langle (i,j,t): i,j \in \mathcal{T}, t \in [0,t_e]\rangle, \tag{5-1}$$

其中 (i,j,t) 表示在时刻 t，有一个资源被从目标 i 转移到了目标 j。（如果目标 i 在初始时刻被保护，那么有 $(i,i,0) \in S$）。令 $D=\langle d_{ij}: i,j \in \mathcal{T}\rangle$，其中 d_{ij} 代表资源从目标 i 转移至目标 j 所需的时间。定义一个二值函数 $q_i(S,t)$，令其指示当安保策略为 S 时，目标 i 在时刻 t 是否被保护。给定 S，$q_i(S,t)$ 在 $t \in [0,t_e]$ 是一个阶段常数函数。令 $\mathcal{S}$ 表示安保方的策略空间。由于安保资源可以在 $[0,t_e]$ 内的任何时刻被转移，$\mathcal{S}$ 是连续的。安保方的混合策略可表示为 $\mathbf{x}=\langle x_S : S \in \mathcal{S}\rangle$。其中 x_S 表示纯策略 S 被使用的概率。令 $A=(i,t_a)$ 表示攻击方的一个纯策略，即在时刻 t_a 发动对目标 i 的攻击。

假设攻击方的目的是造成尽可能大的破坏，而防守方，也就是安保方的目的是使破坏尽可能小。那么显然，博弈为零和的。同样假设攻击方可能造成的破坏的程度由攻击目标的价值决定。如果攻击方使用了策略 $A=(i,t_a)$，而安保方在时刻 t_a 没有保护目标 i，即安保方使用了 $q_i(S,t_a)=0$ 的策略 S，那么攻击就能够成功，攻击方能够获得收益 $v_i(t_a)$，安保方的收益则相应为 $-v_i(t_a)$。否则，若安保方使用了 $q_i(S,t_a)=1$ 的策略 S，那么攻击不能成功。这种情况下，令双方收益均为 0。即攻击方不会获得收益，安保方也不会面临损失。给定一个安保方的混合策略 $\mathbf{x}$ 和一个攻击方的纯策略 $A=(i,t_a)$，攻击方的期望收益可表示为

$$U_a(\mathbf{x},A)=v_i(t_a)\cdot\left(1-\int_{\mathcal{S}} x_S \cdot q_i(S,t_a)\right). \tag{5-2}$$

于是防守方的收益为 $U_d(\mathbf{x},A)=-U_a(\mathbf{x},A)$。在零和条件下，斯塔克伯格均衡等价于纳什均衡，也等价于最大最小均衡，即，均衡条件下，防守方最大化自己的最小收

① 阶段线性函数被广泛用于模拟复杂的函数形式，其精度可通过调整线段个数来控制

益，也就是在考虑到攻击方可能进行最优回应的情况下最大化自己的收益。技术上讲，这等价于最小化攻击方的最大收益。

由于安保部门策略空间的连续性，直接表示 $\mathbf{x}$ 较为困难。因此，本节利用一种紧凑的方式来表示安保部门的混合策略。给定 $\mathbf{x}$，对于任一攻击目标 $i \in \mathcal{T}$，定义一个覆盖函数 $c_i(t)$，该函数表示目标 i 在时刻 t 被保护的概率。

$$c_i(t) = \int_{\mathcal{S}} x_S q_i(S,t). \tag{5-3}$$

$c_i(t)$ 是定义域为 $t \in [0,t_e]$，值域为 $[0,1]$ 的函数。一个混合策略 $\mathbf{x}$ 可由一个覆盖函数组成的覆盖向量 $\mathbf{c}(t) = \langle c_i(t) : i \in \mathcal{T}, t \in [0,t_e] \rangle$ 表示。给定安保方混合策略 $\mathbf{x}$ 和攻击方纯策略 $A = (i,t_a)$，攻击方的收益同样可以用覆盖函数表示，即

$$U_a(\mathbf{c}(t), A) = v_i(t_a) \cdot (1 - c_i(t_a)). \tag{5-4}$$

5.3 转移时间可忽略的情形

本节首先从一个简单情况入手，以解决该问题。在该情形下，安保资源在各目标之间的转移时间可以忽略，即 $d_{ij} = 0$。应对于现实中的场景，这个假设主要对应于资源的转移时间相对于策略的执行时间可忽略不计的情形。下一节将放松这个假设并考虑更具一般性的情况。本节首先提出一种算法，在转移时间可忽略的情形下计算最优的覆盖向量，然后提出一种方法，根据最优覆盖向量，进行单次策略的抽样，以获取能够执行的策略。

5.3.1 计算最优覆盖向量

如果转移时间是可忽略的，那么在时间段 $[0,t_e]$ 内的任何一个时刻都可以被看做一个独立的时刻，任何一个时刻都有 m 个资源正在发挥作用。于是，通过计算各个时刻的" 本地最优" 覆盖向量，就可以计算出总体的最优覆盖向量。技术上讲，当一个在时刻 t_0 的覆盖向量 $\mathbf{c}(t_0)$ 满足以下条件时，其是当前时刻的本地最优覆盖向量。

$$\max_{i \in \mathcal{T}} v_i(t_0)(1 - c_i(t_0)) \leq \max_{i \in \mathcal{T}} v_i(t_0)(1 - c'_i(t_0)), \forall \mathbf{c}'(t_0).$$

上式意味着，如果攻击方在时刻 t_0 发动攻击，那么本地最优覆盖向量使得攻击方能够获得的最大收益最小。接下来，探讨如果计算本地最优向量。令 I 表示一个 “攻击集合”。给定时刻 t_0 和覆盖向量 $\mathbf{c}(t_0)$，攻击该集合内的目标给攻击方带去最大的收益。也就是说，

$$v_i(t_0)(1 - c_i(t_0)) = v_j(t_0)(1 - c_j(t_0)), \forall i,j \in I; \tag{5-5}$$

$$v_i(t_0)(1 - c_i(t_0)) > v_j(t_0)(1 - c_j(t_0)), \forall i \in I, j \in \mathcal{T} \setminus I. \tag{5-6}$$

于是，关于本地最优覆盖向量有以下引理。

引理 1 一个覆盖向量 $\mathbf{c}(t)$ 是 t_0 时刻的本地最优向量当且仅当其满足

1. $\sum_{i\in I} c_i(t_0) = m$,
2. $c_j(t_0) = 0, \forall j \in \mathcal{T} \setminus I$,

证明 $\Leftarrow$ 方向已被 Kiekintveld et al[55] 证明，接下来通过反证法证明 $\Rightarrow$ 方向。假设存在一个 t_0 时刻的本地最优覆盖向量不满足上述方程，即，$\sum_{i\in I} c_i(t_0) < m$。由于 $m \le n$ 以及 $\forall i \in \mathcal{T}, v_i(t) > 0$，有 $\forall i \in I, c_i(t_0) < 1$。于是安保方总是可以降低对于目标 $j \in \mathcal{T} \setminus I$ 的保护并增加对于目标 $i \in I$ 的保护，以降低攻击方能够获得的最大收益，直到 $\sum_{i\in I} c_i(t_0) = m$（在此过程中 I 的大小可能会变化）。 □

接下来考虑计算时刻 $t \in [0, t_e]$ 的最优覆盖向量的方法。首先假设所有目标在所有时刻均处在攻击集合内并计算此时的覆盖向量。显然该假设高估了攻击集合的大小，并可能导致不可行的覆盖向量，例如，有些目标在有些时间段内需要被以负的概率保护。因此，下一步是找出覆盖向量内不可行的部分并加以调整。以下示例简要介绍了直观的计算过程。

示例 1 假设 $t_e = 2$，假设共有 3 个目标 1 个资源。假设目标的重要性函数分别为

$$\begin{cases} v_1(t) = t, \\ v_2(t) = -t + 10, \\ v_3(t) = -0.5t + 5. \end{cases}$$

首先，假设所有目标在所有时刻均处于攻击集合内。根据引理1和攻击集合的定义，控制覆盖函数满足

$$\begin{cases} \sum_{i\in\{1,2,3\}} c_i(t) = 1, \\ v_1(t)\cdot(1-c_1(t)) = v_2(t)\cdot(1-c_2(t)) = v_3(t)\cdot(1-c_3(t)). \end{cases}$$

于是，通过重要性函数 $v_i(t)$，可解得覆盖方程的表达式。根据解出的覆盖方程的表达式，易知对于 $t \in [0,2], c_1(t) < 0, c_2(t) > 0, c_3(t) > 0$。这意味着对于 $[0,2]$ 时间段内，目标 1 不需要被保护。于是，令 $c_1(t) = 0$ 对于 $t \in [0,2]$，并将目标 1 从攻击集合移除。与上一个步骤相同，计算仍处于攻击集合内的目标 2 和 3 的覆盖函数。

$$\begin{cases} \sum_{i\in\{2,3\}} c_i(t) = 1 \\ v_2(t)\cdot(1-c_2(t)) = v_3(t)\cdot(1-c_3(t)). \end{cases}$$

新的函数满足 $c_2(t) \ge 0, c_3(t) \ge 0, \forall t \in [0,2]$。于是覆盖向量 $\mathbf{c} = \langle c_i(t)\rangle$ 在任一时刻 $\forall t \in [0,2]$ 均为本地最优覆盖向量，由于其在任一时刻均满足引理1。

算法 1 描述了 COCO (Computing Optimal Coverage withOut transfer time) 的运行过程。COCO($I, b = 0, e = t_e$) 用于计算时间段 $[b, e)$ 内，给定攻击集合 I 时的覆盖函数。算法首先调用 COCO($I = \mathcal{T}, b = 0, e = t_e$)，并计算攻击集合内的目标的覆盖函数（第 2 行）。E 代表覆盖函数的根组成的向量，其中的元素升序排列，意即，t_k 代表在定义域 (b, e) 内第 k 大的根（第 3 行）。如果 E 是空集，第 5 行检测在定义域 (b, e) 内覆盖函数的符号。如果在定义域内所有覆盖函数符号均为非负，那么覆盖函数可行，算法终止（第 6 行）。否则调整攻击集合，并重复该过程（第 7-9 行）。如果 E 非空，在 (t_k, t_{k+1}) 内，所有覆盖函数均和 t 轴无交点。第 15 行使用集合 I' 记录在 (t_k, t_{k+1}) 具有非负覆盖函数的目标，并将 I' 作为攻击集合，计算各目标的覆盖函数，直到任何目标在任何时间的覆盖函数值都非负为止。

定理 6 COCO 的结果是最优覆盖函数。

证明 本证明指出，在包含时刻 t_0 的时间段的最后一轮计算中，I 是时刻 t_0 的攻击集合，于是定理6可由引理1得证。令 $\mathbf{c}(t_0) = \langle c_i(t_0) \rangle$ 表示最后一轮计算得到的覆盖集合，令 I 表示相应的目标的集合。在此之前的一轮确保了 $\forall j \in \mathcal{T} \setminus I, c_j(t_0) = 0$。

接下来证明 $v_i(t_0)(1 - c_i(t_0)) > v_j(t_0)(\forall i \in I, \forall j \in \mathcal{T} \setminus I)$。只有当算法第 2 行产生了覆盖集合 $\mathbf{c}'(t_0) = \langle c'_i(t_0), \forall i \in I' \rangle$ with $c'_j(t_0) < 0$ 时，目标 j 才不在集合 I 中。此处 I' 是本轮计算中的假设攻击集合。由于每轮计算均不会扩展上一轮得到的攻击集合，可知 I 是 I' 的子集。第 2 行确保了，

1. $\sum_{i \in I} c_i(t_0) = \sum_{i \in I'} c'_i(t_0) = m$,
2. $v_{i_1}(t_0)(1 - c_{i_1}(t_0)) = v_{i_2}(t_0)(1 - c_{i_2}(t_0)), \forall i_1, i_2 \in I$,
3. $v_{i_1}(t_0)(1 - c'_{i_1}(t_0)) = v_{i_2}(t_0)(1 - c'_{i_2}(t_0)), \forall i_1, i_2 \in I'$.

由于 $\forall i \in I, c_i(t_0) \geq 0$ 以及 $c'_j(t_0) < 0$，有

$$\forall i \in I, c_i(t_0) < c'_i(t_0), \tag{5-7}$$

于是

$$v_i(t_0)(1 - c_i(t_0)) > v_i(t_0)(1 - c'_i(t_0)) = v_j(t_0)(1 - c'_j(t_0)) > v_j(t_0). \tag{5-8}$$

最后一个不等式是由于 $c'_j(t_0) < 0$。 □

5.3.2 纯策略取样

COCO 计算得到了最优覆盖函数有可能是连续变化的函数。然而，在执行单次安保策略是，不可能根据覆盖函数的值对所有 $t \in [0, t_e]$ 时的资源分布情况进行取样。本节研究根据 COCO 的计算结果，进行纯策略取样的过程。给定一个覆盖向量 $\mathbf{c}(t)$，首先将时间段 $[0, t_e]$ 离散成一系列的时间点 $\theta = \langle \theta_k \rangle$，使其满足 $\forall t \in (\theta_k, \theta_{k+1}), \forall i \in \mathcal{T}$，$c_i(t)$ 是

Algorithm 10: COCO(I, b, e)

while *true* **do**
  $c_i(t) \leftarrow 1 - \frac{(n-m)\frac{1}{v_i(t)}}{\sum_{j\in I}\frac{1}{v_j(t)}}, \forall i \in I, t \in [b, e)$;
  $E \leftarrow \{t_k : c_i(t_k) = 0, t_k \in (b, e), \forall i \in I\}$;
  if $E = \emptyset$ **then**
    $I' \leftarrow \{i : i \in I, c_i(t) \geq 0, \forall t \in (b, e)\}$;
    if $I' = I$ **then** break;
    else
      $c_i(t) = 0, \forall i \in I \setminus I', \forall t \in [b, e)$;
      COCO(I', b, e);
  else
    $E \leftarrow E \cup \{t_0 = b, t_{|E|+1} = e\}$;
    for $k \in \{0, \ldots, |E|\}$ **do**
      $I' \leftarrow \emptyset$;
      if $c_i(t) \geq 0, \forall i \in I, \forall t \in (t_k, t_{k+1})$ **then**
        $I' \leftarrow I' \cup \{i\}$;
      else $c_i(t) = 0, \forall t \in [t_k, t_{k+1})$;
      COCO(I', t_k, t_{k+1});
return $\mathbf{c} = \{c_i(t) : i \in I, t \in [b, e)\}$;

单调可导的。由于重要性函数是阶段线性函数，COCO 的第 2 行确保了 $c_i(t)$ 只有有限多个不可导点。于是离散化是可行的。采样过程分为以下四步。

第一步： 采样确定在时刻 θ_k 时的安保资源分布。

第二步： 给定时刻 θ_k 时的安保资源分布，采样确定在时间段 (θ_k, θ_{k+1}) 内所发生的转移的起始目标和终点目标。

第三步： 给定第二步的采样结果，采样确定每次转移发生的时间。

第四步： 在时刻 θ_{k+1} 重复步骤 1 - 3。

接下来详细介绍每一步的执行过程。

第一步： 令 $L = \langle L_i : i \in \mathcal{T}, L_i \in \{0, 1\}, \sum_i L_i = m\rangle$ 表示一种资源分布方式，其中 L_i 用于指示目标 i 是否被保护。令 y_L 表示在时刻 θ_k，资源分布 L 被使用的概率。那么，只需知道 y_L 的值，就可以完成第一步的采样。给定 $\mathbf{c}(\theta_k)$，y_L 可通过与此前工作类似的算法计算得到。

第二步：首先说明，即使覆盖函数是连续变化的函数，其也可以通过仅具有有限多次资源转移的纯策略得以实现。命题11从技术上说明了该问题。令

$$\Lambda = \{u : u \in \mathcal{T}, \frac{dc_u(t)}{dt} < 0, \forall t \in (\theta_k, \theta_{k+1})\}$$

代表一组目标，这部分目标的覆盖函数在时间段 (θ_k, θ_{k+1}) 内单调下降；类似的，令

$$V = \{v : v \in \mathcal{T}, \frac{dc_v(t)}{dt} > 0, \forall v \in (\theta_k, \theta_{k+1})\}$$

代表重要性函数在时间段 (θ_k, θ_{k+1}) 内单调上升的一组目标。于是有

命题 11 对于 $t \in (\theta_k, \theta_{k+1})$，基于 $\mathbf{c}(t)$ 的取样仅需考虑如下纯策略，其中资源仅从目标 $u \in \Lambda$ 被转移至目标 $v \in V$，而且一个资源仅被转移一次。

证明 假设 $\mathbf{x}$ 是对应于覆盖向量 $\mathbf{c}(t)$ 的混合策略，为证明命题11，仅需证明存在一个 $\mathbf{x}$，其支持集仅包含命题11所讨论的纯策略。给定一个时间点 t_0，令 ϕ 表示一个纯策略的集合，其中在时刻 t_0，资源从目标 $v \in V$ 被转移至目标 $i \in \mathcal{T}$。令 ψ 表示一个纯策略的集合，其中在时刻 t_0，资源从目标 $i \in \mathcal{T}$ 被转移至目标 $u \in \Lambda$。接下来证明如果在 $\mathbf{x}$ 中，$\exists S \in \phi$ 或者 $S \in \psi$ 满足 $x_S > 0$，那么可以构造另一个混合策略 $\mathbf{x}'$ 与 $\mathbf{c}(t)$ 对应，满足 $\forall S \in \phi \text{ or } S \in \psi, x'_S = 0$。也就是说，$\mathbf{x}'$ 的支持集仅包含那些资源被从目标 $u \in \Lambda$ 转移至目标 $v \in V$ at t_0 的纯策略。

首先考虑那些纯策略 $S \in \phi$，$x_S > 0$。$\forall S \in \phi$ 有

$$\lim_{t \to t_0^-} q_v(S, t_0) = 1; \lim_{t \to t_0^+} q_v(S, t_0) = 0. \tag{5-9}$$

由于目标 v 的覆盖函数在 (θ_k, θ_{k+1}) 内单调增长，于是有

$$\begin{aligned}
c'_v(t_0) &= \lim_{\Delta t \to 0} \frac{c_v(t_0 + \Delta t) - c_v(t_0 - \Delta t)}{2 \cdot \Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\int_{S \in \mathcal{S}} x_S q_v(S, t_0 + \Delta t) - \int_{S \in \mathcal{S}} x_S q_v(S, t_0 - \Delta t)}{2 \cdot \Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\int_{S \in \mathcal{S}/\phi} x_S f_1 + \int_{S \in \phi} x_S \cdot 0 - \int_{S \in \mathcal{S}/\phi} x_S f_2 - \int_{S \in \phi} x_S \cdot 1}{2 \cdot \Delta t} \\
&\quad (f_1 = q_v(S, t_0 + \Delta t), f_2 = q_v(S, t_0 - \Delta t)) \\
&= \lim_{\Delta t \to 0} \frac{\int_{S \in \mathcal{S}/\phi} x_S f_1 - \int_{S \in \mathcal{S}/\phi} x_S f_2 - \int_{S \in \phi} x_S}{2 \cdot \Delta t} \\
&\geq 0.
\end{aligned} \tag{5-10}$$

于是可得

$$\int_{S \in \mathcal{S}/\phi} x_S q_v(S, t_0 + \Delta t) - \int_{S \in \mathcal{S}/\phi} x_S q_v(S, t_0 - \Delta t) > 0. \tag{5-11}$$

由于 $q_i(S,t)$ 是阶段常数函数，取值仅包含 $\{0,1\}$，可知存在纯策略 S，$x_S > 0$，满足

$$\lim_{t\to t_0^-} q_v(S,t_0) = 0; \lim_{t\to t_0^+} q_v(S,t_0) = 1. \tag{5-12}$$

令 φ 代表所有满足上式的纯策略的集合。在这些策略中，在时刻 t_0，资源被从目标 $j \in \mathcal{T}$ 转移至目标 v。对于其他策略 $S \notin \phi$ 和 $S \notin \varphi$，$q_v(S,t)$ 在时间段 $(t_0-\Delta t, t_0+\Delta t)$ 内是常数。于是有

$$c_v'(t_0) = \lim_{\Delta t\to 0} \frac{\int_{S\in\varphi} x_S - \int_{S\in\phi} x_S}{2\cdot\Delta t} \geq 0. \tag{5-13}$$

这意味着

$$\int_{S\in\varphi} x_S - \int_{S\in\phi} x_S \geq 0. \tag{5-14}$$

由于目标在于构造与 $\mathbf{c}(t)$ 对应的混合策略 $\mathbf{x}'$，以保证在该混合策略中所有 ϕ 或 ψ 内的纯策略都不被使用，于是从构造混合策略 $\mathbf{x}^1$，其中 ϕ 内的纯策略均不被使用开始。首先选择一个集合 $\varphi' \subseteq \varphi$，使其满足

$$\int_{S\in\varphi'} y_S = \int_{S\in\phi} x_S, \tag{5-15}$$

$$0 \leq y_S \leq x_S, \forall S \in \varphi'. \tag{5-16}$$

给定方程5-14，必存在集合 φ' 使得以上问题有解。令 ϕ^i 表示 ϕ 的一个子集，其中资源在时刻 t_0 仅从目标 v 被转移至目标 i。于是有

$$\int_{S\in\phi} x_S = \sum_{i\in\mathcal{T}} \int_{S\in\phi^i} x_S. \tag{5-17}$$

综上，以下为构造 $\mathbf{x}^1$ 的步骤。

1. $\forall S \in \phi$，$x_S > 0$，通过删除在 t_0 时刻从 v 发出的资源转移，将 S 转化为 S'。令 $x^1_{S'} = x_S$。
2. $\forall i \in \mathcal{T}$，选择集合 $\varphi_i \subseteq \varphi'$ 使其满足

$$\int_{S\in\varphi_i} z_S^{\varphi_i} = \int_{S\in\phi^i} x_S, \tag{5-18}$$

$$0 \leq z_S^{\varphi_i} \leq y_S, \forall S \in \varphi_i, \tag{5-19}$$

$$\sum_{i\in\mathcal{T}} z_S^{\varphi_i} = y_S, \forall S \in \varphi'. \tag{5-20}$$

给定方程5-15和5-17，必存在集合 φ_i 使上述问题有解。$\forall S \in \varphi_i$，通过将在 t_0 时刻转移到目标 v 的资源转移到目标 i 去，将 S 转化为 S'。令 $x^1_{S'} = z_S^{\varphi_i}$。（前两步的直观意义在于，如果一个资源在时刻 t_0 以 p_1 的概率被从目标 $v \in V$ 转移至目标 $i \in \mathcal{T}$，又有一个资源在此刻以 p_2 的概率被从目标 $j \in \mathcal{T}$ 转移至目标 $v \in V$，

那么可以直接以 $p1$ 的概率将一个资源从目标 j 到目标 i，以避免将资源从 v 转出。方程5-14指出 $p_2 \geq p_1$，使得这一步是可行的。）

3. 对于其他纯策略 $S \in \mathcal{S} \setminus (\phi \cup \varphi')$，令 $x_S^1 = x_S$。

4. 重复步骤 1-3，直到在时刻 t_0 没有资源被从 $v \in V$ 转出。

经过以上步骤后，在 t_0 时刻，一个资源在给定 $\mathbf{x}^1$ 和 $\mathbf{x}$ 时，被保护的概率是相同的。基于 $\mathbf{x}^1$，可构建另一个混合策略 $\mathbf{x}'$，使其与 $\mathbf{c}(t)$ 对应，且不使用任何 ϕ 或 ψ 中的纯策略。$\mathbf{x}^1$ 已保证了没有任何 ϕ 中的策略被使用，因此只需考虑移除那些 $S \in \psi$，$x_S^1 > 0$。由于 $\forall u \in \Lambda$，在 (θ_k, θ_{k+1})，$c_u(t)$ 单调递减，于是必然存在 S，$x_S^1 > 0$，其中资源被从目标 u 转移至其他目标。于是 $\mathbf{x}'$ 可通过与以上步骤类似的步骤构建。于是 $\mathbf{x}'$ 与 $\mathbf{c}(t)$ 对应，且没有 ϕ 或 ψ 内的策略被使用。

基于以上证明，在时间段 (θ_k, θ_{k+1}) 内，资源不会被从该目标 $v \in V$ 转移至其他目标，因此一个资源只能被转移一次。 □

接下来考虑第二步的执行过程。令

$$\Gamma_L = \{(u,v) : u \in \Lambda, v \in V, L_u = 1, L_v = 0\}$$

代表一个“转移集合”，即，$\forall (u,v) \in \Gamma_L$，在时间段 (θ_k, θ_{k+1}) 内，一个资源被从目标 u 转移至了目标 v。因此，只需知道每个 Γ_L 被采用的概率，定义为 $p(\Gamma_L)$，第二步的采样就可以完成。为计算 $p(\Gamma_L)$，首先定义 $p((u,v)|L)$，令其表示在 L 被选定为时刻 θ_k 的资源分布之后，在时间段 (θ_k, θ_{k+1}) 内，一个资源被从目标 u 转移到目标 v 的概率。一旦获取 $p((u,v)|L)$，$p(\Gamma_L)$ 即可通过以下的方程组计算得出。

$$\sum_{\Gamma_L \ni (u,v)} p(\Gamma_L) = p((u,v)|L), \forall L \tag{5-21}$$

为计算 $p((u,v)|L)$，令 $Z((u,v),t)$ 表示在时刻 t 之前，资源被从目标 u 转移到目标 v 的概率。对于 $t \in [\theta_k, \theta_{k+1}]$，$Z((u,v),t)$ 是关于 t 的函数。一旦得到 $Z((u,v),t)$ 的表达式，那么 $p((u,v)|L)$ 即可通过如下方程计算。

$$\sum_L y_L \cdot L_u \cdot (1 - L_v) \cdot p((u,v)|L) = Z((u,v), \theta_{k+1}), \forall u, v \tag{5-22}$$

$$p((u,v)|L) = 0, \forall u \in \Lambda \text{with } L_u = 0, \forall v \in V \text{with } L_v = 1 \tag{5-23}$$

方程5-22是基于全概率公式。方程5-23表明资源仅能够被从 θ_k 被保护的目标转移至 θ_k 时刻未被保护的目标。于是，问题仅在于计算 $Z((u,v),t)$。

计算 $Z((u,v),t)$：首先，一个可行的 $Z((u,v),t)$ 表达式，应满足在 $\forall t \in [\theta_k, \theta_{k+1}]$ 时间内有

$$\sum_{v \in V} Z((u,v),t) = c_u(\theta_k) - c_u(t), \forall u \in \Lambda \tag{5-24}$$

$$\sum_{u \in \Lambda} Z((u,v),t) = c_v(t) - c_v(\theta_k), \forall v \in V \tag{5-25}$$

$$Z((u,v),t) \geq 0, \forall u \in \Lambda, \forall v \in V \tag{5-26}$$

$$\lim_{\Delta t \to 0} \frac{Z((u,v),t+\Delta t) - Z((u,v),t)}{\Delta t} \geq 0, \forall u \in \Lambda, v \in V \tag{5-27}$$

方程5-24 - 5-25基于命题11。方程5-26 - 5-27基于 $Z((u,v),t)$ 的定义。给定一个时间点 $t \in [\theta_k, \theta_{k+1}]$，$Z((u,v),t)$ 的值可通过方程5-24 - 5-27计算得出。但由于时间的连续性，不可能通过求解每个点的值得方式求得 $Z((u,v),t)$ 的表达式。但是，给定方程5-24和5-25，$Z((u,v),t)$ 可表示为覆盖函数的线性组合 ($\forall t \in [\theta_k, \theta_{k+1}]$)，记作，

$$Z((u,v),t) = \sum_{i \in \Lambda} z^i_{uv}(t)(c_u(\theta_k) - c_u(t)) + \sum_{j \in V} z^j_{uv}(t)(c_v(t) - c_v(\theta_k)). \tag{5-28}$$

此处 $z^i_{uv}(t)$ 是与 t 相关侧参数，称作参数方程。通过计算这些参数方程，即可得到 $Z((u,v),t)$ 的表达式。在某一个时刻 $t \in [\theta_k, \theta_{k+1}]$，参数方程的值可通过方程5-24 - 5-27计算。下述命题表述了参数方程的性质。

命题 12 *存在一种对应于 $Z((u,v),t)$ 的参数方程，满足任一 $z^i_{uv}(t)$ 在时间段 $[\theta_k, \theta_{k+1}]$ 内都是阶段常数的。*

命题12的证明基于引理2。

引理 2 *如果 $|\Lambda| > 2$，$|V| > 2$，即存在可行的 $Z((u,v),t)$ 满足 $\lim_{\Delta t \to 0} \frac{Z((u,v),t+\Delta t) - Z((u,v),t)}{\Delta t} > 0, \forall u, \forall v, \forall t \in [\theta_k, \theta_{k+1}]$。*

证明 *为证明引理2，只需证明如果将方程5-27中的 $\geq$ 号换成 $>$ 号，由方程5-24 - 5-27组成的线性规划问题依然有可行解。为证明该问题，构建一个与之等价的线性规划，并证明该线性规划有可行解。给定方程5-28，对任一 $t \in [\theta_k, \theta_{k+1}]$，可将方程5-24 - 5-27中的变量由 $Z((u,v),t)$ 改变为 $z^i_{uv}(t)$，并得到*

$$LP': \sum_{v \in V} z^u_{uv}(t) = 1, \forall u \in \Lambda \tag{5-29}$$

$$\sum_{u \in \Lambda} z^v_{uv}(t) = 1, \forall v \in V \tag{5-30}$$

$$\sum_{v \in V} z^i_{uv}(t) = 0, \forall i \in \mathcal{T} \setminus u, \forall u \in \Lambda \tag{5-31}$$

$$\sum_{u \in \Lambda} z^i_{uv}(t) = 0, \forall i \in \mathcal{T} \setminus v, \forall v \in V \tag{5-32}$$

方程5-26和5-27,将方程5-27中 '≥' 替换为 '>'. (5-33)

基于方程5-28，方程5-29 - 5-32使得 $Z((u,v),t)$ 满足方程5-24 - 5-25。当把 $Z((u,v),t)$ 与方程5-28右侧交换后，方程5-26和5-27即包含变量 $z^i_{uv}(t)$，于是 LP' 与方程5-24 - 5-27等价。LP' 中的变量数量为 $nv = |\Lambda| \cdot |V| \cdot |\Lambda + V|$，方程数量为 $nf = 2|\Lambda| + 2|V| + 2|\Lambda| * |V|$，如果 $|\Lambda| > 2$ 且 $|V| > 2$，那么 $nv > nf$，LP' 有可行解。 □

接下来证明命题12。

证明 如果 $|\Lambda| \leq 2$ or $|V| \leq 2$，基于方程5-24 and 5-25，$Z((u,v),t)$ 在任一 $t \in [\theta_k, \theta_{k+1}]$ 有唯一解，任一 $z^i_{uv}(t)$ 在 $[\theta_k, \theta_{k+1}]$ 内是常数。如果 $|\Lambda| > 2$ 并且 $|V| > 2$，令 $Z((u,v),t)$ 表示基于参数方程 $z^i_{uv}(t_0)$ 的表达式，即 LP' 在 $t = t_0$ 时的解。接下来证明这类 $Z((u,v),t)$ 不仅在时刻 t_0 满足方程5-24 - 5-27，同样在 t_0 前后的一段时间内满足。方程5-29 - 5-32使得 $Z((u,v),t)$ 在所有 $t \in [\theta_k, \theta_{k+1}]$ 满足方程5-24 - 5-25。另外，给定覆盖函数的形式，$\frac{\mathrm{d}Z((u,v),t)}{\mathrm{d}t}$ 的符号不会持续变化。给定在 t_0 时刻 $\frac{\mathrm{d}Z((u,v),t)}{\mathrm{d}t} > 0$，$Z((u,v),t_0) \geq 0$，可知在 t_0 前后的一段时间内有 $Z((u,v),t) > 0$。于是，在 t_0 附近的一段时间内，基于参数方程 $z^i_{uv}(t_0)$ 的 $Z((u,v),t)$ 满足方程5-24 - 5-27，也就意味着阶段常数的参数方程可以得到 $t \in [\theta_k, \theta_{k+1}]$ 中的 $Z((u,v),t)$ 的表达式。 □

给定命题12和引理2，算法11描述了计算参数方程从而计算 $Z((u,v),t)$ 的过程。首先计算时刻 t_0 参数方程的值，随后得到在时间段 $[t_0, \theta_{k+1}]$ 内参数方程为常数的 $Z((u,v),t)$ 的表达式。令 R 表示方程 $\frac{\mathrm{d}Z((u,v),t)}{\mathrm{d}t}$（第 4 行）的根的集合。$Z((u,v),t)$ 的表达式从 t_0 到最小的根的范围内是可行的，由于 $t > t_m$，方程5-27会被违反。在 t_m 时刻，该过程被重复进行，直到 $Z((u,v),t)$ 对任一 $t \in [\theta_k, \theta_{k+1}]$ 都是可行的。

定理 7 算法 2 在有限次循环后计算出 $Z((u,v),t), \forall t \in [\theta_k, \theta_{k+1}]$ 的可行表达式。

证明 首先，基于算法 2 的第 3 行和第 9 行，可知计算出的 $Z((u,v),t)$ 是可行的，也即其满足方程5-24 - 5-26，$\forall t \in [\theta_k, \theta_{k+1}]$。给定方程5-27的约束被设定为 $>$，即有每一轮中均有 $t_m > t_0$。因此循环会在有限次后停止。 □

第三步： 令 $Z(t|L,(u,v))$ 代表时间 t 的累积分布函数，代表在 L 被设定为时刻 θ_k 的资源分布之后，在转移 $\Gamma_L \ni (u,v)$ 被第二步的取样选定后，一个资源是时刻 t 之前 $(t \in [\theta_k, \theta_{k+1}])$ 被从目标 u 转移至目标 v 的概率。只需得到 $Z(t|L,(u,v))$ 的值即可进行第三步的采样。由于从目标 u 到目标 v 的资源转移发生时间与时刻 θ_k 的资源分布情况是独立的，于是有

$$Z(t|L,(u,v)) = Z(t|(u,v)) = \frac{Z((u,v),t)}{Z((u,v),\theta_{k+1})}, \forall t \in [\theta_k, \theta_{k+1}]. \tag{5-34}$$

于是 $Z(t|L,(u,v))$ 可由 $Z((u,v),t)$ 的表达式得到。

Algorithm 11: 计算 $Z((u,v),t)$

$t_0 = \theta_k, k_{uv} = 0$;
while $t_m < \theta_{k+1}$ **do**
  根据方程5-24 - 5-27求解 $z^i_{uv}(t_0)$（将方程5-26中的 0 替换为 k_{uv}，将方程5-27中的 ≥ 替换为 >;
  $R = \langle t : \frac{\mathrm{d}Z((u,v),t)}{\mathrm{d}t} = 0 \text{ and } t \in (t_0, \theta_{k+1})\rangle$;
  if $R = \emptyset$ **then**
    break;
  else
    $t_m = \min_t R; k_{uv} = Z((u,v),t_0)$;
    $z^i_{uv}(t) = z^i_{uv}(t_0), \forall t \in [t_0, t_m); t_0 = t_m$;

5.4 转移时间不可忽略的情形

本节考虑更具一般性的情况，即资源的转移时间不可忽略，$d_{ij} \geq 0$。本节提出一种高效的近似算法来求解最优的安保策略。首先提出一种离散化方法，合理地离散化时间轴，并证明离散之后的防守方收益最多比原连续问题中的最优收益减少 ϵ，然后提出使用网络流的方法求解离散问题的算法。为构建离散化的问题，首先定义 (ϵ,δ)-Mesh 如下。

定义 4 ((ϵ,δ)-Mesh) 一个 (ϵ,δ)-mesh 是一个时间空间 $[0,t_e]$ 内的均匀分布，定义为 $\zeta = \langle t_0, t_1, ..., t_{|\zeta|}\rangle$，其中 $t_0 = 0, t_{|\zeta|} = t_e$，其对于任一 $k = 1, ..., |\zeta|$ 满足 $\delta = t_k - t_{k-1}$ 以及 $var_i(t_{k-1}, t_k) \leq \epsilon$。其中

$$var_i(t_{k-1}, t_k) = \max_{t\in[t_{k-1},t_k]} v_i(t) - \min_{t\in[t_{k-1},t_k]} v_i(t)$$

是 $v_i(t)$ 在时间段 $[t_{k-1}, t_k]$ 内最大值和最小值的差值。

重要性函数的连续性保证了这样的 (ϵ,δ)-Mesh 的存在性，并能够被快速计算出。（对于满足 $|v_i(t) - v_i(t')| \leq K|t - t'|$ 的 Lipschitz 连续的函数，可直接令 $\epsilon = K\delta$。）现在考虑将一个常规的博弈 G 转化为一个离散化的博弈 G_d。首先，假设安保部门只能在固定的时间点转移资源，即 $t_k \in \zeta$，而攻击方也只能在这些固定的时间点发动攻击。于是博弈 G_d 的参与者的策略空间就是博弈 G 的策略空间的一个子集。令 $v_i(t)$ 代表 G 中的重要性函数，利用 (ϵ,δ)-Mesh 来构造 G_d 的重要性函数：$\forall i \in \mathcal{T}$，令

$$v'_i(t) = \max_{t'\in[t_{k-1},t_k]} v_i(t'), \forall t \in [t_{k-1}, t_k) \tag{5-35}$$

代表 G_d 中的重要性函数。值得注意的是，$v_i'(t)$ 在时间段 $[t_{k-1}, t_k]$ 内是常数，并满足 $|v_i'(t) - v_i(t)| \leq \epsilon, \forall t$。又假设 d_{ij} 是 δ 的整数倍，即，$d_{ij} = a_{ij}\delta$，其中 a_{ij} 是一个整数。当 d_{ij} 是一个实数且 δ 值足够小的时候该假设即可成立。为证明 G_d 所对应的攻击方最优收益与原博弈 G 中的攻击方的最优收益的差值，首先引入一个中间博弈 G_b，该博弈中的重要性函数同 G_d 相同，均为 $v_i'(t)$，但是安保方和攻击方可以在任何时刻发起形同，即，博弈参与双方的策略空间均与博弈 G 相同。下面的引理指出了 G_d 和 G_b 中安保方最优收益的关系。

引理 3 令 $\mathbf{x}$ 代表攻击方在博弈 G_d 中一个混合策略，那么 $U_d^{G_d}(\mathbf{x}) = U_d^{G_b}(\mathbf{x})$。其中 $U_d^{G_d}(\mathbf{x})$ 和 $U_d^{G_b}(\mathbf{x})$ 分别为安保方使用策略 $\mathbf{x}$ 而攻击方做出最优回应是安保方在两种博弈中的收益。

这是由于 G_d 和 G_b 博弈中的重要性函数是相同的。但根据引理3，可得到一个更深层的结论。

引理 4 假设资源在任意两个资源中转移所需的时间是 δ 的倍数。如果 $\mathbf{x}$ 是博弈 G_d 中的均衡策略，那么它也是博弈 G_b 中的均衡策略。

证明 首先，该结论并不能直接从3得来，这是由于博弈 G_d 的策略空间仅为博弈 G_b 的策略空间的子集。为证明引理4，此处证明必存在一个博弈 G_b 的均衡策略可被映射到博弈 G_d 均均衡策略。

首先定义映射 Φ，其将纯策略 $S \in G_b$ 映射到另一个纯策略 S'：对于 S 中包含的每一次资源转移，将其开始时间从 t 改变至 $r\delta$，其中 $t \in (r\delta, (r+1)\delta)$，$r \in \mathbb{N}$。此外，将一个混合策略 $\mathbf{x}$ 映射为另一个混合策略 $\mathbf{x}'$，使得如果 $\mathbf{x}$ 以概率 x_S 使用纯策略 S，那么 $\mathbf{x}'$ 以概率

$$x'_{S'} = \int_S x_S I(\Phi(S) = S')$$

使用纯策略 S'，其中 $I(\Phi(S) = S')$ 是一个二值指示函数，用于指示 S 是否可以经由映射 Φ 而被转化为 S'。于是，在 $\mathbf{x}'$ 中，所有资源开始被转移的时间均为 $r\delta, r \in \mathbb{N}$。同时，$\mathbf{x}'$ 也是在博弈 G_d 中的一个可行的混合策略。

接下来说明，如果 $\mathbf{x}$ 是博弈 G_b 的强斯塔克伯格均衡策略（SSE），那么 $\mathbf{x}'$ 就是博弈 G_b 和 G_d 的强斯塔克伯格均衡策略。值得注意的是，安保方在 G_d 中的最优收益不会高于其在 G_b 中的最高收益，这是由于他在 G_d 中的策略空间为 G_b 中的策略空间的子集。于是，只需证明 $\mathbf{x}'$ 是博弈 G_b 中的强斯塔克伯格均衡策略，即可得到 $\mathbf{x}'$ 同样是博弈 G_d 的强斯塔克伯格均衡策略。

由于 $\mathbf{x}$ 是博弈 G_b 中的强斯塔克伯格均衡策略，有 $U_d^{G_b}(\mathbf{x}) \geq U_d^{G_b}(\mathbf{x}')$。接下来说明 $U_d^{G_b}(\mathbf{x}) \leq U_d^{G_b}(\mathbf{x}')$，于是 $U_d^{G_b}(\mathbf{x}) = U_d^{G_b}(\mathbf{x}')$，那么 $\mathbf{x}'$ 就是博弈 G_b 的强斯塔克伯格均衡策

略。令 $\mathbf{c}=\langle c_i(t)\rangle$，$\mathbf{c}'=\langle c'_i(t)\rangle$ 分别表示与 $\mathbf{x}$ 和 $\mathbf{x}'$ 对应的覆盖函数，接下来说明，在时间段 $[r\delta,(r+1)\delta), r\in\mathbb{N}$ 内，在 $\mathbf{c}$ 下一个目标获得的最小保护（被保护的概率最小）不会高于在 $\mathbf{c}'$ 在该目标获得的最小保护。由于博弈参与双方的收益均取决于一个时间段内目标获得的最小保护（因为该时间段内重要性函数已被设定为常数），于是有 $U_d^{G_b}(\mathbf{x})\le U_d^{G_b}(\mathbf{x}')$。给定 $\mathbf{c}$ 和 $\mathbf{c}'$，即有

$$
\begin{aligned}
&\min_{t\in[r\delta,(r+1)\delta)} c_i(t)\\
=&\min_{t\in[r\delta,(r+1)\delta)} \int_S x_S\cdot q_i(S,t)\\
=&\min_{t\in[r\delta,(r+1)\delta)} \Big(c_i(r\delta)+\int_S x_S\cdot\big(q_i(S,t)-q_i(S,r\delta)\big)\Big)\\
=&\min_{t\in[r\delta,(r+1)\delta)} \Big(c_i(r\delta)+\int_{S:\text{ a transfer to } i \text{ ends in } (r\delta,t] \text{ by } S} x_S\ -\\
&\qquad\int_{S:\text{ a transfer away from } i \text{ starts in } (r\delta,t] \text{ by } S} x_S\Big)\\
\le&\min_{t\in[r\delta,(r+1)\delta)} \Big(c_i(r\delta)+\int_{S:\text{ a transfer to } i \text{ ends in } (r\delta,(r+1)\delta) \text{ by } S} x_S\\
&\qquad-\int_{S:\text{ a transfer away from } i \text{ starts in } (r\delta,t] \text{ by } S} x_S\Big)\\
=&\min_{t\in[r\delta,(r+1)\delta)} \Big(c'_i(r\delta)-\int_{S:\text{ a transfer away from } i \text{ starts in } (r\delta,t] \text{ by } S} x_S\Big)\\
=&\min_{t\in[r\delta,(r+1)\delta)} c'_i(t)
\end{aligned}
$$

后两个等式是由于在策略 $\mathbf{x}$ 下所有起始时间或终止时间在 $(r\delta,(r+1)\delta)$ 内的转移，其起始时间或终止时间在 $\mathbf{x}'$ 下都被转换成了 $\mathbf{x}'$ 中的 $r\delta$。 □

现在来考虑 G_d 和 G 中防守方收益的差距。

定理 8 令 $\mathbf{x}'$ 表示 G_d 中的均衡策略，令 $\mathbf{x}$ 表示 G 中的均衡策略，可得

$$U_d^G(\mathbf{x}')-U_d^G(\mathbf{x})\ge-\epsilon,$$

其中 $U_d^G(\mathbf{x}')$ 表示在博弈 G 中，安保方使用 $\mathbf{x}'$，攻击方做出最优回应时安保方的收益，$U_d^G(\mathbf{x})$ 则为博弈 G 中安保方的均衡收益。

证明 这是由于

$$
\begin{aligned}
&U_d^G(\mathbf{x}')-U_d^G(\mathbf{x})\\
=&[U_d^G(\mathbf{x}')-U_d^{G_b}(\mathbf{x}')]+[U_d^{G_b}(\mathbf{x}')-U_d^{G_b}(\mathbf{x})]+[U_d^{G_b}(\mathbf{x})-U_d^G(\mathbf{x})]\\
\ge&[U_d^G(\mathbf{x}')-U_d^{G_b}(\mathbf{x}')]+[U_d^{G_b}(\mathbf{x})-U_d^G(\mathbf{x})]\\
\ge&-|U_d^G(\mathbf{x}')-U_d^{G_b}(\mathbf{x}')|-|U_d^G(\mathbf{x})-U_d^{G_b}(\mathbf{x})|
\end{aligned}
$$

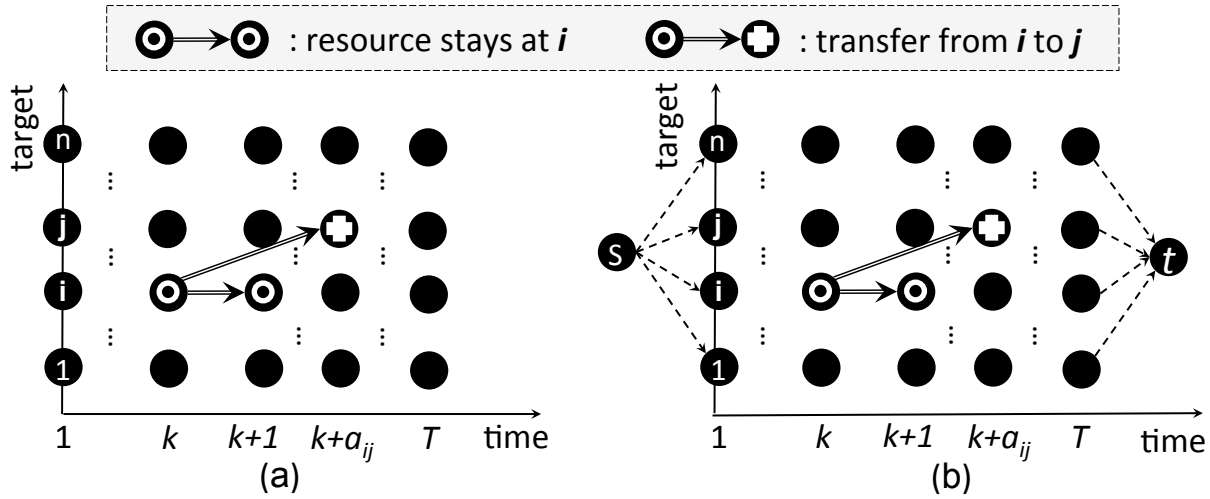


图 5.1 离散博弈的构建

第一个不等式是由于 $U_d^{G_b}(\mathbf{x}') - U_d^{G_b}(\mathbf{x}) \geq 0$，这是因为在 G_d 中，$\mathbf{x}'$ 是比 $\mathbf{x}$ 更好的策略，因此，在 G_b 中，$\mathbf{x}'$ 也是比 $\mathbf{x}$ 更好的策略（引理3）。给定 $\mathbf{x}'$ 时 G_d 的强斯塔克伯格均衡策略，其对于的覆盖函数在任何时间区间 $[t_{k-1}, t_k], \forall k = 1, \ldots, |\zeta|$ 都是阶段常数的。于是防守方的收益取决于这段时间内目标重要性函数的最大值。给定 G_b 中这个时间段内的重要性函数的值等同于 G 中目标的重要性函数在这段时间内的最大值（方程5-35），有 $U_d^G(\mathbf{x}') = U_d^{G_b}(\mathbf{x}')$。由于 $v_i(t)$ 与 $v_i'(t)$ 在给定任意 i, t 时差值不超过 ϵ，有

$$U_d^G(\mathbf{x}) - U_d^{G_b}(\mathbf{x}) \leq \epsilon, \tag{5-36}$$

于是

$$U_d^G(\mathbf{x}') - U_d^G(\mathbf{x}) \geq 0 - \epsilon = -\epsilon. \tag{5-37}$$

□

另外，基于方程5-35，有 $U_d^G(\mathbf{x}') = U_d^{G_d}(\mathbf{x}')$。于是问题就在于求解博弈 G_d 的均衡解。如图5.1所示，G_d 是一个离散化的博弈，其中 x 轴表示 (ϵ, σ)-Mesh，y 轴表示各个目标，于是点 (k, i) 就表示在时刻 t_k 的目标 i。攻击方的一个纯策略即选择图中的一个点。如果在图中增加一个虚拟的起始点 s 和一个虚拟的终止点 t，并使图中每条边的负载都为 1，那么安保方的一个纯策略就是一个图中的 m-单位 $0-1$ 整数 $s-t$ 流，也就是说，如果安保方在时刻 t_k 将一个资源从目标 j 转移至目标 j，那么就在点 (k, j) 和点 $(k + a_{ij}, j)$ 之间构造一条边；如果在时刻 t_k，目标 i 被保护，并且该安保资源随后没有被转走，那么在点 (k, i) 和点 $(k + 1, i)$ 之间构造一条边。以下定理是关于博弈 G_d 中安保方的混合策略的。

定理 9 博弈 G_d 中的一个安保方的混合策略是一个 m-单位部分流，反之亦然。

证明 $\Rightarrow$ 方向显明。$\Leftarrow$ 方向是由于图论中的以下结论：在边的负载为整数的有向图中，m-单位部分流组成的多面体的定点为所有的 m-单位整数流。于是，一个 m-单位部分流可以被分解为 0-1 整数流的凸组合，即纯策略的组合。 □

令 $f(i,j,t_k)$ 代表在时刻 t_k，有资源被从目标 i 转移到目标 j 的概率。最优的安保策略可通过以下被叫做 ADEN (computing Approximately optimal DEfender strategies in games with Nonzero transfer time) 的线性规划算法计算出。

$$\textbf{ADEN}: \quad \min\ U \tag{5-38}$$

$$\text{s.t. } U \geq v_i'(t_k)(1 - f(i,i,t_k)), \forall i \in \mathcal{T}, t_k \in \zeta \tag{5-39}$$

$$\sum_{i \in \mathcal{T}} f(s,i,t_0) = m, \tag{5-40}$$

$$f(s,i,t_0) - \sum_{j \in \mathcal{T}} f(i,j,t_0) = 0, \forall i \in \mathcal{T} \tag{5-41}$$

$$\sum_{j \in \mathcal{T}} f(j,i,t_{k-a_{ji}}) - \sum_{j \in \mathcal{T}} f(i,j,t_k) = 0, \forall i \in \mathcal{T}, t_{k-a_{ji}}, t_k \in \zeta \tag{5-42}$$

值得注意的是，在 t_k 时刻，目标 i 被保护的概率与在 t_k 时刻之后的一个时段留在目标 i 的资源的概率是一致的，也即，$c_i(t_k) = f(i,i,t_k)$。于是目标函数和方程5-39共同保证了该线性规划求出最优解。方程5-40 - 5-42是为了确保 m-单位 $s-t$ 流符合要求。ADEN 求出的覆盖向量 $\mathbf{c} = \langle f(i,i,t_k) : i \in \mathcal{T}, t_k \in \zeta \rangle$ 可被直接映射至一个有有限支持集的混合向量，也就是资源转移只发生在时刻 $t_k (\forall t_k \in \zeta)$ 的纯策略。技术上讲，对于用于在时刻 t_k 保护目标 i 的资源，其保护的下一个目标 j 可根据 $f(i,j,t_k)$ 的值来取样，如果 $j = i$ 就意味着该资源仍然停留在 i 不被转移。如果资源被转移，那么从时刻 t_k 到 $t_{k+a_{ij}}$，该资源就正在转移途中，不被用于保护任何目标。最初的资源分布情况可根据 $f(s,j,t_0)$ 的值来取样实现。该线性规划的规模为 $poly(\frac{t_e}{\delta}n^2)$。当重要性函数为满足 $|v_i(t) - v_i(t')| \leq K|t-t'|, \forall i$ 的 Lipschitz 连续函数时，可令 $\epsilon = K\delta$，于是线性规划的规模为 $poly(\frac{Kt_e}{\epsilon}n^2)$，并使得其结果是原问题最优解的 ϵ 近似。

5.5 实验与分析

本部分首先从攻击方收益的角度评测算法性能。由于博弈的零和性，更低的攻击方收益意味着更高的安保方收益。假设 $t_e = 100$。考虑两种基线算法。第一种是在静态博弈假设下的 ORIGAMI 算法。第二种，被称作 PDT，假设攻击方和防守方只能在预先确定的离散时间点 $\Phi = \{\frac{n \cdot t_e}{7} : n = 0, 1, \cdots, 7\}$ 内进行资源转移（此假设与 [121] 类似。）本部分还测试了算法的可扩展性。

在每个博弈中，重要性函数被认为是至多有两段的阶段线性函数，也就是说，重要性函数的段数在 $\{1,2\}$ 中随机产生。如果一个重要性函数被确定有两段，那么其折点在 $(0,t_e)$ 内随机生成。函数在 0 处和 t_e 处的函数值在 $(0,100]$ 内随机产生。本节通过多组实验检测算法性能。在前两组试验中，转移时间被设定为 0，测定算法在目标个数与资源个数发生变化的情况下的性能。随后检测了 ϵ 值的变化和转移时间的变化对算法 ADEN 性能的影响。最后检测了取样过程的运行时间。

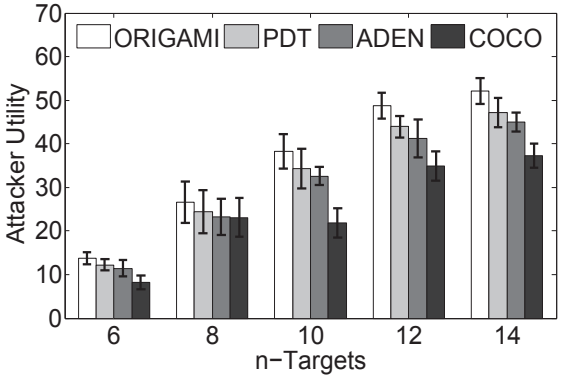


图 5.2 改变目标数量

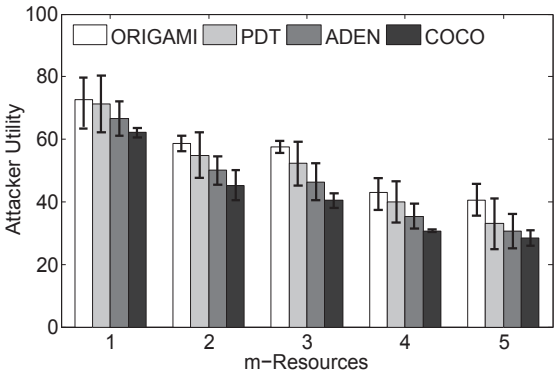


图 5.3 改变资源数量

首先检测算法性能。在图5.2到图5.5中，y 轴表示攻击方的最优收益。图5.2和5.3考虑转移时间可忽略的情况。在图5.2中，x 轴表示目标的数量。在图5.3中，x 轴表示资源的数量。这两幅图的结果显示，不管目标和资源的数量如何变动，ADEN 和 COCO 的结果都优于基线算法。COCO 对应的攻击方收益比 ADEN 更低，由于其计算出的是最优的安保策略。图5.4和5.5进一步检测了 ADEN 的效果。图5.4检测增加 ϵ 的值对 ADEN 的影响。随着 ϵ 值得增加，ADEN 计算出的攻击方最优收益缓慢增加。由于 ϵ 指示的是安保方损失的上界，事实上安保方的收益有可能显著低于 ϵ。ϵ 的值不会影响 PDT 的效用，但是即使 $\epsilon = 30$，ADEN 的效用依然优于 PDT。图5.5检测了增加转移时间对 ADEN 和 PDT 的影响（ϵ 被固定为 10）。x 轴指示转移时间的级别。例如，对于 $x = 10$，转移时间在 $[\frac{10}{2}, 10]$ 内随机选出。图5.5显示随着转移时间增加，ADEN 和 PDT 的效用都变差。转移时间增加对 ADEN 的影响强于其对 PDT 的影响。即使转移时间增加到 30，

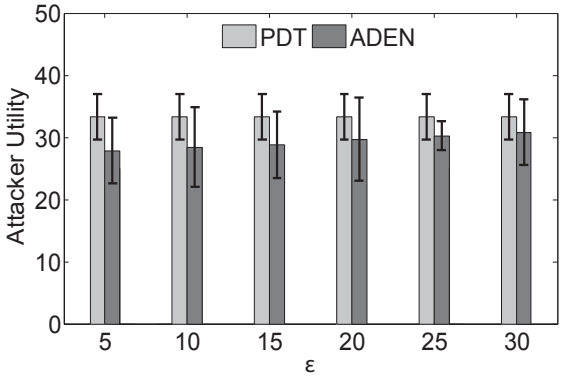


图 5.4 增加 ϵ

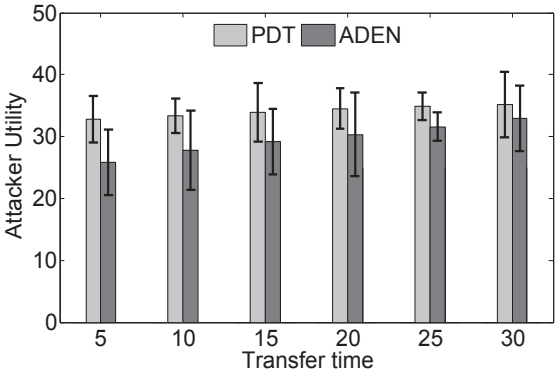


图 5.5 增加 d_{ij}

ADEN 的效用依然优于 PDT。如果转移时间非常长，甚至超过了安保策略的执行时间，那么最优策略即为在静态收益假设下的策略，ADEN 和 PDT 都会获得和 ORIGAMI 一样的效果。

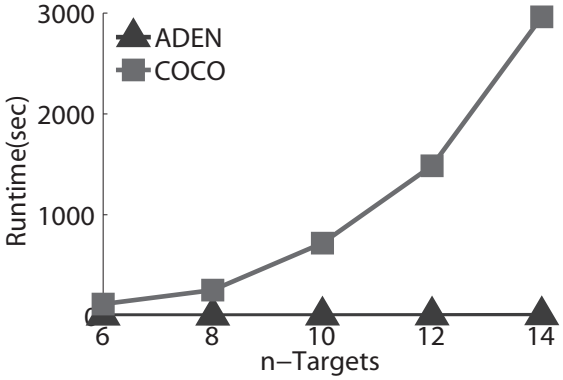


图 5.6　增加目标数量

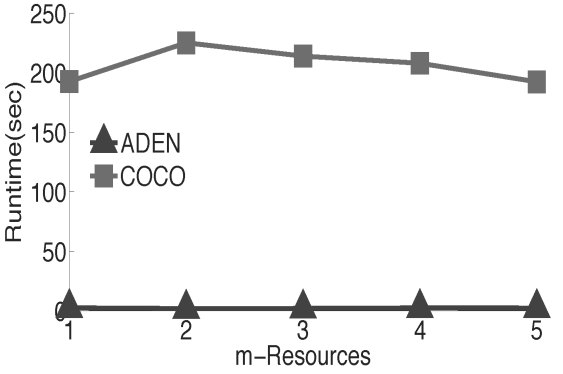


图 5.7　增加资源数量

接下来检测算法的运行时间。在图5.6到图5.9中，y 轴表示算法的运行时间。在图5.6中，x 轴表示目标数量。COCO 的运行时间随着目标数量的增加而明显增长，但 ADEN 的运行时间在目标数量从 6 增长到 14 的过程中未发生明显变化。在图5.7中，x 轴表示资源数量。图中曲线显示，资源数量并不会显著影响 COCO 和 ADEN 的运行时间。图5.8将目标的数量扩展至 100，并展示了 ADEN 在不同的 ϵ. 值下的运行时间。越小的 ϵ 值对应的运行时间越长，运行时间的增长趋势也最明显。当 $\epsilon = 30$ 时，ADEN 可以在 20 分钟内求解有 100 个目标的博弈。图5.8中没有展示 COCO 的运行时间，这是由于其在 $n > 15$ 时即因内存限制而无法运行。图5.9展示了取样过程的运行时间。

5.6　本章小结

本章研究求解动态收益安全博弈中的混合策略均衡，以应对城市安保等问题。本章提出了一个斯塔克伯格博弈模型来对该问题建模，该模型中博弈的参与方均具有连续的策略空间，目标的重要程度是随着时间而变化的，并使用强斯塔克伯格均衡作为解空间。本章提出了算法 COCO 用于求解当转移时间可忽略时安保方的最优策略，并

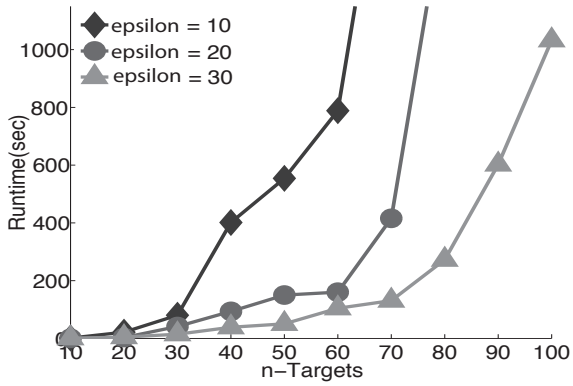


图 5.8　ADEN

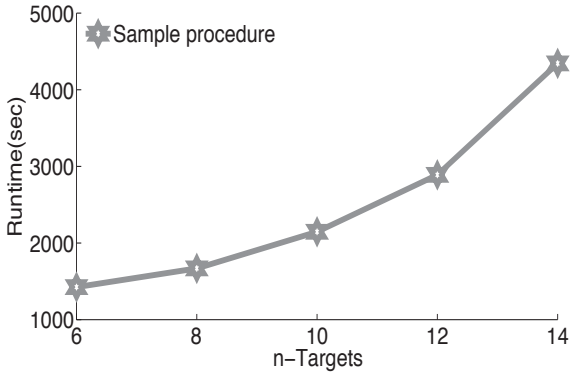


图 5.9　取样过程

基于 COCO 所计算出的紧凑的混合策略表示法，提出了一种方法用于每次执行策略时进行纯策略取样。本章也考虑了更具一般性的转移时间不可忽略的情形，并提出算法 ADEN，基于一种 $(\epsilon - \delta)$-Mesh 的离散方法对博弈进行离散化，求解该情形下的一个 ϵ 近似解。

第六章　图安全博弈

6.1　引言

珊瑚礁是珍贵的自然资源。它们不仅有着极高的美学价值，还支撑着世界上最具活力的生态系统，为众多的海洋生物提供了赖以生存的栖息地（图 6.1）。然而，这些价值也吸引着不法分子进行诸如珊瑚礁挖掘、炸鱼等破坏性活动，这些活动会对珊瑚礁以及他所支持的生态系统产生恶劣的影响。更重要的是，一旦珊瑚礁被破坏，就需要数十年的时间才能逐渐恢复。所以，许多国家建立了海洋生态保护区（marine protected areas，MPAs），通过巡逻队方式来阻止这些破坏性活动 [10]（图6.2，6.3）。

然而，对海洋生态保护区进行高效的巡航一直是一个充满挑战的问题。首先，环保机构通常仅有有限的巡航资源，但却需要保护面积广阔的保护区。例如，中国海南三亚亚龙湾自然保护区的环保部门需要用三艘巡航艇保护超过 85 平方公里的水域。第二，潜在的不法分子可能在一天内的任何时间，选取任一条路径进入自然保护区，并选取一段时长用于进行破坏活动。另外，由于巡航是每天都会进行的，不法分子可能观测到环保部门的巡航策略，并据此选择最隐秘的时间和路径以进行破坏性活动。本章着眼

图 6.1　珊瑚礁生态多样性

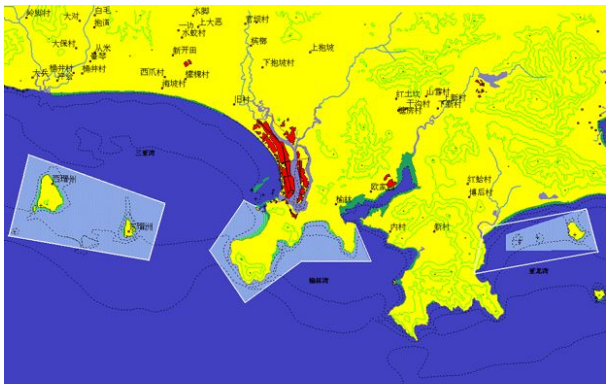


图 6.2　亚龙湾 MPA

图 6.3　亚龙湾 MPA 巡逻船

于为环保部门设计高效的巡航策略以用于海洋生态保护区的巡逻保护。

虽然珊瑚礁及海洋生态系统面临的威胁已受到很多科学家的注意 [10,86]，但此前的工作大多着眼于自然保护区的规划与建设，而不是对于破坏性活动的检测与惩治 [16]。同时，安全博弈理论已被应用于许多现实安全问题中，包括重大赛事安保，城市巡航，以及濒危动物的保护中。在这些问题中，均有一个防守方希望保护一些可能被攻击的目标，一个攻击方意图攻击某个目标。在海洋生态保护的问题中，环保部门可看作是安全博弈的防守方，而意欲进行破坏活动的不法分子则为博弈的攻击方。此处“攻击一个目标”的含义即为选定保护区内的某块区域，并在此进行破坏性活动。

然而，此前关于安全博弈的研究工作并不能直接用于解决海洋生态保护区巡逻的问题，这主要是由于该问题所面临的两点新的挑战。第一，由于该问题中博弈发生的场所是大片连续水域，博弈的参与双方的可行策略均可看做是与时间相关的路线（防守方设定巡航时间和巡航路线，攻击方选择时间和路线到达欲进行破坏活动的区域，并选定破坏时长）。因此，双方均具有巨大的策略空间，使得博弈的求解面临着计算上的挑战。第二，此前关于安全博弈的研究大都假设攻击的进行时间是可忽略的，因此并未考虑选定攻击时长也是攻击方策略的一部分。诚然，在类似引爆炸弹这样的攻击活动中，攻击的发生只在于一瞬间，因此其时间可忽略不计。然而，在破坏海洋生态的问题中，攻击方希望进行的如珊瑚礁挖掘等活动均需要连续时长的保证才有可能获得收益。这两项新的挑战并不能利用此前研究工作的成果解决。

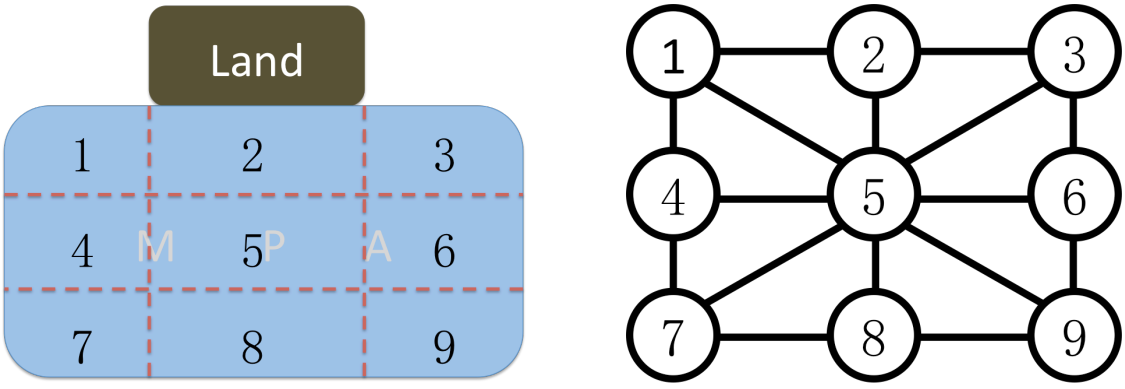
本章主要做出了如下贡献。

- 设计一种新的基于斯塔克伯格的博弈模型对海洋生态保护区的巡逻问题进行建模。在该模型中，博弈参与双方的策略均为基于时间的路线，双方的收益不仅受攻击目标的影响，也受攻击时长的影响。
- 提出一种紧凑的线性规划算法（CLP）来求解博弈均衡。在该算法中，防守方的策略被紧凑表示为图上的 $s-t$ 流。与传统的博弈求解算法相比，变量的数量被大规模减少。
- 提出一种基于策略紧凑表示和 Double Oracle 框架的算法（CDOG）以求解更大规模的博弈。该算法结合了紧凑策略表示中以 $s-t$ 流来减少问题变量的优势，以及 Double Oracle 框架中逐渐扩大博弈规模的优势，能够在避免遍历策略空间的条件性求出博弈均衡。

6.2 问题描述

防守方可将大的海洋生态保护区离散成多个区域，并基于此制定巡航策略。每个区域都是一个可能被攻击的目标。如果一个区域被攻击，即，攻击方成功的在此区域进行了破坏活动，其造成的影响，即双方的收益，与攻击发生的时间和时长均相关。图6.4(a)展示了将一个海洋生态保护区离散成 9 个区域的示例，于是离散后的保护区可

被表示成如图6.4(b)所示的 MPA 图。在巡航中，防守方可以停留在一个区域内巡逻一段时间，从而检测这段时间内也在这个区域的攻击方，也可以在区域之间移动，在移动的路途中检测攻击方。值得注意的是，防守方的路线有可能受现实情况限制而只能起始于某些特定的区域。例如，必须从海岸上的基地出发等。那么在图6.4(a)中，防守方的策略就只能从 2 号水域开始。类似的，其巡航结束后也有可能必须回归特定的水域。



(a) MPA 示例　　(b) MPA 图

图 6.4　包含 9 个区域的 MPA

攻击方的策略是选定一条路线，到达某个区域，并选定一段时长进行攻击。与安保方类似，攻击方的路线起始点同样可能首先。直观上讲，其路线必须从保护区的边缘区域开始。例如，在图6.4(a)中，攻击方不能直接进入 5 号区域。由于策略性的攻击方能够观察到防守方的巡逻策略，他在选择策略时，会综合考虑目标区域的价值以及到达目标的路线被安保方拦截的可能性。本章假设攻击方最终的目标是单个区域。

6.3　模型

本节基于斯塔克伯格模型对海洋生态保护区的巡逻问题建模。该模型下，安保方首先确定一个随机化的混合策略，攻击方能够完全得知此策略，并据此作出最优回应。首先，构建考虑时间线的博弈图。将一个海洋生态保护区定义为 n 个区域的集合 $Z = \{1, 2, \cdots, n\}$。将一天的时间离散成 τ 个时间点 $\mathbf{t} = \langle t_1, \cdots, t_\tau \rangle$，两个时间点间的间隔时间为 δ。假设在两个相邻区域间进行转移所需的时间为 δ 的整数倍（只需 δ 足够小，该假设即可成立）。假设安保方和攻击方的航行速度相同，并且他们只在 $\mathbf{t}$ 内的时间点上考虑进行区域间的移动。定义 $D = \langle d_{ij} \in \{1, 2, \cdots\} \rangle$，其中 d_{ij} 表示从区域 i 转移到相邻的区域 j 所需的时间为 $d_{ij} \cdot \delta$。

为表示双方的策略，构建有向博弈图 $G = \langle V, E \rangle$，其中 V 表示图中点的集合，每个点 $v = \langle i, t_k \in \mathbf{t} \rangle$ 对应时刻 t_k 的某区域 i。当以下的两个条件有一个满足时，即有一条边从点 $\langle i, t_k \rangle$ 指向 $\langle j, t_{k'} \rangle$。

1. $j = i, k' = k + 1$。这样的边被称作停留边。

2. i 和 j 是相邻的区域，并且 $k' = k + d_{ij}$。这样的边被称作移动边。

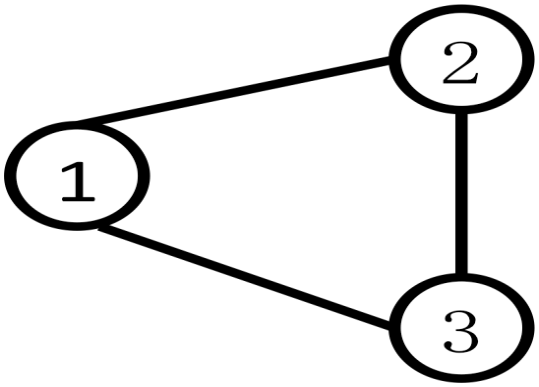


图 6.5　离散 MPA

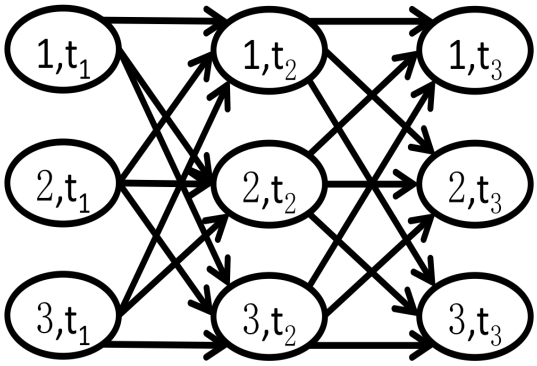


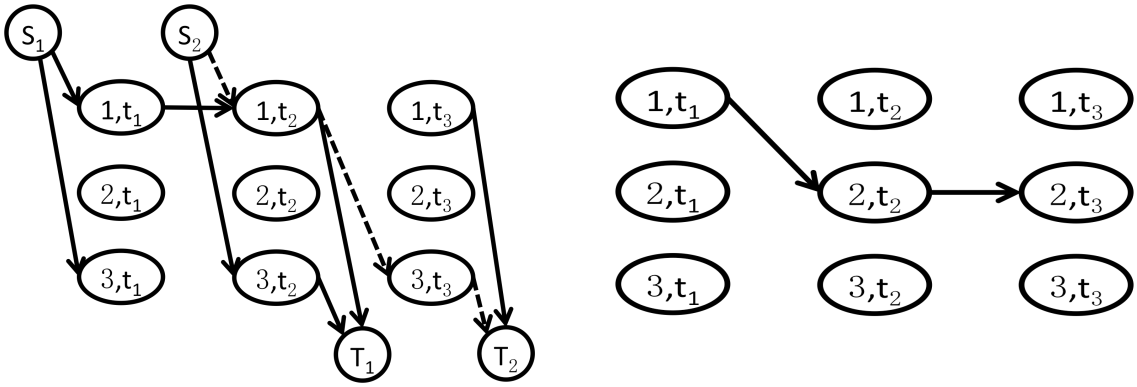
图 6.6　博弈图

考虑图6.5中包含 3 个区域的 MPA 图。令 $\mathbf{t} = \langle t_1, t_2, t_3 \rangle$，$d_{ij} = 1, \forall i, j \in \{1,2,3\}, i \neq j$。于是可得到如图6.6所示的博弈图。连接点 $\langle 1, t_1 \rangle$ 到点 $\langle 1, t_2 \rangle$ 的边表示在时间段 (t_1, t_2) 内，防守方可停留在 1 号区域内巡逻，攻击方也可能停留在 1 号区域内。连接点 $\langle 1, t_1 \rangle$ 到点 $\langle 2, t_2 \rangle$ 的边则说明如果博弈参与者在时刻 t_1 从 1 号区域出发，那么他可在时刻 t_2 到达 2 号区域。

6.3.1　防守方策略

假设防守方攻击 m 个资源，例如 m 条巡逻艇。假设每个资源可持续巡逻的时长为 $\theta\delta$。令 $Z^d \subset Z$ 表示可用于巡航路线开始点与结束点的区域的集合（此处假设巡航路线的起始点集合与结束点集合相同，该模型能够被直观的扩展到起始点集合与结束点集合不同的情况）。由于巡逻的起始时间可以是 $t_{\tau-\theta}$ 之前的任一点，于是向博弈图中增加一组虚拟起始点，即 $S = \langle S_1, S_2, \cdots, S_{\tau-\theta} \rangle$。对于 $S_k \in S$，增加一条边，从 S_k 指向点 $\langle i, t_k \rangle (\forall i \in Z^d)$。类似的，增加一组虚拟的终止点 $T = \langle T_1, T_2, \cdots, T_{\tau-\theta} \rangle$，以使得 $\forall T_k \in T$，存在一条从点 $\langle i, t_{k+\theta} \rangle (\forall i \in Z^d)$ 到 T_k 的边。那么，一个巡航资源 r 对于的一个巡航策略就是一条从 S_k 到 T_k 的网络流。防守方的一个纯策略就是 m 个这样的网络流，即 $P = \{P_r : r \in \{1, \cdots, m\}\}$。防守方的一个混合策略即纯策略上的一个分布，$\mathbf{x} = \langle x_P \rangle$，其中 x_P 表示纯策略 P 被使用的概率。

考虑图6.6中的示例。假设 $Z^d = \{1,3\}$，$\theta = 1$，于是虚拟起始点和终止点可被加上，形成如图6.7(a) 所示的结构，即，S_1 与 $\langle 1, t_1 \rangle$ and $\langle 3, t_1 \rangle$ 中的点连接，表示安保方可在时刻 t_1 从 1 号区域或 3 好区域开始巡航，而 T_1 与 $\langle 1, t_2 \rangle$ and $\langle 3, t_2 \rangle$ 内的点连接，表示从 S_1 开始的巡航应在时刻 t_2 在 1 号区域或 3 号区域结束。任何一个从 S_k 到 T_k 的网络流都是可行的巡航策略，例如，$S_1 \to \langle 1, t_1 \rangle \to \langle 1, t_2 \rangle \to T_1$ 表示防守方在 1 号区域内巡逻，$S_2 \to \langle 1, t_2 \rangle \to \langle 3, t_2 \rangle \to T_2$ 表示防守方从 1 号区域驶向 3 号区域，并在路途中巡逻。



(a) 安保方策略　　(b) 攻击方策略

图 6.7　博弈图中的策略

6.3.2　攻击方策略

假设攻击方的路线起始点为集合 $Z^a \subset Z$。攻击方的一个纯策略包含两部分：第一，一条从起始点到目标区域的路径。第二，到达目标区域后进行破坏性活动的时长。假设攻击的时长是 δ 的整数倍（当 δ 值足够小时该假设可成立）。定义攻击方的纯策略为 $Y = \langle H_Y, A_Y \rangle$，其中 H_Y 是一条从某个起始点到目标点 $\langle i, t_k \rangle$ 的路径，而 A_Y 是一条包含了 l 个临近的停留边的路径，表示在区域 i 进行的攻击活动从时刻 t_k 持续到时刻 t_{k+l}。一旦攻击方成功的执行了攻击策略，珊瑚礁及海洋生态就会被破坏，因此该模型并不关注破坏发生后攻击方的撤退路线。依旧考虑前文提及的有 3 个区域的示例。假设 $Z^a = \{1\}$，图6.7(b) 显示了一个攻击方的可行策略，其中他在时刻 t_1 从 1 号区域进入保护区，在时刻 t_2 到达他的目标区域，2 号区域，然后在 2 号区域进行破坏活动，直到时刻 t_e 停止。由于攻击方在观察到防守方的策略之后进行行动，因此总是存在一个最优的纯策略回应，模型仅考虑攻击方的纯策略。

6.3.3　收益

假设博弈图中的每条边 $e = \langle \langle i, t_k \rangle, \langle i, t_{k+1} \rangle \rangle$ 均对应某一价值 V_e，该价值反映了如果攻击方成功的在区域 i，在时间段 (t_k, t_{k+1}) 进行破坏性活动所能获得的收益。如果攻击方成功执行了策略 $Y = \langle H_Y, A_Y \rangle$，他将获得收益

$$U(Y) = \sum_{e \in A_Y} V_e. \tag{6-1}$$

假设如果攻击方在到达目标区域的路途中或者进行破坏的过程中被防守方成功截获，那么攻击失败，攻击方与安保方获得的收益均为 0。否则攻击成功，攻击方获得的收益为 $U(Y)$，而安保方相应获得 $-U(Y)$。直观上讲，由于环保部门通常不能够检测巡逻过程中遇到的每一条船只，因此即使防守方与攻击方在同一时间处于同一地点，攻击方是否会被检查截获仍然存在一定概率。为表示该概率，模型首先进行两项假设。

假设 1 如果防守方和攻击方在边 e 首次相遇，那么攻击方被拦截的概率，或称拦截因子，为 $f_e \in [0,1]$。

接下来的假设是关于在首次相遇未被拦截的情况下，攻击方与防守方在后续的相遇中被拦截的概率。由于巡逻区域往往也是破坏性活动容易发生的区域，如果环保部门在这样的区域内反复遇到某船只，那么该船只更有可能是意欲进行破坏性活动的不法分子。因此，在后续的相遇中，防守方应以更高的概率拦截已经遇到过但并未拦截的攻击方。

假设 2 假设防守方和攻击方已在边 $e_1, e_2, \cdots, e_{k-1}$ 相遇过，攻击方未被拦截，那么如果双方在边 e_k 再次相遇，那么攻击方被拦截的概率为 $\min\{1, \frac{f_{e_k}}{1-\sum_{i=1}^{k-1} f_{e_i}}\}$。

假设2中所示概率值值域为 $[f_{e_k}, 1]$，这满足后续遇到的船只应以更高的概率被检测的直观印象。基于假设1和2，给定一个巡航资源 r 的巡航路线 P_r，给定一个攻击方的纯策略 Y，假设 P_r 和 Y 共享的边为 $P_r \cap Y = \langle e_1, e_2, \cdots, e_k \rangle$，那么防守方被拦截的概率即为其在各共享边上被拦截的概率的总和，即

$$f_{e_1} + \sum_{e_k \in P_r \cap Y} \max\{0, 1 - \sum_{i=1}^{k-1} f_{e_i}\} \min\{1, \frac{f_{e_k}}{1 - \sum_{i=1}^{k-1} f_{e_i}}\} = \min\{1, \sum_{i=1}^{k} f_{e_k}\}. \tag{6-2}$$

于是，给定一个防守方的纯策略 $P = \{P_r\}$ 和一个攻击方的纯策略 Y，可将攻击方被拦截的概率表示为

$$dp(P, Y) = \min\{1, \sum_{r=1}^{m} \sum_{e \in P_r \cap Y} f_e\}. \tag{6-3}$$

该模型未考虑防守方和攻击方的路线仅在节点处相交，而不共享任何边时，攻击方被拦截的概率。这是由于在博弈图中，节点事实上表示时长为 0。如果双方的路线尽在节点处相交而不共享任何边，则意味着双方在同一处停留的时间为 0，因此模型认为在这种情况下防守方不能拦截攻击方。

基于方程 (6-3)，给定一个防守方纯策略 $\langle P, Y \rangle$，攻击方的期望收益即为

$$U^a(P, Y) = (1 - dp(P, Y))U(Y). \tag{6-4}$$

给定一个防守方的混合策略 $\mathbf{x} = \langle x_P \rangle$ 和一个攻击方的纯策略 Y，攻击方的期望收益可表示为

$$U^a(\mathbf{x}, Y) = \sum_{P} x_P U^a(P, Y), \tag{6-5}$$

相应的，防守方的期望收益即为 $U^d(\mathbf{x}, Y) = -U^a(\mathbf{x}, Y)$。

6.3.4 均衡策略

模型以斯塔克伯格均衡策略为解。在零和博弈的假设下，斯塔克伯格均衡与纳什均衡等价，均可用最大最小的方法求解，也即，在攻击方采取最优回应策略的假设下，计算防守方的最优策略。令 $\mathcal{A}$ 表示攻击方的策略空间，令 $\mathbf{X}$ 表示防守方的混合策略空间。给定一个防守方的混合策略 $\mathbf{x}$，令 $f(\mathbf{x}) \in \mathcal{Y}$ 表示攻击方的最优回应方程。于是，一组均衡策略满足

$$U^a(\mathbf{x}, f(\mathbf{x})) \geq U^a(\mathbf{x}, Y'), \forall Y' \in \mathcal{Y}, \tag{6-6}$$

$$U^d(\mathbf{x}, f(\mathbf{x})) \geq U^d(\mathbf{x}', f(\mathbf{x}')), \forall \mathbf{x}' \in \mathcal{X}. \tag{6-7}$$

方程 (6-6) 表示攻击方在给定的防守方策略下做出最优回应。方程 (6-7) 表示防守方使用最优策略。在零和博弈的假设下，假如有多个攻击方策略均为最优回应，那么无论攻击方选择哪一个，防守方的收益都相同。

6.4 紧凑策略表示线性规划算法

防守方的纯策略数量随着博弈规模的增大而呈指数级增长，本节通过紧凑表示防守策略来解决这一问题。令 c_e 表示博弈图 G 中的边 e 的保护情况，也就是这个边平均享有的巡逻资源数量。给定一个防守方的混合策略 $\mathbf{x}$，有

$$c_e = \sum_P x_P P(e), \forall e \in E, \tag{6-8}$$

其中 $P(e)$ 表示纯策略 P 中经过边 e 的巡逻资源个数。于是，基于方程 (6-8)，给定一个安保方混合策略 $\mathbf{x}$ 和与之相应的覆盖向量 $\mathbf{c} = \langle c_e : e \in E \rangle$，再给定攻击方的纯策略 Y，攻击方的期望收益即可表示为

$$U^a(\mathbf{x}, Y) = U^a(\mathbf{c}, Y) = (1 - \min\{1, \sum_{e \in Y} c_e f_e\}) U(Y). \tag{6-9}$$

给定一个覆盖向量，即可构建一个与之相应的混合策略。于是，求解均衡策略的问题也即求解防守方的最优覆盖向量。为求解覆盖向量，首先考虑将博弈图 G 转化为一个扩展图 EG。EG 是由 G 的一组受限副本组成的，也即 G 的一组子图。不同的子图对应于安保方不同的巡逻起始点。对于对应于起始点 t_k 的子图 G_k，其仅包含 G 内的点 $\langle S_k, T_k, v = \langle i, t_{k'} \rangle : i \in Z, k' \in \{k, \cdots, k + \theta\} \rangle$。于是，任一个起始于 t_k 时刻的巡逻策略都可以被表示为子图 G_k 上的 $S_k - T_k$ 流。令 $z_k(e)$ 表示起始于时刻 t_k 的巡航策略讲过边 e 的平均次数。令 $\Gamma(e)$ 表示包含边 e 的子图的集合，那么 $c_e = \sum_{k:G_k \in \Gamma(e)} z_k(e)$。以下线性规划可用于计算最优的覆盖向量。

$$\text{CLP:} \qquad \max_{\mathbf{c},\mathbf{z}} U \tag{6-10}$$

$$U \leq -(1-\min\{1, \sum_{e\in Y} c_e f_e\})U(Y), \forall Y \tag{6-11}$$

$$c_e = \sum_{k:G_k\in\Gamma(e)} z_k(e), \forall e \in G \tag{6-12}$$

$$\sum_{\langle v',v\rangle\in G_k} z_k(\langle v',v\rangle) = \sum_{\langle v,v''\rangle\in G_k} z_k(\langle v,v''\rangle), \forall G_k \tag{6-13}$$

$$\sum_{S_k\in S}\sum_{i\in Z^d} z_k(\langle S_k, \langle i, t_k\rangle\rangle) = m \tag{6-14}$$

$$\sum_{T_k\in T}\sum_{i\in Z^d} z_k(\langle\langle i, t_{k+\theta}\rangle, T_k\rangle) = m \tag{6-15}$$

约束 (6-13) 使得网络流的限制得以满足。约束 (6-14) 和 (6-15) 使得进入博弈图和流出博弈图的资源总数不超过 m，即巡逻资源个数。约束 (6-11) 表明攻击方使用最优的策略 Y 以最大化自身收益，或与之等价的，最小化防守方的收益 $-(1-\min\{1, \sum_{e\in Y} c_e f_e\})U(Y)$。该约束事实上高估的防守方的收益，这是由于 $1-\min\{1, \sum_{e\in Y} c_e f_e\}$ 事实上高估了拦截概率。这是因为，$\min\{1, \sum_{e\in Y} c_e f_e\}$ 只限制了纯策略内一条边被保护的概率不大于 1，但这使得一个纯策略内可能发生 $\sum_{r=1}^{m}\sum_{e\in P_r\cap Y} f_e > 1$。然而根据方程 (6-3)，该拦截概率也应该被限定在 1 以内。因此，该线性规划算法事实上给出了防守方收益的一个上界。但是，一旦得出覆盖向量，就能够计算攻击方实际上的最优回应。Yin et al. [131] 证明了给定覆盖向量 **c**，即可在多项式时间内构造一个与之相对应的防守方混合策略 **x**，其对每条边的巡航概率均与 **c** 相同。实验显示实际收益与该线性规划算法计算得出的上界非常接近。

6.5 紧凑策略与 Double Oracle 的结合

方程 (6-11) 所表示的约束的数量，也就是攻击方策略的数量，随着博弈规模的增加而呈指数级增长。这使得线性规划算法依然不能解决大规模博弈。为解决该问题，本节提出算法 CDOG，结合了紧凑策略表示法与 Double Oracle 框架 [48]。

Double Oracle 框架的基本思想在于从求解一个受限的小规模博弈入手，逐渐增加求解的博弈规模，在避免遍历整个策略空间的条件下求出均衡解。与策略的紧凑表示结合后，框架首先从一个子图赏的博弈开始，限定双方的策略为该子图上的网络流，并利于 CLP 求解子图上的均衡集。根据以求得的子图均衡解，分别计算在全策略空间中博弈参与双方的最优回应，并扩展子图。重复此过程直至子图不能被扩展。那么子图上的均衡策略也就是原博弈的均衡策略 [78])。这个框架使得有可能在避免遍历全策略空间的情况下求得博弈的均衡解。

对于传统的 Double Oracle 框架，一个问题在于由于子博弈的扩展较慢（每次最多博弈双方各增加一个纯策略），因此算法需用经过较长时间才能收敛，而且，在子博弈被扩展到一定程度以后，单次子博弈的求解时间也较长。例如，[46] 所提出的基于

Algorithm 12: CDOG

1 初始化子图 G';
2 **repeat**
3 $(\mathbf{c}, \mathbf{y}) \leftarrow \text{CLP}(G')$;
4 $P \leftarrow \text{DO}(\mathbf{y})$ // 防守方 Oracle
5 $Y \leftarrow \text{AO}(\mathbf{c})$ // 攻击方 Oracle
6 $G' \leftarrow G' \cup \{P\} \cup \{Y\}$;
7 **until** G' 未被扩展;
8 **return** $(\mathbf{c}, \mathbf{y})$.

Double Oracle 框架的算法求解具有 200 个点的问题就需要 9 个小时，而该问题仅相当于本问题中 10 个区域，20 个时间点的情形。而 CDOG 算法在结合了策略的紧凑表示后，每次循环均至少增加子图的一条边，也就是增加所有通过此边的纯策略，而不是仅增加一个纯策略。而且，由于求解利用网络流的技术，即使增加了多个纯策略，CLP 所需处理的变量个数并不会显著增加，这使得 CDOG 比传统的基于 Double Oracle 的算法更具有可扩展性。CODG 的主要结构如算法12中所示。第1行首先选取原博弈图 G 的的子图 G'。限定限定攻击方和防守方的策略均只能是子图 G' 内的网络流，第3计算该受限博弈的均衡。$\mathbf{c} = \langle c_e \rangle$ 由 CLP 求解得出。$\mathbf{y}$ 是攻击方的混合策略，其支持集为 $\mathbf{Y}'$ 内的纯策略，$\mathbf{y}$ 内各纯策略被使用的概率来自于 CLP 的对偶问题中与方程 (6-11) 对应的变量的值。通过第4行到第7行，CDOG 通过调用两个 Oracle 来计算攻守双方在全策略空间内的最优回应，并据此扩展子图。该过程重复进行，直到子图不再被扩展。如前文所述，Double Oracle 框架本身并非完整的算法，因为两个 Oracle 的具体执行依具体问题而定。接下来研究两个 Oracle 的具体执行过程。

6.5.1 防守方 Oracle

防守方的最优回应，也即全策略空间内的某个纯策略，是原博弈图 G 内的一个网络流。在给定攻击方使用混合策略 $\mathbf{y}$ 的情况下，该策略使得防守方的期望收益最大。防守方 Oracle（DO）根据如下的混合整数线性规划算法来计算防守方的最优回应。

$$\text{DO}: \quad \max_{\mathbf{c},\mathbf{z}} \sum_{Y \in \mathbf{y}} -Y_i(1 - \min\{1, \sum_{e \in Y} c_e f_e\})U(Y) \tag{6-16}$$

$$z_k(e) \in \{0, 1, \cdots, m\}£\neg \quad \forall z_k(e) \tag{6-17}$$

$$\text{Constraints (6-12)–(6-15).} \tag{6-18}$$

约束 (6-17) 确保每个边的覆盖都是整数，于是覆盖向量 $\mathbf{c}$ 对应于一个纯策略。CLP 中的约束 (6-12) - (6-15) 在此处可直接被使用，这是由于无论纯策略还是混合策略，均需满足网络流的基本限定。防守方 Oracle 的目标在于给定攻击方使用的混合策略 $\mathbf{y} = \langle Y_i \rangle$

Algorithm 13: 寻找最优路径 $(\mathbf{c}, v)$

```
Input: c, v, Output: H(v), γ(v));
if i ∈ Z^a then
    H(v) = v, γ(v) = 0;
else
    Λ(v) ← 节点 v 的前驱结点， pre ← −1, min ← ∞;
    for v′ ∈ Λ(v) do
        if H(v′) + c_⟨v′,v⟩ f_⟨v′,v⟩ < min then
            min = H(v′) + c_⟨v′,v⟩ f_⟨v′,v⟩, pre = v′
    H(pre), γ(pre) ← 寻找最优路径 (c, pre);
    H(v) ← H(pre) ∪ ⟨pre, v⟩, γ(v) ← min
```

求得防守方的最优回应，其中 Y_i 是第 i 个攻击方策略被使用的概率，其值来自于 CLP 的对偶问题中与方程 (6-11) 所代表的一组不等式的第 i 个相对于的变量的值。如果防守方 Oracle 求得的最优策略经过了子图 G' 未包含的某些 G 中的边，那么将这些边加入 G' 以扩展该子图。

6.5.2 攻击方 Oracle

由于攻击方策略受到网络流之外的额外限制，例如，策略应结束于停留边，因此攻击方的最优回应不能直接利用网络流的技术来求解。攻击方 Oracle（AO）通过以下三步来求解在给定防守方的混合策略对应的覆盖向量时攻击方的最优回应。

1. 对于博弈图中的每个节点 $v \in V$，计算攻击方到达 v 可行的最优路线 $H(v)$，该最优路线满足攻击方从某起始点出发到达 v 的路途中被拦截的概率最小。
2. 基于 $H(v)$，计算当攻击活动从节点 v 开始时，其最优的持续时长。
3. 综合前两步，选取最优的路径和攻击时长，以组成最优的攻击方策略。

接下来考虑这三个步骤的具体执行过程。

对于第一步，令 $\gamma(v) = \sum_{e \in H(v)} c_e$ 表示当攻击方选择路径 $H(v)$ 时他在路途中被拦截的概率。如果 $v = \langle i, t_k \rangle$ 并且 $i \in Z^a$，那么攻击方可以直接在节点 v 进入博弈图，于是 $H(v)$ 只包含一个节点 v，并且 $\gamma(v) = 0$。如果 $i \notin Z^a$，令 $\Lambda(v)$ 表示一个博弈图内节点 $v' \in G$ 的集合，这些点满足 $\langle v', v \rangle$ 是博弈图 G 内的一条边，于是为到达节点 v，攻击方必须先到达某个节点 v'，再经由边 $\langle v', v \rangle$ 到达节点 v。于是有

$$\gamma(v) = \min_{v' \in \Lambda(v)} \{1, H(v') + c_{\langle v', v \rangle} f_{\langle v', v \rangle}\}. \tag{6-19}$$

令 $v_{pre} = \arg\min_{v' \in \Lambda} \{H(v') + c_{\langle v', v \rangle}\}$,，于是 $H(v) = H(v_{pre}) \cup \langle v_{pre}, v \rangle$，那么 $H(v)$ 可通过算法13所示的过程计算出。那么，未计算所有节点对应的 $H(v)$，每个边需要被遍历

一次，于是第一步的复杂度为 $O(|E|)$。

对于第二步，如果攻击方在节点 $v=\langle i,t_k\rangle$ 开始进行活动并选择活动的持续时长为 l，那么总体来讲，攻击方被拦截的概率取决于 $\gamma(v)$ 和活动的进行过程中被拦截的概率，也就是

$$\iota(v,l)=\min\{1,1-\gamma(v)-\sum_{j\in\{1,2,\cdots l\}}c_{e_j}f_{e_j}\}. \tag{6-20}$$

此处 $e_j=\langle\langle i,t_{k+j-1}\rangle,\langle i,t_{k+j}\rangle\rangle$。该攻击活动能够获得的总价值是所有停留边 e_j 所对应的价值的总和，也就是

$$\ell(v,l)=\sum_{j\in\{1,2,\cdots l\}}V_{e_j}. \tag{6-21}$$

于是期望收益即为 $U(v,l)=(1-\iota(v,l))\cdot\ell(v,l)$。那么，从 v 开始的最优攻击时长就是

$$l_v^{opt}=\arg\max_{l\in\{1,\cdots,\tau-k\}}U(v,l). \tag{6-22}$$

为计算每个节点 $v=\langle i,t_k\rangle$ 开始攻击所对应的最优攻击时长，需要遍历从 v 开始的所有可能的攻击时长，也就是 $\{1,\cdots,\tau-k\}$。于是，第二步的时间复杂度为 $O(n\tau^2)$。给定前两步的结果，第三部只需选择对应攻击方最优收益的节点 v，即，$v=\arg\max_{v'\in V}U(v',l_{v'}^{opt})$，并基于 $H(v)$ 和 l 构建攻击方的最优策略 Y。如果该策略经过了子图 G' 中未包含的边，那么将此边加入 G' 以扩展子图。

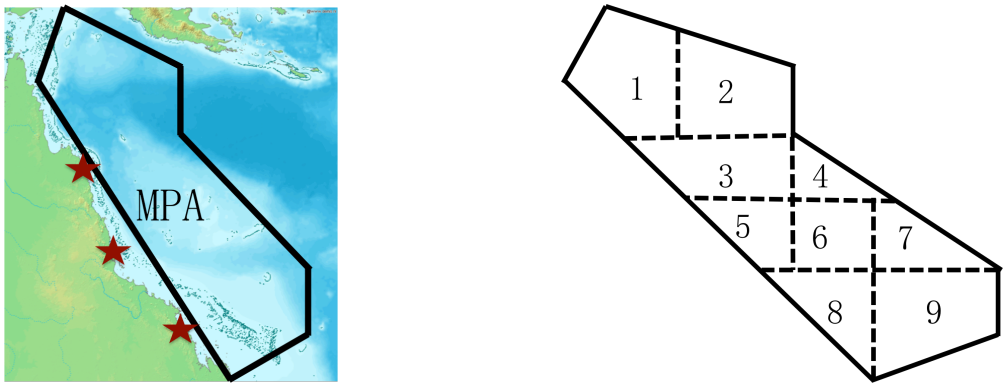
6.6 实验与分析

本节从算法性能，可扩展性与鲁棒性三个方面来检测算法。CLP 通过 Knitro 8.0.0. 求解。图中的每个点都是 30 次采样实验的平均结果。

6.6.1 基本设置

实验对于 MPA 图的设置基于位于澳大利亚昆士兰东海岸的大堡礁保护区的地形图。负责大堡礁的维护工作的机构叫做 The Great Barrier Reef Marine Park Authority (GBRMPA)，该机构的基地位于海岸上（图6.8(a)）。由于出于不同的目的，海洋生态保护区可以被离散成不同的拓扑结构 [115]，因此在每个实验样本数，对该区域随机分割以获取 MPA 结构。总体来讲，每个 MPA 图均代表一种对于大片水域的分割方式，防守方尽可以从靠近海岸一边的区域开始起巡航路线。图6.8(b)显示了将整个保护区分割为 9 个区域的样例，该分割方式可得到如第三节中所示的 MPA 图。

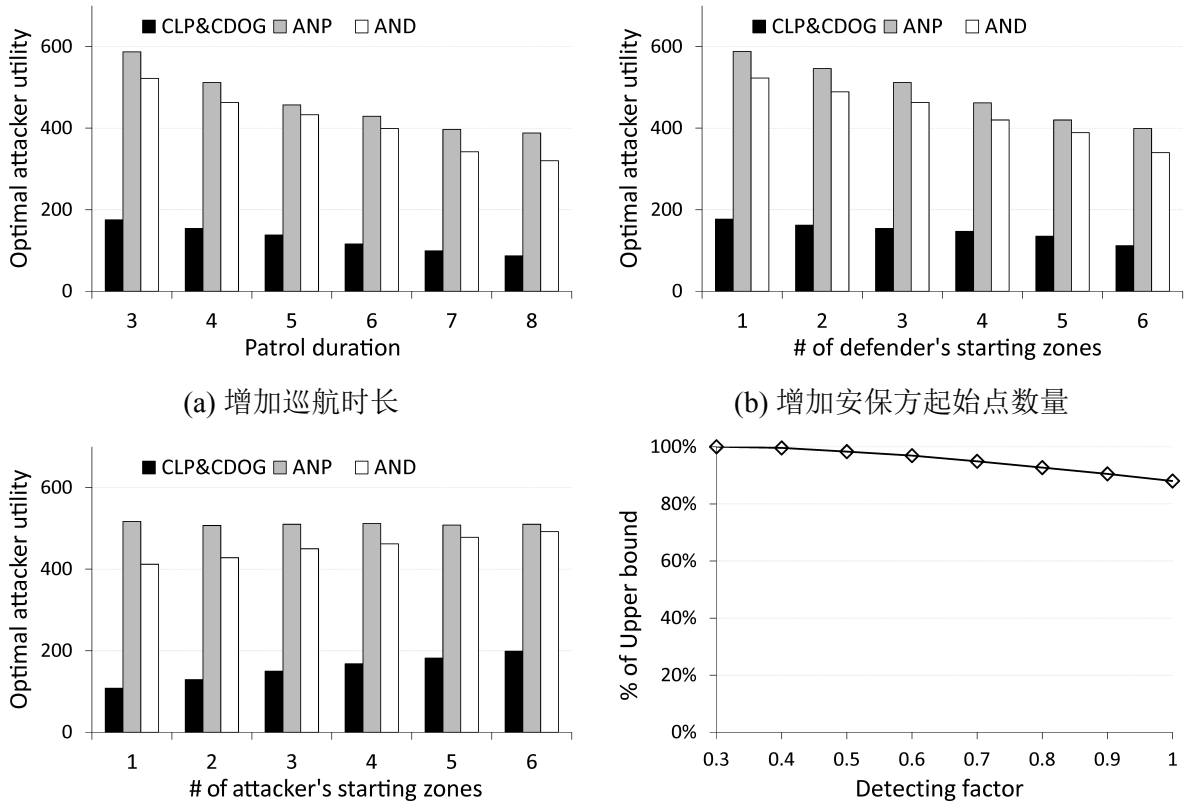
为描述保护区巡航问题中巡航资源的航行速度可能各不相同的性质，在实验中，到达相邻的两个区域所需的时间，也就是 d_{ij}，在 $[1,\frac{|\mathbf{t}|}{2}]$ 内随机产生。其中 $|\mathbf{t}|$ 是博弈中时



(a) 大堡礁 MPA，红星为 GBRMPA　　(b) 离散示例

图 6.8　实验 MPA 图设置

间点的个数。基于该 MPA 图和 d_{ij} 构建博弈图，每个区域有 |t| 个副本。每个边的价值在 (0, 100] 内随机产生。



(a) 增加巡航时长　　(b) 增加安保方起始点数量

(c) 增加攻击方起始点数量　　(d) 上界百分比

图 6.9　算法性能

6.6.2　算法性能

算法性能通过攻击方收益来衡量。在零和博弈的假设下，更低的攻击方收益意味着更高的算法性能。由于此前的工作并不能完全的解决本章提出的问题，本节将提出的算法与两个基线算法 ANP 和 AND 作比较，这两个算法在基于此前工作所做的假设

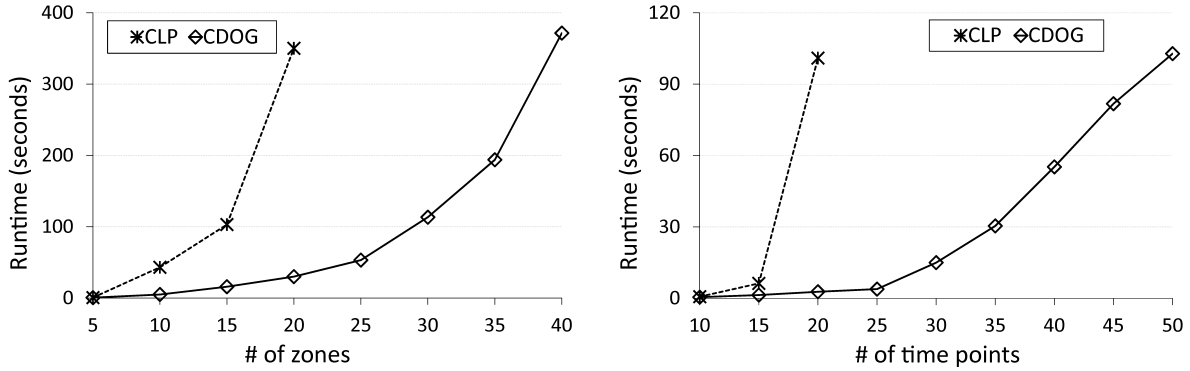
的基础上，解决本章提出的问题。基线算法计算出的防守策略在本问题中也是可行的策略。这两个基线算法的具体情况如下。

1. ANP 的基础假设是，攻击方不需选取路径到达攻击目标，而是可以直接“空降”至目标处。该假设此前的多项研究中被使用 [32]。
2. AND 假设攻击方虽然需要选取路径到达目标，但攻击时间是可忽略的，也即攻击方不选择攻击时长。改价这与此前工作 [131] 的模型类似。

非特殊说明的情况下，实验设置假设有 3 个巡逻资源，9 个区域，时间点有 12 个。

在图6.9(a), 6.9(b)和6.9(c)中，y 轴指示攻击方的最有收益，x 轴分别指示一个巡逻资源的巡逻时长（即 θ 的值），防守方起始点的数量，以及攻击方起始点的数量。由于 CLP 和 CDOG 所计算出的攻击者策略相同，因此他们的结果在同一个柱中显示。无论这三个变量如何取值，CLP 和 CDOG 的结果显著强于 ANP 和 AND 的结果。图6.9(a)显示所有算法求得的攻击方收益随着巡逻时长的减少而增加。图6.9(b)显示随着防守方起始点数量的上升，防守方的可行策略增加，攻击方收益也呈现降低趋势。在图6.9(c)中，随着攻击方起始点数量上升，攻击方的可行策略增加，因此攻击方的收益呈现升高趋势。

图6.9(d)显示了真实的防守方最优收益占 CLP 和 CDOG 所计算得出的攻击方收益上界的比值。x 轴显示拦截因子 f_e 的取值范围，例如，$x = 0.3$ 表示 f_e 在 $(0, 0.3]$ 内随机产生。在这组实验中，巡逻时长被固定为 3。方程 (6-3) 显示，f_e 越小，CLP 和 CDOG 就越不容易高估拦截概率。图6.9(d)显示出真实收益与收益上界的比值有下降趋势。但即使 $f_e = 1$，真实收益与收益上界的比值仍达到 90%，这意味着 CLP 和 CDOG 所计算的收益上界非常接近真实收益。

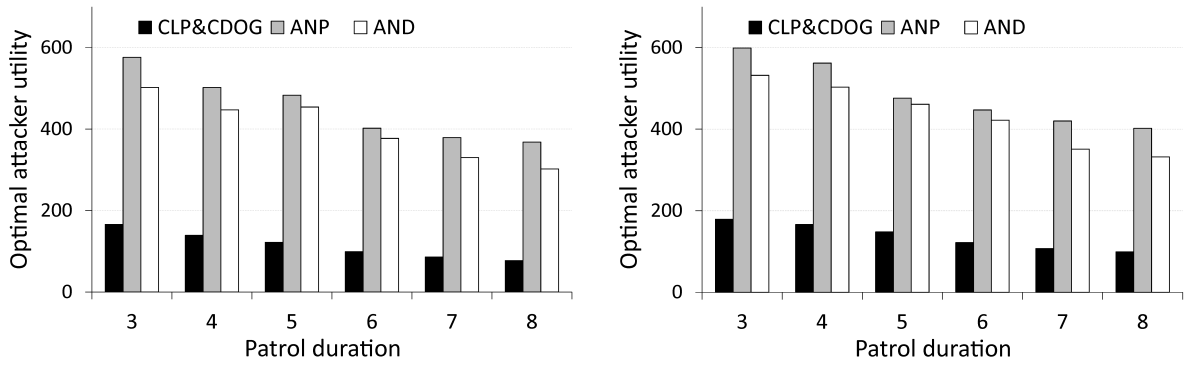


(a) 增加目标数量　　(b) 增加时间点数量

图 6.10　可扩展性

6.6.3　可扩展性

在图6.10(a)和6.10(b)中，y 轴指示算法的运行时间。在图6.10(a)中，x 轴显示区域的数量从 5 增长到 40。在图6.10(b)中，x 轴指示时间点的数量从 10 增长到 50。在这两幅



(a) 观测噪音　　(b) 收益噪音

图 6.11　鲁棒性

图中，CLP 的运行时间都显示出了更明晰的增长趋势。事实上，当目标数量超过 20 个，时间点的数量超过 20 以后，CLP 就会因为内存不足而无法运行。另一方面，CDOG 可以在约 6 分钟内求解有 40 个目标的博弈。本节还检测了在 Double Oracle 框架下各部分的运行时间具体情况。图6.12展示了 CDOG 算法中三个部分运行时间的一个样例，该样例中算法在 10 轮后收敛。CLP 的运行时间随着轮次增加增长明显。但由于两个 Oracle 的问题规模并没有显著增长，因此其运行时间也未受明显影响。

	Iteration									
	1	2	3	4	5	6	7	8	9	10
CLP	4301	5172	5637	6131	6151	6231	6332	6300	6449	6547
DO	1626	1620	1650	1652	1648	1715	1639	1620	1640	1634
AO	328	352	322	323	326	319	320	334	318	320

图 6.12　CDOG 中各部分运行时间

6.6.4　鲁棒性分析

首先考虑攻击方的观测噪音。在真实的防守策略的基础上增加一个均值为 0，标准差为 0.5 的高斯噪音作为攻击方的观测数据。图6.11(a)显示了在于图6.9(a)配置形态的情况下考虑噪音后的算法性能。与图6.9(a)相比，各算法对于的攻击方收益均降低，这是由于观测噪音使得攻击方偏离了最优回应，而在零和假设下，这也就会使得攻击方的收益升高。即使在噪声存在的情况下，CLP 和 CDOG 依然显著优于 ANP 和 AND 的结果。接下来考虑防守方的收益噪音。对于每条停留边，在其原始的价值基础上增加一个标准高斯噪音，作为防守方对于该停留边价值的判断。图6.11(b)显示了在与图6.9(a)配置相同的情况下，考虑收益噪音后各算法的性能。与图6.9(a)相比，算法得到的攻击者收益较高，这是由于收益噪音使得防守方偏离了最优策略。但即使存在收益噪音，CLP 和 CDOG 的效用也明显高于 ANP 和 AND。

6.7 本章小结

本章研究为海洋生态保护区设计最优的巡航策略的问题，以保护珊瑚礁等珍贵的生态资源。本章基于斯塔克伯格模型对该问题进行建模，模型不仅考虑了攻守双方均可采用基于时间的路线作为策略，也考虑了攻击方可选取不可忽略的攻击时长。本章首先提出一种线性规划算法 CLP 来求解博弈均衡，该算法利用网络流来紧凑的表示防守方的纯策略和混合策略，从而大大减少了需处理的变量数量。本章还提出了可扩展性更强的，综合考虑策略的紧凑表示和 Double Oracle 框架的 CDOG 算法，该算法从基于子图的子博弈着手，逐步扩大子图，可在避免遍历全策略空间的情况下求得博弈均衡解。实验结果显示本章提出的算法 CLP 和 CDOG 取得了显著优于基线算法的结果，而 CDOG 比 CLP 有更强的可扩展性。

第七章　总结与展望

7.1　本文总结

关于复杂收益安全博弈的研究在理论上和应用上都具有重大意义。本文研究了三种具有复杂收益形式的安全博弈，包括：1. 非独立、多攻击目标安全博弈，其中各个目标的价值并不是相互独立的，而攻击方可以攻击多个目标；2. 动态收益安全博弈，其中目标的重要程度随时间而变化，安保方可以在任何时刻将资源在目标间转移，攻击也可能发生在任何时间；3. 图安全博弈，其中攻守双方均需要选择图中的路线到达攻击目标，因此安保方可以在攻击方到达目标的路途中对其进行拦截。针对这三种博弈形式，本文研究了保障选举进行，保护重大赛事，城市安保和海洋生态保护区巡航这四个代表性问题，为每个问题进行了建模，并提出了高效的求解博弈均衡的算法。

针对保障选举依法进行的问题，本文考虑了破坏性攻击（不法分子的目标是阻止最初的赢家获胜）、建设性攻击（不法分子的目标是使得某个原本不能获胜的候选人获胜）两种攻击方式。本文分别分析了这两种攻击方式的复杂度，并分析了抵制这两种攻击的复杂度。本文利用斯塔克伯格博弈模型，对保障选举免受这两种攻击影响的问题进行建模。针对抵制破坏性攻击，本文提出了一种线性规划算法，求解抵制攻击，保护选民的最优策略。又提出了一种基于 Double Oracle 框架的更高效的算法，在大多数情况下能够在避免遍历策略空间的情况下即求得最优解。针对抵制建设性攻击，本文提出了一种混合整数线性规划算法以求解抵制攻击的最优策略。本文还提出了一种启发式算法，能够在一部分情况下在多项式时间内求出最优策略。此外，本文还提出了一种近似算法以求解一般性问题中“较好”的策略，该算法能够在绝大多数情况下返回最优解。本文在真实数据集和模拟数据集上进行试验，验证了模型和算法的有效性。

针对重大赛事安保问题，本文设计了一种博弈模型，考虑了安保部门与不法分子均具有连续的策略空间的情况。在此博弈模型的均衡策略下，即安保方的最优策略下，安保方可能遭受的最大的损失最小。本文首先关注了资源的转移时间远小于赛事的进行时间的情形，并提出算法 SCOUT-A 来求解此情形下安保方的最优策略。然后，本文又研究了更具一般性的资源转移时间不可忽略的情况。针对这种情形，本文从离散时间假设着手，提出算法 SCOUT-D 来求解在离散时间假设下的安保方最优策略，又在此基础上设计出算法 SCOUT-C，以求解连续策略空间、资源转移时间不可忽略的情形下的安保方最优策略。最后，本文通过实验验证了算法的有效性。

针对城市安保问题，本文采用具有连续策略空间的斯塔克伯格博弈模型对其进行建模，并以强斯塔克伯格均衡解作为解空间。在此均衡解下，本文首先考虑资源转移时

间可忽略不计的情形，采用紧凑的方法表示安保部门的混合策略，设计算法 COCO 来求解此情形下最优混合策略。基于 COCO 的结果（紧凑表示的混合策略），本文提出一种采样方法，以确定每次进行城市安保所使用的策略。本文随后考虑了资源转移时间不能忽略的情况，并提出算法 ADEN，其能够在多项式时间内求出 ϵ 近似解。实验证明 COCO 和 ADEN 均取得了比现存算法更好的效果。

针对海洋保护区巡逻问题，本文提出了基于图的安全博弈模型，图中的每个节点表示一个时间和地点的二元组，图中的任一条路线均为安保部门或不法分子的可行策略。本文利用网络流的方式来紧凑地表示安保部门的混合策略，并提出一种线性规划算法 CLP 求解安保部门的最优巡航路线。此外，本文还将网络流思想和 Double Oracle 框架相结合，提出 CDOG 算法，高效求解最优策略。实验结果表明 CLP 和 CDOG 均取得了明显好于现存算法的效果，CDOG 的可扩展性也明显好于已知的最好算法。

综上所述，本文研究了三种复杂收益安全博弈，关注了这三种博弈中的四个代表性问题。针对每一个问题，本文均采用了新的建模方式，以解决此前研究工作中的多种假设所造成的局限性。本文提出了高效的算法以求解各模型中的最优安保策略。实验证明本文所提出的模型和算法取得了比现有研究成果更好的效果，拓展了安全博弈论领域的研究边界。

7.2 下一步工作展望

安全博弈的理论研究以及相关应用已在学术界和工业界取得了令人瞩目的成就。然而，现存的研究依然存在一些局限，还有很多问题尚未被触及。本文从三种具有复杂收益形式的安全博弈着手，对该领域的研究进行了一定的扩展。接下来本人将从以下三个方面进一步深化该研究。

第一是关于安全博弈与选举的更多探索。本文研究了利用安全博弈理论，设计安保策略，使得多数制选举免受攻击影响的问题。虽然多数制是较常见的选举制度，对多数制选举的研究至关重要，但不可否认的是，其他选举形式，例如优序投票制（每个选民自行依照喜好排序候选人，逐次淘汰最少被列为第一顺位的候选人，直到某个候选人得到过半数的第一顺位，即认为该候选人获胜），孔多赛制（将每个候选人与其他所有候选人一一比较，击败所有其他候选人的便是赢家）等，也在很多现实问题中发挥着重要的作用。不同的投票制度被攻击的难度各不相同，保障其免受攻击的最优策略也不能用同样的算法计算。因此，关于安全博弈与选举的研究尚有很多问题值得探索。

第二是安全博弈中的人机结合优化问题。在现实世界的安保中，安保方的策略往往被要求满足一些特殊限制，例如某些时刻需要保护某些目标等。在此前的工作中，这些限制通常被直接加入优化问题的约束条件中，从而求得严格符合限制要求的解。然而，这样的解有可能带来较差的安保方收益，而仅需放松一些限制，就有可能使得安保方收益大幅度提高。因此，建立人机结合的优化模型，使得模型能够实时从安保方

获得可放松的限制的信息，并据此计算更好的安保策略，是一个极具潜力的研究方向。[2] 中介绍了该方向的问题和挑战——设计可解决大规模问题的高效算法，分析增加限制与放松限制所对应的收益变动等，都是极具挑战的重要问题。

第三是重复博弈与大数据在安全博弈研究中的应用。一些策划与进行相对简单，且对应的惩罚力度较低的犯罪活动，如乱砍滥伐、偷猎野生动物等，往往以较高的频率发生，难以通过一次性博弈模型进行数学描述。因此，利用重复博弈的理论模型对其进行建模，设计具有可持续性的安保策略，具有重大的现实意义。同时，因为这些活动的高频性，安保方能够掌握较多关于此类犯罪活动的历史数据。这些数据可被用于探索攻击方的行为模式，使得安保方能够获取更多关于攻击方的收益情况、理性程度与行为能力的信息，从而设计出更具针对性的安保策略。

总而言之，安全博弈的相关研究拓展了对博弈理论的理解，产生了众多意义广泛的应用项目。由于该研究领域尚年轻，仍存在许多待发掘、待解决的问题，具有重大的研究和应用价值。本人将在接下来的研究工作中进一步探索安全博弈中的问题，探索安全博弈与社会选择、机器学习等其他学科的交叉应用，致力于取得更有价值的理论与应用成果。

参考文献

[1] Bo An, Matthew Brown, Yevgeniy Vorobeychik, and Milind Tambe. Security games with surveillance cost and optimal timing of attack execution. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent systems (AAMAS)*, pages 223–230, 2013.

[2] Bo An, Manish Jain, Milind Tambe, and Christopher Kiekintveld. Mixed-initiative optimization in security games: A preliminary report. In *AAAI Spring Symposium: Help me help you: Bridging the gaps in human-agent collaboration*, pages 8–11, 2011.

[3] Bo An, Fernando Ordóñez, Milind Tambe, Eric Shieh, Rong Yang, Craig Baldwin, Joseph DiRenzo III, Kathryn Moretti, Ben Maule, and Garrett Meyer. A deployed quantal response-based patrol planning system for the us coast guard. *Interfaces*, 43:400–420, 2013.

[4] Bo An, Milind Tambe, Fernando Ordonez, Eric Anyung Shieh, and Christopher Kiekintveld. Refinement of strong stackelberg equilibria in security games. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, 2011.

[5] Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 61–78, 2015.

[6] Jonathan Bannet, David W. Price, Algis Rudys, Justin Singer, and Dan S. Wallach. Hack-a-Vote: Security issues with electronic voting systems. *IEEE Security and Privacy*, 2(1):32–37, 2004.

[7] John J. Bartholdi, Craig A. Tovey, and Michael A. Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8):27–40, 1992.

[8] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 57–64, 2009.

[9] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence*, 184:78–123, 2012.

[10] David R Bellwood, Terry P Hughes, C Folke, and M Nyström. Confronting the coral reef crisis. *Nature*, 429(6994):827–833, 2004.

[11] Colin J Bennett and Kevin Haggerty. *Security games: surveillance and control at mega-events*. Routledge, 2014.

[12] Nadja Betzler and Johannes Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theoretical Computer Science*, 410(52):5425–5442, 2009.

[13] Satarupa Bhattacharjya. Low turnout and invalid votes mark first post war general polls. http://www.sundaytimes.lk/100411/News/nws_16.html, 2010.

[14] Sayan Bhattacharya, Vincent Conitzer, and Kamesh Munagala. Approximation algorithm for security games with costly resources. In *International Workshop on Internet and Network Economics*, pages 13–24, 2011.

[15] Branislav Bošanskỳ, Albert Xin Jiang, Milind Tambe, and Christopher Kiekintveld. Combining compact representation and incremental generation in large games with sequential strategies. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pages 812–818, 2015.

[16] Josie Carwardine, Carissa J Klein, Kerrie A Wilson, Robert L Pressey, and Hugh P Possingham. Hitting the target and missing the point: target-based conservation planning in context. *Conservation Letters*, 2(1):4–11, 2009.

[17] Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Combinatorial voter control in elections. In *International Symposium on Mathematical Foundations of Computer Science*, pages 153–164, 2014.

[18] Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Elections with few voters: Candidate control can be easy. In *AAAI Conference on Artificial Intelligence*, pages 2045–2051, 2015.

[19] Vincent Conitzer. Computing game-theoretic solutions and applications to security. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2106–2112, 2012.

[20] Vincent Conitzer and Dmytro Korzhyk. Commitment to correlated strategies. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, pages 632–637, 2011.

[21] Vincent Conitzer, Jérôme Lang, Lirong Xia, et al. How hard is it to control sequential elections via the agenda? In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 103–108, 2009.

[22] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90, 2006.

[23] Vincent Conitzer and Tuomas Sandholm. New complexity results about nash equilibria. *Games and Economic Behavior*, 63(2):621–641, 2008.

[24] Vincent Conitzer, Toby Walsh, and Lirong Xia. Dominating manipulations in voting with partial information. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, pages 638–643, 2011.

[25] Vincent Conitzer and Makoto Yokoo. Using mechanism design to prevent false-name manipulations. *AI magazine*, 31(4):65–78, 2010.

[26] Edith Elkind and Helger Lipmaa. Hybrid voting protocols and hardness of manipulation. In *International Symposium on Algorithms and Computation*, pages 206–215, 2005.

[27] Gábor Erdélyi, Edith Hemaspaandra, and Lane A. Hemaspaandra. More natural models of electoral control by partition. In *Algorithmic Decision Theory*, pages 396–413, 2015.

[28] Gábor Erdélyi, Markus Nowak, and Jörg Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Mathematical Logic Quarterly*, 55(4):425–443, 2009.

[29] Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40(1):305–351, 2011.

[30] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Weighted electoral control. In *International Conference on Autonomous Agents and Multi-Agent Systems*, pages 367–374, 2013.

[31] Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Llull and copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[32] Fei Fang, Albert Xin Jiang, and Milind Tambe. Optimal patrol strategy for protecting moving targets with multiple mobile resources. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent systems (AAMAS)*, pages 957–964, 2013.

[33] Fei Fang, Albert Xin Jiang, and Milind Tambe. Protecting moving targets with multiple mobile resources. *Journal of Artificial Intelligence Research*, 48:583–634, 2013.

[34] Fei Fang, Thanh H Nguyen, Rob Pickles, Wai Y Lam, Gopalasamy R Clements, Bo An, Amandeep Singh, Milind Tambe, and Andrew Lemieux. Deploying Paws: Field optimization of the protection assistant for wildlife security. In *Proceedings of the Twenty-Eighth Innovative Applications of Artificial Intelligence Conference*, 2016.

[35] Fei Fang, Peter Stone, and Milind Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2589–2595, 2015.

[36] Jiarui Gan, Bo An, and Chunyan Miao. Optimizing efficiency of taxi systems: Scaling-up and handling arbitrary constraints. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 523–531, 2015.

[37] Jiarui Gan, Bo An, and Yevgeniy Vorobeychik. Security games with protection externalities. In *AAAI*, pages 914–920, 2015.

[38] Jiarui Gan, Bo An, and Yevgeniy Vorobeychik. Security games with protection externalities. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pages 914–920, 2015.

[39] Jiarui Gan, Bo An, Haizhong Wang, Xiaoming Sun, and Zhongzhi Shi. Optimal pricing for improving efficiency of taxi systems. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2811–2818, 2013.

[40] Jens Grossklags, Nicolas Christin, and John Chuang. Secure or insure?: a game-theoretic analysis of information security games. In *Proceedings of the 17th international conference on World Wide Web*, pages 209–218, 2008.

[41] Qingyu Guo, Bo An, Yevgeniy Vorobeychik, Long Tran-Thanh, Jiarui Gan, and Chunyan Miao. Coalitional security games. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 159–167, 2016.

[42] Nika Haghtalab, Fei Fang, Thanh H Nguyen, Arunesh Sinha, Ariel D Procaccia, and Milind Tambe. Three strategies to success: Learning adversary models in security games. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 308–314, 2016.

[43] John C Harsanyi and Reinhard Selten. A generalized nash solution for two-person bargaining games with incomplete information. *Management Science*, 18(5-part-2):80–106, 1972.

[44] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5):255–285, 2007.

[45] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, 55(4):397–424, 2009.

[46] Manish Jain, Vincent Conitzer, and Milind Tambe. Security scheduling for real-world networks. In *Proceedings of the 12th International Conference on Autonomous agents and Multiagent systems (AAMAS)*, pages 215–222, 2013.

[47] Manish Jain, Erim Kardes, Christopher Kiekintveld, Fernando Ordónez, and Milind Tambe. Security games with arbitrary schedules: A branch and price approach. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, 2010.

[48] Manish Jain, Dmytro Korzhyk, Ondřej Vaněk, Vincent Conitzer, Michal Pěchouček, and Milind Tambe. A double oracle algorithm for zero-sum security games on graphs. In *The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 327–334, 2011.

[49] Manish Jain, Kevin Leyton-Brown, and Milind Tambe. The deployment-to-saturation ratio in security games. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

[50] Albert Xin Jiang, Thanh H Nguyen, Milind Tambe, and Ariel D Procaccia. Monotonic maximin: A robust Stackelberg solution against boundedly rational followers. In *International Conference on Decision and Game Theory for Security*, pages 119–139, 2013.

[51] Albert Xin Jiang, Ariel D Procaccia, Yundi Qian, Nisarg Shah, and Milind Tambe. Defender (mis) coordination in security games. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 220 – 226, 2013.

[52] Albert Xin Jiang, Zhengyu Yin, Chao Zhang, Milind Tambe, and Sarit Kraus. Game-theoretic randomization for security patrolling with dynamic execution uncertainty. In *Proceedings of 12th International Conference on Autonomous Agents and Multiagent systems (AAMAS)*, pages 207–214, 2013.

[53] Michael Kearns and Luis E Ortiz. Algorithms for interdependent security games. In *Advances in Neural Information Processing Systems*, 2003.

[54] Christopher Kiekintveld, Towhidul Islam, and Vladik Kreinovich. Security games with interval uncertainty. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent systems*, pages 231–238, 2013.

[55] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 689–696, 2009.

[56] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, pages 805 – 810, 2010.

[57] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Security games with multiple attacker resources. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 273 – 279, 2011.

[58] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Security games with multiple attacker resources. In *International Joint Conference on Artificial Intelligence*, pages 273–279, 2011.

[59] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Solving stackelberg games with uncertain observability. In *The 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1013–1020, 2011.

[60] Dmytro Korzhyk, Zhengyu Yin, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research (JAIR)*, 41:297–327, 2011.

[61] Andreas Krause, Alex Roper, and Daniel Golovin. Randomized sensing in adversarial environments. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2133 – 2139, 2011.

[62] Jun-young Kwak, Debarun Kar, William B Haskell, Pradeep Varakantham, and Milind Tambe. Building thinc: User incentivization and meeting rescheduling for energy savings. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent systems*, pages 925–932, 2014.

[63] Jean-François Laslier and Karine Van der Straeten. A live experiment on approval voting. *Experimental Economics*, 11:97–105, 2008.

[64] Aron Laszka, Jian Lou, and Yevgeniy Vorobeychik. Multi-defender strategic filtering against spear-phishing attacks. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[65] Aron Laszka, Bradley Potteiger, Yevgeniy Vorobeychik, Saurabh Amin, and Xenofon Koutsoukos. Vulnerability of transportation networks to traffic-signal tampering. In *ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, pages 1–10, 2016.

[66] Aron Laszka, Yevgeniy Vorobeychik, and Xenofon D Koutsoukos. Optimal personalized filtering against spear-phishing attacks. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pages 958–964, 2015.

[67] Joshua Letchford and Vincent Conitzer. Computing optimal strategies to commit to in extensive-form games. In *Proceedings of the 11th ACM conference on Electronic commerce (EC)*, pages 83–92, 2010.

[68] Joshua Letchford and Vincent Conitzer. Solving security games on graphs via marginal probabilities. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*, pages 591–597, 2013.

[69] Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. Learning and approximating the optimal strategy to commit to. In *International Symposium on Algorithmic Game Theory*, pages 250–262, 2009.

[70] Joshua Letchford, Liam MacDermed, Vincent Conitzer, Ronald Parr, and Charles L Isbell. Computing optimal strategies to commit to in stochastic games. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1380 – 1386. Citeseer, 2012.

[71] Joshua Letchford and Yevgeniy Vorobeychik. Optimal interdiction of attack plans. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 199–206, 2013.

[72] Yuqian Li, Vincent Conitzer, and Dmytro Korzhyk. Catcher-evader games. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 329 – 337, 2016.

[73] Hong Liu, Haodi Feng, Daming Zhu, and Junfeng Luan. Parameterized computational complexity of control problems in voting systems. *Theoretical Computer Science*, 410(27):2746–2753, 2009.

[74] Hong Liu and Daming Zhu. Parameterized complexity of control problems in maximin election. *Information Processing Letters*, 110(10):383–388, 2010.

[75] Jian Lou and Yevgeniy Vorobeychik. Equilibrium analysis of multi-defender security games. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 596–602, 2015.

[76] Jian Lou and Yevgeniy Vorobeychik. Decentralization and security in dynamic traffic light control. In *Proceedings of the Symposium and Bootcamp on the Science of Security*, pages 90–92, 2016.

[77] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

[78] H. Brendan Mcmahan, Geoffrey J Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *International Conference on Machine Learning*, pages 536–543, 2003.

[79] Curtis Menton. Normalized range voting broadly resists control. *Theory of Computing Systems*, 53(4):507–531, 2013.

[80] Curtis Menton and Preetjot Singh. Control complexity of schulze voting. In *International Joint Conference on Artificial Intelligence*, pages 286–292, 2013.

[81] Tyler Moore, Allan Friedman, and Ariel D Procaccia. Would a cyber warrior protect us: Exploring trade-offs between attack and defense of information systems. In *Proceedings of the 2010 workshop on New security paradigms*, pages 85–94, 2010.

[82] Kien C Nguyen, Tansu Alpcan, and Tamer Basar. Security games with incomplete information. In *2009 IEEE International Conference on Communications*, pages 1–6, 2009.

[83] Thanh Hong Nguyen, Amulya Yadav, Bo An, Milind Tambe, and Craig Boutilier. Regret-based optimization and preference elicitation for Stackelberg security games with uncertainty. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, pages 756–762, 2014.

[84] Thanh Hong Nguyen, Rong Yang, Amos Azaria, Sarit Kraus, and Milind Tambe. Analyzing the effectiveness of adversary modeling in security games. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*, 2013.

[85] Swetasudha Panda and Yevgeniy Vorobeychik. Stackelberg games for vaccine design. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1391–1399, 2015.

[86] John M Pandolfi, Roger H Bradbury, Enric Sala, Terence P Hughes, Karen A Bjorndal, Richard G Cooke, Deborah McArdle, Loren McClenachan, Marah JH Newman, Gustavo Paredes, et al. Global trajectories of the long-term decline of coral reef ecosystems. *Science*, 301(5635):955–958, 2003.

[87] David Parkes and Lirong Xia. A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs. In *AAAI Conference on Artificial Intelligence*, pages 1429–1435, 2012.

[88] Praveen Paruchuri, Jonathan P Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, and Sarit Kraus. Playing games for security: an efficient exact algorithm for solving bayesian stackelberg games. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent systems (AAMAS)*, pages 895–902, 2008.

[89] Praveen Paruchuri, Jonathan P Pearce, Milind Tambe, Fernando Ordonez, and Sarit Kraus. An efficient heuristic approach for security against multiple adversaries. In *Proceedings of the 6th international joint conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 181–188, 2007.

[90] James Pita, Manish Jain, Janusz Marecki, Fernando Ordóñez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Deployed Armor protection: The application of a game theoretic model for security at the Los Angeles International Airport. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*, pages 125–132, 2008.

[91] James Pita, Manish Jain, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15):1142–1171, 2010.

[92] James Pita, Milind Tambe, Chris Kiekintveld, Shane Cullen, and Erin Steigerwald. GUARDS: Game theoretic security allocation on a national scale. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 37–44, 2011.

[93] James Pita, Milind Tambe, Christopher Kiekintveld, Shane Cullen, and Erin Steigerwald. GUARDS —innovative application of game theory for national airport security. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2710–2715, 2011.

[94] Rob Powers and Yoav Shoham. Learning against opponents with bounded memory. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 817–822, 2005.

[95] Yundi Qian, William B Haskell, Albert Xin Jiang, and Milind Tambe. Online planning for optimal protector strategies in resource conservation games. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent systems (AAMAS)*, pages 733–740, 2014.

[96] RT. Election day bombings sweep Pakistan: Over 30 killed, more than 200 injured. https://www.rt.com/news/pakistan-election-day-bombing-136/, 2013.

[97] Eric Shieh, Manish Jain, Albert Xin Jiang, and Milind Tambe. Efficiently solving joint activity based security games. In *Proceedings of the 23rd International Joint conference on Artificial Intelligence (IJCAI)*, pages 346–352, 2013.

[98] Eric Anyung Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. PROTECT: An application of computational game theory for the security of the ports of the United States. In *AAAI*, 2012.

[99] Leo K Simon and Maxwell B Stinchcombe. Equilibrium refinement for infinite normal-form games. *Econometrica: Journal of the Econometric Society*, pages 1421–1443, 1995.

[100] Troels Bjerre Sørensen, Melissa Dalis, Joshua Letchford, Dmytro Korzhyk, and Vincent Conitzer. Beat the cheater: Computing game-theoretic strategies for when to kick a gambler out of a casino. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, pages 798–804, 2014.

[101] Noah D Stein, Asuman Ozdaglar, and Pablo A Parrilo. Separable and low-rank continuous games. *International Journal of Game Theory*, 37:475–504, 2008.

[102] Emma J Stokes. Improving effectiveness of protection efforts in tiger source sites: Developing a framework for law enforcement monitoring using MIST. *Integrative Zoology*, 5:363–377, 2010.

[103] Peter Stone, Gal A Kaminka, Sarit Kraus, Jeffrey S Rosenschein, et al. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, page 1504–1509, 2010.

[104] Milind Tambe. *Security and game theory: Algorithms, deployed systems, lessons learned*. Cambridge University Press, 2011.

[105] Jason Tsai, Yundi Qian, Yevgeniy Vorobeychik, Christopher Kiekintveld, and Milind Tambe. Bayesian security games for controlling contagion. In *Social Computing (SocialCom), 2013 International Conference on*, pages 33–38, 2013.

[106] Yevgeniy Vorobeychik, Bo An, Milind Tambe, and Satinder P Singh. Computing solutions in infinite-horizon discounted adversarial patrolling games. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 2014.

[107] Yevgeniy Vorobeychik, Steven Kimbrough, and Howard Kunreuther. A framework for computational strategic analysis: Applications to iterated interdependent security games. *Computational Economics*, 45(3):469–500, 2015.

[108] Yevgeniy Vorobeychik, Michael Z Lee, Adam Anderson, Mitch Adair, William Atkins, Alan Berryhill, Dominic Chen, Ben Cook, Jeremy Erickson, Steve Hurd, et al. Fireaxe: the dhs secure design competition pilot. In *Proceedings of the 8th Annual Cyber Security and Information Intelligence Research Workshop*, page 21, 2013.

[109] Yevgeniy Vorobeychik and Joshua Letchford. Securing interdependent assets. *Autonomous Agents and Multiagent Systems*, 29(2):305–333, 2015.

[110] Yevgeniy Vorobeychik and Joshua Letchford. Securing interdependent assets. *Journal of the Autonomous Agents and Multiagent Systems*, 29(2):305–333, 2015.

[111] Yevgeniy Vorobeychik and Satinder P Singh. Computing stackelberg equilibria in discounted stochastic games. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

[112] Bo Waggoner, Lirong Xia, and Vincent Conitzer. Evaluating resistance to false-name manipulations in elections. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1485–1491, 2012.

[113] Zhiyu Wan, Yevgeniy Vorobeychik, Weiyi Xia, Ellen Wright Clayton, Murat Kantarcioglu, Ranjit Ganta, Raymond Heatherly, and Bradley A Malin. A game theoretic framework for analyzing re-identification risk. *PloS one*, 10(3):e0120592, 2015.

[114] Zhen Wang, Yue Yin, and Bo An. Computing optimal monitoring strategy for detecting terrorist plots. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[115] Matthew E Watts, Ian R Ball, Romola S Stewart, Carissa J Klein, Kerrie Wilson, Charles Steinback, Reinaldo Lourival, Lindsay Kircher, and Hugh P Possingham. Marxan with zones: software for optimal conservation based land-and sea-use zoning. *Environmental Modelling & Software*, 24(12):1513–1521, 2009.

[116] Scott Wolchok, Eric Wustrow, Dawn Isabel, and J. Alex Halderman. Attacking the Washington, DC internet voting system. In *International Conference on Financial Cryptography and Data Security*, pages 114–128, 2012.

[117] Lirong Xia and Vincent Conitzer. A sufficient condition for voting rules to be frequently manipulable. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pages 99–108, 2008.

[118] Lirong Xia and Vincent Conitzer. Stackelberg voting games: Computational aspects and paradoxes. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, pages 921–926, 2010.

[119] Lirong Xia, Vincent Conitzer, and Ariel D Procaccia. A scheduling approach to coalitional manipulation. In *Proceedings of the 11th ACM conference on Electronic commerce (EC)*, pages 275–284, 2010.

[120] Yanhai Xiong, Jiarui Gan, Bo An, Chunyan Miao, and Ana LC Bazzan. Optimal electric vehicle charging station placement. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2662–2668, 2015.

[121] Haifeng Xu, Fei Fang, Albert Xin Jiang, Vincent Conitzer, Shaddin Dughmi, and Milind Tambe. Solving zero-sum security games in discretized spatio-temporal domains. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1500–1506, 2014.

[122] Haifeng Xu, Rupert Freeman, Vincent Conitzer, Shaddin Dughmi, and Milind Tambe. Signaling in bayesian stackelberg games. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 150–158, 2016.

[123] Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. SATzilla: Portfolio-based algorithm selection for sat. *Journal of Artificial Intelligence Research*, 32(1):565–606, 2008.

[124] Rong Yang, Benjamin Ford, Milind Tambe, and Andrew Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. In *Proceedings of the 13th International Conference on Autonomous agents and Multiagent systems*, pages 453–460, 2014.

[125] Rong Yang, Christopher Kiekintveld, Fernando OrdóñEz, Milind Tambe, and Richard John. Improving resource allocation strategies against human adversaries in security games: An extended study. *Artificial Intelligence*, 195:440–469, 2013.

[126] Yue Yin and Bo An. Efficient resource allocation for protecting coral reef ecosystems. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 531–537, 2016.

[127] Yue Yin, Yevgeniy Vorobeychik, Bo An, and Noam Hazon. Optimally protecting elections. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 538–545, 2016.

[128] Yue Yin, Haifeng Xu, Jiarui Gan, Bo An, and Albert Xin Jiang. Computing optimal mixed strategies for security games with dynamic payoffs. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 681–687, 2015.

[129] Zhengyu Yin, Albert Xin Jiang, Matthew Paul Johnson, Christopher Kiekintveld, Kevin Leyton-Brown, Tuomas Sandholm, Milind Tambe, and John P Sullivan. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems. In *The 24th Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pages 2348–2354, 2012.

[130] Zhengyu Yin, Albert Xin Jiang, Milind Tambe, Christopher Kiekintveld, Kevin Leyton-Brown, Tuomas Sandholm, and John P Sullivan. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine*, 33(4):59, 2012.

[131] Zhengyu Yin and Milind Tambe. A unified method for handling discrete and continuous uncertainty in bayesian stackelberg games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 855–862, 2012.

[132] Chao Zhang, Victor Bucarey, Ayan Mukhopadhyay, Arunesh Sinha, Yundi Qian, Yevgeniy Vorobeychik, and Milind Tambe. Using abstractions to solve opportunistic crime security games at scale. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 196–204, 2016.

[133] Michael Zuckerman, Piotr Faliszewski, Vincent Conitzer, and Jeffrey S Rosenschein. An NTU cooperative game theoretic view of manipulating elections. In *International Workshop on Internet and Network Economics*, pages 363–374, 2011.

致　谢

博士生涯渐进尾声，在此我由衷地感谢所有在我读博期间关心和帮助过我的老师、同学、家人和朋友。犹记得当时安老师的一场讲座，激起了我对于多智能体研究的兴趣。而“安全博弈”这个课题，更让我仿佛看到了“羽扇纶巾谈笑间，樯橹灰飞烟灭”的场景。五年半的时光让我逐渐认识到，科研工作的日常，不是潇洒的运筹帷幄，而是坚韧的砥砺前行。是各位师长同侪的陪伴与支持，让我在这条道路上勇敢前进，一边成长，一边收获。

我要感谢我的导师安波老师。最初跟随安老师做研究时，我对如何建模，如何行文都无甚概念，又往往幼稚浅薄而不自知，是安老师一步一步地将我引入了正确的道路。如果不是安老师一字一句地帮我修改论文，我不可能有机会参与 AAAI，IJCAI 这样的 AI 盛会；如果不是安老师一页一页地帮我把关 PPT，我更不可能在会议上，众多专家学者面前，胸有成竹的介绍自己的工作。安老师还带我和各国学者合作，帮我争取到了出国访学的机会，让我拓展了眼界，丰富了经历。这几年来我的所有成长，背后都有安老师的耕耘。师恩不敢忘，未来我将再接再厉，以不枉费安老师的付出。

我要感谢我的导师山世光老师。是山老师的支持与帮助，让我能够顺利度过博士生涯中的种种关卡。是山老师的鼓励与信任，让我有机会申请种种奖学金，让我在高手林立的计算所中逐渐自信起来。山老师待人接物的儒雅态度，工作科研的专业精神，潜移默化中影响着我，使我受益匪浅。在我以后的工作生活中，山老师将一直是我立身立业的楷模。

我要感谢我在 Vanderbilt University 访学期间的导师 Yevgeniy (Eugene) Vorobeychik 老师。虽然相处时间不过半年有余，但 Eugene 缜密的思维习惯，井井有条的工作方式，平易近人的态度，都让我印象深刻，见贤思齐。Eugene 的帮助与支持，让我在美国顺利的继续科研，未受到环境变迁的影响。和 Eugene 相处的经历将永远提醒我，不忘初心，不负流年。

我要感谢尊敬的史忠植老师。我大学期间就曾在图书馆借阅史老师的著作，史老师在我心中一直是高山仰止的前辈。研一时史老师教授的高级人工智能课程，于我是打开了一扇通向未知领域的大门。史老师认真的治学态度，敬业的工作精神，时时激励着我不畏艰难，奋勇前进。

我要感谢在我的论文写作中给予过我无私帮助的 Manish Jain、Noam Hazon、Jun Zhuang、Haifeng Xu 和 Fei Fang。犹记得 Manish 在我的论文初稿中手写的批注“You should have a storyline”，让我第一次意识到学术论文也要讲一个好故事。我要感谢在我博士论文的写作过程中给予过我宝贵意见的罗四维老师、沈一栋老师、杜军平老师、李

华老师、卜东波老师、孙晓明老师、唐平中老师、马少平老师、何清老师、毛文吉老师和陈熙霖老师，是你们的意见与鼓励使这篇博士论文得以成型。我还要感谢在我硕士、博士集中学习期间不辞辛劳开设课程的各位老师，是你们的辛勤工作让我得以打下日后科研的基础。

我要感谢我的师弟甘家瑞和荣江，你们的陪伴与帮助是我沮丧困惑时温暖的力量。我要感谢杨曦和马刚在我遇到种种问题时给予我的帮助。我要感谢实验室的所有同学和 704 班的所有同学，是你们让这段旅程多姿多彩。

我要感谢实验室的胡兰萍老师和田卫平老师对我学习和生活中的帮助。感谢研究生部的李琳老师、周世佳老师、张平老师、李丹老师、宋守礼老师和冯钢老师等各位老师在我的学习过程中给予的支持和帮助。

我要感谢我的父母，是你们让我相信，我可以成为任何人。我要感谢我的挚友冉倚，是你的陪伴，让我有幸总能和人分享这一路的风景。

作者简介

姓名：殷越　　性别：女　　出生日期：1989.4.6　　籍贯：河南

2011.8 – 现在　　中国科学院计算技术所攻读博士研究生
2007.8 – 2011.6　　北京信息科技大学本科生

【攻读博士学位期间发表的论文】

[1] **Y. Yin**, Y. Vorobeychik, B. An, N. Hazon. Optimally protecting elections. In *Proc. of the 25th International Joint Conference on Artificial Intelligence* (IJCAI), pages 538–545, 2016

[2] **Y. Yin**, B. An. Efficient resource allocation for protecting coral reef ecosystems. In *Proc. of the 25th International Joint Conference on Artificial Intelligence* (IJCAI), pages 531–537, 2016

[3] Z. Wang, **Y. Yin**, B. An, Computing optimal monitoring strategy for detecting terrorist plots, In *P*roc. of the The 30th AAAI Conference on Artificial Intelligence (AAAI), pages 637–643, 2016

[4] **Y. Yin**, H. Xu, J. Gan, B. An, A. Jiang, Computing optimal mixed strategies for security games with dynamic payoffs, In *Proc. of the 24th International Joint Conference on Artificial Intelligence* (IJCAI), pages 681–687, 2015

[5] **Y. Yin**, B. An, and M. Jain, Game-theoretic resource allocation for protecting large public events, In *Proc. of the 28th AAAI Conference on Artificial Intelligence* (AAAI), pages 826–833, 2014

[6] **殷越**，安波，史忠植. 基于博弈论的重大公共活动安保策略设计算法. 中国科学 F 辑, 信息科学 (已录用)

【攻读博士学位期间的获奖情况】

[1] 2016 年获得国家奖学金

[2] 2015 年被评为中国科学院计算技术研究所“三好学生标兵”

[3] 2015 年被评为中国科学院计算技术研究所“三好学生”

[4] 2014 年获得中国科学院计算技术研究所所长奖学金优秀奖