doi:10.11959/j.issn.1000-436x.2015027

基于随机松弛优选策略的网络脆弱性弥补算法

赵光胜[1]，程庆丰[1]，孙永林[2]
(1. 解放军外国语学院 语言工程系，河南 洛阳 471003; 2. 国防科技大学 计算机学院，湖南 长沙 410073)

摘 要：为了在大规模网络中构建代价最小的脆弱性弥补方案，提出了一种基于随机松弛优选策略的网络脆弱性弥补算法 (MCNHA-SLOS)，并分析了算法的有效性。MCNHA-SLOS 是一种近似最优算法，通过在全部弥补方案空间的一系列随机松弛子空间中进行迭代计算，使近似最优弥补方案必定落入低代价弥补方案空间中。实例分析和仿真结果表明，MCNHA-SLOS 具有高效、精度可控、渐近最优等特点，能够应用于大规模网络环境。
关键词：网络脆弱性；攻击图；网络脆弱性弥补；随机松弛优选
中图分类号：TN915.08 **文献标识码**：A

Minimum-cost network hardening algorithm based on stochastic loose optimize strategy

ZHAO Guang-sheng[1], CHENG Qing-feng[1], SUN Yong-lin[2]
(1. Department of Language Engineering, PLA University of Foreign Languages, Luoyang 471003, China;
2. College of Computer Science, National University of Defense Technology, Changsha 410073, China)

Abstract: To construst a minimum-cost network hardening (MCNH) scheme in large-scale network, a stochastic loose optimize strategy based algorithm (MCNHA-SLOS) was proposed, and its effectiveness was analyzed. MCNHA-SLOS was a near-optimal approximation algorithm, which could achieve iterative computations in the array of sparse spaces of the whole plan space, so that the near-optimal scheme must exist in the low cost plan space. Instantiation analysis and experimental results show that the MCNHA-SLOS algorithm to be efficient, precision controllable and asymptotically optimal, and thus very applicable for large-scale network.
Key words: network vulnerability; attack graph; minimum-cost network hardening; stochastic loose optimize strategy

1 引言

计算机网络技术的飞速发展深刻影响着人类的生产生活方式。然而在享受互联互通和信息共享带来的便捷与效益的同时，网络技术发展过程中对安全性的忽视，导致了网络环境中存在各式各样的安全隐患，严重威胁着网络运营及合法用户的信息安全。从根本上讲，敌手通常利用网络中存在的脆弱性逐步渗透并控制若干网络节点，达到恶意目的。因此，寻找并弥补威胁网络关键资源的关键脆弱性，是提高网络安全性的一种有效方法。网络脆弱性弥补分析(MCNHA, minimum-cost network hardening analysis)以寻找并弥补威胁关键目标的代价最小的网络脆弱性为目标，是一个 NPC 问题，目前尚无足够有效的算法应对大规模复杂网络[1~10]。

MCNHA 涉及的问题包括：如何确定弥补方案空间，如何判定弥补方案有效，如何确定弥补方案代价，最重要也是最困难的是如何求解代价最小并且有效的弥补方案。确定弥补方案空间的关键在于确定可能会对关键目标构成威胁的全部脆弱性，一

收稿日期：2013-10-11；**修回日期**：2014-07-22
基金项目：国家高技术研究发展计划(“863”计划)基金资助项目(2009AA01Z432)；国家自然科学基金资助项目(60873215, 61003303)；信息保障技术重点实验室开放基金资助项目(KJ-13-109)
Foundation Items: The National High Technology Research and Development Program of China (863 Program) (2009AA01Z432); The National Natural Science Foundation of China(60873215,61003303); Foundation of Science and Technology on Information Assurance Laboratory (KJ-13-109)

种有效的方法是使用攻击图[1~10]确定可能威胁到关键目标的全部脆弱性，它能充分反映给定网络环境中的脆弱性之间的利用依赖，攻击图也可以确定弥补方案的有效性判定法则。弥补方案的代价取决于弥补方案的实施难度，以及相关脆弱性弥补后对网络环境的影响的评估。求解代价最小并且有效的弥补方案在于如何快速地从 $2^{O(n)}$规模的弥补方案空间中选出代价最小的有效的方案，这是一个 NPC 问题。因此，设计高效的策略，在有限的计算资源下快速寻找代价尽可能小的有效弥补方案，是 MCNHA 应对大规模复杂网络的唯一出路。

基于以上分析，与 MCNHA 相关的研究工作主要涉及攻击图构建与分析和代价最小化弥补 2 方面，下面分别介绍这 2 方面的研究现状。

攻击图构建与分析技术是 MCNHA 的基础[1~10]，攻击图作为一种网络防御分析的工具，其基本思想已提出 10 多年。具体地，Swiler[11]于 1998 年最早提出了攻击图模型，使用攻击模型来描述一致的攻击行为，并通过手工方式从目标状态反向的生成攻击图。文献[12~14]引入了自动分析技术，提高了攻击图的构建效率。Lippmann [15,16]总结了过去的攻击图分析技术，并提出了基于攻击图的网络安全评估与弥补方法。Ou[17,18]提出了单目标攻击图构建算法，并发现了含圈攻击路径现象。陈峰[7,8,19,20]提出了一种生成多目标攻击图的算法，并从威胁概率的角度分析攻击路径，得出了长攻击路径在实际网络攻击中通常不会发生的结论。尚大鹏[21]以评估网络整体安全性为目标，提出了通过限定攻击步数和状态节点可达性的策略，降低攻击图的复杂度，进一步提高攻击图的生成效率。此外，研究者们还提出了多种方法来增强攻击图的可理解性和可视化效果[22~24]。攻击图的构建与分析经历了从手工到自动分析，从简单到复杂的过程，现有的技术虽然在构建效率、可理解性和可视化等方面仍存在不足，但基本满足 MCNHA 的需求。

在代价最小化弥补方面，Jha[1]基于状态攻击图寻找保障目标网络关键信息资产安全的最小安全措施集。Noel S 等[2]提出了基于属性攻击图的最小成本安全弥补措施集，但是该方法不能应用大型的具有含圈攻击路径的攻击图。Wang [3]解决了攻击图的含圈问题，但不能应用于大规模攻击图。Homer[4]提出了一种基于逻辑攻击图的自动化网络配置管理方法，但在攻击图规模较大时应用情况仍然不佳。司加全[5]提出了一种可操作性较强的基于安全损失关键度最大优先原则的网络安全增强策略，但没有分析在大规模复杂网络环境中的应用情况。陈峰[6~8]提出了基于二叉决策图的最优弥补集精确计算方法和基于贪婪策略的近似计算方法，前者较 Wang 的方法有较大性能提升，但仍只能应用于小规模网络环境；后者引入了 *n*-有效路径的概念，是一种可应用于大规模网络的计算方法，但适用范围仍有较大限制。Albanese[9]提出了一种时间效能近似最小的方案，可以获得较高的近似比率，但没有考虑求解所用的计算资源和求解精度问题。

针对 MCNHA，本文提出了一种基于随机松弛优选策略的网络脆弱性弥补分析算法(MCNHA-SLOS, minimum- cost network hardening algorithm based on stochastic loose optimize strategy)，依据概率论的思想，在弥补方案空间的一系列随机松弛子空间中迭代求解代价最小的弥补方案，并依据其有效性判定结果更新近似最优弥补方案，使其逐步逼近最优弥补方案。算法具有求解速度快、耗费/收益比高、精度可控等优点，适合在大规模网络环境中应用。

2 基本概念和前提假设

为了突出网络脆弱性弥补分析的重点和难点，且方便后文叙述，本节明确若干基本概念、前提假设和符号约定。

弥补方案空间。由所有可能的弥补方案组成的集合，记为 *Plan-Space*。假定目标网络中可能威胁到关键目标的脆弱性总数为 n，分别用$\{1, 2, \cdots, n\}$标记这 n 个脆弱性，则可用 n 位二进制数来表示 *Plan-Space*，这样每个弥补方案 *Plan* 对应一个 n 位二进制数，为“1”的位置表示对应的脆弱性需要弥补，即$(0)_2 \leqslant (Plan)_2 \leqslant (2^n-1)_2$，其中，$(0)_2$是平凡无效弥补方案，对任何弥补目标都无效，而$(2^n-1)_2$则是平凡有效弥补方案，对任何弥补目标都有效，同时弥补代价也最高。

最优弥补方案。弥补方案空间中代价最小的有效弥补方案，记为 *Opt-Plan*。

近似最优弥补方案。近似最优方法求解得到的有效弥补方案，记为 *Approx-Opt-Plan*。

弥补方案代价计算函数。计算弥补方案代价的函数，记为 *Cost*()。

弥补方案有效判定函数。判定弥补措施是否有效的函数，记为 *Valid*()。

弥补方案松弛子空间。从 *Plan-Space* 中随机选取若干项构成的子空间，记为 *Sparse-Space*。

优势弥补方案空间和优势比率。依据 *Cost*()和优势比率 $P_{superior}$ 可将弥补方案空间 *Plan-Space* 划分成 2 部分，低代价部分称为优势空间，记为 *Superior-Space*。*Superior-Spac*e 中的弥补方案至少比 *Plan-Space* 中|*Plan-Space*|(1−$P_{superior}$)个弥补方案的代价低，$P_{superior}$ 体现了用户对近似最优弥补方案 *Approx-Opt-Plan* 的代价最优性期望，$P_{superior}$ 越小，用户对 *Approx-Opt-Plan* 的代价最优性期望越高。

有效弥补方案空间和有效比率。依据 *Valid*()可将弥补方案空间 *Plan-Space* 划分成有效弥补方案空间和无效弥补方案空间，将前者记为 *Valid-Space*，|*Valid-Space*|与|*Plan-Space*|之比称为有效比率，记为 P_{Valid}。反映了关键目标受威胁程度，P_{Valid} 越高关键目标越容易受到攻击。

目标空间。优势空间和有效空间相交的部分，记为 *Goal-Space*。*Goal-Space* 中的弥补方案既满足优势比率的要求，又满足有效性的要求，体现了用户对 *Approx-Opt-Plan* 期望的可满足性，*Goal-Space* 的规模愈小，用户的期望愈难满足，当 *Goal-Space* 为空时，表明用户的期望无法满足。

目标达成概率。*Approx-Opt-Plan* 落入 *Goal-Space* 中的概率。

前提假设 1。弥补方案空间 *Plan-Space* 已知。对于给定的网络环境和关键目标，通过攻击图自动生成算法可以生成各种形式的攻击图，依据攻击图可以确定关键目标的弥补方案空间。如果将攻击图中存在攻击路径到达关键目标的脆弱性组成的集合称为相关脆弱性集合，那么相关脆弱性集的幂集即为弥补方案空间。为了突出网络脆弱性弥补分析的重点和技术难点。从大规模的 *Plan-Space* 中寻找落入 *Goal-Space* 中的 *Plan*，约定 *Plan-Space* 已知。

前提假设 2。弥补方案有效性判定函数 *Valid*()已知。如果将能够到达关键目标的每条攻击路径表示成脆弱性的析取子句，那么所有析取子句的合取即构成关键目标的弥补方案有效性判定逻辑表达式 *Valid*()，对于一个给定的 *Plan*，*Valid*(*Plan*)=1 就表示 *Plan* 有效。为了突出重点和技术难点，约定 *Valid*()已知。

前提假设 3。弥补方案代价函数 *Cost*()已知。因为弥补方案的代价取决于弥补方案的实施难度以及相关脆弱性的弥补对网络环境影响的评估，所以确定弥补方案的代价是一个技术含量低且相对主观的复杂过程。为了突出重点和技术难点，约定 *Cost*()已知。

符号约定。为了叙述方便，约定文中出现的符号及其含义，如表 1 所示。

表 1　　符号约定

符号	含义
Plan	弥补方案
Approx-Opt-Plan	近似最优有效弥补方案
Opt-Plan	最优有效弥补方案
Plan-Space	弥补方案空间
Sparse-Space	弥补方案松弛子空间
Valid-Space	有效弥补方案空间
Superior-Space	优势弥补方案空间
Goal-Space	目标弥补方案空间
GeneratePlan()	随机弥补方案产生器
Valid()	弥补方案有效性判定函数
Cost()	弥补方案代价计算函数
N_{Sparse}	松弛子空间规模
$N_{iterate}$	迭代次数
P_{Valid}	有效比率
$P_{Superior}$	优势比率
P_{Goal}	目标达成概率

3　随机松弛优选原理

网络脆弱性弥补分析的重点和技术难点在于如何从大规模的 *Plan-Space* 中快速寻找到落入 *Goal-Space* 中的 *Plan*，实际上这是一个状态空间搜索问题。对于大规模的复杂网络环境，*Plan-Space* 可能很大，为了加速收敛，提出了一种随机松弛优选策略 (SLOS, stochastic loose optimize strategy)，并提出了一种基于 SLOS 的网络脆弱性弥补分析算法 (MCNHA-SLOS, minimum-cost network hardening algorithm based on stochastic loose optimize strategy)。SLOS 的基本原理称为随机松弛优选原理 (SLOP, stochastic loose optimize principal)，首先描述并证明 SLOP。

随机松弛优选原理：将一个集合 *Universe* 划分成 2 个子集 *Low* 和 *High*，*Low* 中元素的个数与 *Universe* 中元素的个数之比为 P_L；从 *Universe* 中随机选取一个元素，则该元素属于集合 *Low* 的概率为 P_L，属于集合 *High* 的概率为 1−P_L；从 *Universe* 中

随机选取 N 个元素，则其中包含 *Low* 中元素的概率为 $1-(1-P_L)^N$。因此无论集合 *Universe* 有多少元素，也无论 P_L 有多小，只要取一个适当大小的数 N，就可以保证 $1-(1-P_L)^N\approx 1$，也就是从 *Universe* 中随机选取 N 个元素，其中至少有一个元素存在于集合 *Low* 中的概率几乎为 1。

随机松弛优选原理的证明非常简单，如图 1 所示。

前提条件:

$|Low|/|Universe| = P_L$;

$r_1, r_2, \cdots, r_N \in Universe$;

$r_1, r_2, \cdots, r_N$是*Universe*中一组随机元素

结论:

$P(\{ r_1, r_2, \cdots, r_N \} \cap Low \neq \varnothing) = 1-(1-P_L)^N$,

其中$P(event)$表示事件$event$发生的概率

证明:

∵ $r_1, r_2, \cdots, r_N$是*Universe*中一组随机元素.

∴ $P(r_1 \in Low) = P(r_2 \in Low) = \cdots = P(r_N \in Low) = P_L$。

∴ $P(r_1 \notin Low) = P(r_2 \notin Low) = \cdots = P(r_N \notin Low) = 1-P_L$。

又∵ 事件$r_1 \notin Low, r_2 \notin Low, \cdots, r_N \notin Low$相互独立。

∴ $P(\{r_1, r_2, \cdots, r_N\} \cap Low = \varnothing)$

$= P((r_1 \notin Low) \cap (r_2 \notin Low) \cap \cdots \cap (r_N \notin Low))$

$= P(r_1 \notin Low) P(r_2 \notin Low) \cdots P(r_N \notin Low)$

$= (1-P_L)^N$

∴ $P(\{ r_1, r_2, \cdots, r_N \} \cap Low \neq \varnothing)$

$= 1-P(\{ r_1, r_2, \cdots, r_N \} \cap Low = \varnothing)$

$= 1-(1-P_L)^N$

图 1 随机松弛优选原理证明

4 网络脆弱性弥补分析算法

4.1 算法思想

MCNHA-SLOS 思想：为了同时满足代价最小和有效弥补 2 项指标，MCNHA-SLOS 迭代地应用 SLOP，每次迭代都从 *Plan-Space* 的一个 N_{Sparse} 规模的 *Sparse-Space* 中选取代价最小的弥补方案 *Min-Plan*，并依据 *Valid*(*Min-Plan*)更新近似最优方案 *Approx-Opt-Plan*，通过 $N_{iterate}$ 次迭代使 *Approx-Opt-Plan* 落入 *Goal-Space* 的概率 P_{Goal} 几乎为 1。

对于给定的优势比率 $P_{superior}$，通过 $N_{iterate}$ 次 N_{Sparse} 规模的 *Sparse-Space* 迭代，目标达成概率 P_{Goal} 为 $(1-(1-P_{Superio})^{N_{Sparse}})(1-(1-P_{Valid})^{N_{iterate}})$。可以证明无论 $P_{Superior}$ 和 P_{Valid} 有多小，都可以选取适当的 $N_{iterate}$ 和 N_{Sparse} 使 P_{Goal} 几乎为 1。

MCNHA-SLOS 巧妙地运用了随机松弛优选策略，将全空间的优化问题转化为阶段性的松弛子空间的优化问题，将 NP 难度的求解问题转化为精度可控的、可快速求解的迭代运算。

4.2 算法描述

MCNHA-SLOS 的伪代码如图 2 所示。首先，将 2^n-1 赋予近似最优弥补方案 *Approx-Opt-Plan*，表示所有的脆弱性都需要弥补，这必然是有效的，并且具有最大的弥补代价。接下来进行 $N_{iterate}$ 轮迭代，在每轮迭代中，首先将 *Approx-Opt-Plan* 赋予临时的代价最小方案 *Min-Plan*；再利用 *GeneratePlan*()产生 N_{Sparse} 个随机方案 *Plan*，每产生一个 *Plan*，计算其代价 *Cost*(*Plan*)，并与 *Min-Plan* 的代价 *Cost*(*Min-Plan*)进行比较，若 *Cost*(*Plan*)＜*Cost*(*Min-Plan*)，则将 *Min-Plan* 更新为 *Plan*；N_{Sparse} 次迭代结束后，判定 *Min-Plan* 的有效性，若 *Valid*(*Min-Plan*)为真，则将 *Approx-Opt-Plan* 更新为 *Min-Plan*。$N_{iterate}$ 轮迭代结束后，输出近似最优弥补方案 *Approx-Opt-Plan*，算法结束。

```
Start:
    Approx-Opt-Plan ← 2^n−1;
    For round = 1 to N_iterate;
        Min−Plan ← Approx−Opt−Plan;
        For count = 1 to N_Sparse;
                Plan = GeneratePlan();
                If( Cost(Plan) < Cost(Min−Plan) )
                        Min-Plan ← Plan;
        If( Valid(Min−Plan) )
                Approx−Opt−Plan ← Min−Plan;
    Output Approx−Opt−Plan;
End
```

图 2 MCNHA-SLOS 算法伪代码

4.3 算法分析

精确求解 MCNH 问题是在 *Plan-Space* 中寻找 *Opt-Plan*，其理论搜索量为 2^n 次，在冯·诺依曼计算结构下，当 n 较大时该问题是不可解的。而 MCNHA-SLOS 算法则是通过 $N_{iterate}N_{Sparse}$ 次搜索以很高的概率得到落入 *Goal-Space* 的 *Approx-Opt-Plan*。具体地，P_{Goal}、$P_{Superior}$、P_{Valid}、N_{Sparse}、$N_{iterate}$ 之间关系满足式(1)。

$$P_{Goal} = \left(1-\left(1-P_{Superior}\right)^{N_{Sparse}}\right)\left(1-\left(1-P_{Valid}\right)^{N_{iterate}}\right) \quad (1)$$

从式(1)可以看出，在网络环境和关键目标给定的情况下，P_{Valid} 是固定的；在目标空间 *Goal-Space* 确定的情况下，$P_{Superior}$ 也是固定的。因此无论 $P_{Superior}$ 和 P_{Valid} 有多小，总可以选取适当的 N_{Sparse} 和 $N_{iterate}$，使 P_{Goal} 几乎为 1，也就是使 *Approx-Opt-*

Plan 几乎必定落入 *Goal-Space*，达到用户的期望。

通俗地讲，MCNHA-SLOS 算法将 2^n 量级的精确求解问题，转化成了 $N_{iterate}N_{Sparse}$ 量级的近似求解问题，将不可能在有效时间内求得最优解的问题转化为可以在有效时间内获得满意的近似最优解问题，并且可以依据拥有的计算资源和可容忍的时间控制求解精度，非常适合在大规模网络环境下应用。

5　实例分析

本节通过一个简单的实例演示 MCNHA-SLOS 算法的计算流程，如图 3 所示。

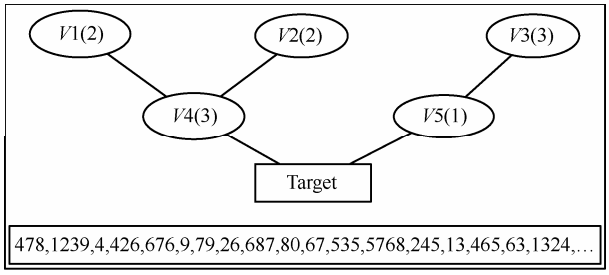


图 3　网络脆弱性弥补分析实例

图 3 中有 $V1$~$V5$ 共 5 个脆弱性可能威胁到关键目标 Target，弥补这 5 个脆弱性的代价分别为(2, 2, 3, 3, 1)，有效性判定函数 $Valid(Plan)=(V1||V4)\ \&\&\ (V2||V4)\ \&\&\ (V3||V5)$。简化攻击图下方是一串随机数，可以转化为随机生成的弥补方案。不难看出代价最小弥补方案为$(00011)_2$，即弥补脆弱性 $V4$ 和 $V5$ 可以有效防护关键目标 Target，且弥补代价最小，有效比率 P_{Valid} = 15/32≈0.47，*Plan-Space* 中的方案按弥补代价排序后的结果如表 2 所示，其中加粗显示的为有效弥补方案。

在本实例的计算过程中，将 N_{Sparse} 取为 2，并且将 *GeneratePlan*() 取为 Rand()%2^5，MCNHA-SLOS 的具体演算过程如图 4 所示。

Start
Approx-Opt-Plan = $(11111)_2$;
round= 1;
　　Sparse-Space = { 478%2^5=$(11110)_2$, 1239%2^5=$(10111)_2$};
　　Min-Plan = (10111)2;
　　Approx-Opt-Plan = (10111)2;
round= 2;
　　Sparse-Space = { 4%2^5=$(00100)_2$, 426%2^5=$(01010)_2$};
　　Min-Plan = $(00100)_2$;
round= 3;
　　Sparse-Space = { 679%2^5=$(00111)_2$, 9%2^5=$(01001)_2$};
　　Min-Plan = $(01001)_2$;
round= 4
　　Sparse-Space = { 79%2^5=$(01111)_2$, 26%2^5=$(11100)_2$};
　　Min-Plan = $(11100)_2$;
　　Approx-Opt-Plan = $(11100)_2$;
round= 5;
　　Sparse-Space = {687%2^5=$(01111)_2$, 67%2^5=$(00011)_2$, };
　　Min-Plan = $(00011)_2$;
　　Approx-Opt-Plan = $(00011)_2$;
Output: *Approx-Opt-Plan* = $(00011)_2$;
End

图 4　MCNHA-SLOS 算法示例演算

在实例中，如果令 $P_{Superior}=0.5$，则 *Goal-Space* $=\{(00011)_2, (11001)_2\}$，$P_{Goal}=(1-(1-0.5)^2)\times(1-(1-0.47)^5)\approx 0.72$，而实际的计算结果 *Approx-Opt-Plan* = $(00011)_2 \in$ *Goal-Space*，实际上已经找到了最优方案，而搜索量是 5×2 = 10，较 32 的全空间搜索量要少很多。

6　实验分析

为了便于分析攻击图技术的相关算法，设计实现了目标网络建模演示系统(Net-MD, network modeling and demonstrating system)和网络脆弱性综合分析系统(Net-VA, network vulnerability analyzing system)。Net-MD 可用于快速构建各种规模的模拟网络环境，并将网络环境信息存储在数据库中；Net-VA 可获取数据库中的网络环境数据，应用不同的算法生成目标

表 2　　**依代价排序的弥补方案空间**

Cost(*order*)	*Plan*	*Cost*(*order*)	*Plan*	*Cost*(*order*)	*Plan*	*Cost*(*order*)	*Plan*
0(1)	00000	4(9)	11000	6(17)	**00110**	8(25)	**10110**
1(2)	00001	4(10)	00101	6(18)	10101	8(26)	**01110**
2(3)	10000	4(11)	**00011**	6(19)	**10011**	8(27)	**11101**
2(4)	01000	5(12)	10100	6(20)	01101	8(28)	11011
3(5)	00100	5(13)	10010	6(21)	**01011**	9(29)	10111
3(6)	00010	5(14)	01100	7(22)	**11100**	9(30)	01111
3(7)	10001	5(15)	01010	7(23)	11010	10(31)	11110
3(8)	01001	5(16)	**11001**	7(24)	**00111**	11(32)	11111

网络的攻击图，并将攻击图存储在数据库中。

实验分为 2 组，第一组为 MCNHA-SLOS 的参数分析，通过观察在不同参数设置下的实验结果，明确各参数在算法中所起的作用，制定合理的参数设定策略；第二组在合理的参数设定策略基础上对比 MCNHA-SLOS 算法与其他算法的性能差异。考虑到 MCNHA-SLOS 算法的统计特性，采取多次实验取平均数据的方法。为了讨论方便，在实验中假定所有脆弱性的弥补代价相同，都取为 1，因此弥补方案的代价只取决于包含的脆弱性数量。

6.1 参数分析

利用 Net-MD 构建了 200 个网络节点规模的模拟网络环境，并以粗线框的 4 个服务器为关键目标，如图 5 所示。利用 Net-VA 生成了 30 个脆弱性规模的攻击图，如图 6 所示。并基于此图对 MCNHA-SLOS 进行参数分析。

6.1.1 N_{Sparse} 和 $N_{iterate}$ 分析

针对以上网络环境和攻击图，首先观察在 N_{Sparse} 和 $N_{iterate}$ 取不同值时 MCNHA-SLOS 的计算结果，并分析 N_{Sparse} 和 $N_{iterate}$ 的最佳取值原则，为此将 *GeneratePlan*()固定为 Rand()%2^{30}。

首先，固定 N_{Sparse} 为 16，随着 $N_{iterate}$ 的增加，观察 *Approx-Opt-Plan* 的弥补代价变化情况。如图 7 所示，左边部分代表 *Approx-Opt-Plan* 的平均代价，右边部分代表花费的时间，可以看出，随着 $N_{iterate}$ 的增加，计算时间的消耗逐渐增长，而 *Approx-Opt-Plan* 的平均代价则稳步降低。说明算法迭代的次数越多获得的平均代价就越低，越能接近最低代价。

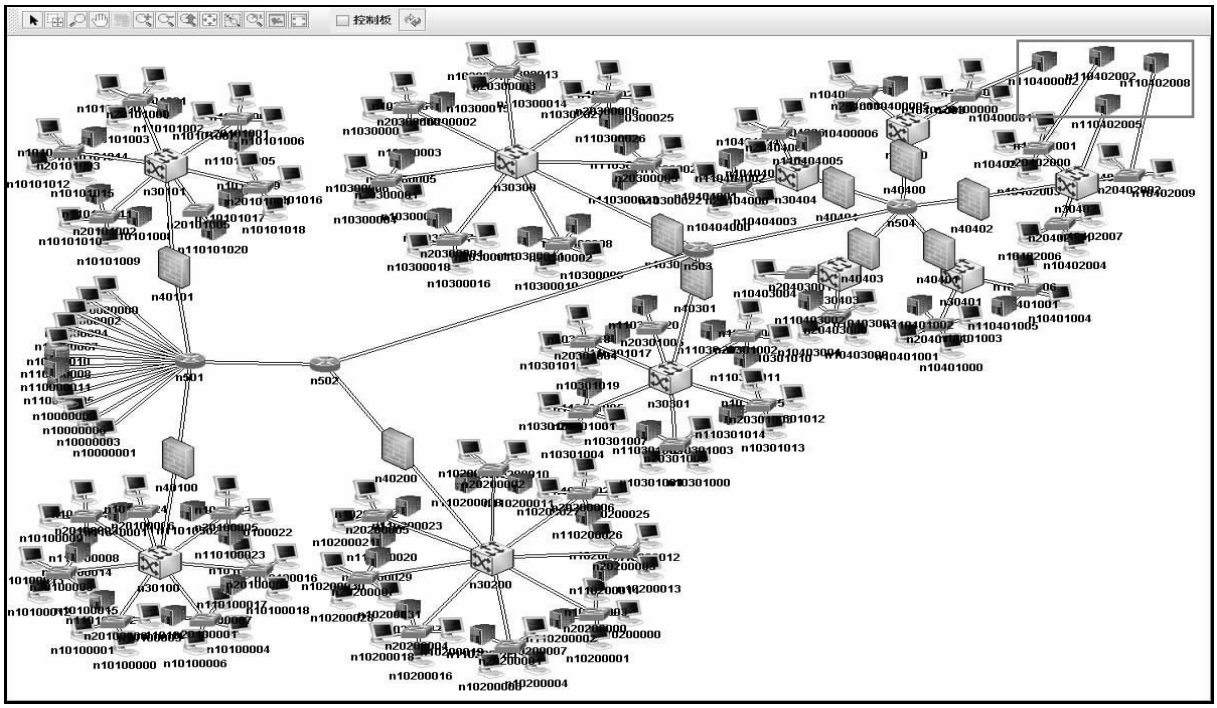


图 5 模拟网络环境

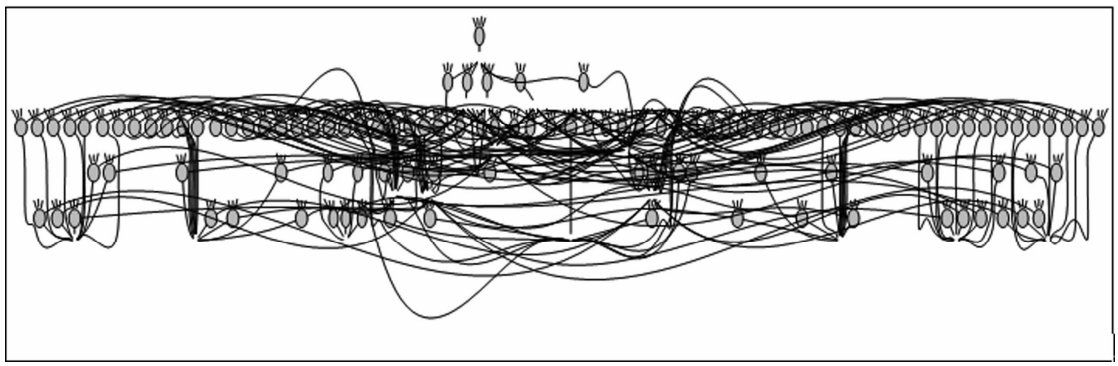


图 6 攻击图

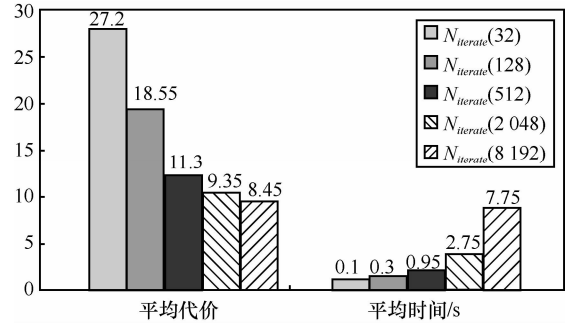


图 7　MCNHA-SLOS 算法耗费收益对比

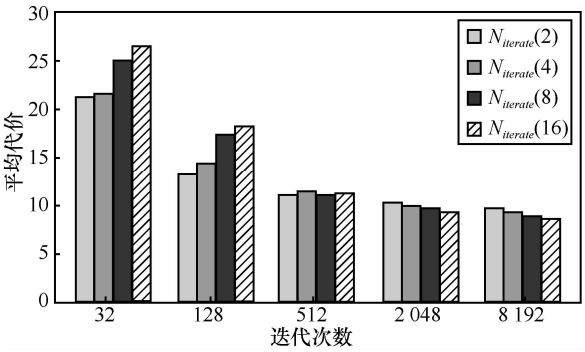


图 8　不同 N_{Sparse} 的比较

从表 3 所列详细实验结果可以看出，当 $N_{iterate}$ 为 32 时，有多达 17 次平凡有效弥补方案出现，同时也有代价低至 10 的弥补方案出现；而当 $N_{iterate}$ 增加到 8192 时，*Approx-Opt-Plan* 的弥补代价基本稳定在 8 和 9。这充分体现了 MCNHA-SLOS 的统计特性，即使计算投入量很小，也有可能获得较好的有效弥补方案，随着计算资源投入的增加，所获得的有效弥补方案的弥补代价在统计意义上稳步下降。

接下来，观察不同的 N_{Sparse} 取值对 *Approx-Opt-Plan* 平均代价的影响，以确定 N_{Sparse} 的取值原则。如图 8 所示，当 $N_{iterate}$ 较小时(如 32, 128)，N_{Sparse} 的取值越小，*Approx-Opt-Plan* 的平均代价越小；当 $N_{iterate}$ 较大时(如 2 048, 8 192)，N_{Sparse} 的取值越大，*Approx-Opt-Plan* 的平均代价越小，但差别并不显著。结合表 4 所示的详细数据，其中“()”中表示的是平均计算时间(s)，可以看出，当 $N_{iterate}$ 较大时，*Approx-Opt-Plan* 的平均代价随着 N_{Sparse} 的增加而降低，而计算时间在增加，总体上 *Approx- Opt-Plan* 的平均代价随着计算时间投入($N_{iterate}N_{Sparse}$)的增加而稳步下降，N_{Sparse} 的取值对算法的影响不大。综合考虑，认为 N_{Sparse} 应当尽量取小一点。

6.1.2　*GeneratePlan*()分析

在上文中都将 *GeneratePlan*()取为 Rand()%2^n，实际上这并不是一个好的策略，从第 5 节的实例分析中可以看出，弥补方案空间低代价区的有效方案只有 2 个，而 Rand()%2^n 所产生的 *Plan* 均匀分布于 *Plan-Space*。每轮迭代通过在 N_{Sparse} 个 *Plan* 中选取代价最低的 *Min-Plan*，它为有效方案的概率会很低，这将导致 *Approx-Opt-Plan* 很长时间才能得到更新。相反，如果弥补方案空间低代价区的有效方案较多，*Min-Plan* 为有效方案的概率就会较高，这样 *Approx-Opt-Plan* 就会经常更新。如果能够依据 *Plan-Space* 中有效弥补方案的比率 P_{Valid}，改变由 *GeneratePlan*()控制生成的 *Plan* 的分布范围，就能够提高 MCNHA-SLOS 的效率。

在 *Plan* 的定长二进制表示下，$(Plan)_2$ 包含的“1”越多，则 *Plan* 有效的概率就越高，用 *Density*(*Plan*)表示$(Plan)_2$ 中“1”的个数与$(Plan)_2$ 长度的比值，称其为方案 1 密度；并为 *GeneratePlan*()引入参数 *density* 来控制其所产生方案的 1 密度。新的 *GeneratePlan*()定义为

表 3　**MCNHA-SLOS 实验结果**

$N_{iterate}$	实验结果	平均代价	平均时间/s
32	30, 12, 30, 30, 30, 30, 12, 30, 30, 30, 10, 30, 30, 30, 30, 30, 30, 30, 30, 30	27.2	0.1
128	11, 30, 12, 10, 10, 11, 30, 10, 12, 10, 30, 30, 11, 30, 30, 11, 11, 30, 30, 12	18.55	0.3
512	10, 10, 10, 11, 8, 30, 13, 8, 10, 12, 10, 11, 12, 12, 11, 9, 10, 9, 11, 9	11.3	0.95
2 048	9, 9, 10, 9, 10, 10, 8, 10, 9, 9, 9, 10, 10, 10, 10, 8, 9, 9, 10, 9	9.35	2.75
8 192	9, 8, 9, 8, 9, 8, 9, 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 8, 8, 8	8.45	7.75

表 4　**不同 N_{Sparse} 的比较**

N_{Sparse}	$N_{iterate}$(32)	$N_{iterate}$ (128)	$N_{iterate}$ (512)	$N_{iterate}$ (2 048)	$N_{iterate}$ (8 192)
N_{Sparse}(2)	21.7(0.05)	13.25(0.2)	11.05(0.45)	10.2(1)	9.5(2.15)
N_{Sparse} (4)	22.1(0.05)	14.45(0.2)	11.45(0.6)	9.85(1.5)	9.1(3.1)
N_{Sparse} (8)	25.7(0.1)	17.7(0.3)	11.1(0.75)	9.6(2.05)	8.75(4.5)
N_{Sparse} (16)	27.2(0.1)	18.55(0.3)	11.3(0.95)	9.35(2.75)	8.45(7.75)

$$GeneratePlan(density)=(x_1x_2,\cdots,x_n)_2 \quad (2)$$

其中，如果 Rand()%10 < *density*×10，则 x_i=1，如果 Rand( )%10 ≥ *density*×10，则 x_i=0。而 *GeneratePlan*() =Rand()%2^n 实际上是当 *density*=0.5 时式(2)的退化形式。本节固定 N_{Sparse} 为 5，观察 *density* 的不同取值对 *Approx-Opt-Plan* 平均代价的影响。

对于图 5 所示的网络环境和图 6 所示的攻击图，实验结果如图 9 所示。当 *density* 取 0.3 时，由 *GeneratePlan*()产生的 *Plan* 有效的概率较低，*Approx-Opt-Plan* 难以更新，因此当 $N_{iterate}$ 较小时，*Approx-Opt-Plan* 的平均代价维持在较高的水平(包含较多的平凡有效弥补方案)，但当 $N_{iterate}$=8 192 时，*Approx-Opt-Plan* 的平均代价迅速下降至较低的水平；当 *density* 取 0.7 时，由 *GeneratePlan*()产生的 *Plan* 有效的概率较高，*Approx-Opt-Plan* 比较容易更新，因此当 $N_{iterate}$ 较小时，*Approx-Opt-Plan* 的平均代价就达到了较低的水平，但由于 *density* 较大，由 *GeneratePlan*()产生的 *Plan* 的代价始终维持在较高的水平，当 *Approx-Opt-Plan* 的平均代价降至一定程度后，就很难再有明显下降；当 *density* 取 0.5 和 0.45 时，由 *GeneratePlan*()产生的 *Plan* 有效的概率比较合理，使 *Approx-Opt-Plan* 的平均代价随着 $N_{iterate}$ 的增加能够迅速下降，并且当 $N_{iterate}$ 较大时也能维持比较明显的下降趋势，相比较而言，*density* 取 0.45 较 0.5 具有更好的效果。

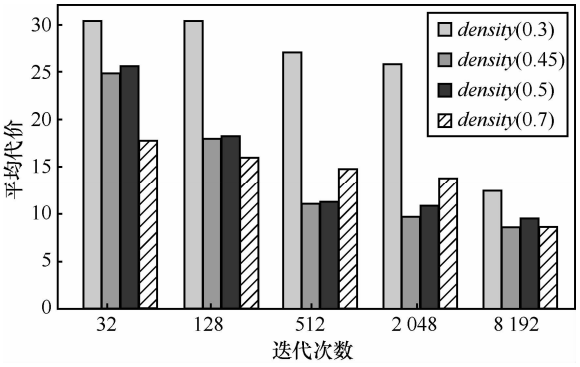


图 9 不同 *density* 的比较

合理地选取 *density*，实际上取决于网络环境和攻击图本身，当攻击图中到达关键目标的路径较密集时，关键目标就越容易遭受攻击，同时也越难弥补，这种情况下 *density* 就应该选取较大的值；而当到达关键目标的路径较稀疏时，则应选取较小的 *density*。

6.2 算法对比

目前为止，已有不少针对 MCNH 问题的求解方法，但真正能够应用于大规模复杂网络的却很少，其中陈峰提出的近似求解算法 weighted-Greedy[7,8]具有较好的效果。因此将 MCNHA-SLOS 与 weighted-Greedy 进行对比。为公平起见，在相同的软硬件环境(Intel Core Duo T7500 2.2 GHz CPU, 2 GB RAM, Windows XP)中运行 2 种算法。Weighted-Greedy 算法通过|*C*|×|*L*|来控制算法的复杂度，其中 *C* 表示脆弱性利用的所有初始前提属性，为方便讨论，假定每个脆弱性只有唯一的前提初始属性，而 *L* 代表所有的 *n*-有效攻击路径。为了对比 2 种算法在不同问题难度下的性能，利用 Net-MD 构建了 5 个不同的网络环境：Net_1，200 个网络节点，10 个脆弱性；Net_2，200 个网络节点，20 个脆弱性；Net_3，200 个网络节点，30 个脆弱性；Net_4，200 个网络节点，40 个脆弱性；Net_5，200 个网络节点，50 个脆弱性。利用 Net-VA 构建了相应的攻击图，并统计出相应的 *n*-有效攻击路径的个数分别为：16、116、244、975 和 2 297。

实验中，为 MCNHA-SLOS 选择适当的 *density*，将 N_{Sparse} 取为 5，并依据|*C*|×|*L*|来设定相应的 $N_{iterate}$，使 2 个算法具有相近的计算量，在此基础上对比 2 个算法求得 *Approx-Opt-Plan* 的平均代价。如图 10 所示，当网络环境和攻击图的复杂程度较低时，weighted-Greedy 较 MCNHA-SLOS 有更好的计算结果，但随着问题复杂程度的增加，MCNHA-SLOS 的优势逐渐显现，在 Net_4 和 Net_5 下，MCNHA-SLOS 得到的 *Approx-Opt-Plan* 的平均代价明显低于 weighted-Greedy，并且这种趋势随着问题复杂程度的增加越来越明显。通过对比实验，MCNHA-SLOS 能够适用于大规模复杂网络环境下的 MCNH 问题求解。

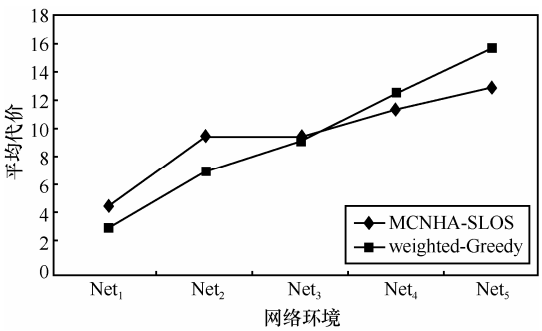


图 10 MCNHA-SLOS 和 weighted-Greedy 算法比较

7 结束语

MCNH 一直受到研究者的关注，目前针对它的研究工作主要集中 2 个方面：一方面是如何构建高效攻

击图，包括智能化、可视化等；另一方面是在现有的攻击图技术的基础上，如何使用其他辅助手段进行高效的、低代价的弥补分析。随着网络安全问题的日益凸显以及网络规模和复杂性的扩大，对这些问题的研究还会更加深入和复杂，也一定会受到持续关注。

本文讨论了 MCNH 问题涉及的基本问题，并针对其 NP 难解性，提出了一种近似最优求解算法 MCNHA-SLOS，运用随机松弛优选策略，将全空间的优化问题转化为阶段性的松弛子空间的优化问题，将 NP 难度的问题求解转化为精度可控的、可快速求解的迭代运算。网络维护者可依据现有计算资源，选择适当的参数，利用 MCNHA-SLOS 算法获得满意的近似最优弥补方案。实验表明，MCNHA-SLOS 较现有算法在求解复杂 MCNH 问题时具有显著优势，能够适用于大规模复杂网络环境。MCNHA-SLOS 虽然能够适用于大规模复杂网络环境，但仍有需要深入研究的地方，下一步的研究工作主要是：1）N_{Sparse} 和 $N_{iterate}$ 之间的关系；2）如何对 *density* 进行合理取值使算法具有更好的性能；3）在可实现规模的真实环境中测试。

参考文献：

[1] JHA S, SHEYNER O, WING J M. Two formal analyses of attack graphs[A]. Proceedings of 15th IEEE Computer Security Foundations Workshop[C]. 2002.

[2] NOEL S, JAJODIA S, O'BERRY B, JACOBS M, *et al*. Efficient minimum-cost network hardening via exploit dependency graphs[A]. Proceedings of 19th Annual Computer Security Applications Conference[C]. 2003.86-95.

[3] WANG L Y, NOEL S, JAJODIA S. Minimum-cost network hardening using attack graphs[J]. Computer Communications, 2006,29(18): 3812-3824.

[4] HOMER J, *et al*. From Attack Graphs to Automated Configuration Management-An Iterative Approach[R]. Kansas State University Technical Report, 2008.

[5] SI J Q, ZHANG B, MAN D P, *et al*. Approach to making strategies for network security enhancement based on attack graphs[J]. Journal on Commurtications, 2009,30(2):123-128.

[6] CHEN F, WANG L Y, SU J S. An efficient approach to minimum-cost network hardening using attack graphs[A]. Proceedings of the 4th International Conference on Information Assurance and Security[C]. 2008.209-212.

[7] CHEN F, ZHANG Y, SU J S, *et al*. Two formal analyses of attack graphs[J]. Journal of Software, 2010,21(4): 838-848.

[8] CHEN F. A Hierarchical Network Security Risk Evaluation Approach Based on Multi-goal Attack Graph[D]. National University of Defense Technology, 2008.

[9] ALBANESE M, JAJODIA S, NOEL S. Time-efficient and cost- effective network hardening using attack graphs[A]. Proceedings of the 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)[C]. 2012.1-12.

[10] DIAMAH A, MOHAMMADIA M, BALACHANDRAN B. Network security evaluation method via attack graphs and fuzzy cognitive maps[J]. Intelligent Decision Technologies. 2012, 16: 433-440.

[11] SWILER L P, PHILLIPS C, ELLIS D, *et al*. Computer-attack graph generation tool[A]. Proceedings of DARPA Information Survivability Conference &Exposition II[C]. 2001.307-321.

[12] SHEYNER O, HAINES J, JHA S, *et al*. Automated generation and analysis of attack graphes[A]. Proceedings of IEEE Symposium on Security and Privacy[C]. 2002.273-284.

[13] SHEYNER O. Scenario Graphs and Attack Graphs[D]. Carnegie Mellon University, 2004.

[14] AMMANN P, WIJESEKERA D, KAUSHIK S. Scalable, graph-based network vulnerability analysis[A]. Proceedings of the 9th ACM Conference on Computer and Communications Security[C]. 2002.217-224.

[15] LIPPMANN R P, *et al*. An Annotated Review of Past Papers on Attack Graphs[R]. MIT Lincoln Laboratory, 2005.

[16] LIPPMANN R P, INGOLS K W, SCOTT C, *et al*. Evaluating and Strengthening Enterprise Network Security Using Attack Graphs[R]. ESC-TR-2005-064, MIT Lincoln Laboratory, 2005.

[17] OU X M, GOVINDAVAJHALA S, APPEL A W. MulVAL: a logic-based network security analyzer[A]. Proceedings of 14th USENIX Security Symposium[C]. 2005.8.

[18] OU X M, BOYER W F, MCQUEEN M A. A scalable approach to attack graph generation[A]. Proceedings of 13th ACM conference on Computer and Communications Security[C]. 2006.336-345.

[19] CHEN F, TU R, ZHANG Y, *et al*. Two scalable approaches to analyzing network security using compact attack graphs[A]. Proceedings of International Symposium on Information Engineering and Electronic Commerce[C]. 2009.90-94.

[20] CHEN F, SUN J S, HAN W B. AI planning-based approach of attack graph generation[J]. Journal of PLA University of Science and Technology, 2008,9(5):460-465.

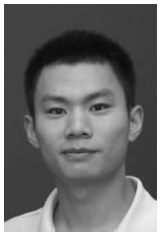
[21] MAN D P, ZHOU Y, YANG W, *et al*. Method to generate attack graphs for assessing the overall security of networks[J]. Journal on Commurtications, 2009,30(3):1-5.

[22] HOMER J, VARIKUTI A, OU XM, *et al*. Improving attack graph visualization through data reduction and attack grouping[A]. Proceedings of 5th International Workshop on Visualization for Cyber Security[C]. 2008.68-79.

[23] HARBORT Z, LOUTHAN G, HALE J. Techniques for attack graph visualization and interaction[A]. Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research[C]. 2011.

[24] ALHOMIDI M A, REED M J. Attack graphs representations[A]. Proceedings of 4th Computer Science and Electronic Engineering Conference (CEEC)[C]. 2012. 83-88.

作者简介：



赵光胜（1984-），男，河南濮阳人，硕士，解放军外国语学院讲师，主要研究方向为网络攻防、网络信息安全。

程庆丰（1979-），男，辽宁朝阳人，博士，解放军外国语学院副教授，主要研究方向为密码学与信息安全。

孙永林（1984-），男，陕西西安人，国防科技大学博士生，主要研究方向为网络安全、移动安全。