文章编号:1007－6735(2026)05－0645－12　DOI: 10.13255/j.cnki.jusst.20251111001

# 基于 DoDAF 的随机需求驱动系统设计与评估

马义翔，　倪　枫，　刘　姜，　付　锐，　窦宇博
（上海理工大学 管理学院，上海 200093）

**摘要：**针对随机需求环境下集成体系(SoS)架构难以兼顾规范化描述与动态决策的问题，提出一种贯通国防部体系架构框架(DoDAF)建模、形式化映射与 DQN 求解的设计评估方法。依据 SysML 2.0 规范建立 DoDAF 视图；将视图产品提取的现实系统任务、资源、状态和规则映射为动态规划要素；最后，通过 DQN 算法解决动态规划问题获得算法分析角度下相关策略的最优解集。创新地建立了从 DoDAF 架构视图到学习型决策模型的可追溯映射。以应急管理系统作为流程化建模及方法验证的实例，证明了该映射方法能够指导 SoS 生成决策，以此应对不同环境下产生的随机需求。

**关键词：**国防部体系架构框架；深度强化学习；SoS；可执行建模
**中图分类号：**N 945　　**文献标志码：**A

# Design and evaluation of stochastic demand-driven systems based on DoDAF

Ma Yixiang,　Ni Feng,　Liu Jiang,　Fu Rui,　Dou Yubo
(*Business School, University of Shanghai for Science and Technology, Shanghai 200093, China*)

**Abstract:** To address the difficulty of balancing standardized description and dynamic decision-making in system of systems (SoS) architectures under stochastic demand environments, a design and evaluation method integrating the department of defense architecture framework (DoDAF) modeling, formal mapping, and deep Q-network (DQN) solution was proposed. DoDAF viewpoints were first developed in accordance with the SysML 2.0 specification. The missions, resources, states, and rules of the real-world system extracted from the viewpoint products were mapped to dynamic programming elements. Finally, the DQN algorithm was employed to solve the dynamic programming problem and to derive a

**收稿日期:** 2025−11−11
**基金项目:** 国家自然科学基金资助项目(12371508)；教育部产学合作协同育人项目(241103760141654)；上海市大学生创新创业训练计划资助项目(SH2025085，XJ2025167)
**第一作者:** 马义翔(2000—)，男，硕士研究生．研究方向：系统科学、系统分析与集成．E-mail：yixiangma000@163.com
**通信作者:** 倪　枫(1982—)，男，副教授．研究方向：系统科学、系统分析与集成．E-mail：nifeng@usst.edu.cn

**引文格式:** 马义翔，倪枫，刘姜，等．基于 DoDAF 的随机需求驱动系统设计与评估[J]．上海理工大学学报，2026，48(5)：645-656.
**Citation:** Ma Yixiang, Ni Feng, Liu Jiang, et al．Design and evaluation of stochastic demand-driven systems based on DoDAF[J]．Journal of University of Shanghai for Science and Technology, 2026, 48(5)：645-656.

set of optimal policies. The principal innovation lay in establishing a traceable mapping from DoDAF architecture viewpoints to a learning-based decision model. An emergency management system was used as a case study to demonstrate the proposed modeling and verification process. The results confirm that the mapping method can support the SoS in making effective decisions in response to stochastic demands arising from different environmental conditions.

**Keywords:** *department of defense architecture framework* (*DoDAF*); *deep reinforcement learning* (*DRL*); *system of systems* (*SoS*); *executable model*

“系统之系统”[1]自概念提出以来始终是系统工程领域的核心议题。与传统集成系统不同，在多域军事行动、大规模应急响应等日益复杂的应用场景下，高鲁棒性的系统架构愈发关键[2-3]。而为了应对此类复杂问题，2009年5月，美国国防部颁布了全新的国防部体系架构框架(department of defense architecture framework, DoDAF)2.0版本，对原有的架构导向系统工程方法加以改进[4]。该框架以数据为中心，依托元模型与数据支撑，提供8类视角、52个视图开发工具，并从模型、视图、视角3个维度逐层细化，推动体系架构由顶层概念设计走向系统工程实践[5-6]。其产品能够有效规划分布式组件、协调各组件功能，保障从高层作战需求到技术规范的可追溯性，以及结构、行为与功能表达的严谨性。然而，该框架虽优于描述性建模，却缺乏对外部环境变化下集成体系(system of systems, SoS)动态策略进行量化评估的固有机制。针对这一不足，研究在已有工作基础上，采用体系化建模映射算法元素的流程化方法，指导SoS应对复杂问题。

## 1 文献综述

目前，国内外学者已就非孤立系统的体系架构展开了广泛研究。随着研究的深入，Sillmann等[7]针对传统系统建模方法的局限，提出一种改良的遗传算法。该算法依据搜寻、评估与优化流程，以SoS的物理属性(容量、时序、输入输出等)为基础，将一组约束条件作为生成备选架构的参照，从而得到满足演化需求的SoS架构方案，实现候选方案的自动筛选，在节约分析资源的同时提升系统构建效率。

Raman等[8]提出了一个集成化的决策学习框架(decision learning framework)，用于指导复杂系统与SoS的架构设计决策。该框架渐进式地积累组织的架构知识库，使决策者能够在多次决策循环中持续学习，并借助机器学习方法将历史决策经验量化为不确定性评估的依据。这一研究的价值在于，它将架构设计从一次性的“最优解求解”转变为持续演化的“学习型决策过程”，为应对SoS长期演化中的不确定性提供了方法论支撑。此外，Brings等[9]提出了基于物理系统量化特征模型动态生成SoS的方法。Yin等[10]针对灾难不确定性及其引发的SoS综合能力衰减问题，提出了两阶段鲁棒优化算法，先处理可能影响救援的关键问题，再规划救援行动与救援区域，以减轻未知因素对救援效果的扰动，有效解决动态不确定性问题。

基于模型的系统工程(model-based systems engineering，MBSE)方法在现实系统中运用时，已有研究采用基于代理、事件仿真以及图论建模的方法来描述SoS架构。Mori等[11]提出了一套基于SysML(systems modeling language)综合视角、面向SoS的架构建模方法，并在智能电网场景中验证了该方法的适用性，展示了SysML在SoS架构描述中的表达力与实用性。在系统仿真模拟中，多项研究通过量化规则构建体系架构与实体单元之间的映射关系，以比较并选择适宜的系统实施方案[12-14]。Kang等[15]从网络中心战出发，提出了一种SoS模型转换的方法，在保证通信效应分析精度的同时显著缩短了仿真执行时间。这一研究表明，机器学习方法不仅可以辅助架构决策，还可以直接嵌入架构建模与仿真流程中。Robinson[16]将体系内部的状态变化按时间登记为离散事件序列，最后通过建模仿真以及相应的事件演化规则，呈现系统随时间变化的演化方向及关键事件。

因此，如何有效地将 MBSE 提取的信息转化为形式化模型，逐渐成为此类研究的重点。Guariniello 等[17] 整合 MBSE 方法的工具集，以此更有效地分析、评估和比较 SoS 架构，对不同抽象层次上的演化能力进行量化分析，从而扩展 MBSE 的分析深度和能力。同时，弥补单一工具可能造成分析指标不足而导致对系统全貌的观察出现纰漏。Xia 等[18] 提出了专注于系统组件表现的模型驱动建模方法，通过将现实系统及其元模型的信息通过 DoDAF 视图再到需求评价仿真模型，从而全面评价分析系统的构建体系情况。该方法可在 SoS 建造前评估体系架构，进一步降低研发成本和风险。Hayano 等[19] 将架构研究的视角延伸到离散事件系统的分散式监督控制领域，这一工作在控制理论层面为 SoS 中分散决策体的协调控制提供了架构层面的解决方案，拓展了“架构”一词从“结构描述”到“控制逻辑组织”的内涵边界。在此基础上，Tao 等[20] 提出了 WFT-net 的概念，将数据操作纳入工作流网的语义框架，通过数据精化定义了 WFT-net 的触发规则，从而避免了无界数据导致的状态爆炸问题。而 Huff 等[21] 探索了规则需求、系统框架、专家建议等现实建模元素从 DoDAF 视图产品到整数规划问题之间的联系。Fang 等[22] 结合波浪模型给出了一个 SoS 架构随着时间进化的演化过程，提出了从 DoDAF 视图产品到近似动态规划问题的映射方法。

DQN（deep Q-learning）算法是一种将深度神经网络与 Q-learning相结合的强化学习算法，其核心是利用深度神经网络近似强化学习中的动作价值函数[23-24]，解决传统近似方法中参数难以选取的问题，并缓解近似动态规划在高维状态空间中面临的维数灾难。该算法通过两个关键机制提升训练稳定性与效率：一是经验回放，将系统交互过程中产生的状态转移数据存储于缓冲池中，并通过随机采样打破数据间的时序相关性，提高数据利用效率；二是双重网络，通过定期更新一个独立的目标网络来生成稳定的训练目标值，有效缓解价值估计过程中的发散问题。在基于 DoDAF 架构的 SoS 动态决策场景中，DQN 能够依据资源分布、威胁等级和任务进度等系统状态，自主学习近似最优的调度或协同策略，从而在随机需求驱动下实现系统整体效能的优化。

然而，针对 DoDAF 模型向其他可分析模型转换的研究相对较少，且现有方法存在诸多限制，难以满足现实系统高维要素的分析要求。尽管已有文献从概念、目标、组件、演化、映射关系和构建方法等方面研究 SoS 架构，但对环境变化下的架构决策仍缺少探索。因此，本文结合 MBSE 思路，在 DoDAF 基础上，以近似动态规划在高维状态空间中的函数近似方法解决状态空间爆炸的思路为启发，探索 DoDAF 模型要素到 DQN 要素的映射路径，并通过流程化建模减少量化分析中的信息丢失，提供可供参考的随机需求驱动系统在不同阶段与外界环境的互动决策思路。

## 2　随机需求驱动系统架构设计方法

本文提出了一种流程化建模方法，使目标对象由 DoDAF 体系建模过渡到数学形式化表述，并通过求解动态规划问题生成相应方案。如图 1 所示，在 SoS 分析的开始阶段，DoDAF 体系模型能够提取系统的主要要素，如任务目标、任务能力需求、系统组件能力评价。同时，DoDAF 模型可利用资源交换关系视图、系统能力关系矩阵等工具描述系统组件之间的联系。随后，从上述模型中提取关键动态规划要素并建立映射关系，使 DoDAF 视角下的 SoS 能够利用动态规划算法生成应对不同环境的随机需求方案。

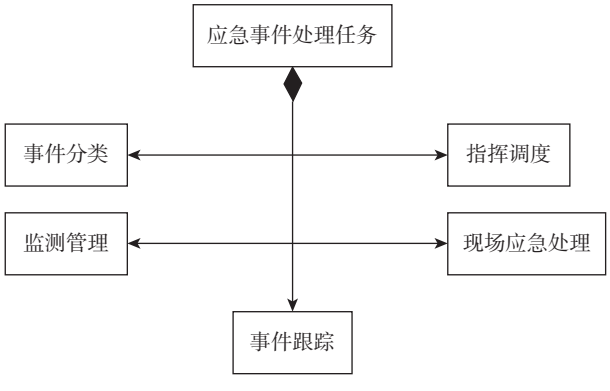


**图 1　流程化建模方法**

**Fig. 1　Process-oriented modeling method**

为了说明建模与分析流程，本文以城市应急管理系统为例进行流程演示。城市发生突发公共安全事件时，应急管理系统通过多渠道预警监测子系统获取灾情信息，并将其上报至相关专业部门进行处理。其行为转换与任务目标能够体现随机需求对系统的影响，符合本文方法的应用范围。随着城市基础设施不断完善、人口日益集

中，一旦发生公共安全事件，处置不当可能给设施和受灾群众造成重大损害。然而，由于政府机构职能的分化越发细致，加上应急事件种类的增多，现有应急系统架构需要具备良好的鲁棒性与灵活性，才能够迅速、有效地应对复杂多变的突发事件。

## 2.1 建立应急管理系统 DoDAF 模型

以数据为中心是 DoDAF 2.0 的显著特征，本文方法旨在通过建立必要的 DoDAF 模型量化 SoS 的主要属性，并利用强化学习算法，使应急管理系统在较长时间跨度内实现抗灾收益最大化。表 1 列出了本文使用的 DoDAF 主要视图工具及其表示方法、用途，包括 OV-5a(行动分解树)、OV-5b(行动模型)、SV-7(系统测量模型)、SV-10a(系统规则模型)、OV-6a(行动规则模型)、SV-10b(系统状态转换说明)、SV-5b(系统与行动追溯矩阵)、SV-4(系统功能描述)。其中，DoDAF 模型主要使用矩阵、UML 以及 SysML 相关的图来表示，选择 8 个视图，共同构成一个从行动问题空间到系统解决方案空间的规范化、模型化桥梁，文中的 SysML 与 UML 分别用于描述不同视图以发挥各自的核心优势。鉴于 SysML 2.0 相关指南已公布，本文相关视图均采用 2023 版 SysML v2 Language Graphical Notation标准。

**表 1 应急事件管理系统体系架构模型**

**Tab. 1 Architecture model of emergency incident management system**

| DoDAF视图 | 目标 | 表示形式 | 其他形式 |
|---|---|---|---|
| OV-5a: 行动分解树 | 识别行动 | 模块定义图 | — |
| OV-5b: 行动模型 | 识别行动 | SysML 活动图 | — |
| SV-7: 系统测量模型 | 提取关键的系统组件要素 | 矩阵图 | — |
| SV-10a: 系统规则模型 | 识别系统及行动的限制条件 | — | 文本描述 |
| OV-6a: 行动规则模型 | 识别系统及行动的限制条件 | — | 文本描述 |
| SV-10b: 系统状态转换说明 | 识别系统组件的状态情况 | UML状态图 | — |
| SV-5b: 系统与行动追溯矩阵 | 构建系统组件与行动关系 | 矩阵图 | — |
| SV-4: 系统功能描述 | 描述SoS的组件及其功能 | 模块定义图 | — |

如图 2 所示，OV-5a 分解了任务目标，系统行动分解为探测、追踪、分类、交战、指挥模块。该视图将行动分为较为粗糙的颗粒度以追求快速地理解行动内容。基于该视图的分解，如图 3 所示，OV-5b 视图则更加充分地展示了各个行动模块之间信息的传递，以及各个活动相互之间的关系，包含终止活动以及传递中的各个活动归属方，如预警监测中心、民情处理中心、指挥与控制中心、应急事件处理中心和后勤中心。

现实中的活动需由对应的 SoS 组件协同完成，因此，需要借助 SV-4 视图对各系统组件功能加以描述。不同类型的系统组件承担价值各异的系统功能，SV-5b 视图则刻画了系统组件与行动中各任务内容之间的对应关系，详见表 2。表中，在对应功能下打“√”，表示系统组件具备这项功能。比如：指挥中心负责信息获取，同时还负责人员调度；民情处理中心负责对应急事件性质的初步排查以及获取相关信息；后勤中心则更多地负责事件善后，减少后期影响。

SV-7 视图展示了各个系统的重要指标，在本研究的城市应急管理系统中，以 SV-7 视图为基准，人员属性通过综合能力表现、专业适配度、空闲与否、在岗年限以及应对权限等参数表征。该视图所选指标旨在简化实验验证与系统组件，以服务于后续动态规划要素的映射选取，其数据范围和属性均按算法对状态空间离散化的要求作了相应简化。如图 4 所示，SV-10b 视图说明了系统组件对各应急事件的响应，其中，每个状态转移都表示该系统组件通过采取行动进入新状态。该视图主要表示了响应外部事件以及对应服务功能的时间顺序，便于展示系统组件状态的更迭。OV-6a 和 SV-10a 则能够全面描述规则对系统和活动的限制，如系统组件的占用限制、对不同系统组件随机事件的应答奖励情况等。

## 2.2 映射 DoDAF 模型元素至动态规划元素

动态规划是一种灵活的算法结构，其主要目的在于通过求解 Bellman 最优方程获得多阶段决策的最优策略方案。其中，近似动态规划是为了解决状态空间爆炸而产生的状态值函数，而非表格值，以此解决状态空间的存储和计算问题。在生

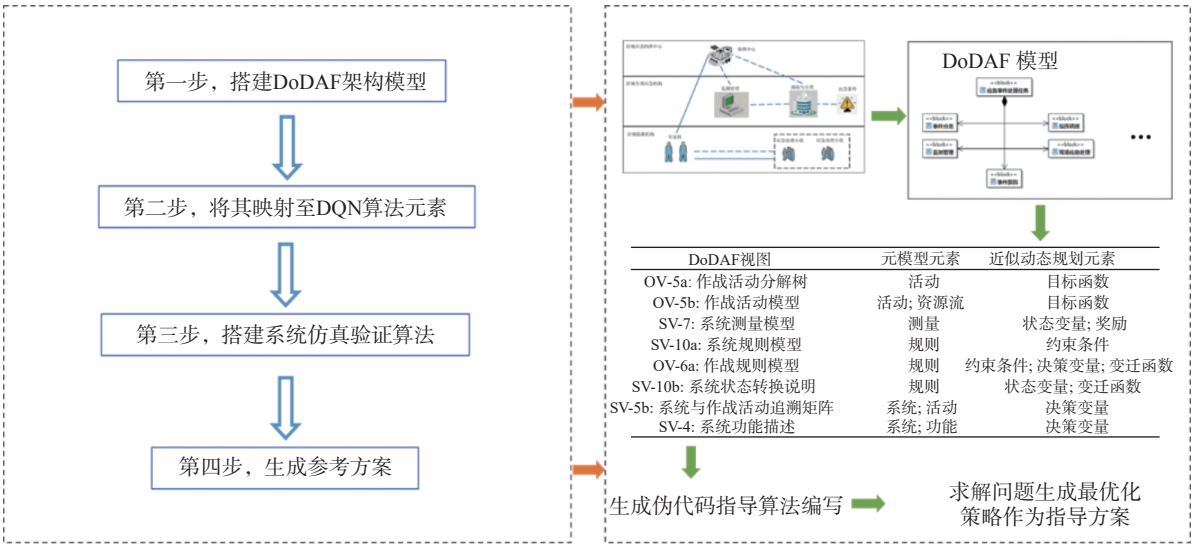


**图 2　行动分解树（OV-5a）**

**Fig. 2　Operational activity decomposition tree（OV-5a）**

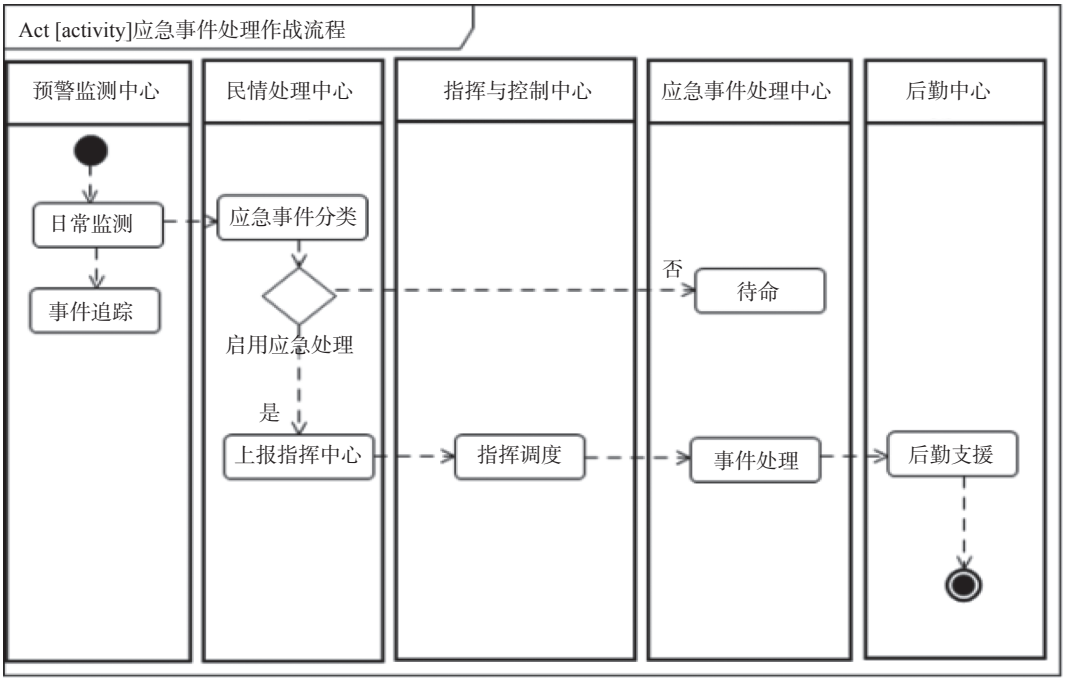


**图 3　行动模型（OV-5b）**

**Fig. 3　Operational activity model（OV-5b）**

成系统架构决策之前，需先映射 DoDAF模型至动态规划形式化元素，包括状态变量、决策变量、目标函数、状态变迁、额外随机信息、奖励以及约束条件。表 3 展示了 DoDAF 模型视图到动态规划建模所需元素借由 DoDAF 元模型生成的映射关系。DoDAF 元模型要素包括了活动、资源、能力、测量、系统、规则等核心实体；OV-6a 和 SV-10a 中则包含了活动、系统、规则等元模型要素。

首先，状态变量是动态规划算法中最重要的部分，是获取策略和变迁函数以及目标函数所需的基本变量。本文采用 SV-7 中获取的系统组件参数来表示初始的资源属性，且对应资源的属性应是向量的形式，除此之外，随机生成的需求也是状态变量的一部分。

对应系统组件参数属性情况，可以用向量 $\boldsymbol{b}$ 表示：

$$\boldsymbol{b}=\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}=\begin{pmatrix} \text{职业表现} \\ \text{学业能力} \\ \text{闲置与否} \end{pmatrix} \tag{1}$$

式中，$b_1,b_2,b_3$ 表示 3 个不同的资源属性，可以根

**表 2 系统与行动追溯矩阵（SV-5b）**

**Tab. 2 System–activity traceability matrix（SV-5b）**

| 系统 | 监测管理 | 事件分类 | 现场信息更新 | 关键信息获取 | 事件处理 | 人员调度 | 事件善后 |
|---|---|---|---|---|---|---|---|
| 预警监测中心 | √ | | √ | √ | | | |
| 民情处理中心 | | √ | | √ | | | |
| 指挥中心 | | | | √ | | √ | |
| 现场处理小组 | | | | √ | √ | | |
| 后勤中心 | | | | √ | | | √ |

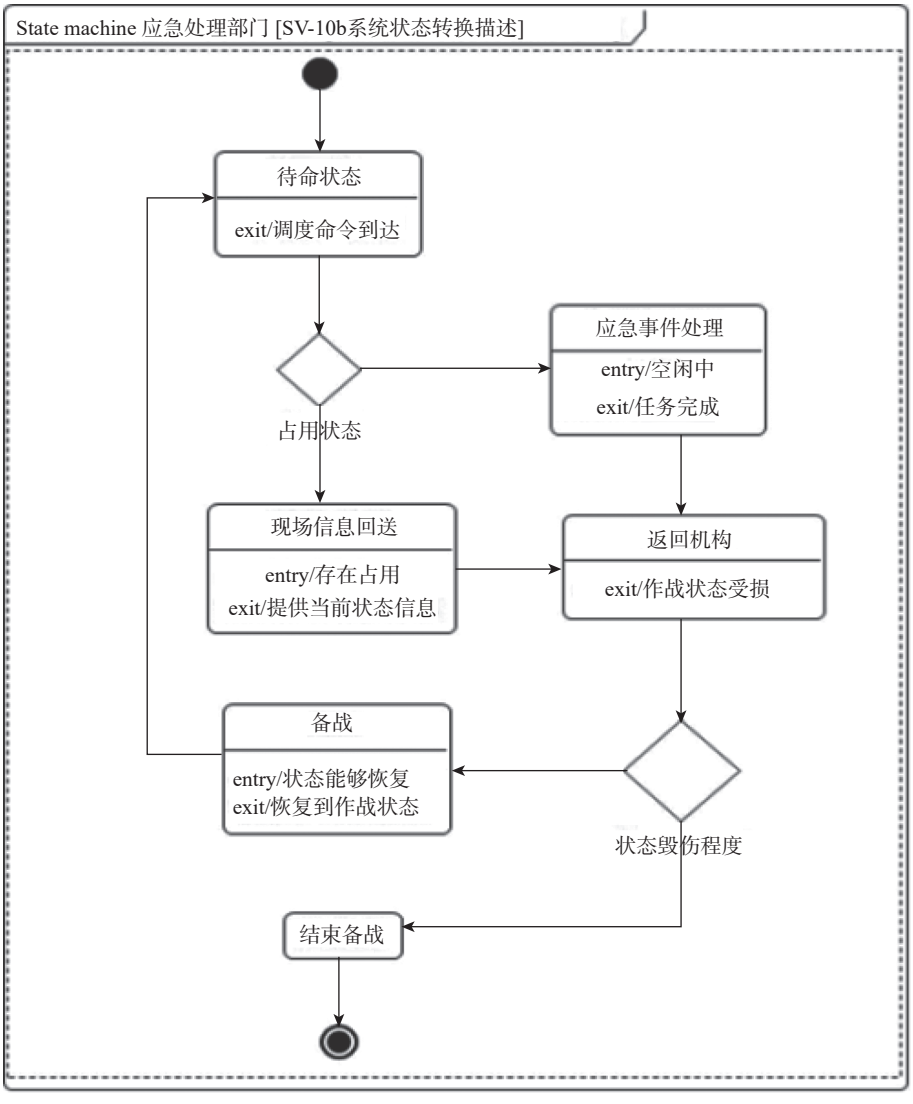


**图 4 系统状态转换描述（SV-10b）**

**Fig. 4 Systems state transition description（SV-10b）**

据 SV-7 视图中的评价要素进行自定义，在表征阶段允许字符串，只要符合离散化要求。

额外随机信息往往是指状态动作会对状态空间产生影响的具有确定分布的随机变量，本文选用变量$\widehat{\mathcal{D}}_{t,d}$表示 $t$ 阶段新生成的具有 $d$ 类型的额外随机任务。因此，案例中的随机需求变量可用向量 $\boldsymbol{d}$ 表示：

$$\boldsymbol{d}=\begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}=\begin{pmatrix} \text{优先级水平} \\ \text{向下包容与否} \\ \text{时间消耗} \end{pmatrix} \tag{2}$$

式中，$d_1,d_2,d_3$表示 3 个不同的需求属性，可以根据现实应急事件的要素进行自定义。

**表 3　DoDAF 模型元素映射至算法元素**

**Tab. 3　Mapping of DoDAF model elements to algorithm elements**

| DoDAF视图 | 元模型元素 | 近似动态规划元素 |
|---|---|---|
| OV-5a: 行动分解树 | 活动 | 目标函数 |
| OV-5b: 行动模型 | 活动; 资源流 | 目标函数 |
| SV-7: 系统测量模型 | 测量 | 状态变量; 奖励 |
| SV-10a: 系统规则模型 | 规则 | 约束条件 |
| OV-6a: 行动规则模型 | 规则 | 约束条件; 决策变量; 变迁函数 |
| SV-10b: 系统状态转换说明 | 规则 | 状态变量; 变迁函数 |
| SV-5b: 系统与行动追溯矩阵 | 系统; 活动 | 决策变量 |
| SV-4: 系统功能描述 | 系统; 功能 | 决策变量 |

由于外在环境多变，且难以通过任一系统架构模型语言进行表述，本文的外在随机信息倾向于通过专家意见获得要素信息，且利用过往信息进行计算，从而获取随机分布。当获得额外信息和系统组件参数时，当前状态变量的组成为

$$S_t = S^{b,d}(b_{i,t},\ \widehat{\mathcal{D}}_{t,d}),\ i=1,\ 2,\ 3 \tag{3}$$

式中：$S_t$表示 $t$ 时刻的状态变量；$S^{b,d}$表示以 $t$ 时刻 $i$ 资源属性和 $j$ 需求属性所组成的函数。

决策变量联系的是状态变量的变迁，说明在应急管理系统中，何种系统组件是针对随机应急事件进行行动的表示。本文主要将应对紧急事件的人员组作为视图 OV-5 中的系统组件，对应着算法中的决策变量。

状态变迁函数表示状态在动作和额外信息干涉下的变化，基于应急事件对系统组件的占用情况，以及决策对资源属性可能会产生的影响来确定。为了体现随机事件的紧迫性以及重要性，本文中每阶段随机需求都会重新生成，对应的状态变迁方程如下所示：

$$S_t^x = S_{\mathrm{M},x}(S_t,\ x_{t,i,d}) \tag{4}$$

$$S_{t+1} = S_{\mathrm{M},D}(S_t^x,\ \widehat{\mathcal{D}}_{t+1,d}) \tag{5}$$

式中：$x_{t,i,d}$为布尔变量，表示在 $t$ 时刻用 $i$ 组人员去解决 $d$ 类型任务的动作；$S_t^x$表示 $t$ 时刻且进行动作 $x_{t,i,d}$之后的状态变量；$S_{\mathrm{M},x}$表示以 $x_{t,i,d}$和状态 $S_t$作为自变量的函数；$S_{\mathrm{M},D}$表示以$S_t^x$和随机需求 $\widehat{\mathcal{D}}_{t+1,d}$作为自变量的函数；M 表示函数为分段函数，分段是为了计算动作后再计算随机信息。

本案例中的第 $i$ 个成员属性变迁函数根据以下不同情况进行分类：

$$\boldsymbol{b}_{i,t+1} = \begin{pmatrix} b_{i,t+1,1} \\ b_{i,t+1,2} \\ b_{i,t+1,3} \end{pmatrix} = \begin{cases} \begin{pmatrix} b_{i,t,1} \\ b_{i,t,2} \\ b_{i,t,3} \end{pmatrix},\ x_{t,i,d}=0,\ b_{i,t,3}=0 \\ \begin{pmatrix} b_{i,t,1} \\ b_{i,t,2} \\ b_{i,t,3}-1 \end{pmatrix},\ x_{t,i,d}=0,\ b_{i,t,3}\neq 0 \\ \begin{pmatrix} b_{i,t,1} \\ b_{i,t,2} \\ b_{i,t,3}=d_{t,3} \end{pmatrix},\ x_{t,i,d}=1,\ b_{i,t,3}=0 \end{cases} \tag{6}$$

大括号包括的公式中：第 1 行表示该人员动作没有发生且占用为 0，此时状态保持不变；第 2 行表示该人员动作没有发生且占用不为 0，则根据当前时间，减去 1 单位；第 3 行表示选择该人员发生动作，与此同时，该人员必须是空闲状态，状态$b_{i,t,3}$选择工作状态，时间即为随机需求的占用时长。本文每阶段随机需求各属性按离散均匀分布生成，由于各属性的分布可以根据实际情况定义，故随机需求的生成此处不再赘述。

构筑合适的目标函数和奖励是得出良好结论的关键，因此，要仔细考量各种情况的奖励值。CV-6 视图能力映射展示出了各个 SoS 组件对应的行动，所求贝尔曼最优公式的表达如下所示：

$$V(S_t) = \max \mathrm{E}\left[\sum_{t=0}^{T}\gamma^t C_t(S_t,\ x_t)\right] \tag{7}$$

式中：$V$ 表示 $S_t$ 状态下的累计奖励值函数；$C_t$ 表示 $S_t$ 状态下做出 $x_t$ 行为的一组奖励值函数；$\gamma$ 表示未来折扣系数，用来调整未来阶段的收益对状态值的影响，可以根据现实案例的要求进行取值。

约束条件中的具体信息来自先前总结的 OV-6a 和 SV-10a 的 DoDAF 产品，也包括 DQN 算法对变量的限制、不同情况下分类的奖励取值范围，以及一些其他需求生成或者状态空间维度的限制。因奖励规则可以根据案例自行决定，故主要展示本案例下状态空间的限制。设$\boldsymbol{P}$为人员属性矩阵，有 $i$ 个人员，每个人员有 $j$ 个属性，且 $i,j\in\mathbb{Z}^+$，在本文中 $i$=4，$j$=3，具体矩阵如下所示：

$$\boldsymbol{P} = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \\ p_{4,1} & p_{4,2} & p_{4,3} \end{bmatrix} \tag{8}$$

式中：$P_{i,1}\in\{0,1,2\}$表示专业契合度，以离散数字

代替字符串，表示不同的专业契合度；$P_{i,2}\in\{0,1,2,3\}$表示处理人员的在岗时长；$P_{i,3}\in\mathbb{Z}^{+}$表示占用时间，其上限由随机生成的需求属性所决定。设$\boldsymbol{D}$为需求属性矩阵，每个需求有$k$个属性，本文$k$取值为3，具体矩阵如下所示：

$$\boldsymbol{D}=\begin{bmatrix} d_1 & d_2 & d_3 \end{bmatrix} \tag{9}$$

式中：$d_1\in\{0,1\}$表示需求的紧急程度，0表示不紧急，1表示紧急；$d_2\in\{0,1\}$表示能否由非专业的人员去完成；$d_3\in\{1,2,3\}$表示具体占用阶段数。

设$\boldsymbol{S}$为状态空间矩阵，由人员属性矩阵和需求属性矩阵拼接而成，即

$$\boldsymbol{S}=\begin{bmatrix} \boldsymbol{P}^{\mathrm{T}} & \boldsymbol{D}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{10}$$

式中，$\boldsymbol{P}$为4×3的矩阵，$\boldsymbol{D}$为1×3的矩阵，因此，$\boldsymbol{S}$为5×3的矩阵。

### 2.3 建立应急管理系统的深度学习算法

动态规划算法是一种通过将复杂问题分解为子问题，并存储子问题解来高效求解最优决策的算法范式，其本质上是通过把整体最优的问题分解为求解每个阶段及其未来收益总合最优。自贝尔曼提出多阶段最优决策的数学框架之后，经过多年发展，衍生出了多种计算状态值的方式，以适应不同的收敛场景。特别是当前人工智能领域广泛使用的近似动态规划结合双重神经网络，并辅以Q值学习的算法，解决了长久以来动态规划学术群对于状态空间和决策空间爆炸，以及状态值拟合函数参数选取的难题。通过替换算法迭代求值的目标，使其从传统离散的表格存储状态值过渡到能够存储所有状态空间值的状态值函数，通过逼近获得近似值函数，减少相应的计算复杂度，也因此获得接近最优的方案。首先，确立动态规划策略目标求解的贝尔曼最优公式：

$$V(S_t)=\max \mathrm{E}\left[C_t(S_t,x_t)+\gamma\overline{V}(S_{t+1}|S_t)\right] \tag{11}$$

式中，$\overline{V}$表示未来近似值函数。期望符号E内的第1个式子表示$t$阶段做出动作后的回报值，第2个式子表示折扣后的未来状态值。

为了能够增量逼近需要的近似值函数，DQN算法采用了RM(Robbins-Monro)算法，即

$$w_{t+1}=w_t+\alpha_t\nabla_w\mathcal{J}(w) \tag{12}$$

式中：$w_t$表示$t$阶段对未知参数的估计值；$\alpha_t$表示可调节的折扣参数；$\mathcal{J}(w)$表示损失函数，计算如下：

$$\mathcal{J}(w)=\mathrm{E}\left[\left(\mathcal{R}+\gamma\max_{a\in\mathcal{A}(S')}\hat{q}(S',a,w)-\hat{q}(S,\mathcal{A},w)\right)^2\right] \tag{13}$$

式中：$\mathcal{R}$表示当前阶段奖励值；$\gamma$是根据情况调整的未来参数；$S'$表示后置状态；$S$表示前置状态；$\mathcal{A}$表示动作集合；$\hat{q}$表示近似Q值函数。括号内表示贝尔曼最优方程和求取的拟合函数的差值。

式(12)实际上是贝尔曼最优误差，该值最理想的情况应当为0。为了更好地进行偏导运算，DQN算法通过以下假设将函数$y$中主网络所得参数$w$转换成目标网络参数$w_{\mathrm{T}}$，计算公式为

$$y\doteq\mathcal{R}+\gamma\max_{a\in\mathcal{A}(S')}\hat{q}(S',a,w) \tag{14}$$

采用双重网络设计架构的损失函数，其梯度表示如下：

$$\nabla_w\mathcal{J}(w)=-2\mathrm{E}\left[\left(\mathcal{R}+\gamma\max_{a\in\mathcal{A}(S')}\hat{q}(S',a,w_{\mathrm{T}})-\hat{q}(S,\mathcal{A},w)\right)\nabla_w\hat{q}(S,\mathcal{A},w)\right] \tag{15}$$

将式(14)代入后进行深度学习算法训练，可获得一个输入状态和动作就给出合适Q值的代理，DQN算法通过建立主网络和目标网络，并且固定变量迭代更新的方法逐步实现最优的同时，达到了计算的简便性和有效性的平衡。

### 2.4 应急管理系统算法流程

为了生成随机需求驱动系统的解决方案，首先，需要解决先前数学形式化映射部分。然后，通过DQN算法解决复杂状态空间的动态规划问题，得出需要的解决方案，也就是贝尔曼最优公式所求的最优策略。算法流程如下：

**步骤1** 选择一个行为策略，并用它产生经验池$\mathcal{B}$：

$$\mathcal{B}=\left\{\left(S,a,r,S'\right)\right\}$$

**步骤2** 设$w_{\mathrm{T}}$为目标网络需要更新的参数，$w$为主网络所需要更新的参数，每次迭代需要对经验池中的样本进行随机的均匀采样。

**步骤3** 将每个采样$(S,a,r,S')$代入式(15)，求出目标值：

$$y_{\mathrm{T}}\doteq\mathcal{R}+\gamma\max_{a\in\mathcal{A}(S')}\hat{q}(S',a,w_{\mathrm{T}}) \tag{16}$$

**步骤4** 对主网络进行参数的更新，使损失函数达到最小值。结束步骤4后返回步骤2，直到完成$N$次后进入步骤5。

**步骤5** 在进行$N$组采样后，将主网络中的参

数代入到目标网络的参数中，完成后继续进行步骤 2。

**步骤 6**　满足任一算法终止条件：达到最大迭代次数、回报长期不提升、参数变化足够小、贝尔曼残差足够小，便输出结果，否则返回步骤 2。

## 3　实验验证和分析

### 3.1　算法环境搭建

本文以应急事件管理系统为案例，通过 DoDAF 体系建模提取关键信息后，进行动态规划元素映射，获取本文所需用例要求后，进行 DQN 算法的搭建。以 SV-7 视图为基准，选取其中的专业适配度、空闲与否、在岗年限以及应对权限等作为状态变量。随机需求则根据紧急程度、应对人员的专业要求程度及处理占用时间 3 个属性进行仿真。为保证算法的探索效率以及简化运算，将各属性按离散均匀分布，并且每阶段独立产生，未完成的委托将不会延续到下一阶段。该设定确保不会因为系统应答上限而影响随机应急需求的及时响应要求，从而，可以更好地求取 SoS 应对随机事件最优策略的代理。

实验算法通过 Matlab R2023A 进行，主要调用 Deep Learning Toolbox 配合自定义 Simulink 环境求解所需 Agent。硬件环境：AMD Ryzen 7 5800x3d CPU；NVIDIA GeForce RTX 4070 GPU；Memory16.0GB；操作系统为 Windows 10。

### 3.2　超参数选取

本文为了能够覆盖随机生成的需求所处的状态空间，采用探索性策略算法，进行离线数据集生成，收集不同 action-state pair 所生成的状态变量值。此外，在自定义环境下生成数据集时，会将所有生成数据存储于经验回放池中，故而经验回放池大小决定了每次迭代选取的路径目标 Q 值的覆盖情况。为了在随机组员状态和随机事件需求情景下，DQN 算法模拟应急管理系统在应对随机事件时能更好地发挥指挥功能，选取具有较高稳定性的代理进行人员指挥控制，并且选择近 20 组不同情景平均窗口奖励作为选择合适代理的参考。在 DQN 算法中，确定超参数的值对 Q 值的拟合函数性能具有重要意义，但是超参数设置要求比较宽泛，且对实际效果的影响往往较难锁定。

本文尝试选取表现更好的超参数，对于应急事件管理系统的案例实例，分析不同网络结构、学习率、经验池大小、每次训练选取的样本数下的验证结果。在本例中经验池的选取在超过 $10^3$ 之后，窗口平均奖励趋于稳定，经验池增大并不会带来更多的奖励提升。全连接神经元数量在 32 时窗口奖励值降低，而在 64 和 128 时保持合适水平，同时为避免随机需求产生的状态空间导致的过拟合风险，避免使用超高单位的神经元数量。最后，本文选取表 4 所示的超参数作为训练代理的参数。

**表 4　超参数**

**Tab. 4　Hyperparameters**

| 超参数 | 数值 |
|---|---|
| 软更新系数 | $10^{-3}$ |
| 迷你批次 | 64 |
| 折扣因子 | 0.9 |
| 经验缓冲区长度 | $10^6$ |
| 全连接层 | 128 |

### 3.3　多案例结果比较

为了更贴合现实案例中随机需求造成的人员占用情况，比较两种需求占用时长设置对代理总奖励的影响。每阶段需求占用时长分别为 1～3 和 1～5。当需求占用时长不同时，人员状态恢复的可用时间随之变化，这一点直接反映在图 5 和图 6 所示的近 20 组窗口平均累积奖励差异中，需求占用时间越长，代理越需要优先满足救灾需求。即使当前人员组并非最优选择，也可能派出专业程度相对较低但处于空闲状态的人员处理应急事件。可以预见，窗口平均值会随随机应急事件占用时间的增加而进一步下降。因此，提高处理人员的综合素质和处理效率，是提升应急管理系统综合能力的重要途径。

### 3.4　短时段下 Agent 策略选择分析

每次实验中的人员状态均按离散均匀分布生成，并从中选取 3 种差异较明显的应急事件处理小组。在该情况下，使用训练得到的最优 Agent 进行短时段测试，以考察其泛化能力，具体人员状态如表 5 所示。应急事件根据紧急程度、应对人员的专业要求程度及处理占用时间 3 个属性进行仿真。每个属性均按离散均匀分布，并且每阶段独立产生，未完成的委托将不会延续到下一阶段。本次实验选择的应急事件占用时间为 1～3，应急事件组 Code_1 和 Code_2 的人员分配随机需

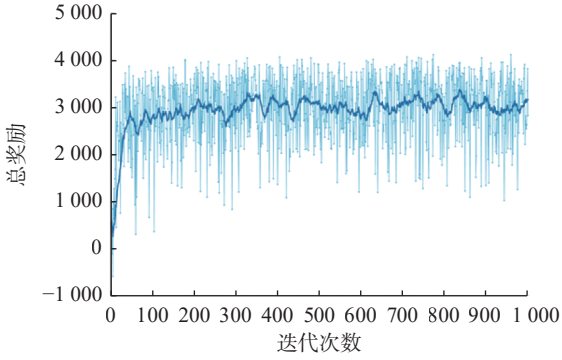


**图 5 随机需求占用为 3 时总奖励**

**Fig. 5 Total reward at a stochastic demand occupancy duration of 3**

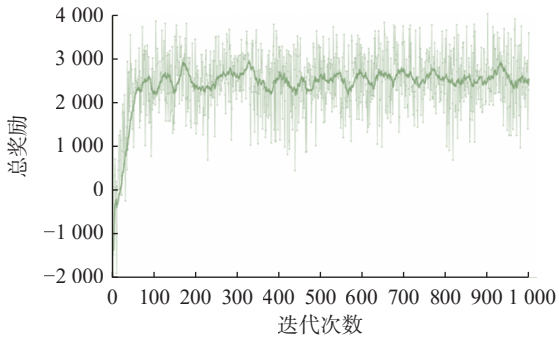


**图 6 随机需求占用为 5 时总奖励**

**Fig. 6 Total reward at a stochastic demand occupancy duration of 5**

求情况如图 7 和图 8 所示。

由图 7 可知，为提高随机需求驱动 SoS 处理应急事件的综合效果，实例训练得到的代理更倾向于选择专业程度较高、在岗年限较长的人员处理相关应急事件。由图 8 可知，在阶段 30 的任务处理中，拥有较高人员素质的组明显有着更高的分配次数。同时，当应急事件占用时间较短时，人员状态恢复速度明显高于任务处理压力。因此，素质差距较大的人员可能始终未被调用。尽管，在多数现实案例中，充分利用人力资源有助于提高综合效益，但该实例说明，在应急事件处理中，能够快速高效处理事件的高素质人员会更受指挥中心的青睐。这样的情况也将会在应急事件处理难度飙升的情况下发生明显变化。当应急事件占用时间明显大于人员状态恢复供给时间时，应急管理系统也会采取增加应急处理反应小组人员数量，以及提高人员综合素质的做法，以应对压力。因此，该方法模拟的在随机需求驱动

**表 5 随机产生的组员状态**

**Tab. 5 Randomly generated member status**

| 行动组代号 | 1号人员 | 2号人员 | 3号人员 | 4号人员 |
|---|---|---|---|---|
| Code_1 | 专业/资历最深 | 正常/在岗较长 | 不专业/新人 | 不专业/资历最深 |
| Code_2 | 专业/资历最深 | 专业/新人 | 不专业/在岗较长 | 专业/在岗较长 |
| Code_3 | 正常/新人 | 专业/在岗较长 | 正常/新人 | 正常/资历最深 |

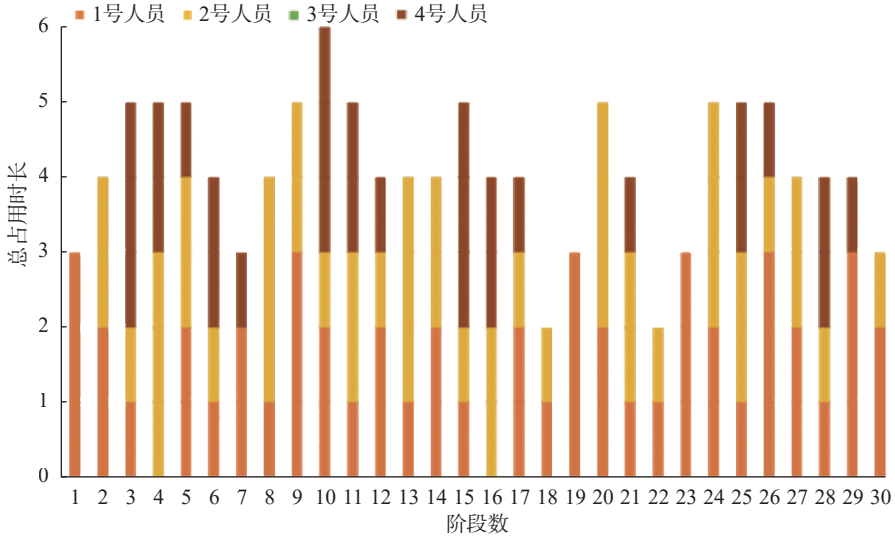


**图 7 行动组 1 人员分配情况**

**Fig. 7 Personnel allocation situation of group 1**

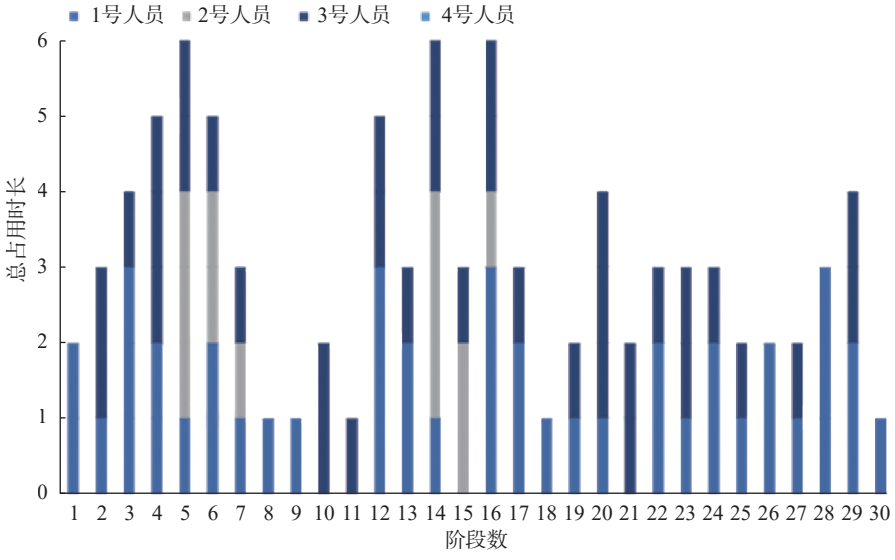


**图 8　行动组 2 人员分配情况**

**Fig. 8　Personnel allocation situation of group 2**

下的 SoS 演化也会有一定程度上的参考价值。

该案例的最优策略结果表明，代理倾向于选择效益较高的人员组，这种情况在追求最优效益的现实案例中具有不错的指导意义，证明了 DQN 方法寻得的随机需求驱动系统处理应急事件的稳定性和有效性，但仍需关注过度使用系统组件造成的损耗，以及由此导致的系统综合能力下降。一种简化处理方式是将人员疲劳恢复时间纳入随机需求处理时间，并假设人员经过若干阶段后恢复正常工作效率。然而，难以避免的是，事件响应的速度以及处理效率会由于人员综合能力的情况而受到影响，这些都需依靠进一步的实验研究来进行探讨。

## 4　结　论

长期以来，无论何种领域，MBSE 越来越成为用于分析不同 SoS 架构的首选。本文应用基于模型的系统工程方法，运用 SysML 通用模型语言对 DoDAF 模型架构中的各个视图进行建模，以提取深度学习算法中随机需求驱动系统所需要的要素。以应急管理系统为例，构建了从 DoDAF 模型到 DQN 要素的完整映射框架。随后，利用形式化要素构建可在 Matlab 中运行的算法，最后求解获得现实案例中所需的最优代理，保证应急管理系统能够针对不同外部环境和随机事件自动生成相应策略。尽管该方法尚未实现全流程自动化，但该方法可以帮助减少系统建模信息的损失，并最大程度保留算法需要的资源、系统、规则和活动等重要信息，帮助更好地理解这类随机需求驱动的 SoS 问题。

本文的建模与映射方法同样适用于规模更大或细节更丰富的系统，其主要难点在于 DoDAF 模型与算法要素之间仍依赖人工映射。因此，将 DoDAF 模型的数学形式化编译过程部分或完全自动化，可作为未来研究方向之一。实现该目标仍需深入厘清 DoDAF 体系元模型与强化学习算法要素之间的逻辑关系。

**参考文献：**

[1] Maier M W. Architecting principles for systems-of-systems[J]. Systems Engineering, 1998, 1(4): 267−284.

[2] U. S. Department of Defense. Quadrennial defense review report[R]. Washington: Department of Defense, 2001.

[3] Ge B F, Hipel K W, Fang L P, et al. An interactive portfolio decision analysis approach for system-of-systems architecting using the graph model for conflict resolution[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2014, 44(10): 1328−1346.

[4] Department of Defense. SUBJECT: the department of defense architecture framework (DoDAF) version 2.0[R]. Washington: Department of Defense, 2009.

[5] 苗学问, 董骁雄, 钱征文, 等. 基于 DoDAF 的航空装备智能保障系统体系结构建模 [J]. 系统工程与电子技术, 2024, 46(2): 640−648.

[6] Wang S K, Li J Z, Zhao J T, et al. Research on architecture design and description methods based on

DoDAF[M]//Angrisani L, Arteaga M, Chakraborty S, et al. Lecture Notes in Electrical Engineering. Singapore: Springer, 2024: 366–372.

[7] Sillmann B, Glock T, Ghassemi R, et al. A multi-objective optimization approach for analysing and architecting system of systems[C]//Proceedings of the 2018 Annual IEEE International Systems Conference. Vancouver: IEEE, 2018: 1–8.

[8] Raman R, D'Souza M. Decision learning framework for architecture design decisions of complex systems and system-of-systems[J]. Systems Engineering, 2019, 22(6): 538−560.

[9] Brings J, Daun M. Towards automated safety analysis for architectures of dynamically forming networks of cyber-physical systems[C]//Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops. Seoul Republic of: Association for Computing Machinery, 2020: 258–265.

[10] Yin Y Q, Xu X R, Wang D J, et al. Two-stage recoverable robust optimization for an integrated location-allocation and evacuation planning problem[J]. Transportation Research Part B: Methodological, 2024, 182: 102906.

[11] Mori M, Ceccarelli A, Lollini P, et al. Systems-of-systems modeling using a comprehensive viewpoint-based SysML profile[J]. Journal of Software: Evolution and Process, 2018, 30(3): e1878.

[12] 王俊龙, 楼京俊, 阮旻智, 等. 面向任务的舰船装备备件利用率计算模型 [J]. 系统工程与电子技术, 2023, 45(5): 1420−1428.

[13] 黄苾, 代飞, 王亚博, 等. 基于 Petri 网分析编排的可实现性 [J]. 计算机集成制造系统, 2020, 26(6): 1548−1556.

[14] 黄凤兰, 倪枫, 刘姜, 等. 基于 HCPN 的复杂 BPMN 协作模型数据流建模与验证 [J]. 计算机集成制造系统, 2024, 30(5): 1754−1769.

[15] Kang B G, Seo K M, Kim T G. Communication analysis of network-centric warfare via transformation of system of systems model into integrated system model using neural network[J]. Complexity, 2018, 2018(1): 6201356.

[16] Robinson S. Simulation: the practice of model development and use[M]. 2nd ed. London: Palgrave Macmillan, 2014.

[17] Guariniello C, Fang Z M, Davendralingam N, et al. Tool suite to support model based systems engineering-enabled system-of-systems analysis[C]//Proceedings of the 2018 IEEE Aerospace Conference. Big Sky: IEEE, 2018: 1–16.

[18] Xia X K, Wu J, Liu C, et al. A model-driven approach for evaluating system of systems[C]//Proceedings of the 2013 18th International Conference on Engineering of Complex Computer Systems. Singapore: IEEE, 2013: 56–64.

[19] Hayano A, Takai S. A general architecture for intersection-based decentralized supervisory control of discrete event systems[J]. IEEE Transactions on Automatic Control, 2024, 69(1): 674−680.

[20] Tao X Y, Liu G J, Yang B, et al. Workflow nets with tables and their soundness[J]. IEEE Transactions on Industrial Informatics, 2020, 16(3): 1503−1515.

[21] Huff J, Medal H, Griendling K. A model-based systems engineering approach to critical infrastructure vulnerability assessment and decision analysis[J]. Systems Engineering, 2019, 22(2): 114−133.

[22] Fang Z M, Zhou X Z, Song A N. Architectural models enabled dynamic optimization for system-of-systems evolution[J]. Complexity, 2020, 2020(1): 7534819.

[23] Garaffa L C, Basso M, Konzen A A, et al. Reinforcement learning for mobile robotics exploration: a survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(8): 3796−3810.

[24] Xin J, Zhao H, Liu D, et al. Application of deep reinforcement learning in mobile robot path planning[C]//Proceedings of the 2017 Chinese Automation Congress. Jinan: IEEE, 2017: 7112–7116.

（编辑：丁红艺）