基于深度强化学习的救护车动态重定位调度研究①

刘冠男[1]，曲金铭[1]，李小琳[2*]，吴俊杰[1]
(1. 北京航空航天大学经济管理学院，北京 100191；2. 南京大学商学院，南京 210093)

摘要：救护车是挽救患者生命的重要医疗资源，合理调配有限的救护车资源可以降低呼叫响应时间，提高医疗服务水平. 本文面向救护车动态重定位调度问题，提出了一种基于强化学习的调度策略结构. 为解决传统强化学习所面临的高维状态空间的挑战，本文基于深度 Q 值网络(DQN)方法，提出了一种考虑多种调度交互因子的算法 RedCon-DQN，以在给定环境状态下得到最优的重定位调度策略. 在此基础上，本文还提出了急救网络弹性概念，以评估各站点对全局救护优化目标的影响力. 最后，基于南京市 2016 年～2017 年的实际救护车呼叫及响应数据，构造了环境交互模拟器. 在模拟器中通过大规模数据实验，验证了模型得到的调度策略相比已有方法的优越性，并分析了不同时段下调度策略的有效性及其特点.

关键词：强化学习；DQN；救护车调度；重定位
中图分类号：TP18 **文献标识码**：A **文章编号**：1007－9807(2020)02－0038－15

0 引 言

救护车作为一种重要的医疗资源，守护着患者的生命线，提前 1min 的应急响应就可能挽救更多生命. 然而，囿于我国医疗资源的稀缺性，无论是在地理和资源相对匮乏的偏远地区，还是在供不应求的城市枢纽地区，都面临着不同程度的急救响应不及时、救护车响应范围受限等问题. 事实上，发达国家一般要求市内的救护车响应时间低于 10min，但我国目前的救护车响应时间远高于此标准，例如福建省 2015 年日平均响应时间为 28.32min/次[1]. 而提高院前急救的响应速度，除了增添应急救护资源如新建急救网点、购进救护车等长期基础建设的方法之外，高效率地使用已有应急资源是更加节约有效的选择. 因此，在限定已有应急资源的约束条件下，如何进行救护车资源的调配整合，成为一个重要的研究问题.

基于我国地域广、行政范围跨度大的特点，国内普遍的院前急救的调度方式是以行政范围设立一个或多个集中调度急救中心. 以南京市为例，南京市内目前急救中心网络内设有 60 个急救站点(其中 7 个自管站、35 个分站)，形成了以“急救中心为基础、分站为骨干”的“独立院前急救型”模式. 它依靠医院建立多个分站，构建辐射范围足够的急救网络，以实现统一调度、分散救治，但同时也限制了救护车在结束应答之后必须返回原站点. 目前实际应用的调度策略主要采取的是“就近原则”，即利用离呼叫发出地点最近站点派送救护车进行响应. 然而，呼叫需求的产生是分散且不确定的，因此仅依靠就近原则的响应方式，往往只能响应有限的需求；而当产生新的用车需求时，只能从更远的站点派车，这就大大地增加了响应时间. 特别地，站点之间通常供需不均衡，这也导致了各站点间的响应水平差异较大，影响到整体的应急响应时间和效率. 由此可见，现行的救护车

① 收稿日期：2019－06－04；修订日期：2019－12－21.
基金项目：国家自然科学基金资助项目(71701007；71531001；71490723；71725002；U1636210).
通讯作者：李小琳(1978—)，女，吉林长春人，博士，副教授. Email：lixl@nju.edu.cn

调度响应往往缺乏全局、动态的视角,因而仍存在大量可优化空间. 近年来,研究领域也开始探索动态情境下的救护车调度策略,提出了一系列实时优化的方法来设计派车、站点之间的车辆调配等问题[2-6].

从我国医疗救护的现实情况来看,医院站点之间进行救护车的动态重定位(redeployment) 是一种现实可行的、操作相对简单且预期能带来显著效果的急救调度方式. 因此,本研究试图针对站点之间供需不均衡的问题,在有限救护车数量、限定调度范围、限定时序优先顺序的约束条件下,设计救护车站点之间的调度策略以提升救护车整体的响应效率. 考虑到救护车调度中动态时变的环境以及面向动态状态下的调度寻优目标,该问题可以被形式化为一种强化学习(reinforcement learning) 结构,即在动态环境(environment) 与动作(action) 的交互过程获取奖励(reward) ,并以最优化全局奖励为目标学习产生动作策略. 然而,针对现实场景中的救护车重定位调度问题,其状态空间及对应的动作空间组合较大,会面临较高的时空复杂度的严峻挑战. 而在面向全局的实时动态调度需求时,要实现优化目标则更加困难. 为解决这种高维度状态空间下的奖励值函数估计问题,深度强化学习在近年来得到了巨大的发展,并被应用到了各类实际的动态决策问题中. 其基本的思路是利用深度神经网络来估计强化学习中的奖励值函数,从而建立状态、动作与奖励值之间的映射关系.

本文基于 Deep Q-Network(DQN) 的深度强化学习方法,以最小化救护车平均响应时间为目标,根据各时刻的环境状态进行站点之间救护车的动态重定位调度. 为规避低效甚至无效的调度,本文扩展了传统的 DQN 算法,提出了一种考虑调度交互因子的算法 RedCon-DQN,以在决策过程中考虑一些外部环境对调度智能体之间信息交互等影响. 此外,应注意不同站点承载的应急能力有所不同; 特别是由于各站点之间的供需不均衡,会导致某些站点成为救护车全局响应时间优化的瓶颈. 然而,目前相关文献中尚缺乏对站点响应能力的评价测度. 为此,基于交通弹性网络理论[7-11],本文提出了急救网络弹性测度,以评价各救护站点的响应能力,从而帮助识别全局救护响应的瓶颈,为应急响应资源的调配等问题提供决策依据. 在此基础上,本文利用南京市 2016 年 ~ 2017 年救护呼叫及响应数据构造了环境交互模拟器,并通过大规模实验验证了提出的调度算法的有效性,并分析了其在不同时间段的表现. 同时,对各救护站点的急救网络弹性进行度量,分析了典型的瓶颈站点,为未来的站点选址、资源分配等管理问题提供了决策依据.

1 文献综述

1.1 救护车调度问题研究

救护车调度问题是一个重要的研究问题,研究者在该领域进行了大量的探索. 其中一种典型的调度策略是静态视角下的救护车站点分配,将有限的救护车辆固定分配到各个站点上,从而尽可能覆盖更多的救护需求,即 MEXCLP 问题[12],一般而言可利用 Lookup tables 进行求解[13]; Lee[14] 从复杂网络研究的中心性原则出发,提出了一种适用于各种苛刻的紧急状况(例如灾难等) 的救护车调度策略,王付宇等[15] 以带三角函数变异的离散型萤火虫优化算法解决震后伤员救援车辆两阶段优化问题. 但显然,这种静态策略无法适应环境和需求的动态变化. 因而近年来研究也开始着重关注救护车的动态调度策略问题. 在动态环境下,救护车并不固定依附于某个特定站点,而采用重定位(redeployment) 、重定向(relocation) 等策略进行救护车的调度[2-4]. Zhang 等[2] 将该问题视为马尔科夫决策,在模拟系统随机性的前提下进行动态调度,Jagtenberg 等[3] 利用启发式算法进行了动态救护车重定位问题的研究,Barneveld 基于改进的 MEXPREP,并使用 Compliance table 方法来进行救护车重定向问题的研究[4]; Maxwell 等[5] 和 Schmid[6] 等针对实时调度问题的复杂性,提出利用 Approximate Decision Process(ADP) 来优化调度策略; Gendreau 等提出了一个动态模型用以解决实时重定向问题[16],还提出了整数线性规划模型来解决 MEXCLP 重定

位问题[17]. 此外,学者考虑伤情优先级的角度来进行救援路径的优化[18-20],并且考虑急救站的紧急程度来进行数据驱动的动态调度[21],还有学者针对不同的优化目标,如最大化在一定时间阈值内进行救援的比例[5]、最大化期望覆盖面积[17]等进行了研究. 总体而言,现有的救护车调度研究中,主要采用的是基于单一目标的优化方法. 针对动态规划下动作空间维度较高的特点,有学者提出了以近似动态规划法、利用静态规划和计算边界进行动态部署等方法来解决计算复杂度问题[22-23],但总体计算效率仍然较低. 因此,这类方法对于高维复杂状态空间的动态调度问题存在一定的局限,未能根据动作与环境交互的结果进行调度策略的学习; 同时,由于动作空间的高维特征,对于优化目标的估计也面临较大的挑战.

1.2　基于深度强化学习的应用研究

强化学习是一种无监督学习方法,能够通过和环境的交互来自我学习和更新,也即一种不断试错学习、通过得到的评价性信息来不断修正自己的行为的机器学习算法. 而深度强化学习方法是将深度学习和强化学习方法进行结合的算法,通过神经网络结构的引入,可以实现动作的直接输出. 2013 年 DeepMind 公司的 Mnih 等[24]开创性地提出了 DQN(Deep Q-Network) 之后,深度强化学习迅速成为了研究热点. DQN 通过引入经验回放机制,直接接收高维输入值并进行学习,使得算法能够在各种视频游戏的表现上超过人类. 紧随其后,研究者提出了大量基于 DQN 的改进算法,包括了基于策略的算法(DPPO[25]、DDPG[26]),基于奖励值和策略的算法(A2C、A3C[27]) 等. 与此同时,关于强化学习的研究也从单智能体向多智能体扩展,即多智能体系统(multi-agent system,MAS) . 多智能体改变了以往由单个智能行为对象与环境交互改变状态信息的情况,演变为多个智能行为体对象共同与环境交互,并且互相影响的系统[28],而同时多智能体之间的协调也成为一种学习问题[29]. 实际上,已有研究在考虑多智能体协同时基于了监督学习的方式,提出了宏观策略上的通信方式[30]. 在这种方式下,可以将监督学习下的其他智能体的标签作为目标智能体的特征输入,但这一方法不适用于无监督学习环境下的无标签协同问题.

随着深度强化学习的广泛应用,也有研究者将强化学习应用于调度等管理问题中. 例如,Lin 等[31]以收益最大化为目标,考虑实时动态的城市出行用车需求,基于 DQN 设计了对共享出租车平台(如滴滴出行) 的车辆重定位策略. Wang 等[32]提出将深度强化学习于有监督学习方法相结合,并利用深度循环神经网络(RNN) 来进行诊疗方案的动态推荐. Wei 等[33]利用深度强化学习来训练交通信号灯的控制策略,以期降低排队和拥堵,并提供策略的解释性. Ardi 等[34]进行了多智能体和深度强化学习的结合使用,用于模拟游戏中不同智能体对同一物体的共同操作. 这种基于多智能体强化学习结构适用于考虑协同合作的调度问题,而其中的挑战和焦点则在于如何高效地建立多智能体之间的信息沟通,从而更高效的实现调度优化目标. 特别是面向实际的救护调度的限制条件和环境约束,更加需要设计特定的结合环境因素的智能体间的状态交互,以实现高效的调度.

2　救护车的动态重定位调度问题

如前所述,制约救护车资源使用不均衡的原因之一在于目前的调度策略往往没有提前进行动态的负载平衡,也就是没有根据实际的动态需求在各站点进行合理有效的车辆调度,以实现总体响应时间的优化. 换言之,如果能将救护车辆资源提前调配至需求旺盛区域,可以能更好地响应呼叫需求,即将闲置救护车在不同站点之间进行重定位(redeployment) . 这种调度方式的优点在于,系统可以根据当前的环境需求状态,在各站点之间进行闲置救护车的重新部署,将救护车从潜在的低需求站点向高需求站点,从而保证高需求站点附近的呼叫需求能被快速响应,实现总体响应时间的降低. 这种调度策略是在事前发生的,一旦完成调度就能迅速和最近的呼叫需求进行匹配,从而降低响应时间.

因此,有必要根据动态的环境状态来进行调度策略的寻优,例如根据当前站点的救护车辆数,

以及近期周边的呼叫需求量,来综合决定是否需要向该站点增派车辆,以及从哪个站点进行车辆的重定位调度. 然而,需求总是处于动态变化之中,难以准确预测特定地区的急救需求,这使得基于实际需求状态进行调度也更加困难. 为此,本文将救护车重新部署调度问题形式化为一个强化学习问题. 具体而言,救护车的重定位调度的强化学习可以被定义为: 给定集中调度中心的信息平台上的呼叫信息、车辆状态、所属站点等信息,通过救护车实时的重新部署调度,实现地区救护车平均响应时间的最优化. 其中,反映响应时间全局最优化的通常包括三个指标: 呼叫响应率、全局平均响应时间和黄金时间比例. 具体而言,本文研究的重定位调度策略即为: 在时刻 t,从站点 X 向站点 Y 调配救护车.

为了简化起见,本研究以一天为周期,将 10min 作为调度周期,从而可以将全天 24h 划分为 $T=144$ 个时间区间. 在每个时间区间内,调度中心会根据各站点实时的状态信息和呼叫信息,来对车辆进行调度动作,调度动作会在各站点之间的救护车进行重新部署和调动. 与此同时,按照近邻原则和先到先响应的原则,对车辆和呼叫订单之间进行匹配,而实际的急救响应时间则为救护车出发至急救点,最后回到站点的所有时间. 问题的本质即是决定在当前时间点每个医院分站应当有多少辆救护车,能够最大限度的第一时间响应呼叫,将呼叫响应的时间最小化.

本文涉及到的呼叫响应及救护车辆调度时间线如图 1 所示. 值得注意的是,本文所研究的重定位的调度动作并不是由一个特定的急救事件触发的,而是根据系统当前各个站点的救护车数量和需求情况实时进行的; 换言之,系统按照一定的调度周期来进行调度动作,而不是依赖于某一特定的急救事件. 在本模型中,救护的响应时间是指救护车从站点出发到急救地点,最后回到站点从而完成整个救护过程的时间,不包含 120 应答与车辆分配的时间. 此外,由于救护车性质的特殊性,结合现实情况,本文假定救护车在完成所在急救站点的呼叫应答后,通常会返回原站点. 因此不考虑车到人不走、车到其他医院等特殊情况,同时忽略救护车损坏等随机情况.

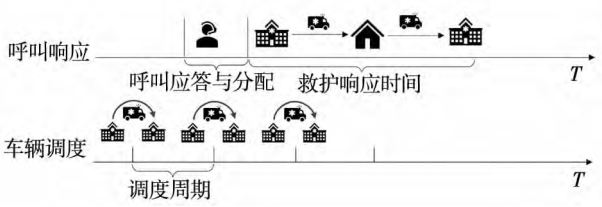


图 1 呼叫响应及车辆调度时间线

Fig. 1 The timeline of emergency response and ambulance scheduling

如上所述,本文使用 N 个智能体参与的马尔科夫决策过程 $G(N,S,A,R,P,\gamma)$ 来描述救护车调度问题,其中的参数 N 为救护车数量,S 为状态空间,A 为策略空间,R 为执行动作后的奖励函数值,P 为状态转移概率,γ 表示远期奖励值在当期的折扣率. 具体来说,问题 G 中的主要元素的定义如下.

1) 智能体(agent)

医院急救站点中的可用救护车,即一个在线的、不在应答呼叫过程中的救护车是本文认为的可调度的智能体,其中在应答过程中的救护车无法响应急救中心的调度. 对一个站点在选择响应急救呼叫的救护车的方式具有随机性,因此假设同一个时间节点在同一个医院站点的救护车完全同质,对同质救护车的调度均可被认为是相同的调度策略. 在该问题中总智能体数量恒定为 N,但是可被调度的数量 N_t 是随着时间 t 不断变化的.

2) 状态空间(state)

各时刻 t 均具有一个全局状态 $s_t \in S$,对单个救护车智能体而言,其状态信息包含了所处医院站点 g_j,所在站点在该时刻的呼叫数量 m_{jt},及所在站点可用救护车数量 n_{jt}. 值得注意的是,相同站点下具有同质性的智能体具有相同的局部状态信息,而每条状态信息中的离散特征如时间、医院站点编号等均采取独热编码(one-hot encoding) 进行转换.

3) 动作空间(action)

本研究针对救护车所采取的调度动作包括: 救护车保留在原站点或重定位到 K 个近邻站点. 因此对于一辆可调度的救护车来说,其状态空间大小为 $K+1$. 在 t 时刻,针对各站点的救护车,可采取的动作为 $a_t \in A = A_1 \cup A_2 \cup \cdots \cup A_{N_t}$. 为简

化起见,假设调度动作在 t 时间点上立刻发生,不考虑重定位调度动作的时间成本,但考虑非时间成本 c; 因此,重定位的目标站点选取需考虑到站点之间的距离因素,本研究取 $K = 4$,针对每个救护车的动作空间大小都为 5,数值范围在 0 – 4 之间. 其中 $a_t = 0$ 代表救护车留在当前医院站点,后四个数值分别代表调度到最近邻的四个医院,例如 $a_t = 3$ 代表转移到所在医院站点的第三近邻医院站点.

4) 奖励(reward)

强化学习中奖励信息是由状态和动作共同决定的,即对一个智能体采取特定的动作后可以获取的奖励为 $r_i \in R = S \times A$. 由上文关于同质救护车和共享的局部状态的描述可知,转移到相同医院站点的救护车享有相同的奖励. 基于全局救护车平均响应时间最小化的目标,将救护车 i 在 t 时刻所在的医院站点的平均应答时间 r_t^i 设置为奖励值. 与常规的奖励值不同的是,这里的目标是最小化奖励函数. 因此,每个救护车的目标在于最小化自己在折扣率 γ 下的期望奖励 $[\sum_{l=0} \gamma^l r_{t+l}^i]$. 由于 t 时刻的奖励值取决于救护车所处的医院站点,因此 r_t^i 也可以表述为 $r_t(g_j)$. 为了避免单智能体极端追求各自的目标最优化,这里奖励函数设置为医院急救站点的平均响应时间,而非救护车的平均响应时间. 同时,在实际的救护车调度中,每次调度动作会带来额外的非时间成本,例如救护车自身折旧、出现故障概率以及车辆配备工作人员等因素. 同时,由于重定位调度的距离较短,每次调度产生的成本可以认为相同. 因此,在每次进行调度时,引入调度成本 c,于是奖励值可以表示为 $r_t(g_j) + c$.

基于以上关于救护车重定位问题的强化学习的结构定义,该问题可以被认为是一辆救护车智能体在动态变化的呼叫需求环境中,在获得环境状态信息后,执行了某种动作与环境交互,环境受到动作影响并返回对智能体的奖励和下一个环境信息,从而构成一个完整的单步迭代强化学习,如图 2 所示.

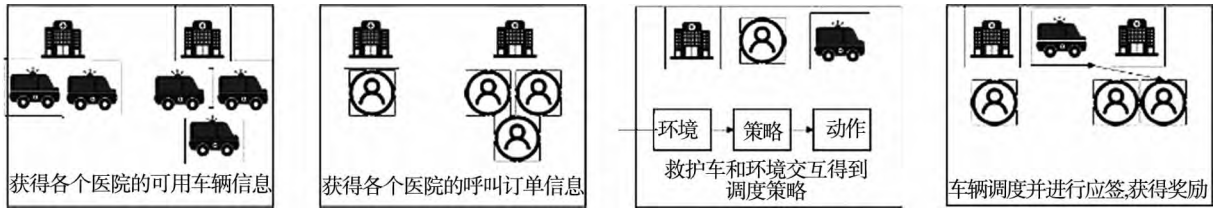


图 2　单步迭代过程

Fig. 2 The iterative process in single step

3　面向重定位调度的交互性多智能体强化学习算法

3.1　Deep Q-value Network(DQN) 算法

Q-learning 是一种基于时序差分学习的强化学习算法[37],其基本思想是根据当前环境状态执行 ε – 贪心策略,执行动作得到新的状态奖励函数值,进而利用动态规划思想,通过后继状态来更新 Q 函数值.

在动作数量和状态数量都比较低维的情况下,Q 值列表的计算和更新都是比较容易的. 但是当两者之一的维度空间较高时,计算的时间与空间需求都会迅速增长,导致 Q 值的更新的复杂度较高. 对于救护车重定位调度问题,每个站点的救护车数目都处于动态变化中,状态空间极大; 用传统的 Q 值表难以维护状态和对应的动作. 有鉴于此,本文采用 Deep Q-network (DQN) 算法来进行策略学习. DQN 是一种将 Q-learning 算法和深度神经网络有效结合起来的一种强化学习算法,同样基于时序差分学习的思路,并利用深度神经网络来建立状态动作与奖励,即 Q 函数值(Q-value) 之间的映射函数关系. 一般而言,Q-network 第一层输入网络输入节点代表的是状态值向量,中间层节点代表神经网络的隐藏层,输出节点代表对应的 Q 值. 在训练 DQN 时,通过执行动作与环境交互得到奖励函数 Q 值作为标签,并记录在经验回放池(experience replay) 中; 利用抽样产生批

量的训练样本进行 Q 网络的参数估计. 具体对于单个救护车智能体来说,其优化的目标损失函数如下

$$\underset{s_t^i,a_t^i s_{t+1}^{i+1},a_{t+1}^{i+1}}{E}\left[Q(s_t^i,a_t^i;\theta) - (r_t^i + 1 + \gamma \max_{a_t^{i+1}} Q(s_{t+1}^i,a_{t+1}^i;\theta')) \right]^2 \quad (1)$$

其中 θ 和 θ' 分别代表实际 Q 网络和目标 Q 网络(target Q-network)的参数. 在本文的问题当中,每个救护车智能体也可以按照上述思想进行独立调度,但调度的预期奖励往往会和实际得到奖励产生差距,这是因为全局状态会受到所有救护车的联合调度动作影响,奖励会与针对单个救护车进行重定位动作的期望奖励值有所不同.

3.2 考虑重定位调度策略的交互 DQN(RedCon-DQN)算法

基于第 2 节中提到的救护车同质性的条件和概念,将同质的概念迁移到 Q 值奖励函数上

$$Q(s_t^i,a_t^i) = Q(s_t,g_d) \quad (2)$$

式(2)是在说明同时间点、同医院站点的 Q 值是相同的,这背后呈现了同质救护车和同质奖励的理念. 这也解释了在基于 Q 值选择动作的本算法中,为什么同质救护车的动作选择相同. 因此,式(1)可以调整为

$$\left[Q(s_t,g_d;\theta) - (r_{t+1}(g_d) + \gamma \max_{g_p \in Ner(g_d)} Q(s_{t+1},g_p;\theta')) \right]^2 \quad (3)$$

其中 $Ner(g_d)$ 表示站点 g_d 附近的近邻站点集合. 通过这样的设计,动作和状态的组合空间大小得到降低,只需要考虑按照医院站点数来计算即可. 但是,对于救护车重定位调度的优化目标来说,考虑到调度过程中站点之间调度的距离,救护车的可用性,以及多辆救护车之间的协调等实际因素,还需要进一步补充相关的约束条件,避免低效、冗余的调度策略. 为此,需要对传统的 DQN 算法进行扩展,在奖励函数中融合考虑如下因子.

1) 地理交互因子

有效的重定向调度是指可以在有限的时间内将救护车在不同的急救站点之间进行重定位. 因此认为,在设定的最近邻站点中超出一定距离的站点应被视作无效站点,无法进行调度动作. 这是由于调度时间过长,会与降低全局平均应答时间的最终目标相悖. 根据实际情况设定距离阈值为 10km. 基于此提出了地理交互因子 G_{gj},其定义如下

$$[G_{gj}]_k = \begin{cases} 1, & \text{若调度到 } g_d \text{ 是有效调度} \\ 0, & \text{其他} \end{cases} \quad (4)$$

其中 g_d 是位于 g_j 中的救护车采取第 k 个动作所转移到的医院站点,该因子是可以预先加载和记录的. 显然,对于留在本地的动作不需要考虑距离的问题,对应的交互因子恒为 1; 而其余调度动作的地理交互因子是需要根据站点之间的距离进行计算.

2) 动作交互因子

对于动作空间上的动作来说,调度对于任何救护车都应该是可行的,但是救护车应答呼叫的时间往往要长于时间区间(10min),所以对于正在应答呼叫的救护车而言,在它们的应答行为完成之前,它们的目的地都是原医院站点,所以应当取消它们在当前时间的调度资格. 由此引入动作交互因子 O^i,其定义如下

$$O_t^i = \begin{cases} 1, & \text{若救护车 } i \text{ 不在应答状态} \\ 0, & \text{其他} \end{cases} \quad (5)$$

该动作因子是针对救护车的实时更新的是否能够进行调度的信号,当救护车在线不处于应答状态的时候,对车辆进行调度,否则将拒绝进行调度,默认动作为留在原有医院站点.

3) 信息交互因子

对于调度的目标来说有正面效果的动作才是有效动作,因为涉及到多辆救护车的动作调度,因此需要设计多智能体信息交互的渠道,来保证调度行为的有效性. 为此,引入了信息交互因子 C_{t,g_j},其定义如下

$$[C_{t,g_j}]_k = \begin{cases} 1, & \text{若 } Q(s_t,g_i) \geqslant Q(s_t,g_j) \\ 0, & \text{其他} \end{cases} \quad (6)$$

其中 g_i 是位于 g_i 中的救护车采取第 k 个动作所转移到的医院站点. 信息交互因子的作用在于取消两个医院站点之间互相调度的行为,比如 H01 医院站点的 089 号救护车调度到 H02,同时 H02 的

090 号救护车调度到 H01,这样的调度行为是被信息交互因子所限制的,因为从全局优化的角度上来看,对于医院可用车辆数量没有改变,所以它是无效的调度. 该因子能够让它限制双边调度,而要求调度策略体现全局一致性,即单向调度. 在信息交互因子的限制下,同一站点的可调度救护车会产生一致动作,即向使得 Q 值最优的临近站点进行调度; 同时,不同调度路程会引起调度时间成本的不同,因此本文对调度距离进行了限制,即仅在范围10km的距离限制进行车辆的调度,因此可以认为,从不同距离的两个临近医院点向同一站点的调度行为是等价的.

算法 1　RedCon-DQN 算法

Algorithm 1 RedCon-DQN

联合动作获取:
1. 输入全局状态 s_t
2. 计算 $Q(s_t, g_j)$
3. for 1,2,…,N_t do
4.　通过式(2) 计算 Q^i
5.　计算交互因子 $G_{gj}, O_t^i, C_{t,gj}$
6.　通过式(7) 计算有效 Q 值
7.　通过 ε - 贪心策略返回动作值
8. end for
9. 返回 joint_action{ }

DQN 迭代
1. 初始化经验容器和预训练网络参数值
2. for *iter* = 1,2,…,max_interation do
3.　for t = 1,…, T do
4.　指定 S_t 通过联合动作获取得到 joint_action
5.　joint_action 输入到环境模拟器当中与环境交互得到每次救护车调度的 r_t 和 s_{t+1}
6.　将所有 agent 的交互信息保存到经验回放容器
7.　end for
8.　for 1,2,…,*M*1 do
9.　获取容器中存放的经验交互信息
10.　计算 target Q-network
11.　利用随机梯度下降算法更新实际 Q-network 的参数
12.　end for
13. end for

综合以上三类交互因子,调度策略优化的有效奖励值可以重新被定义为

$$q(s_t^i) = Q(s_\tau^i) \times G_{g_j} \times O_t^i \times C_{t,gj} \tag{7}$$

基于以上更新后的 Q 值进行的动作选择能够保证在全局视角下,同质的救护车采取的行为是严格一致、单调的. 这在单智能体的 DQN 算法里是难以达成的. 也是 DQN 算法适用性改进以用于救护车集中调度的关键所在,即智能体信息交互的关键通道, 算法 1 展示了考虑重定位调度因子的 DQN (Redeployment Contextual DQN, RedCon-DQN) 算法.

3.3　急救网络弹性测度

网络弹性被定义为网络在提供可接受的服务水平的同时,能够适应、吸收、预测并迅速从导致链路关闭、节点关闭和容量降低的破坏性事件中恢复的能力[2-6]. 基于以上定义,本文提出急救网络弹性测度,用以评价各急救站点响应水平. 具体来说,可以定义为由于应急事件发生导致的医院站点突然从调度中心的急救网络下线的事件对急救网络的冲击的回复能力. 因此,判断一个站点下线对急救网络的影响能力,只需要通过计算剩余站点的响应时间相比正常情况的延迟时间即可. 具体而言,假设急救网络中有 M 个站点 $H_1, H_2, \cdots, H_M$,在对站点 H_i 的急救网络弹性进行评价时,则在调度中取消该站点及原有附属的急救车辆,利用 RedCon-DQN 算法下进行调度并对所有需求呼叫进行响应,得到新的平均响应时间,并计算与原有情况下的平均响应时间取差值,因此 $D(H_i)$ 定义如下

$$D(H_i) = \frac{\sum_{k \neq i} AT(H_k)}{M-1} - \frac{\sum_{k=i}^{M} AT(H_k)}{M} \tag{8}$$

其中 $AT(H_K)$ 是站点 H_K 的平均响应时间.

4　救护车重定位的环境交互模拟器构造

4.1　数据概况及统计描述

本文基于南京市2016年6月 ~ 2017年5月的真实急救呼叫及救护车响应数据进行环境交互模拟器的构建. 数据主要包括了实时呼叫订单流水号、救护车标号、实时位置经纬度、实时方向、实时速度等信息. 本文只考虑南京市范围内的应答信息,不考虑有出入境的特殊情况; 只考虑正常情况下的应答情况,不考虑超长时长和超长距离的呼叫信息.

通过将南京市三级及以上医院和急救中心信息②与真实急救呼叫数据中的医院站点进行比对,并经人工校对,确定了 37 所医院为实际的急救站点. 同时,由实际急救数据可知,南京市 2016 年 ~ 2017 年实际运营的救护车数量为 108 辆. 图 3 展示了南京市急救中心站点的分布情况. 而总体来说,全年的呼叫平均响应时间为 28.02min.

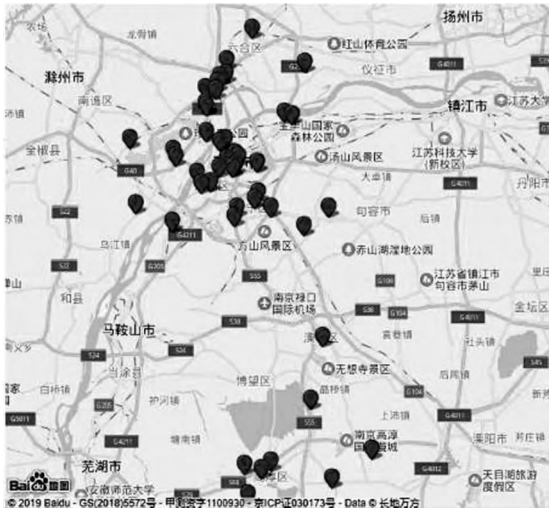


图 3　南京市急救中心官方分站点分布图

Fig. 3 The distribution of official emergency stations in the city of Nanjing

如图4 所示,以 1 天为周期,10min 为一个时间窗口,可以得到在 $T = 144$ 的时间划分中的呼叫数量分布. 可以发现,从 $T = 0$ 到 $T = 20$(0:00 ~ 3:20) 的时间区间内,呼叫数量一直处于下降趋势,一直到 $T = 30$(5:00) 左右降到最低点; 随后开始回升在 $T = 60$(10:00) 处达到峰值,在 $T = 60$ 到 $T = 80$(10:00 ~ 13:00) 之间呈现骤降,在 $T = 80$ 到 $T = 144$ 之间呈现缓慢而稳定的下降趋势. 南京市内救护车日均出车次数 379 次 / 天,也就是最终要控制日均订单生成数量近似于 379 次 / 天.

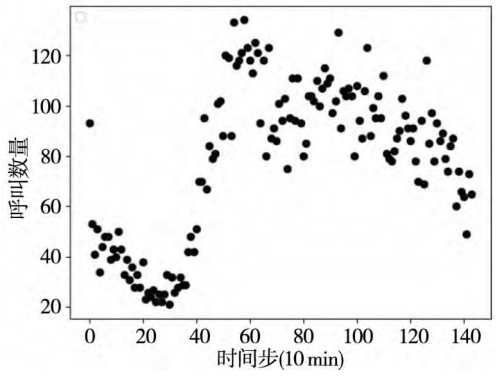


图 4　南京市呼叫数量的时间分布图

Fig. 4 The distribution of the emergency calls in Nanjing

4.2　呼叫数据拟合

在实际呼叫场景中,呼叫的时间、地点不受外界调度因素影响,因此在构造模拟器之前,本文首先根据历史呼叫订单数据从呼叫的时间分布、距离、车辆速度等方面进行拟合,从而得到符合现实调度场景的数据环境,拟合结果如图 5 所示.

4.2.1　呼叫距离拟合

基于实际呼叫信息,以及与响应站点的距离、响应时间等信息,所有呼叫距离分布如图 5(a) 所示,可以发现呼叫订单主要分布在20km 以下的区间范围内,整体可以用一个指数分布进行拟合.

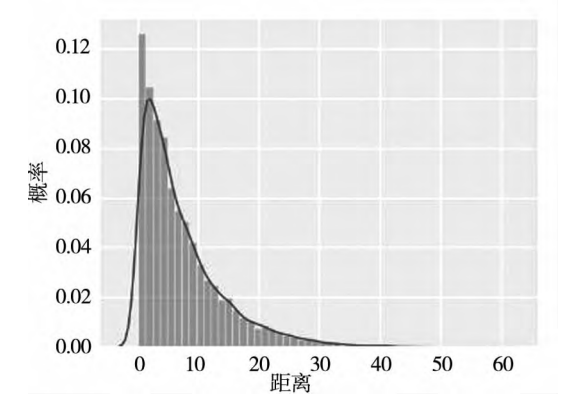
4.2.2　呼叫时间分布拟合

考虑到各个医院的实际日平均订单数量根据医院级别和地域有很大区别,在进行各医院站点的呼叫数量拟合时采取分医院拟合的方式,每个医院享有自己的分布参数. 在使用混合高斯模型对呼叫数量的拟合过程中,本文发现,当采用由 6 个高斯分布拟合的效果较好,拟合效果如图 5(b) 所示.

4.2.3　车辆平均行驶速度的分布计算

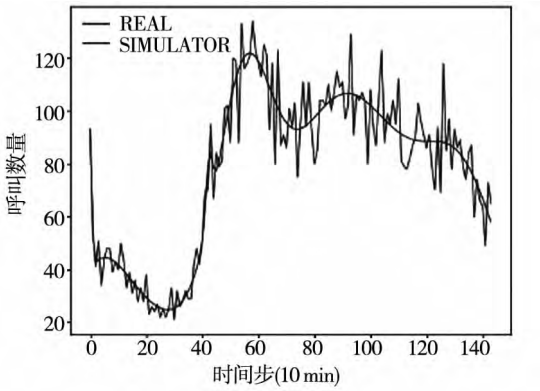
考虑急救呼叫的响应时间与距离之间的关系强烈依赖于车辆行驶速度的结论,对单位天时间分布上的车辆行驶平均速度进行计算,获得的车辆行驶平均速度的分布如图 5(c) 所示,并通过捕获不同时间段的平均速度来拟合不同时段的真实的急救应答时间. 从计算结果可以看出在呼叫订单的数量高峰时段 $T = 40$ 到 $T = 60$、$T = 80$ 到 $T = 120$ 之间,行驶速度产生了明显的下降,这与人们的认知相符合. 基于此,本文提出了“距离 - 速度”架构来对真实的呼叫和响应进行拟合.

② 数据来源: 南京市 2017 年卫生统计年鉴.



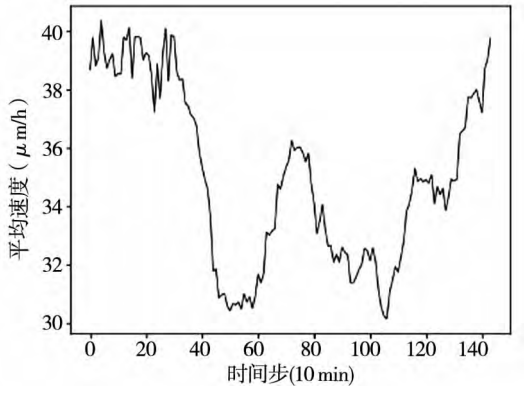
(a) 呼叫距离拟合

(a) Emergency distance fitting



(b) 呼叫时间拟合

(b) Emergency time fitting



(c) 救护车行驶速度

(c) Speed of ambulances

图 5　呼叫数据拟合

Fig. 5 The fitting of emergnecy data

4.2.4　拟合结果

在“距离 - 速度”的拟合框架下,用生成的订单的路径距离除以所在时间段的车辆行驶平均速度,以获得订单基础时间估计,再用拟合的全部订单数量时序数据和全部订单时间时序数据来对呼叫应答时间进行修正,从而获得更加精准的环境模拟器. 最终的拟合效果对比如图 6 和图 7 所示. 研究对环境交互模拟器的订单生成情况进行实际统计检验,呼叫数量分布拟合的 R^2 = 0.905 551, Pearson 相关系数为 0.951 605; 呼叫订单响应时间分布拟合 R^2 = 0.830 052, Pearson 相关系数为 0.911 082,可以认为呼叫数据的拟合情况比较符合实际,贴近现实中南京市急救呼叫订单的分布和响应状况.

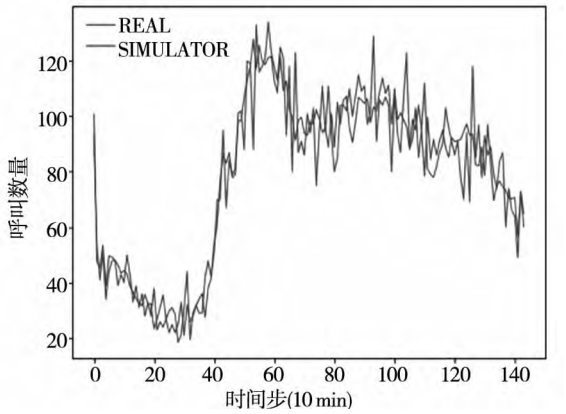


图 6　呼叫订单数量分布拟合

Fig. 6 Fitting for the number of emergency calls

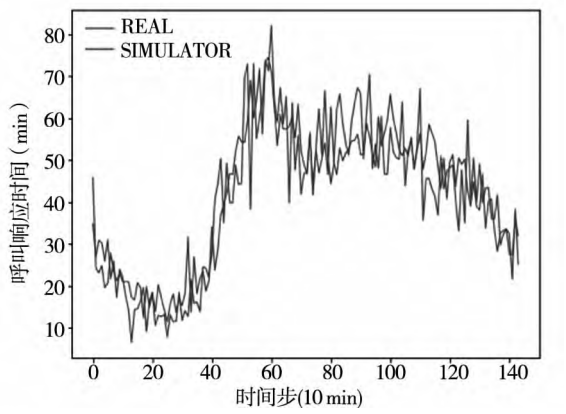


图 7　呼叫响应时间分布拟合

Fig. 7 Fitting for the response time of emergency calls

表 1　实验参数表

Table 1 Experimental parameters

迭代数	状态长度	动作空间数	学习率	激活函数	ε 随机数	gamma	Batch_size	Final loss
2 000	255	5	1e - 3	Tanh	0.9	0.95	50	0.014 801

4.3 环境交互功能设置

根据上述呼叫数量与时间、距离分布的拟合结果,可以生成呼叫订单,并基于此进行救护车站点之间的重定位调度与状态更新、呼叫订单的分配以及奖励值计算.

救护车调度与状态更新: 每一个时刻,模拟器生成呼叫数据(时间、位置等) ,并基于状态信息(各站点救护车可用数量、是否在途等) 进行车辆调度,进而更新救护车所属站点及各站点救护车数量等状态信息. 同时返回交互因子的计算信息,以获取联合动作.

呼叫订单分配: 当调度工作完成后,立刻开始呼叫订单和救护车的匹配应答工作. 同站点的所有救护车随机匹配订单,如果呼叫订单没有剩余,则结束匹配,返回奖励值,即医院站点的平均响应时间; 如果呼叫订单有剩余但最近站点无可用救护车,进入并按远近顺序遍历近邻医院进行应答匹配; 当遍历完成,仍未应答的订单会归入未响应队列,此时呼叫订单消除,被认为是无法响应的订单.

奖励值计算: 在状态更新时进行奖励值计算,即局部平均响应时间计算,当订单分配到近邻医院站点时,环境模拟器会更新订单时长,产生一个等待时间,用以惩罚订单承接不及时,这一部分的奖励会算在订单产生的医院节点当中,提高该地区的平均响应时间,用以激励救护车向本医院站点进行调度. 同时向全局更新每一步的全局时间,用以计算最终的评价指标,同时统计完成订单的情况.

5 实验结果

5.1 实验设置

在实验中,通过环境交互模拟器生成呼叫的时间、距离等数据,具体包括每个时刻上的呼叫订单信息,包括呼叫数量、每个订单到医院的距离、时长、所属医院. 每天的模拟订单数量平均为 384 次 / 天,接近真实数据值. 实验用 2 000 天的调度数据做平均响应时间和初次应答响应率的调度效果检验,保证数据和结果的稳定性与鲁棒性.

5.2 对比算法与评价指标

5.2.1 对比算法

在实验中将 RedCon-DQN 所得到调度策略同如下方法进行比较.

随机分配(Random) : 每个救护车采取随机动作分配到近邻医院站点,不与其他救护车进行任何交互.

基于规则的调度(Rule-based) : 各时刻对救护车所在站点以及近邻站点的响应率进行排序,将救护车调度到可调度范围内长期(即 5 个时间步长) 响应率低下的医院站点. 该方法即为现行南京市调度策略

Q-learning 算法[35] : 基础的 Q-learning 算法使用 ε - 贪心策略,当取消智能体的信息交互时,智能体获得的信息将被减少到时间节点和所在位置.

Sarsa 算法[36,37] : 基础的基于值迭代的 Sarsa 算法,其他设置与 Q-learning 算法相同.

DQN 算法[24] : 各智能体之间没有信息沟通,相当于分散独立的自我调度方式,采取的参数设置与表 1 相同.

5.2.2 评价指标

1) 全局平均响应时间: 全局呼叫的总响应时间除以全部应答数量,越低反映响应效率越高.

2) 初次应答响应率: 全局急救呼叫中由距离呼叫地最近的医院站点响应的比率.

3) 黄金时间比例: 完成的所有急救呼叫订单中,时长处于急救黄金时间阈值 20min 以下的订单数量占全部订单数量的比例.

5.3 调度策略实验结果

利用各算法策略在模拟环境中进行救护车的重定位调度,得到的响应结果如表 2 所示. 可以发现,本文所提出的 RedCon-DQN 算法表现最好,相比次优的 Sarsa 算法在平均响应时间上减少了约 2 min.

表 2　调度策略实验结果对比

Table 2 Comparison of experimental results of scheduling strategy

算法	平均响应时间(min)	初次响应率	黄金时间比例
随机动作	20.000 9 ± 5.992 6	0.584 6 ± 0.223 5	0.584 7 ± 0.169 4
基于规则	20.340 3 ± 5.997 8	0.435 4 ± 0.164 5	0.550 0 ± 0.158 4
Q-learning	19.912 3 ± 6.218 2	0.674 8 ± 0.236 6	0.602 4 ± 0.170 2
Sarsa	19.583 4 ± 6.234 9	0.718 5 ± 0.223 6	0.629 0 ± 0.175 9
DQN	19.468 7 ± 6.155 0	0.684 0 ± 0.209 7	0.603 0 ± 0.171 8
RedCon-DQN	17.629 9 ± 5.645 4	0.732 1 ± 0.220 2	0.715 8 ± 0.200 3

RedCon-DQN 算法相比传统 DQN 算法而言在平均响应时间和初次响应率上均有明显的提升,说明考虑信息的交互可以有效规避掉低效的调度,提升响应水平. 同时还可以发现,利用深度神经网络进行奖励值函数估计的 DQN 算法相比 Q-learning 接近并且也有一定的提升,说明引入的神经网络能准确估计奖励值函数,并且实现目标优化.

为了进一步分析 RedCon-DQN 调度策略的特点,在调度实验中分时段计算在不同调度策略下的全局平均响应时间. 如图 8 所示,RedCon-DQN 算法的调度优良性主要体现在5:00 ~ 16:00 的时间段内,说明了 RedCon-DQN 调度算法的优势集中体现为能够较好地应对高峰值的冲击,能够在救护需求旺盛时较好地实现不同救护站点之间的负载平衡. 表现次佳的Sarsa算法在时间线上的表现相对于 RedCon-DQN 来说比较不稳定,起伏波动比较大. 而随机动作算法、基于规则的算法、Q-learning 算法等没有体现出明显的时段特征,表现出了相对的不稳定性. 相对的不稳定性是由于策略选择的随机性以及智能体信息的不完备性所带来的波动性.

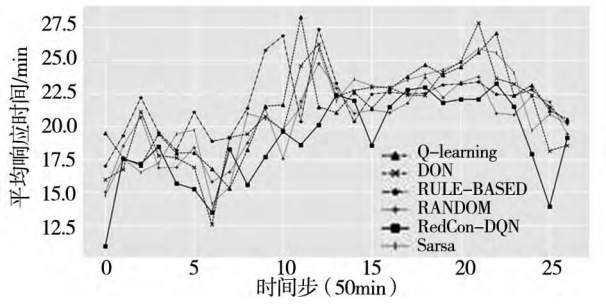


图 8　各调度算法下一天内的平均响应时间

Fig. 8 The average response time in one day

类似地,对于全天周期内的初次响应率而言,如图 9 所示,RedCon-DQN 除了在凌晨时段受到无订单分布的零响应率的影响造成的初次响应率的下跌之外,在全天内 RedCon-DQN 的表现都比较平稳. 但是与图 8 对比发现,在某些时间段初次响应率在一天内的变化趋势与平均响应时间的变化趋势正好相反. 这主要是由于算法的优化目标是全局的平均响应时间,因此当初次响应率下降时不会及时做出策略调整,但当对全局平均响应时间造成了负面影响时,算法才会进行策略的变化以改善平均响应时间,但同时可能又会对特定站点的初次响应率带来负面影响. 对于不稳定的其他算法而言,这种不平衡的状况始终在反复振动,没有达到均衡状态的表现. Sarsa 算法在初次响应率上的表现接近于 RedCon-DQN,远优于其他算法. 而其他算法波动幅度较大,说明调度策略无法保持一个稳定的状态.

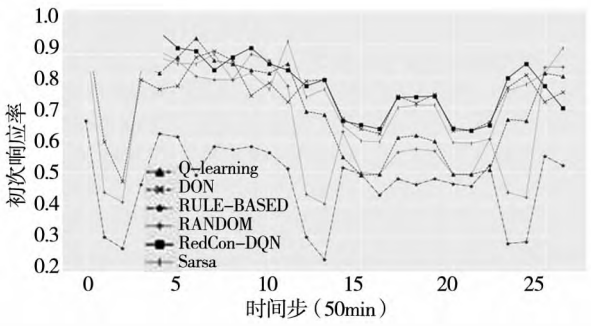


图 9　各调度算法下的一天周期内初次响应率

Fig. 9 The initial response rate in one day

此外,从图 10 上来看,RedCon-DQN 在黄金时间比例上的表现要明显优于其他算法,并且从波动性的角度上看,RedCon-DQN 比较平滑,因此得到 RedCon-DQN 在黄金时间比例这一评价指标上依然表现良好的结论.

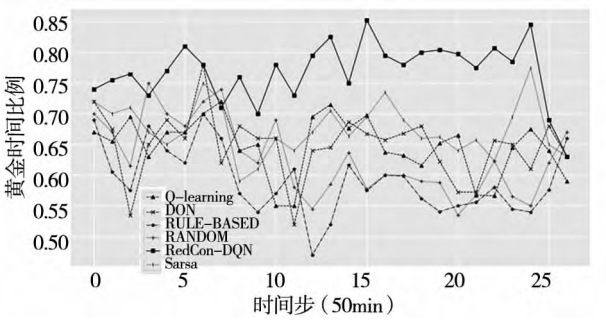
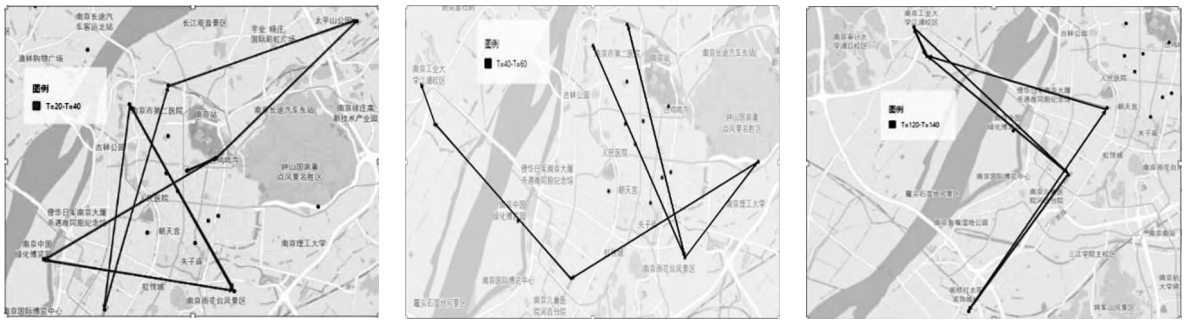


图 10　各调度算法下的一天周期内黄金时间比例

Fig. 10 The ratio of golden period of in one day

5.4　典型时间段调度策略分析

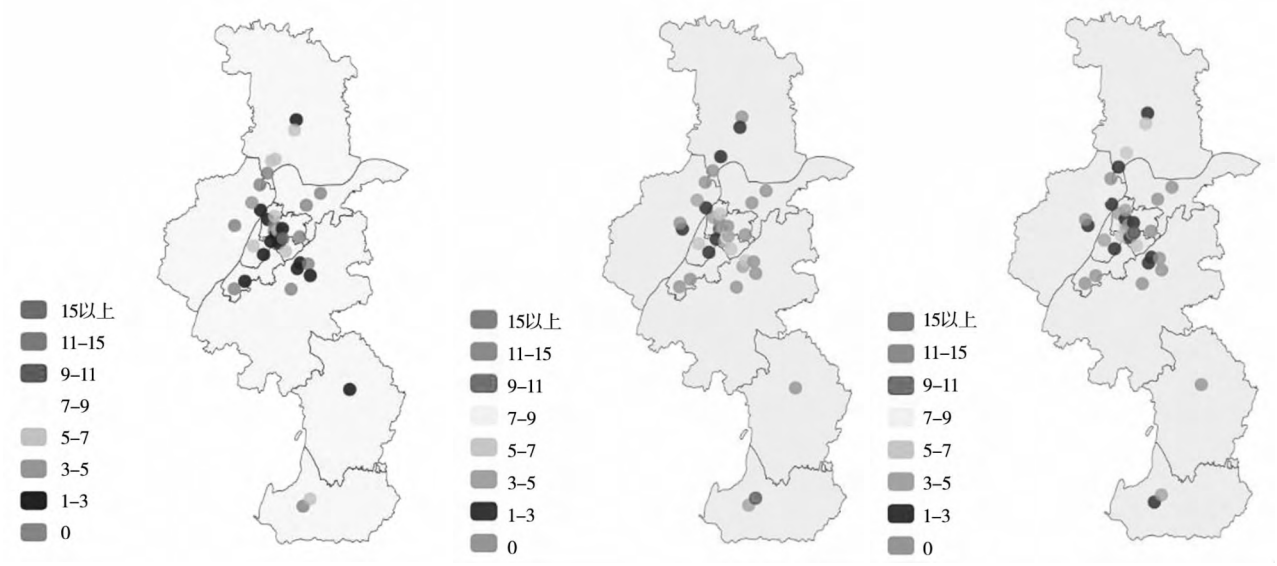
由调度实验结果可见，RedCon-DQN 在需求高峰时间段能够具有较好的响应表现. 为了能解释调度策略并理解重定位动作的实际意义，随机抽取一个救护车在 3 个关键时间段内的调度路径并绘制在地图上，如图 11 所示; 而对应地，在图 12 中，用不同颜色来表示各站点在对应时间段调度后的救护车数量. 从图 12(a) 中可以看到，在呼叫数量较少的时间段 3:00 ~ 6:00(T = 20 ~ 40) ，车辆的调度范围更大，需求分散; 而对应的救护车数量分布集中于有呼叫的站点，调动策略会使得车辆随时有可能向有呼叫需求的医院站点调度. 而在需求较高的时间段 6:00 ~10:00(T = 40 ~ 60) ，救护车的调动次数较多，但调动的地理范围有限; 而由图 12(b) 可知，救护车此时的分布在地理上也较为均匀，这主要是因为在该时间段需求相对集中，算法倾向于在小范围内进行救护车辆频繁调度来及时响应需求. 而在夜晚 T =120 ~ 140(20:00 ~ 23:00) ，救护车的调度越倾向于稳定和重复，而救护车分布较为分散.



(a) 3:00 ~ 6:00　　(b) 6:00 ~ 10:00　　(c) 20:00 ~ 23:00

图 11　某救护车辆在 RedCon-DQN 算法下在典型时间段的调度路径

Fig. 11 The scheduling path of an ambulance under RedCon-DQN algorithm in typical periods



(a) 3:00 ~ 6:00　　(b) 6:00 ~ 10:00　　(c) 20:00 ~ 23:00

图 12　RedCon-DQN 算法下典型时间段的救护车地理分布

Fig. 12 The geographic distribution of ambulances under RedCon-DQN algorithm in typical periods

5.5　急救网络弹性分析

根据 3.3 节中的式(8) ,在用 RedCon-DQN 进行调度之后可以计算各救护站点的急救网络弹性,其中弹性值最大的五个节点以及它们的测度(平均响应延迟时间) 如表 3 所示. 这五个医院节点分别是溧水区人民医院、南京市江宁医院、南京同仁医院、南京高淳区人民医院、南京栖霞区医院. 这五所医院的共同点在于都位于非市中心区域,但周边急救需求量很高. 这类医院地处偏远,缺乏足够近的近邻医院来及时完成救护车的动态调度. 因此,当这些站点附近产生较高的救护需求时,容易产生供不应求的状况,只能由距离较远的站点派车响应,导致响应时间加长,对全局平均响应时间仍然是负面影响. 所以从救护车调度中心的角度来看,应对此类站点的运营情况加以关注,通过建立更加密集的辅助站点,及时增派车辆,加强急救网络的健壮性.

表 3　急救网络弹性值最大的五个医院站点(min)

Table 3 The top 5 hospitals sites with the largest elastic value of emergency network (minutes)

医院站点	H24	H18	H33	H25	H16
弹性测度	7.365 9	7.212 3	7.005 8	6.785 6	6.312 6

为了探究弹性不同的各医院的特征表现,本文选取了弹性值最大和最小的站点进行比对,结果如图 13 所示. 结果显示弹性测度大的医院站点的救护车数量极差比较小,分布稳定; 弹性测度小的医院站点的救护车数量极差大,且允许站点救护车实时数量为 0,这是弹性测度小的医院站点可以依赖近邻站点的原因. 结果与本文的定义以及认知比较符合.

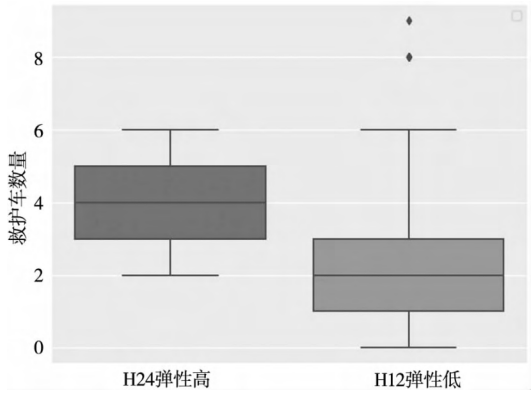


图 13　不同弹性的站点救护车数量浮动对比

Fig. 13 The comparison of the number of ambulances in stations with different elastic values

6　结束语

急救需求具有动态时变的特点,动态的救护车调度策略可以在有限的救护资源水平下实现应急响应水平的优化. 为此,本文研究了在动态需求环境下,对救护车在不同急救站点之间进行重定位,以最优化全局平均响应时间的问题,并提出了一种基于强化学习的结构. 在考虑多种实时调度交互因子的基础上,提出了一种改进的深度强化学习算法 RedCon-DQN. 借鉴交通网络弹性的概念,设计了一个考虑站点响应时延的急救网络弹性指标,度量急救站点对全局响应水平的影响.

基于南京市 2016 年 ~ 2017 年全年急救响应数据,构造了能准确拟合呼叫数量、地点分布的交互环境模拟器. 在模拟器生成的数据中利用不同的调度策略进行救护车重定位,提出的算法在全局平均响应时间和初次响应率的指标上均优于已有算法; 同时,本文提出的算法在需求高峰期有更明显的优势. 此外,通过急救网络弹性分析得到的瓶颈站点对于医疗资源配置有重要的借鉴意义. 未来的研究中可以将模型扩展,将救护车调度中更加现实的因素,如故障处理、二次调度、救护车的异构性、急救站点的容量限制等考虑到调度策略中来.

参 考 文 献:

[1]张宇杰, 陈　兵, 林才经. 福建省救护车调度情况调查[J]. 中华急诊医学杂志, 2015, 24(10) : 1103 – 1105.
Zhang Yujie, Chen Bing, Lin Caijing. Investigation on ambulance dispatching in Fujian Province[J]. Chinese Journal of Emergency Medicine, 2015, 24(10) : 1103 – 1105. (in Chinese)

[2] Zhang O, Mason A J, Philpott A B. Simulation and Optimisation for Ambulance Logistics and Relocation [C]. Washington: INFORMS 2008 Conference, 2008.
[3] Jagtenberg C J, Bhulai S, Van d M R D. An efficient heuristic for real-time ambulance redeployment [J]. Operations Research for Health Care, 2015, 4: 27 – 35.
[4] Barneveld T C V. The minimum expected penalty relocation problem for the computation of compliance tables for ambulance vehicles [J]. Informs Journal on Computing, 2016, 28(2): 370 – 384.
[5] Maxwell M S, Restrepo M, Henderson S G, et al. Approximate dynamic programming for ambulance redeployment [J]. INFORMS Journal on Computing, 2010, 22(2): 266 – 281.
[6] Schmid V. Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming [J]. European Journal of Operational Research, 2012, 219(3): 611 – 621.
[7] Miramontes M, Jessica L. Transportation Network Resilience: Evaluation of Performance Measures [M]. ETD Collection for University of Texas, El Paso, Ann Arbor: ProQuest Dissertations Publishing, 2016.
[8] Pavlos T, Yiannis X. Resilience in transportation systems [J]. Procedia-Social and Behavioral Sciences, 2012, 48: 3441 – 3450.
[9] Hong C, Jasmine S, Nan L. Strategic investment in enhancing port-hinterland container transportation network resilience: A network game theory approach [J]. Transportation Research Part B, 2018, 111: 83 – 112.
[10] Zhang X, Miller-Hooks E, Denny K. Assessing the role of network topology in transportation network resilience [J]. Journal of Transport Geography, 2015, 46: 35 – 45.
[11] 汪定伟, 叶伟雄. 交通网络弹复度与易碎度的测算与分析 [J]. 控制理论与应用, 2010, 27(7): 849 – 854.
Wang Dingwei, Ye Weixiong. Evaluation and analysis of resilience and frangibility for transportation networks [J]. Control Theory & Applications, 2010, 27(7): 849 – 854. (in Chinese)
[12] Daskin M S. Application of an expected covering model to emergency medical service system design [J]. Decision Sciences, 1982, 13(3): 416 – 439.
[13] Alanis R, Ingolfsson A, Kolfal B. A Markov chain model for an EMS system with repositioning [J]. Production & Operations Management, 2013, 22(1): 216 – 231.
[14] Lee S. The role of centrality in ambulance dispatching [J]. Decision Support Systems, 2012, 54(1): 282 – 291.
[15] 王付宇, 叶春明, 王 涛, 等. 震后伤员救援车辆两阶段规划模型及算法研究 [J]. 管理科学学报, 2018, 21(2): 68 – 79.
Wang Fuyu, Ye Chunming, Wang Tao, et al. Research on two stage planning model and algorithm of wounded rescue vehicle after earthquake [J]. Journal of Management Sciences in China, 2018, 21(2): 68 – 79. (in Chinese)
[16] Gendreau M, Laporte G, Frédéric S. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation [J]. Parallel Computing, 2001, 27(12): 1641 – 1653.
[17] Gendreau M, Laporte G, Semet F. The maximal expected coverage relocation problem for emergency vehicles [J]. Journal of the Operational Research Society, 2006, 57(1): 22 – 28.
[18] 孔 林, 张国富, 苏兆品, 等. 基于改进蚁群算法的救护车应急救援路径规划 [J]. 计算机工程与应用, 2018, 54(13): 153 – 159.
Kong Lin, Zhang Guofu, Su Zhaopin, et al. Ambulance emergency rescue routing planning for improved ant colony algorithm [J]. Computer Engineering and Applications, 2018, 54(13): 153 – 159. (in Chinese)
[19] 王 晶, 刘昊天, 黄 钧. 考虑伤情分类的灾后创伤伤员救治与转运路径优化研究 [J]. 中国管理科学, 2017, 25(8): 114 – 122.
Wang Jing, Liu Haotian, Huang Jun, et al. Research on the route optimization of ambulance treatment and transportation after disaster based on the injured classification [J]. Chinese Journal of Management Science, 2017, 25(8): 114 – 122. (in Chinese)
[20] Knyazkov K, Oerevitsky I, Mednikov L, et al. Evaluation of dynamic ambulance routing for the transportation of patients with acute coronary syndrome in Saint Petersburg [J]. Procedia Computer Science, 2015, 66: 419 – 428.
[21] Ji S, Zheng Y, Wang W, et al. Real-time ambulance redeployment: A data-driven approach [J]. IEEE Transactions on Knowledge and Data Engineering, 2019, PP(99): 1 – 1.
[22] Maxwell M S, Restrepo M, Henderson S G, et al. Approximate dynamic programming for ambulance redeployment [J]. INFORMS Journal on Computing, 2010, 22(2): 266 – 281.
[23] Marla L, Yue Y, Ramayya K. Data-Driven Omniscient Bounds and Greedy Policies for Ambulance Allocation and Dynamic

Redeployment [EB/OL]. SSRN Electronic Journal, https://papers.ssrn.com/sol3/papers.cfm? abstract_id = 3009043, 2017 - 03 - 12.
[24] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning [J]. arXiv Preprint arXiv: 1312.5602, 2013.
[25] Heess N, Dhruva T B, Sriram S, et al. Emergence of locomotion behaviours in rich environments [J]. arXiv Preprint arXiv: 1707.02286, 2017.
[26] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning [J]. arXiv Preprint arXiv: 1509.02971, 2015.
[27] Wu Y, Mansimov E, Grosse R B, et al. Scalable Trust-Region Method for Deep Reinforcement Learning Using Kronecker-Factored Approximation [C]. California: Neural Information Processing Systems, 2017: 5279 - 5288.
[28] Busoniu L, Babuska R, Schutter B. A comprehensive survey of multiagent reinforcement learning [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2008, 38(2): 156 - 172.
[29] Syafiie S, Tadeo F, Martinez E. Model-free learning control of neutralization processes using reinforcement learning [J]. Engineering Applications of Artificial Intelligence, 2007, 20(6): 767 - 782.
[30] Wu B. Hierarchical Macro Strategy Model for MOBA Game AI [C]. Hawaii: The AAAI Conference on Artificial Intelligence, 2019, 33: 1206 - 1213.
[31] Lin K, Zhao R, Xu Z, et al. Efficient Large-Scale Fleet Management Via Multi-Agent Deep Reinforcement Learning [C]. London: KDD, 2018, July: 1774 - 1783.
[32] Wang L, Zhang W, et al. Supervised Reinforcement Learning with Recurrent Neural Network for Dynamic Treatment Recommendation [C]. London: KDD, 2018, July: 2447 - 2456.
[33] Wei H, Zheng G, Yao H, et al. Intellilight: A Reinforcement Learning Approach for Intelligent Traffic Light Control [C]. London: KDD, 2018, July: 2496 - 2505.
[34] Ardi T, Tambet M, Dorian K, et al. Multiagent cooperation and competition with deep reinforcement learning [J]. PLOS ONE, 2017, 12(4): e0172395.
[35] Watkins C J, Dayan P. Q-learning [J]. Machine Learning, 1992, 8(3 - 4): 279 - 292.
[36] Ben J A. Learning from delayed rewards [J]. Robotics and Autonomous Systems, 1995, 15(4): 233 - 235.
[37] Rummery G A, Niranjan M. On-Line Q-Learning Using Connectionist Systems. Department of Engineering [R]. Cambridge: University of Cambridge, 1994.

Dynamic ambulance redeployment based on deep reinforcement learning

LIU Guan-nan[1], *QU Jin-ming*[1], *LI Xiao-lin*[2*], *WU Jun-jie*[1]
1. School of Economics and Management, Beihang University, Beijing 100191, China;
2. School of Business, Nanjing University, Nanjing 210093, China

Abstract: Ambulance is one of the most crucial medical resources to save patients' lives. Appropriate allocations of limited ambulances to different emergency stations can effectively lower the response time and lift medical service quality. In view of this, we propose a reinforcement learning based scheduling structure to resolve the dynamic ambulance redeployment problems. In order to address the challenges aroused from high-dimensional state spaces, we propose RedCon-DQN by considering multiple scheduling interactive factors, which is based on Deep Q-value Network (DQN) and can output the optimized redeployment policy given specific environment. In addition, we propose a measurement, emergency-network resilience to evaluate the influences of each individual emergency station on the global optimization objectives. Finally, we construct a environment interactive simulator based on the emergency calls and response data of Nanjing from 2016 to 2017. We validate the advantages of the proposed redeployment policy over the state-of-the-art methods, and further analyze the effectiveness and characteristics in different time periods.

Key words: reinforcement learning; DQN; ambulance scheduling; redeployment