

doi: 10.12052/gdutxb.230104

面向停驻车辆辅助边缘计算的服务可靠增强算法

陈明秋, 黄家乐, 武继刚

(广东工业大学 计算机学院, 广东 广州 510006)

摘要: 停驻车辆辅助边缘计算(Parked Vehicle Assisted Edge Computing, PVEC)可充分利用停驻车辆闲置资源,有效缓解车载边缘计算中的资源供需矛盾。然而,因停车行为具有不确定性,车主可突然中止车辆对外提供的计算服务。这将增大用户获取可靠计算服务的难度。为此,本文提出一个面向PVEC的服务可靠增强问题,并引入任务复制技术,将该问题转化为任务副本卸载问题,目标是最小化任务副本的平均服务时延。同时,证明了该问题的NP难解性。为求解问题,提出一种贪婪算法(Greedy Algorithm, GA),优先为具有较大数据量的任务选取一组服务时延最小且满足其服务可靠需求的停驻车辆。此外,还提出一种增强遗传算法(Enhanced Genetic Algorithm, EGA),以进一步优化算法GA的解。实验结果表明,针对用户服务可靠需求变化的情形,与现有的冗余最小化算法及非任务复制算法相比,算法EGA和GA有效降低了任务副本的平均服务时延。

关键词: 停驻车辆辅助边缘计算; 服务可靠增强; 任务复制; 任务副本卸载

中图分类号: TP391

文献标志码: A

文章编号: 1007-7162(2024)04-0080-09

Algorithms for Service Reliability Guarantee in Parked Vehicle Assisted Edge Computing

Chen Ming-qiu, Huang Jia-le, Wu Ji-gang

(School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: Parked vehicle assisted edge computing (PVEC) is effective to alleviate the imbalance between supply and demand of resources in vehicular edge computing, by utilizing the idle resources in parked vehicles (PVs). However, the computing services provided by the PVs can be abruptly aborted due to uncertain parking behaviours. This makes it hard to meet the requirements of users on service reliability. To address this issue, this paper formulates an optimization problem for service reliability guarantee. Then, a task replication technique is introduced to transform the formulated problem into a replication offloading problem, with the goal of minimizing the average completion time of task replications. The NP-hardness of the formulated problem is proved. A greedy algorithm (GA) is proposed to solve the formulated problem, by carefully offloading the replicas of the tasks with large data sizes to the PVs, which can provide the computing services with service guarantee and short completion time. Meanwhile, an enhanced genetic algorithm (EGA) is proposed to refine the solution generated by the proposed algorithm GA. Experimental results show that the proposed GA and EGA algorithms outperform the baseline algorithms in terms of the average completion time of task replications for different requirements of users on service reliability.

Key words: parked vehicle assisted edge computing; service reliability; task replication; replication offloading

随着智能交通的迅猛发展,涌现出大量车载智能应用,如自动驾驶和碰撞避免^[1]。然而,受制于资源容量与续航需求,车辆难以保障车载智能应用的低时延响应需求。针对该问题,现有研究提出一种车载

边缘计算(Vehicular Edge Computing, VEC)范式^[2-3],其可将车载任务卸载至网络边缘,实现车载智能应用的低时延运行。然而,受制于资源容量,VEC中的边缘服务器难以应对日益增长的车载智能应用请求^[4]。

收稿日期: 2005-08-12

基金项目: 国家自然科学基金资助项目(62202108, 62072118); 广东省基础与应用基础研究基金资助项目(2021B1515120010)

作者简介: 陈明秋(1997-),女,硕士研究生,主要研究方向为边缘计算和车联网, E-mail: ElevenChen11@126.com

通信作者: 武继刚(1963-),男,教授,博士,主要研究方向为边缘计算和智能计算, E-mail: asjgwucn@outlook.com

根据城市停车管理研究报告^[5],私家车日均停车时间超20 h。当车辆处于停车状态时,车载资源将处于闲置状态。此时,充分利用停驻车辆(Parked Vehicle, PV)的闲置资源,可有效缓解VEC中的资源供需矛盾。受此启发,现有研究提出了一种停驻车辆辅助边缘计算(Parked Vehicle Assisted Edge Computing, PVEC)范式^[6-7],其可利用PV的闲置资源,扩充VEC中边缘服务器的资源储备。

服务可靠度是衡量用户服务体验的重要指标,具体指用户任务在执行过程中不被中止的概率^[8-9]。通常,具有较低中止概率的计算服务具有较高的服务可靠度。然而,在PVEC中,车主的停车行为具有不确定性,如停车时间长短不一。受此影响,PV可在不同时刻随时中止对边缘服务器的资源共享,从而中止对用户的服务供给。因此,在固定的服务周期内,PV提供的计算服务具有差异化服务可靠度。受制于资源容量,PV难以向服务周期内的所有用户提供计算服务。另外,用户的服务可靠需求存在差异性。若简单地卸载任务至具有较低服务可靠度的PV上进行处理,用户将难以获得高可靠度的计算服务,并将遭受低质量的服务体验。因此,亟需寻求一种可有效保障用户服务可靠需求的任务卸载方案,提高用户的服务体验质量。

任务复制技术是一种高效容错技术,其利用冗余调度策略,保障任务的服务可靠需求^[10]。因此,本文将引入任务复制技术,确保用户的服务可靠需求。具体而言,为每个用户任务创建多个副本,并将这些副本交由多辆PV进行处理。若某一副本的计算过程正常结束,其对应的任务可被成功执行^[11-12]。换言之,在PVEC中,当存在PV完成任务副本的计算过程时,该副本对应的用户将获得可靠的计算服务。

若利用任务复制技术保障用户服务可靠需求,本文需解决下述挑战:(1) 如何确定任务副本的数量。当副本数量过多时,任务将占用多辆PV的闲置计算资源。这将导致其他任务难以获取足够的计算资源,实现其副本的放置与处理。另外,用户具有服务可靠需求。若副本数量过少,用户将难以获取高可靠度的计算服务。(2) 如何确定任务副本的放置位置。不合理的副本放置方案将导致具有充足计算资源或高服务可靠度的PV中出现资源和服务的过度抢占。这将极大延长任务的服务时延,从而导致用户时延需求得不到满足。

为应对上述挑战,本文将提出一个服务可靠增强问题,并将该问题建模为任务副本卸载问题。该问

题将制定任务副本卸载策略,在满足用户的服务可靠需求与延迟需求,以及PV中可用计算资源约束的前提下,实现任务副本的平均服务时延最小化。本文的主要贡献如下。

(1) 提出一个面向PVEC的服务可靠增强问题,优化任务副本的平均服务时延。引入任务复制技术,将所提问题转化为任务副本卸载问题。同时,证明了该问题的NP难解性。

(2) 提出一种贪婪算法,优先为具有较大数据量的用户任务选取一组服务时延最小且满足用户服务可靠需求的PV。同时,还提出一种增强遗传算法,进一步优化贪婪算法的解。

(3) 实验结果表明,增强遗传算法和贪婪算法具有明显的性能优势。针对用户服务可靠需求变化的情形,相较于冗余最小化算法与非任务复制算法,所提算法都可有效降低任务副本的平均服务时延。

1 相关工作

在PVEC研究领域中,现有工作主要聚焦于如何提高PV中闲置资源利用率^[7,13-17]。文献[13]研究了一个面向用户任务和PV的双向匹配问题,并提出了一种基于契约理论的激励机制,促进PV的资源共享。文献[14]研究了一个面向PVEC中用户任务的部分计算卸载问题,并设计了一种次梯度算法,确定用户计算任务的最优卸载率,以及制定边缘服务器的计算资源分配方案,从而实现系统效用的提升。文献[7]研究了一个面向用户任务与PV中资源容器的匹配博弈问题。同时,文献[7]提出了一种资源分配策略,在满足PV计算资源约束的基础上,为任务分配合适的资源容器,从而实现用户与PV的总效用最大化。文献[15]提出了一个面向PVEC的社会福利最大化问题,并设计了一种低复杂度联合算法,在满足用户预算需求的基础上,制定用户任务部署策略与PV计算资源配置策略。针对PV间负载不均问题,文献[16]设计了一种主动负载均衡机制,提高PV的资源利用率,进而实现PV间负载均衡。文献[17]研究了一个面向停车场的车位预测问题,并提出了一种多代理深度强化学习算法,实现停车场运营方与PV的总体经济效益最大化。

在任务复制研究领域中,现有工作侧重于求解副本数量推断问题与副本位置放置问题^[12,18-22]。针对边缘网络,文献[18]分析了数据块访问频率、副本数量和副本位置三者间的关系,并提出了一种自适应副本放置策略,实现了用户的低时延数据访问。针对

边云协作计算网络,文献[19]研究了一个面向数据副本创建与分配的联合优化问题,并提出了一个动态副本生成方法,优化网络中全部数据的访问成本。文献[20]提出了一种副本放置均衡策略,实现数据节点的负载均衡。针对边缘云文件系统中数据写入延迟高、数据可用性低等问题,文献[21]提出了一种延迟自适应副本同步策略,实现数据写入速度与数据质量的共同提升。文献[12]研究了一个 workflow 调度问题,并设计了一个基于博弈论的副本调度模型,提高系统的安全性与可靠性。文献[22]提出了一个面向 VEC 的任务复制问题,该问题旨在综合考虑车辆密度、任务到达率等因素,实现用户任务副本数量的最优推断。

然而,上述工作尚未考虑停车行为对 PV 服务可靠度的影响。为填补这一空白,本文研究了一个面向 PVEC 的服务可靠增强问题,以提高用户所获计算服务的可靠度,进而确保用户的服务可靠需求。

2 系统模型

2.1 网络模型

本文考虑一个任务复制技术赋能的 PVEC 系统。如图1所示,该系统由一个基站和多辆 PV 构成。基站中部署有边缘服务器(Edge Server, ES),而 PV 处于停车状态,可通过无线链路与 ES 进行数据传输。令 $V = \{v_1, v_2, \dots, v_m\}$ 表示 PV 集合,其中 v_i 为第 i 辆 PV, m 为 PV 数量。令 $S = \{s_1, s_2, \dots, s_n\}$ 表示用户集合,其中 s_j 为第 j 个用户, n 为用户数量。同时,每个用户仅产生一个任务。ES 采用任务复制技术,可为任务创建多个副本。这些副本将被卸载至多辆 PV 上进行处理。如图1所示,在停车场内,ES 创建了任务1的2个副本,并将其分别卸载至 PV a 和 PV b 上进行处理。在处理过程中,若 PV b 突然离开停车场,其上的副本处理将被中止。但是, PV a 仍处于停车状态,可继续处理其

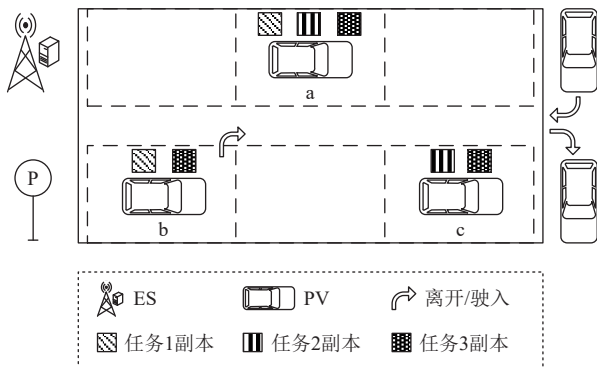


图1 任务复制技术赋能的PVEC系统

Fig.1 Task replication based PVEC system

上的副本。因此,任务1仍可被成功执行。

2.2 服务可靠度模型

根据文献[6-7],本文选用 PV 继续保持停车状态的概率,表征 PV 的服务可靠度。令 σ_i 和 P_i 分别为 v_i 的累计停车时间和服务可靠度。服务可靠度 P_i 的定义为,在累计停车 σ_i 分钟的前提下, v_i 在未来 T 分钟内继续保持停车状态的概率,即

$$P_i = P\{t \geq \sigma_i + T | t \geq \sigma_i\} \quad (1)$$

式中: t 为所有 PV 累计停车时间,其值不可大于所有 PV 累计停车时间的最大值 $t^{\max}$,即 $t \in [0, t^{\max}]$ 。

为求得式(1),需计算概率 $P\{t \geq \sigma_i\}$ 与概率 $P\{t \geq \sigma_i + T\}$,计算公式分别为

$$P\{t \geq \sigma_i\} = \int_{\sigma_i}^{t^{\max}} f(t) dt = 1 - F(\sigma_i) \quad (2)$$

$$P\{t \geq \sigma_i + T\} = \int_{\sigma_i + T}^{t^{\max}} f(t) dt = 1 - F(\sigma_i + T) \quad (3)$$

式中: $f(t)$ 和 $F(t)$ 分别为所有 PV 累计停车时间 t 的概率密度函数和分布函数,且有 $F(t) = \int_0^t f(t) dt$ 。

根据贝叶斯公式,将式(2)~(3)代入式(1),可计算出 v_i 的服务可靠度 P_i 。计算公式为

$$P_i = \frac{1 - F(\sigma_i + T)}{1 - F(\sigma_i)} \quad (4)$$

2.3 任务复制模型

令 $X = \{x_{i,j} | v_i \in V, s_j \in S\}$ 表示任务副本卸载策略。其中, $x_{i,j}$ 为任务 s_j 的副本是否被卸载至 v_i 上执行。若是,则 $x_{i,j} = 1$,否则, $x_{i,j} = 0$ 。

假设每个任务副本仅可被卸载至一辆 PV 上执行,则有

$$\sum_{i=1}^m x_{i,j} = 1 \quad (5)$$

由此可知,每个任务的副本数量不可大于 PV 数量,即

$$1 \leq \theta_j \leq m \quad (6)$$

式中: θ_j 为任务 s_j 的副本数量,其计算公式为

$$\theta_j = \sum_{i=1}^m x_{i,j} \quad (7)$$

任务副本总数量的计算公式为

$$\varpi = \sum_{j=1}^n \theta_j \quad (8)$$

用户所获计算服务的服务可靠度由执行任务副本的多辆 PV 的服务可靠度共同决定,计算公式为

$$R_j = 1 - \prod_{i=1}^m (1 - x_{i,j} P_i) \quad (9)$$

2.4 时延模型

任务副本的服务时延由3部分组成:(1) ES将副本卸载至PV的传输时延;(2) PV执行副本的计算时延;(3) PV将计算结果回传至ES的传输时延。

根据香农定理,可计算出ES与 v_i 之间的传输速率,计算公式为

$$r_i = B \log_2 \left(1 + \frac{p_0 h_i^{-\varepsilon}}{N_0} \right) \quad (10)$$

式中: B 为信道带宽, p_0 为ES的发送功率, N_0 为环境高斯白噪音, $h_i^{-\varepsilon}$ 为信道增益, h_i 为ES与 v_i 之间的欧氏距离, ε 为路径损耗参数。

ES将任务 s_j 的副本卸载至 v_i 的传输时延 $t_{i,j}^{\text{tra}}$ 中,其由副本数据量与传输速率共同决定,计算公式为

$$t_{i,j}^{\text{tra}} = \frac{d_j}{r_i} \quad (11)$$

式中: d_j 为副本数据量,即任务 s_j 的数据量。

v_i 可同时执行多个副本。假设PV的计算资源由其上副本平均分配。此时, v_i 分配给任务 s_j 的副本的计算资源为

$$f_{i,j} = f_i^{\max} / \sum_{j=1}^n x_{i,j} \quad (12)$$

式中: $f_i^{\max}$ 为 v_i 的计算资源容量。PV执行副本的计算时延 $t_{i,j}^{\text{exe}}$ 由副本的工作负载与 v_i 分配给副本的计算资源共同决定,计算公式为

$$t_{i,j}^{\text{exe}} = \frac{\lambda d_j}{f_{i,j}} \quad (13)$$

式中: λ 为每比特数据的计算负载^[23]。

与副本数据量相比,副本计算结果的数据量非常小,可忽略不计^[7]。因此,本文忽略副本计算结果的传输时延。此时,副本的服务时延 $t_{i,j}$ 为

$$t_{i,j} = t_{i,j}^{\text{tra}} + t_{i,j}^{\text{exe}} \quad (14)$$

综上,任务副本的平均服务时延计算公式为

$$\varphi(X) = \frac{1}{\omega} \sum_{i=1}^m \sum_{j=1}^n x_{i,j} t_{i,j} \quad (15)$$

2.5 问题定义

本文将服务可靠增强问题建模为任务副本卸载问题,目标是 minimized 副本的平均服务时延。该问题的形式化描述为

$$\mathcal{P}: \min \varphi(X)$$

$$\text{s.t. (5) ~ (8)}$$

$$x_{i,j} t_{i,j} \leq t_j^{\max}, \forall v_i \in V, \forall s_j \in S \quad (16)$$

$$\sum_{j=1}^n x_{i,j} f_{i,j} \leq f_i^{\max}, \forall v_i \in V \quad (17)$$

$$R_j \geq R_j^{\min}, \forall s_j \in S \quad (18)$$

$$x_{i,j} \in \{0, 1\}, \forall v_i \in V, \forall s_j \in S$$

其中,式(16)保证任务的服务时延不大于其最大容忍时延,式(17)确保PV分配给任务的计算资源不超过其计算资源容量,式(18)确保任务的服务可靠需求可得到满足。

定理1 问题 $\mathcal{P}$ 是NP难问题。

证明 本文将问题 $\mathcal{P}$ 归约为0-1背包问题,从而证明其NP难解性。具体而言,假设用户可接受无限大的服务时延,且每辆PV的服务可靠度都为1。由此可知,用户的时延需求与服务可靠需求都可得到满足。因此,可忽略问题 $\mathcal{P}$ 的时延约束(16)、服务可靠约束(18)以及副本数量约束(6)。基于此,问题 $\mathcal{P}$ 可被转化为一个PV计算资源容量受限的任务卸载问题 $\mathcal{P}'$ 。依据背包问题的定义,在问题 $\mathcal{P}'$ 中,任务可等同于物品,PV可等同于背包,PV计算资源容量可等同于背包容量,PV分配给任务的计算资源可等同于物品重量,任务的服务时延可等同于物品价值。因此,PV上的任务卸载问题可被归约为0-1背包问题。由于背包问题是NP难问题^[24],因此PV上的任务卸载问题具有NP难解性。此外,由此问题 $\mathcal{P}'$ 中存在多辆PV,因此其求解难度更大。综上,问题 $\mathcal{P}$ 是NP难问题。

3 算法设计

由定理1可知,难以在多项式时间复杂度内求得问题 $\mathcal{P}$ 的最优解。因此,本文将提出一个贪婪算法(Greedy Algorithm, GA)和一个增强遗传算法(Enhanced Genetic Algorithm, EGA),近似求解 $\mathcal{P}$ 。

3.1 贪婪算法

算法GA的核心思想是,具有较大数据量的任务优先交由一组服务时延最小且满足用户服务可靠需求的PV进行处理,具体步骤如下。

步骤①:根据数据量大小,对所有用户任务进行降序排序,并构建相应的有序队列 S 。

步骤②:取出 S 中首个任务 s_j ;根据式(13),计算各PV分配给 s_j 的计算资源;根据分配给 s_j 的计算资源大小,对所有PV进行降序排序,并构建相应的有序队列 V 。

步骤③:取出 V 中首辆 v_i ;尝试将 s_j 的副本卸载至 v_i 上执行;根据式(14),计算 v_i 上各副本的服务时延,并判断副本对应用户的时延需求是否得到满足;若满足,转至步骤④,否则,转至步骤⑤。

步骤④:根据式(9),计算 s_j 所获计算服务的服务

可靠度 R_j ,并判断其用户服务可靠需求是否得到满足;若不满足,转至步骤⑤,否则,转至步骤⑥。

步骤⑤:从 V 中剔除 v_i ,并判断 V 是否为空队列;若是,则撤销 v_i 中 s_j 的副本放置,并转至步骤⑥,否则,转至步骤③。

步骤⑥:从 S 中剔除 s_j ;若 S 中存在元素,则重复步骤②~⑥。

3.2 增强遗传算法

算法EGA将算法GA的解作为其初始解。同时,算法EGA将依次执行染色体选择、染色体交叉、染色体变异、染色体修正等操作,优化初始解,从而寻求问题 P 的全局最优解,具体步骤如下。

(1) 染色体编码:采用二进制编码方式,将任务副本卸载策略编码为染色体。染色体长度为 nm ,其中 n 为任务数量, m 为PV数量。如果任务 s_j 的副本被卸载至 v_i 上执行,则染色体中第 $(i-1)n+j$ 位置的等位基因值置为1,否则,置为0。此外,任务副本的总数量由染色体中值为1的等位基因个数决定。

(2) 种群初始化:首先,将算法GA的贪心解编码为一条染色体。然后,利用随机策略生成多条满足问题约束的染色体。最后,贪心解染色体和随机策略染色体共同组成初始种群。其中,每条染色体代表种群的一个个体。

(3) 适应度函数:选用任务副本的平均服务时延的相反数,衡量个体的适应度值,计算公式为

$$\text{fitness} = -\varphi(X) \quad (19)$$

式中:fitness为个体适应度值。若个体中问题约束得不到满足,则该个体的适应度值被置为负无穷。

(4) 染色体选择:随机选取种群中2个个体,作为下一代种群的2个父亲。

(5) 染色体交叉:采用两点交叉方法,实现父代染色体的交叉互换。具体而言,随机生成2个在区间 $[1, nm]$ 中的整数 a 和 b 。然后,互换两条父代染色体中区间 $[low, high]$ 的基因片段。其中, $low = \min\{a, b\}$, $high = \max\{a, b\}$ 。

(6) 染色体变异:随机生成一个在区间 $[1, nm]$ 中的整数,标记染色体中产生变异的等位基因位置。该位置的等位基因值将由1/0变异为0/1。

(7) 有效性判断:在前述3个染色体操作结束后,需判断子代染色体中用户的时延需求与服务可靠需求,以及副本数量约束是否得到满足。若不满足,则子代染色体是无效染色体,否则,子代染色体为有效染色体。

(8) 染色体修正(Chromosome Correction, CO):为提高算法收敛性,需对无效染色体执行修正操作。具体而言,在无效染色体中,筛选出不满足用户服务可靠需求的等位基因。该等位基因值将被置为0,其对应任务将交由可满足用户时延需求与服务可靠需求的PV执行。该操作的伪代码如算法1所示。其中,根据算法1中3~10行,等位基因筛选操作的时间复杂度为 $O(n)$ 。根据算法1中15~24行,为一个任务构建有序PV队列的时间复杂度为 $O(m \log_2 m)$,任务副本放置操作的时间复杂度为 $O(m)$ 。在最坏情况下,服务可靠需求得不到满足的用户数目为 n 。因此,前述操作的时间复杂度为 $O(nm \log_2 m + nm)$ 。综上可知,算法1的时间复杂度为 $O(nm \log_2 m)$ 。

算法1 $[I'] = CO(I, V, S)$

- 1) 输入:无效染色体 I ,PV集合 V ,用户集合 S 。
- 2) $U \leftarrow \emptyset$;
- 3) for $j \leftarrow 1$ to n do
- 4) 根据式(9),计算 R_j ;
- 5) if $R_j < R_j^{\min}$ then
- 6) 构建PV集合 F_j ,其中PV都具有 s_j 的副本;
- 7) 在 I 中,将与 s_j 关联的等位基因值置为0;
- 8) $U \leftarrow U \cup \{s_j\}$;
- 9) end if
- 10) end for
- 11) while $U \neq \emptyset$ do
- 12) 随机选取 U 中一个任务 s_k ;
- 13) $W \leftarrow V \setminus F_k$;
- 14) 对于 W 中每辆PV,当 s_k 副本放置于其上时,根据式(13),计算其分配给 s_k 的计算资源;
- 15) 根据分配给 s_k 的计算资源大小,对 W 中所有PV进行降序排序,并构建相应的有序队列 W' ;
- 16) while $R_k < R_k^{\min}$ and $W' \neq \emptyset$ do
- 17) 取出 W' 中首辆 v_l ;
- 18) 根据式(11)、(12)和(14),计算 $t_{l,k}$;
- 19) if $t_{l,k} \leq t_k^{\max}$ then
- 20) 根据式(9),更新 R_k ;
- 21) 将 I 中 $(l-1)n+k$ 位置的等位基因值置为1;
- 22) end if
- 23) $W' \leftarrow W' \setminus \{v_l\}$;
- 24) end while
- 25) $U \leftarrow U \setminus \{s_k\}$;
- 26) end while
- 27) 输出:有效染色体 I' 。

(9) 下一代种群生成:选用下述2种方式,生成下

一代个体。第一种方式是通过染色体交叉、染色体变异以及染色体修正三大操作生成新个体。第二种方式是精英保留策略,即父代种群中具有最高适应度值的个体将被直接保留。

4 实验设置与分析

4.1 实验设置

实验参数的具体设置如下所示:根据文献[2-3, 6-7],信道带宽 B 设为10 MHz,ES的发射功率 p_0 设为0.1 W,高斯白噪声 N_0 设为-174 dB/Hz。PV数量 m 和任务数量 n 的取值范围分别设为[60, 140]和[60, 100]。PV计算资源容量的取值范围设为[2, 3] GHz, PV与ES的欧氏距离 h_i 的取值范围设为[5, 200] m。任务数据量 d_i 、最大容忍时延 $t_{ij}^{\max}$ 和用户服务可靠需求 $R_j^{\min}$ 的取值范围分别设为[100, 300] KB、[1, 3] s和[0.6, 0.95]。每比特数据的计算负载 λ 设为500 CPU cycles/bit,时间跨度 T 设为30 min。此外,在所提算法EGA中,种群规模设为150,最大迭代次数设为1 000,染色体的交叉率和变异率分别设为0.95和0.9。

实验选用开放数据网站dataACT^[25]和剑桥数据平台^[26]提供的真实停车时间数据集,分别记为ACT和Cambridge。根据统计分析,ACT中约98.21%的PV停车时间小于250 min, Cambridge中约98.49%的PV停车时间小于500 min。因此,在拟合数据集的停车时间概率密度曲线时,本文将ACT和Cambridge中最长累计停车时间分别设为250 min和500 min。如图2(a)和图2(b)所示,分别绘制了ACT和Cambridge中PV停车时间的频率分布直方图,并拟合了相应的概率密度曲线。在95%的置信区间下,设 $f_A(t)$ 和 $f_C(t)$ 分别表示ACT和Cambridge中PV停车时间的概率密度函数。

$$f_A(t) = 0.86e^{-0.18t} + 0.17e^{-0.02t} \quad (20)$$

$$f_C(t) = 0.005e^{-\frac{(t-150.37)^2}{15316.01}} \quad (21)$$

根据式(20)、(21)和(4),可计算PV的服务可靠度。经过多次重复计算,ACT和Cambridge中PV服务可靠度的取值分别坐落在区间[0.39, 0.97]和[0.22, 0.95]。

本文选取非任务复制算法(Replication-Free Algorithm, RFA)^[6]和冗余最小化算法(Redundancy Minimization Algorithm, RMA)^[27],与所提算法进行性能对比。在不采用任务复制技术的前提下,算法RFA优先选择恰好满足用户服务可靠需求的PV,执行用户任务。若存在服务可靠需求无法得到满足的

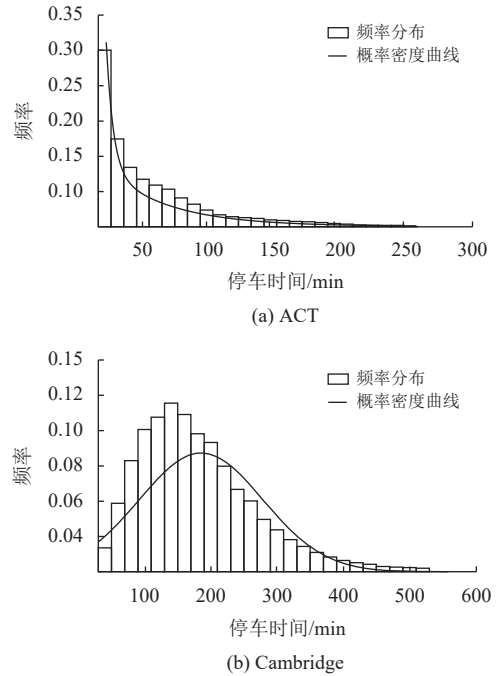


图2 停车时间频率分布和概率密度

Fig.2 Frequency distribution and probability density of the parking time

用户,算法RFA将放弃执行该用户任务。算法RMA采用任务复制技术,为用户任务创建多个副本。对于某个用户,在其服务可靠需求得到满足之前,算法RMA将贪婪地选取具有较高服务可靠度的PV,执行其任务副本。此外,本文采用Python语言编写所提算法与对比算法的实验代码,并在配置有Intel(R) Core(TM) i7-6 700 CPU与8 GB RAM的主机上开展实验。

4.2 实验分析

图3展示了算法EGA的收敛性。随着遗传进化代数的增加,副本的平均服务时延逐渐减小。这是因为在遗传进化过程中算法EGA不断优化解的质量。由图3可知,在600次遗传进化后,算法EGA开始收敛。

图4展示了在PV数量变化的情形下算法EGA、GA、RFA以及RMA在副本平均服务时延方面的性能对比。由图4可知,随着PV数量的增大,4种算法的平均服务时延都不断减小。具体而言,随着PV数量的增大,系统的总体计算资源容量不断增多,任务副本可获计算资源也随之增多。这将减少任务副本的服务时延。如图4(a)所示,在ACT中,与算法RMA相比,EGA和GA分别降低了65.56%和61.24%的平均服务时延。如图4(b)所示,在Cambridge中,与算法RMA相比,EGA和GA分别降低了50.11%和47.19%的平均服务时延。

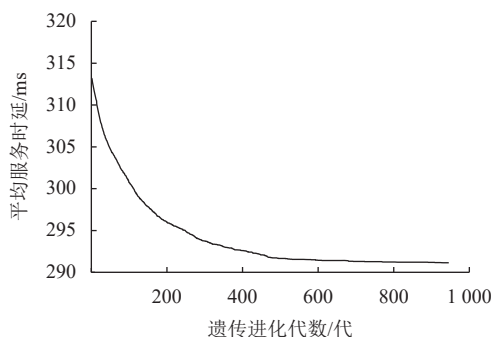
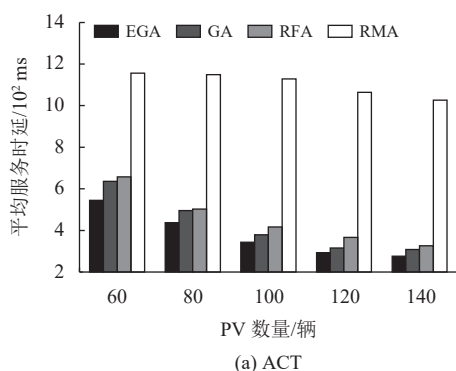
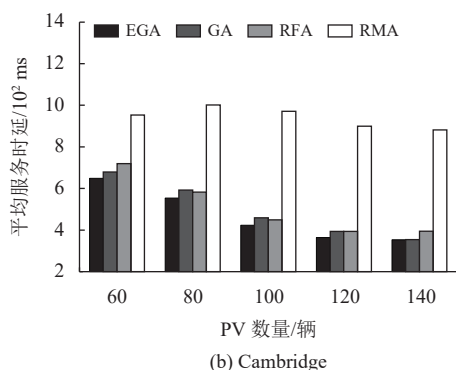


图3 算法EGA的收敛性

Fig.3 Convergence of the EGA



(a) ACT



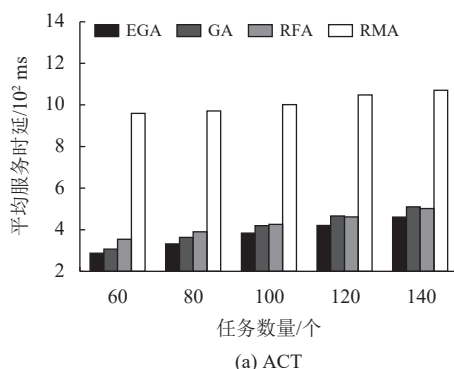
(b) Cambridge

图4 平均服务时延与PV数量

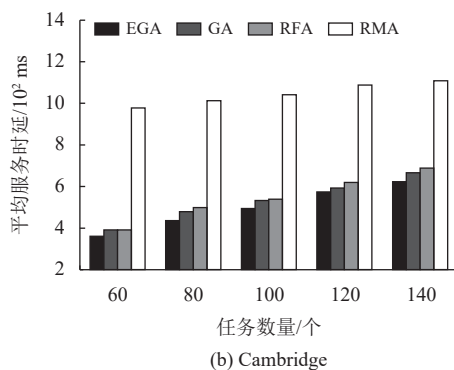
Fig.4 Average completion time vs the number of PVs

图5展示了在任务数量变化的情形下算法EGA、GA、RFA以及RMA在副本平均服务时延方面的性能对比。如图5所示,随着任务数量的增多,4种算法中副本的平均服务时延都不断增大。具体而言,当PV中计算资源容量保持不变时,随着任务数量增多,任务可获计算资源将不断减少。这将延长任务的服务时延。如图5(a)所示,在ACT中,与算法RMA相比,EGA和GA分别降低了63.46%和59.80%的平均服务时延。如图5(b)所示,在Cambridge中,与算法RMA相比,EGA和GA分别降低了52.42%和49.14%的平均服务时延。

图6展示了在PV服务可靠度变化的情形下算法



(a) ACT



(b) Cambridge

图5 平均服务时延与任务数量

Fig.5 Average completion time vs the number of tasks

EGA、GA以及RMA在副本平均服务时延方面的性能对比。如图6所示,随着PV服务可靠度的增大,平均服务时延不断减少。具体而言,随着PV服务可靠度的增大,系统的总计算资源容量不断增大,任务副本可获得的计算资源随之增多。这将降低任务副本平均服务时延。如图6(a)所示,在ACT中,与算法RMA相比,EGA和GA分别降低了66.08%和64.70%的平均服务时延。如图6(b)所示,在Cambridge中,与算法RMA相比,EGA和GA分别降低了55.99%和52.53%的平均服务时延。

图7展示了在用户服务可靠需求变化的情形下算法EGA、GA、RFA以及RMA在任务副本的平均服务时延方面的性能对比。如图7所示,随着用户服务可靠需求的提高,副本的平均服务时延不断增大。具体而言,当用户服务可靠需求变高时,用户任务的副本数量将变多。这将减少各副本的计算资源,从而增加副本的时间开销。如图7(a)所示,在ACT中,与算法RFA相比,EGA和GA分别降低了44.49%和30.58%的平均服务时延。与算法RMA相比,EGA和GA分别降低了73.22%和66.51%的平均服务时延。如图7(b)所示,在Cambridge中,与算法RFA相比,EGA和GA分别降低了39.82%和28.20%的平均服务时延。与算法MRA相比,EGA和GA分别降低了65.21%和58.49%

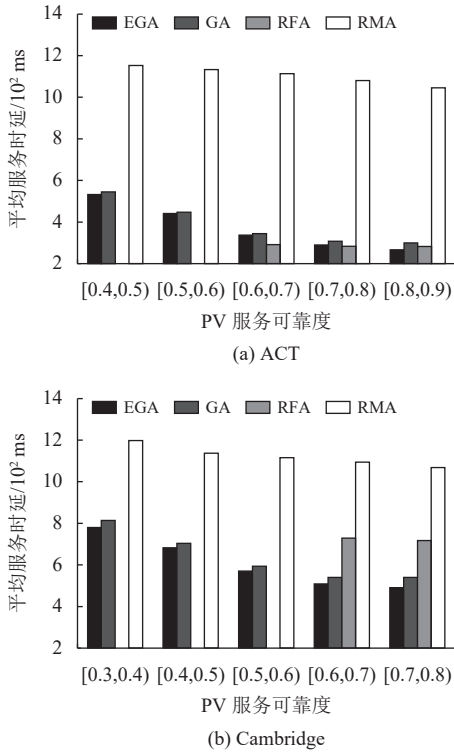


图6 平均服务时延与PV服务可靠度
Fig.6 Average completion time vs reliability of PVs

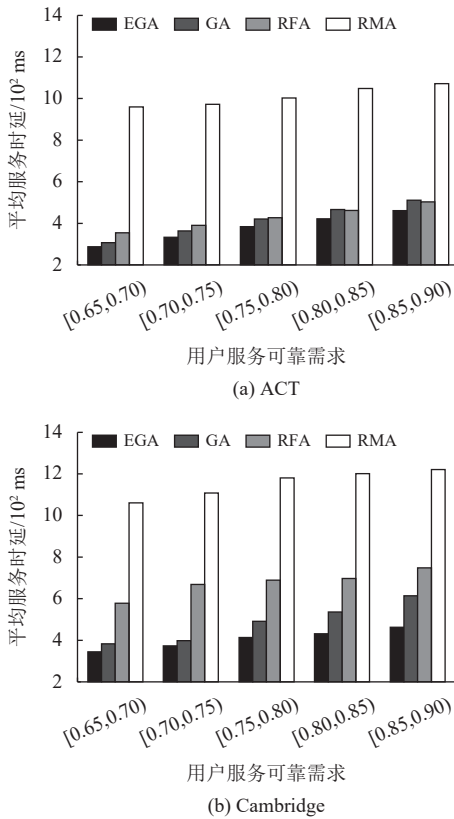


图7 平均服务时延与用户服务可靠需求
Fig.7 Average completion time vs reliability requirements of users

的平均服务时延。

图8展示了在PV服务可靠度变化的情形下算法EGA、GA以及RFA在任务完成率方面的性能对比。如图8(a)所示,在ACT中,随着PV服务可靠度的增大,算法RFA的任务完成率从0逐渐增至84.33%,而算法EGA和GA的任务完成率都为100%。如图8(b)所示,在Cambridge中,随着PV服务可靠度的增大,算法RFA的任务完成率从0逐渐增至78.33%,而算法EGA和GA的任务完成率都为100%。这是因为RFA没有采用任务复制技术,且服务可靠度低的PV难以满足用户的高服务可靠需求。此时,RFA将放弃执行相关的用户任务。

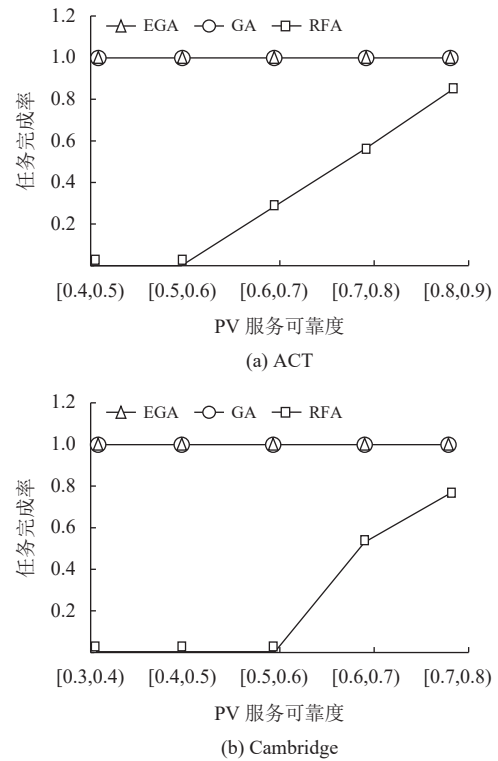


图8 任务完成率与PV服务可靠度
Fig.8 Completion rate vs reliability of PVs

5 结论

本文研究了一个面向PVEC的服务可靠增强问题,并引入任务复制技术,将其转化为任务副本卸载问题。同时,证明了该问题的NP难解性。为求解问题,提出了一种增强遗传算法和一种贪婪算法。实验结果表明,本文所提算法具有明显的性能优势。在数据集ACT中,当用户服务可靠需求变化时,与算法RMA相比,EGA和GA分别降低了73.22%和66.51%的平均服务时延,而与算法MRA相比,EGA和GA分别降低了73.22%和66.51%的平均服务

时延。在数据集Cambridge中,当PV服务可靠度变化时,与算法RFA相比,算法EGA和GA都提高了279.75%的任务完成率。

参考文献:

- [1] KESHAVAMURTHY P, PATEROMICHELAKIS D, DAHLHAUS D, *et al.* Edge cloud-enabled radio resource management for cooperative automated driving [J]. *IEEE Journal on Selected Areas in Communications*, 2020, 38(7): 1515-1530.
- [2] WU Y L, WU J G, CHEN L, *et al.* Fog computing model and efficient algorithms for directional vehicle mobility in vehicular network [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 22(5): 2599-2614.
- [3] WU Y L, WU J G, CHEN L, *et al.* Load balance guaranteed vehicle-to-vehicle computation offloading for min-max fairness in VANETs [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(8): 11994-12013.
- [4] WANG X J, NING Z L, GUO S, *et al.* Imitation learning enabled task scheduling for online vehicular edge computing [J]. *IEEE Transactions on Mobile Computing*, 2022, 21(2): 598-611.
- [5] LITMAN T. Parking management: strategies, evaluation and planning[M]. Victoria, BC, Canada: Victoria Transport Policy Institute, 2006.
- [6] HUANG X M, YE D D, YU R, *et al.* Securing parked vehicle assisted fog computing with blockchain and optimal smart contract design [J]. *IEEE/CAA Journal of Automatica Sinica*, 2020, 7(2): 426-441.
- [7] HUANG X M, YU R, XIE S, *et al.* Task-container matching game for computation offloading in vehicular edge computing and networks [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 22(10): 6242-6255.
- [8] ZHOU A, WANG S, CHENG B, *et al.* Cloud service reliability enhancement via virtual machine placement optimization [J]. *IEEE Transactions on Services Computing*, 2017, 10(6): 902-913.
- [9] SILIC M, DELAC G, SRBLJIC S, *et al.* Prediction of atomic web services reliability for QoS-aware recommendation [J]. *IEEE Transactions on Services Computing*, 2015, 8(3): 425-438.
- [10] DONGARRA J, JEANNOT E, SAULE E, *et al.* Bi-objective scheduling algorithms for optimizing makespan and reliability on heterogeneous systems[C]// ACM Symposium on Parallel Algorithms and Architectures. San Diego: ACM, 2007: 280-288.
- [11] CHEN L, XU J. Task replication for vehicular cloud: contextual combinatorial bandit with delayed feedback[C]// IEEE Conference on Computer Communications. Paris: IEEE, 2019: 748-756.
- [12] SUN Y, ZHOU S, NIU Z. Distributed task replication for vehicular edge computing: performance analysis and learning-based algorithm [J]. *IEEE Transactions on Wireless Communications*, 2021, 20(2): 1138-1151.
- [13] ZHOU Z Y, LIU P J, FENG J H, *et al.* Computation resource allocation and task assignment optimization in vehicular fog computing: a contract-matching approach [J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(4): 3113-3125.
- [14] PHAM X Q, HUYNH T, HUH E, *et al.* Partial computation offloading in parked vehicle-assisted multi-access edge computing: a game-theoretic approach [J]. *IEEE Transactions on Vehicular Technology*, 2022, 71(9): 10220-10225.
- [15] HUANG X M, LI P, YU R. Social welfare maximization in container-based task scheduling for parked vehicle edge computing [J]. *IEEE Communications Letters*, 2019, 23(8): 1347-1351.
- [16] AHMED J K, JABER I N. Proactive load balancing mechanism for fog computing supported by parked vehicles in IoV-SDN [J]. *China Communications*, 2021, 18(2): 271-289.
- [17] HUANG X, LI P, YU R, *et al.* Fedparking: a federated learning based parking space estimation with parked vehicle assisted edge computing [J]. *IEEE Transactions on Vehicular Technology*, 2021, 70(9): 9355-9368.
- [18] LI C, WANG Y, TANG H, *et al.* Flexible replica placement for enhancing the availability in edge computing environment [J]. *Computer Communications*, 2019, 146: 1-14.
- [19] SHAO Y L, LI C L, FU Z, *et al.* Cost-effective replication management and scheduling in edge computing [J]. *Journal of Network and Computer Applications*, 2019, 129: 46-61.
- [20] LI C, WANG Y, TANG H, *et al.* Dynamic multi-objective optimized replica placement and migration strategies for SaaS applications in edge cloud [J]. *Future Generation Computer Systems*, 2019, 100: 921-937.
- [21] LI C, SONG M, ZHANG M, *et al.* Effective replica management for improving reliability and availability in edge-cloud computing environment [J]. *Journal of Parallel and Distributed Computing*, 2020, 143: 107-128.
- [22] TAO M, OTA K, DONG M. DSARP: dependable scheduling with active replica placement for workflow applications in cloud computing [J]. *IEEE Transactions on Cloud Computing*, 2020, 8(4): 1069-1078.
- [23] WU Y L, WU J G, CHEN L, *et al.* Efficient task scheduling for servers with dynamic states in vehicular edge computing [J]. *Computer Communications*, 2020, 150: 245-253.
- [24] KELLERER H, PFERSCHY U, PISINGER D. Knapsack problems[M]. Berlin: Springer, 2004.
- [25] ACT Government Open Data Portal dataACT, Smart parking lots[EB/OL]. (2017-07-28) [2023-05-25]. <https://www.data.act.gov.au/Transport/Smart-Parking-Stays/3vsj-zpk7>.
- [26] Cambridgeshire Insight Open Data, Cambridge city parking data[EB/OL]. (2019-11-12) [2023-09-18]. <https://data.cambridgeshireinsight.org.uk/dataset/cambridge-city-parking-data>.
- [27] LIU X Y, JIANG J H, LI L. Computation offloading and task scheduling with fault-tolerance for minimizing redundancy in edge computing[C]// IEEE International Symposium on Software Reliability Engineering Workshops. Wuhan: IEEE, 2021: 198-209.

(责任编辑: 杨耀辉 英文审核: 费伦科)