摘　要

混合云通过整合公有云丰富的资源和私有云严密的数据隐私保护，成为处理隐私数据的理想平台。然而，混合云中调度隐私任务仍存在安全性与效率难以平衡问题，由私有云独立处理虽安全但资源受限，而公私云协作处理虽高效但可能泄露隐私。已有研究依靠隐私约束保证安全性，例如限制数据只在私有云处理，或虽允许协作处理但只能使用高计算开销的隐私加密算法。然而，仅依靠隐私约束难以平衡安全与效率，一方面，即便满足跨云协作隐私约束条件的数据在实际处理中仍存在泄漏的风险；另一方面，在混合云高负载时，固定的约束导致资源利用率下降，进而使任务完工时间延长、效率下降。

针对混合云隐私任务调度中安全性与效率难以平衡的问题，本文构建了混合云中动态细粒度的隐私任务调度模型，为安全和效率的平衡提供完工时间、安全性与成本三大量化指标。再使用多目标隐私任务调度算法，同时优化这三个指标，并输出一组在完工时间（效率）与安全性上满足非支配条件的调度方案集合。可以通过用户手动干预或者自动的多标准决策等方法，从调度方案集合中选择安全与效率间的平衡点。

本文主要工作包括：

提出了一种动态细粒度的隐私任务调度模型。该模型通过线性工作流技术完善了公私云协作的动态任务建模方法，支持在调度过程中灵活插入加密与验证子任务，充分利用混合云资源，从而提升混合云任务执行效率。同时，通过引入细粒度隐私标签机制，将数据拥有者多样化的隐私需求映射为对特定加密算法组合的要求，从而支持处理来自不同数据拥有者的多样化隐私数据。此外，基于数据大小与加密算法安全系数，提出了量化评估混合云系统安全性的指标。最终，将隐私任务调度问题建模为完工时间、安全性及成本的三目标优化问题，为平衡效率与安全提供理论依据。

设计了一种考虑卸载窗口的非支配排序遗传算法 NSGA-OW，完成完工时间、安全性与成本的协同优化。针对跨云协作中虚拟机空闲时段问题，提出了卸载窗口首次适应填充算法（OW-FF），通过动态检测并利用任务卸载至公有云时的等待时段，插入后续任务，提高私有云资源利用率。还针对混合云隐私任务调度问题改进了多目标遗传算法，改进的遗传算法框架 NSGA-OW 采用混合编码策略，设计了虚拟机分块多点交叉算子与负载感知变异算子，加速收敛并提升调度质量。算法优先优化安全性与完工时间目标，满足用户偏好，最终输出一组在完工时间、安全性与成本上最优的 Pareto 前沿集合，以便用户选择安全性与效率平衡点。

通过实验验证，本文的模型与算法表现出了显著的优势。与传统未考虑公私云

协作、细粒度隐私标签以及多目标优化的调度方法相比，本文方法在完工时间上降低了 52.7%，在安全性指标上提升了 31.9%，同时保证了成本不劣于对比算法。此外，NSGA-OW 算法在收敛速度与 Pareto 前沿解集质量上均优于传统多目标元启发式算法，进一步验证了其有效性。

关键词：云计算，调度算法，多目标优化，数据隐私

ABSTRACT

Hybrid cloud, by integrating the abundant resources of public cloud and the stringent data privacy protection of private cloud, has emerged as an ideal platform for processing private data. However, scheduling privacy tasks in hybrid cloud environments still faces challenges in balancing security and efficiency. Processing tasks solely in the private cloud ensures high security but suffers from resource limitations, while public-private cloud collaboration improves efficiency but introduces potential security risks. Existing studies rely on privacy constraints, such as restricting data processing to the private cloud or enforcing computationally intensive encryption algorithms during collaboration. However, such rigid constraints struggle to achieve equilibrium between security and efficiency: on one hand, data complying with privacy constraints may still face security risks during processing; on the other hand, fixed constraints under high hybrid cloud workloads leading to prolonged task completion times and reduced efficiency.

To address the challenge of balancing security and efficiency in hybrid cloud privacy task scheduling, this paper proposes a dynamic fine-grained privacy task scheduling model, which provides three quantitative metrics: makespan, security, and cost to guide the equilibrium between safety and efficiency. A multi-objective privacy task scheduling algorithm is further developed to optimize these metrics simultaneously, generating a set of non-dominated scheduling solutions that satisfy the trade-off between makespan (efficiency) and security. Users may select the optimal balance point from this Pareto-optimal solution set through manual intervention or automated multi-criteria decision-making methods.

The main contributions of this work include:

A dynamic fine-grained privacy task scheduling model leverages linear workflow technology to construct a dynamic task framework for public-private cloud collaboration. This model supports flexible insertion of encryption and verification subtasks during scheduling, enhancing hybrid cloud efficiency. Additionally, a fine-grained privacy labeling mechanism dynamically adapts multiple privacy encryption algorithm groups by mapping data owners ' privacy requirements to specific cryptographic combinations. By integrating data size and encryption security metrics, a quantifiable hybrid cloud security assessment framework is established.

The privacy task scheduling problem is formulated as a three-objective optimization framework comprising makespan, security, and cost, thereby establishing a theoretical foundation for balancing the trade-offs between system efficiency and data security.

A novel non-dominated sorting genetic algorithm with offload window optimization (NSGA-OW) is designed to co-optimize makespan, security, and cost. For virtual machine idle time during cross-cloud collaboration, the Offload Window First-Fit Fill (OW-FF) algorithm dynamically detects and utilizes waiting intervals for task offloading to insert subsequent tasks, improving private cloud resource utilization. Enhanced with hybrid encoding and novel genetic operators VM block-based multi-point crossover and load-aware mutation, the NSGA-OW framework accelerates convergence and enhances scheduling quality. Prioritizing security and makespan optimization based on user preferences, the algorithm outputs a Pareto front of optimal solutions, allowing users to select security-efficiency trade-offs.

Experimental validation demonstrates the superiority of the proposed model and algorithms. Compared to traditional methods lacking public-private cloud collaboration, fine-grained privacy labeling, or multi-objective optimization, our approach achieves a 52.7% reduction in makespan and a 31.9% improvement in security metrics while maintaining cost competitiveness. Furthermore, NSGA-OW outperforms conventional multi-objective metaheuristic algorithms in convergence speed and Pareto front quality, further validating its effectiveness.

Keywords: Cloud computing, Scheduling algorithms, Multiobjective optimization, Data privacy

第一章　绪论

1.1 研究背景及意义

近年来，随着数字经济的迅猛发展，云计算作为关键信息基础设施，在政府治理与企业数字化转型中发挥着核心作用。根据中国信息通信研究院发布的《云计算蓝皮书（2024 年）》，全球云计算市场正以 18.6% 的年复合增长率快速扩张，预计到 2027 年市场规模将突破万亿美元[1]。在这一背景下，混合云作为云计算的重要模式，通过整合私有云的安全可控性与公有云的弹性扩展优势，成为具有隐私数据处理需求企业数字化转型的重要选择。企业可将隐私数据部署于私有云以确保完全控制权，同时利用公有云的弹性算力进行临时扩容，既满足隐私安全要求，又能快速响应市场需求。随着企业对混合云隐私数据处理需求的持续增长，华为云、阿里云、亚马逊 AWS 等主要厂商纷纷推出成熟的混合云解决方案，进一步推动了混合云技术的普及与应用。总之，混合云凭借其架构中独特的可控性和灵活性等优势，逐渐成为处理隐私数据的理想平台。

尽管混合云在隐私数据处理上具备灵活性和可控性优势，但混合云中的隐私安全依旧是制约大规模推广部署的重要挑战。首先，混合云需管理多个云平台，其复杂的资源组成导致攻击面显著扩大，数据泄露事件如酒店消费信息和订票网站信息，表明混合云环境中的部分资源存在安全风险；其次，混合云必须满足严格的合规性要求，在隐私数据传输过程中采用符合中国《个人信息保护法》和欧盟 GDPR 等法规的加密技术，并针对不同监管需求动态选择适配的加密策略；最后，特定行业的隐私数据需要强化保护措施，例如医疗行业在将患者记录迁移至云平台时，必须满足 HIPAA 规范，以确保患者健康信息的隐私与安全得到有效保障。为应对这些挑战，现有混合云任务调度研究通过引入隐私加密算法，围绕数据传输保密性、用户认证以及数据完整性三个方面构建安全机制，从而显著提升混合云的隐私保护性能[2-7]。除此之外，存在一些尚未在混合云任务调度中被充分利用的新型隐私加密算法。例如，同态加密适用于投票系统中的密文计算，数据脱敏技术在大数据分析场景中表现突出，而局部模糊处理则可以有效遮挡人物图像、车牌路牌等敏感信息。这些算法虽然能够提供更强的安全机制，然而，这些新型隐私加密算法存在复杂加密流程、额外计算开销以及仅适用于特定类型隐私数据的局限性，使得现有隐私任务调度算法难以有效整合和应用这些技术。综上所述，通过在混合云任务调度中引入隐私加密机制，可以部分解决混合云部署中遇到的隐私安全问题，提高混合云的部署率。

混合云在隐私数据处理领域展现出的独特潜力使其成为全球数字化转型进程中

的重要支撑力量，各国政府与企业不仅通过政策框架加速技术落地，更将其纳入国家数字化发展战略的核心议程。美国发布《国家网络安全战略》，强化云基础设施安全与弹性建设；欧盟在《欧洲 2030》战略规划中提出 75% 的企业需采用云计算、大数据与人工智能技术，以提升数字竞争力；中国印发《算力基础设施高质量发展行动计划》，推动云服务模式整合算力资源，实现多元算力高效协同。随着隐私数据处理需求的持续增长，混合云环境下的隐私任务调度研究成为提升数据处理效率、保障数据安全的关键课题，其重要性日益凸显。

1.2 国内外研究现状

本章从隐私任务调度中保护隐私数据的方法，以及云计算中单目标和多目标任务调度的研究进展两个方面，对现有文献进行系统性梳理与对比分析，从而明确当前研究的核心问题与发展方向。

1.2.1 隐私任务调度研究现状

隐私任务调度是隐私数据处理中的关键问题，现有隐私任务调度研究主要通过资源限制、安全评估与安全加固三种方法保护隐私数据安全：资源限制方法通过限制敏感数据只使用可靠的计算资源处理，而将非敏感数据分配到所有计算资源；安全评估则根据隐私数据的风险级别和计算资源的安全评级，动态选择计算资源；安全加固机制利用加密算法增强数据安全性，允许更多任务在多样化的计算环境中执行，以提高资源效率。这三种方法之间存在着从“静态隔离”到“动态评估”再到“主动加固”的递进关系。

在资源限制方法研究方向，当前研究通过设定任务执行位置约束实现隐私数据的保护。[2-3,8-13]，通过对任务施加资源限制，确保隐私任务仅在安全的平台中执行，避免隐私数据泄露。

Stavrinides 等人[11]构建的混合云隐私调度框架，将包含敏感数据的任务袋分配至私有云资源，考虑截止时间和安全约束的任务调度问题，优化资源分配可以节约高成本并确保数据安全。Sharif 等人[10]研究了隐私工作流调度问题，重点关注资源限制策略以实现隐私保护。该研究以医疗环境中的隐私任务调度为例，通过将含有患者身份信息的任务分配至私有云执行，确保了数据安全性。同时，利用公有云资源处理非敏感任务，在保障隐私的基础上有效降低了成本并兼顾了任务截止时间约束。张敏禹[13]针对 Spark 工作流任务提出的两阶段协同框架，通过引入弹性资源抢占机制与动态时间窗松弛技术，在保障隐私数据在私有云处理的前提下，优化了截止期达成率并提高了资源消耗均衡性。

资源限制方法虽通过执行位置约束实现了隐私保护，但其也存在混合云资源利

用不充分的不足。制约了不同计算环境之间的协同能力与混合云资源动态调配的优势，难以适应复杂多变的负载与安全态势。Stavrinides[11]的研究表明，当任务中隐私任务的比例从 25% 上升至 75% 时，截止时间违反率提升了 37%，凸显了资源限制策略在动态环境下的局限性。为了更充分利用混合云资源，另一些研究考虑安全评估，通过评估服务提供商与的可信度与隐私数据安全需求[14-19]，将一部分安全需求数据的任务交给可信的云服务提供商，从而更充分的利用混合云资源。

Asghari 等人[16]构建的云服务组合方法通过多维度服务分级与信任评级体系，实现隐私保护级别与云服务商安全能力的自动化匹配，提升了整体服务质量。针对跨国场景中的合规性挑战和多级数据保护要求，Zhou 等人[14]设计一种进程映射方法，通过动态约束条件的组合优化算法，在任务调度和数据跨境传输符合隐私保护法规与传输时延控制，通过数学建模和启发式算法设计，优化了通信成本和部署性能。在多方数据融合场景领域，刘圣龙等人[20]系统梳理了多方协同计算架构下的隐私保护技术演进路径，提出覆盖数据全生命周期的安全技术，其分析表明后续研究应着重解决跨域信任建立、计算-通信开销优化等关键问题。邓慧娜[21]提出车联网边缘计算环境的轨迹隐私动态防护模型，通过马尔可夫驱动的任务迁移策略与迁移成本约束的联合优化，在保证低时延服务供给的同时，建立面向边缘计算攻击的多层级隐私防护体系。

安全评估研究通过不同方案确定了服务提供商与的可信度与隐私数据安全需求，从而充分利用混合云资源。然而，当前研究是针对特定应用场景设计的隐私度量方法。未能有效考虑不同地域的隐私法规或用户对数据的细粒度隐私需求[22]，设计一种支持不同数据的细粒度的安全策略。此外，研究不止考虑资源限制和评估，还通过引入数据加密与认证机制增强混合云资源的安全性。增强安全性可以使公有云资源有能力处理部分隐私数据。采用固定加密算法[5-6])，虽能减少隐私泄露风险有些研究考虑在任务调度中考虑对数据加密保护[2-7]，与加密算法结合，保护数据安全性，从而将更多任务交给公有云处理。

雷剑[23] 提出隐私工作流增强框架采用分级加密策略，依据任务敏感度动态选择不同等级的加密算法，增强了隐私工作流在混合云中传输数据的安全性，在保持数据隐私性的前提下提升混合云整体资源利用效率。Li 等人[24]针对当前移动边缘计算领域的研究多聚焦于任务卸载与性能优化，却普遍忽视企业多媒体在无线传输中的安全风险的问题。通过 Lyapunov 优化模型动态调配加密策略与计算资源，在保证安全性的前提下将移动边缘计算环境下降低传输时延与能耗。Chen 等人[4]提出安全敏感中间数据的工作流调度框架，通过预计算资源空闲时段进行任务复制减少数据传输与加密对后续任务启动时间的影响，实现实现安全约束下完工时间与成本的联合优化。Hammouti 等人[25]提出了一种针对混合云环境的任务调度策略，通过引入数据加

密与认证机制增强混合云资源的安全性和经济性，重点研究了这些安全服务对任务总成本和截止时间的影响。

这些研究考虑在任务调度中引入安全增强机制，考虑使用加密算法对数据加密，并对结果进行验证。然而这些研究在这些安全增强机制产生的额外开销考虑不足。实际上，在加密与解密中会引入额外的任务依赖，例如公有云的处理任务依赖于私有云的加密任务，而私有云的验证任务又依赖于公有云的处理任务。这种额外的依赖会造成任务结构和执行顺序发生动态变化，同时私有云和公有云之间会交换数据，而现有研究对于这方面的建模不足。

还有些研究有研究考虑在调度期间，任务拓扑可动态改变的调度任务。Stavrinides等人研究了分布式系统中具有动态变化结构的线性工作流[26-27]（Linear Workflow, LW）及其安全感知调度技术[28]。他们提出了一种基于条件允许部分计算的调度方案，并在高风险任务需要安全资源处理、低风险任务可灵活分配的背景下，设计了两种路由技术。通过仿真实验，研究了不同安全资源比例及任务风险概率对性能的影响，还验证了动态任务插入与删除策略在提升系统性能方面的有效性。然而，他们的工作未将线性工作流应用于混合云协作任务场景，也未能通过动态插入加密与验证任务将高计算开销隐私加密算法与混合云的高效整合。

通过对隐私任务调度相关研究的梳理，本文发现现有的在混合云环境中隐私调度研究面临若干关键挑战。首先，仅限私有云处理隐私任务的调度策略[5,8]虽在一定程度上保障了数据安全，但却导致资源利用率低下，未能充分发挥混合云的优势。其次，现有研究在安全评估方面[6,14]未能有效考虑用户对数据的细粒度隐私需求，限制了策略在实际应用中的灵活性与有效性。此外，考虑隐私加密算法的任务调度研究因未能适应动态任务结构变化[29]，且传统 DAG 模型也难以表示由加密引发的动态拓扑变化[11]，从而导致该调度方法的实际有效性受到制约。为解决这一问题，本文还探索了其他领域中能够表示动态任务结构的建模方法[26]，以弥补隐私加密算法在动态任务模型上的不足。最后，现有研究多将隐私安全作为约束条件，而缺少对安全性的量化评估，难以在安全与效率之间实现有效权衡[2-3,30]。综上所述，当前隐私任务调度研究在资源利用、细粒度隐私需求、公私云协作中精准动态建模以及安全性量化评估等方面存在不足，难以满足混合云环境中日益复杂的隐私保护与高效处理需求。

1.2.2 云计算任务调度现状

本小节通过单目标与多目标优化两种优化问题建立方式分析云计算任务调度现状。

单目标优化作为云计算任务调度领域的传统方法，主要通过对时间、成本或能效等单一核心指标的优化实现资源高效配置[31]。研究通常围绕单一目标建立数学模

型，同时将资源、截止时间以及安全性等维度作为约束条件进行折中处理，以此兼顾系统多方面的性能需求。

例如，在混合云安全调度场景中，多数研究通过嵌入安全性约束，将隐私敏感任务强制调度至私有云执行。Wang 等人[3]提出的调度策略即为典型代表。其将安全约束与最早截止时间优先调度算法融合，实现任务完成率与安全性的联合优化，实验数据显示该方法在数据敏感场景下可降低 15% 的任务丢弃率。而在边缘计算场景中，张智峰[32]利用软件定义网络实现任务卸载与资源动态调配，并通过资源约束与执行位置约束兼顾了边-端服务器的负载均衡，实现了 19% 的时延降幅。

以上研究表明，单目标优化方法通过联合次要目标作为约束的策略，虽能简化问题且在某些场景下取得良好调度效果，但在混合云隐私调度等复杂场景中存在显著局限：静态约束模式无法适应动态需求，例如在安全风险时期，用户希望选择更高的安全保护。且未准确处理目标间的冲突。这些缺陷推动了任务调度研究向多目标协同优化方向的转变。

多目标优化涉及同时优化两个或多个冲突的目标。在云计算中，任务通常需要在不同目标之间进行权衡，例如在成本、性能、能源效率和用户满意度之间取得平衡，Khan 的[33]综述指出，云协同研究已涵盖 20 类调度指标，其中安全合规性、隐私保护强度等新兴指标正逐步成为目标函数的重要构成。目前的多目标优化问题有两种解决方案，第一种是通过加权求和法将多目标问题转化为单目标优化问题再使用单目标调度算法进行调度。另一种是使用多目标调度算法，会生成一组帕累托最优解，其中没有一个目标可以改进而不使另一个目标恶化。

Laili 等人[34]，针对工业物联网场景下大规模任务的云-边缘协作调度问题，提出了一种基于加权求和的多目标优化方法，通过将任务完成时间与能源消耗合并为单一目标函数实现高效调度。该方法结合并行分组-合并进化算法，提升了任务分配的均衡性与计算效率。如 Sun 等人[9]针对混合云工作流调度问题，构建总延迟-私有云能耗-公有云成本的三目标优化模型，提出基于鲈鱼群算法的双阶段优化策略，通过探索 Pareto 前沿解集与贪心搜索的协同机制，在仿真实验中实现能耗与经济成本的均衡下降。[35]NSGA-II 非支配排序多目标优化能力的基础上，通过定性指标确定搜索方向，加快了优化算法在用户感兴趣的目标上的搜索速度。Mousavi 等人[36]基于云计算任务调度算子，论文在经典 NSGA-II 算法中引入了一个新的重组算子，从而提出了一种定向非支配排序遗传算法（D-NSGA-II）。该新算子能够调节种群的选择压力，有效平衡算法在全局探索与局部开发之间的能力。Mangalampalli 等人[37]提出一种基于深度强化学习的多目标优化调度框架，通过优先级感知与 DQN 动态决策机制显著降低科学工作流调度场景的完工时间与能源消耗。

根据以上研究可以发现，使用加权求和方法的多目标优化算法将多目标问题转

化为单目标优化问题，但其调度效果高度依赖于加权系数的设定，而加权系数往往难以准确确定，难以适应本文考虑的隐私与效率难以权衡的场景。而基于元启发式算法的多目标优化方法通过模拟生物进化过程中的选择、交叉与突变机制，能够同时优化多个目标，并为用户提供一组可比较的非支配调度方案，从而在隐私任务调度场景中展现出更强的适应性和优势。

本小节通过对混合云中单目标和多目标任务调度的研究进行分析，发现单目标调度在平衡相互冲突优化目标方面存在不足，而基于加权求和的多目标优化算法又面临难以确定加权系数的挑战。基于上述问题，本研究选择基于元启发式算法的多目标优化方法，以有效权衡效率与安全性目标。这一方法不仅能够克服传统单目标调度的局限，还能避开加权系数设定的问题，为混合云任务调度问题提供质量更高的调度方案。

1.3 论文研究内容和主要工作

本文研究混合云中隐私任务调度问题，针对已有混合云隐私任务调度研究中安全性与效率难以权衡的问题，以最小化完工时间、成本并最大化系统安全性为目标，建立混合云中动态细粒度的隐私任务调度模型，建立多目标优化问题。并针对该优化问题，设计多目标隐私任务调度算法，同时优化三个目标，并为用户提供一组可比较的非支配调度方案。最后，通过实验验证模型与算法的有效性。本文的主要研究如下：

1. 提出一种混合云动态细粒度隐私任务调度模型，旨在解决隐私任务调度中的效率与安全性平衡难题。该模型通过线性工作流技术构建公私云协作的动态任务模型，支持在调度过程中灵活插入加密与验证子任务，提升跨云协作能力；同时引入细粒度隐私标签机制，动态适配多类隐私加密算法组，满足差异化隐私需求。以数据主权合规为例，模型依据中国、美国和欧洲三地的隐私加密算法偏好，为不同数据选择对应的加密算法组。结合数据大小和算法安全性，量化评估混合云系统的安全水平，并将任务调度问题建模为完工时间、安全性及成本的三目标优化框架，为效率与安全的权衡提供理论依据。

2. 设计了一种考虑卸载窗口的非支配排序遗传算法 NSGA-OW，完成完工时间、安全性与成本的协同优化。通过制定虚拟机空闲窗口检测启发式规则提升资源利用率，并改进遗传算子加快收敛速度且改善调度质量，最终输出一组在完工时间、安全性与成本上最优的 Pareto 前沿，为用户提供多样的调度方案。

3. 通过实验验证，本文的模型与算法表现出了显著的优势。与传统未考虑协作、细粒度隐私标签以及多目标优化的方法相比，本文方法在完工时间上提高了

52.7%，在隐私安全性上提升了 31.9%，同时保证了成本不劣于对比算法。此外，NSGA-OW 算法在收敛速度与 Pareto 前沿解集质量上均优于传统多目标元启发式算法，进一步验证了其有效性。

1.4 论文结构安排

本文对混合云中隐私任务调度问题进行研究，考虑混合云环境中细粒度隐私数据管理与公私有云间协同处理任务建立混合云隐私调度模型，并将优化问题建模为三目标优化问题。并针对考虑隐私与协作的任务调度三目标优化优化问题，设计并改进多目标优化算法求解。并进行充分实验验证所提出的模型和算法的有效性。论文的组织结构如图1.1所示。本文主要划分为以下章节：

第一章，绪论。首先分析混合云作为隐私数据处理平台的架构优势，并说明混合云隐私任务调度的价值。通过系统梳理云计算任务调度与隐私保护技术的研究现状，按优化目标和隐私策略进行分类对比，指出现有研究在多目标均衡优化、细粒度隐私适配以及公私云协作方面的不足。通过对比分析，明确本文研究方向。最后提出本文主要研究内容，并确定文章的组织结构。

第二章，介绍相关基础知识。涵盖多目标优化、混合云架构及其隐私保护技术。重点分析 NSGA-II 的遗传算子与精英选择策略在离散优化中的应用，探讨混合云的特点及隐私保护技术的应用，为后续研究提供理论支持。

第三章，提出混合云动态细粒度隐私任务调度模型。解决安全性量化不足、隐私需求适配困难及跨云协作效率低下的问题。通过建立混合云资源模型、设计动态任务模型、构建隐私标签机制及安全性量化评估模型，最终将问题形式化为多目标优化框架。

第四章，设计多目标优化算法 NSGA-OW。针对虚拟机空闲资源利用率低的问题，提出卸载窗口首次适应填充算法 OW-FF，并设计混合编码方案与负载感知的遗传算子组合，优化完工时间与运营成本。最后，提出动态偏好精英选择策略，提升关键目标的搜索效率。

第五章，通过实验验证模型与算法的有效性。从多目标优化性能、任务量与数据规模影响、隐私需求影响及加密开销影响四个方面展开分析。实验结果表明，所提模型与算法在收敛速度、解集质量及调度效率方面均优于对比方法。

第六章，总结和展望。对本文所提出的调度方法研究进行分析总结，说明现有研究中的不足之处，并对未来的研究方向进行预测。

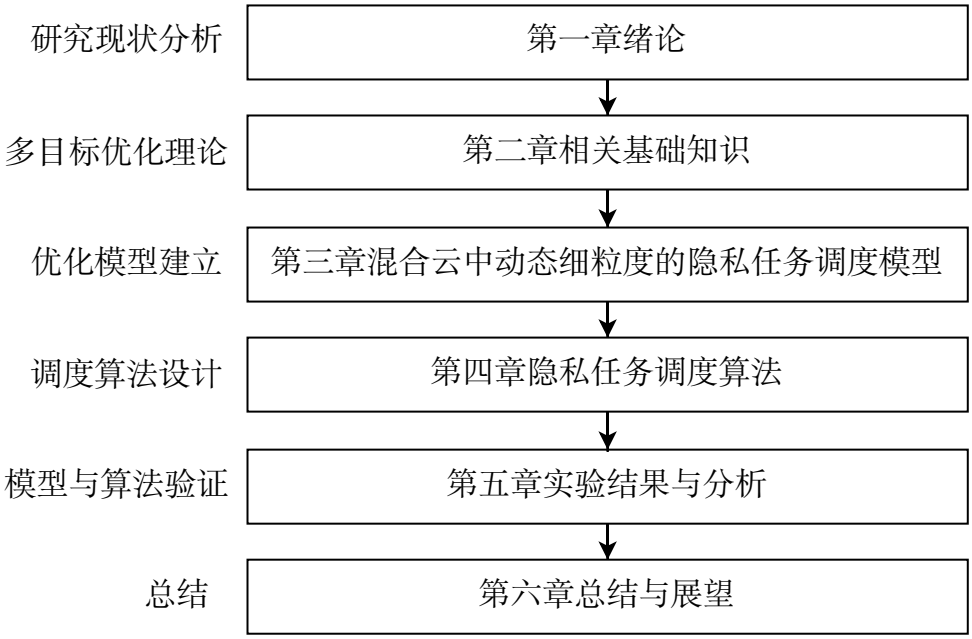

图 1.1　论文组织结构图

1.5 本章小结

本章介绍了混合云作为隐私数据处理平台的研究背景与意义，系统梳理了当前研究现状，指出了现有研究在多目标优化方面尚存不足，未能有效平衡安全性与效率关系，同时在细粒度隐私保护与公私云精准协作机制方面也存在问题。基于此，本研究明确了解决上述问题的方向，提出了具体研究内容，并确定了文章的组织结构框架。

第二章　相关基础知识

本章介绍了混合云的基本概念、部署方式以及现有混合云环境下任务调度研究中常用的隐私加密算法；阐述了用于平衡混合云安全性与效率的多目标优化理论；并以 NSGA-II 算法框架为例进行分析总结，给出元启发式多目标优化算法的原理及其改进方向。为后续章节中隐私任务调度模型的设计与算法优化提供了理论基础。

2.1 混合云概述

云计算作为基于互联网的计算服务模式，通过整合分布式服务器资源向用户提供按需访问的计算能力。这种模式使得计算资源的获取如同水电等公共设施般便捷，其特征是将物理硬件与软件资源抽象为标准化服务，用户仅需关注服务功能需求而无需管理底层基础设施。2006 年亚马逊推出的弹性计算云（EC2）标志着云服务的突破性进展，其虚拟化计算资源的租赁机制至今仍是行业标杆。

混合云作为云计算发展的重要方向，通过结合私有云、本地设施和公有云构建出灵活的资源管理模式。还能借助本地就近部署减少网络延迟还可以充分挖掘既有设备的潜力，从而增强系统稳定性并节省运营成本。混合云通过统一平台管理各类资源，当业务需求突然增加时，其独有的” 云爆发” 功能会自动调用公有云资源，及时应对短期的高强度计算需求。这种架构还能满足数据管理法规和企业安全的需求，让隐私敏感的数据始终留在安全环境中。

2.1.1 基本概念与术语

根据 NIST 标准定义，混合云基础设施由两个或以上独立云实体构成，其中既包含公有云与私有云的协作形式，也支持多个公有云或多个私有云的组合形式。鉴于实际应用中私有云与公有云整合的普遍性，本文所研究的混合云特指通过统一任务调度系统实现单公有云和单私有云资源协同的云架构。该架构通过统一的管理平台消除异构环境的差异，既支持私有云资源池的敏感数据处理能力，又可动态调度公有云资源池满足弹性扩展需求，最终在控制成本的同时提高混合云的安全性，并满足隐私数据安全需求。

本文所使用的混合云系统相关技术术语介绍如下：

- **公有资源池**：由第三方服务商托管的多租户共享型云计算资源集合，通过互联网提供标准化服务。其采用弹性架构设计，可根据需求动态调整资源规模，具有按需付费、全局可用性强、运维成本低等特点，适合处理低敏感类数据、波

动性业务负载。

- **私有资源池**：组织内部专用的物理或虚拟化资源集群，通过本地数据中心或隔离云环境实现全生命周期管理。其具备物理隔离、定制化配置和细粒度安全管控特征，可确保核心业务数据处理过程的合规性，主要服务于高敏感性或受监管的固定工作负载。

- **混合云资源**：整合公有资源池与私有资源池形成的协同计算架构，通过隐私加密算法实现跨域资源调度。具体而言，采用隐私保护算法对隐私数据进行预处理，将加密后的数据调度至公有资源池的虚拟机执行，同时将核心数据保留在私有资源池处理，以确保安全性[23,30]。

- **虚拟机**：基于虚拟化技术实现的逻辑计算单元，可跨异构硬件平台部署独立操作系统环境。在混合云场景中，私有资源池的虚拟机通常采用定制化安全增强配置，而公有资源池的虚拟机则强调快速部署与横向扩展能力，二者通过统一镜像标准实现工作负载的混合编排与跨云处理。

2.1.2 混合云的部署与应用

混合云的核心应用价值体现在其“云爆发”功能中。当本地私有云资源接近满载时，系统能够自动将额外的计算任务转移至公有云，这种动态资源分配机制有助于避免服务中断。例如电商促销活动或在线政务服务等具有明显峰谷特征的应用场景，混合云通过预先设定的规则自动扩展公有云资源，既能维持核心业务在私有环境的稳定运行，又可临时调用云端资源处理辅助任务，从而在保障服务质量的同时避免过度硬件投资。

部署混合云需遵循三个基本原则：首先根据数据敏感度对业务进行分类管理，将涉及核心隐私的金融交易等系统固定部署在私有云的专用虚拟机中，而数据分析等非敏感任务则采用公有云资源动态部署。其次建立统一的监控体系，实时评估私有云资源使用状况，当负载指标如处理器或内存使用率或者完工时间达到预设阈值时，启动公有云虚拟机来处理新增负载。最后确保安全策略的跨云同步，在扩展公有云资源时同步实施数据传输加密、访问权限控制等防护措施，形成完整的隐私保护。

该架构在实际应用中展现出多个典型场景：教育机构将核心教学数据保留在私有云，而将视频直播等高峰时段的转码任务分配至公有云虚拟机；医疗信息系统将患者原始数据存储在私有云加密空间，同时将脱敏后的样本上传至公有云进行科研分析；企业将业务系统日常备份至公有云存储，通过定期快照确保应急恢复能力。这些模式共同印证了混合云在安全性和灵活性上的平衡能力。

混合云的部署特性可归纳为三个层面：通过私有云实现敏感数据的物理隔离，满

足法规对隐私保护的强制性要求；借助公有云构建弹性计算资源池，用于处理突发性和非关键任务；最终通过统一调度机制形成跨云协同能力。当私有云负载接近饱和时，系统自动将可公开处理的任务转移至公有云，这种智能调度策略在税务申报、节日促销等周期性高峰场景中展现出显著优势，既保障了核心系统的稳定运行，又降低了基础设施的闲置成本。

2.1.3 混合云中隐私加密算法

在混合云环境下的任务调度研究中，数据隐私保护主要考虑用户认证、数据机密性与完整性验证三个方向。任务调度系统通过量化评估隐私加密算法的安全等级（以 0 至 1 区间表征防护强度，1 为最高防护），动态选择隐私加密算法，以实现精准的安全调控。这一机制在任务调度领域的相关研究中得到了广泛应用[2-3]。

用户认证模块根据防护需求选择认证协议，如 HMAC 或 AES-CMAC，以实现高效的身份验证。机密性保护通过动态适配加密策略实现，非敏感数据采用 SEAL 等流式加密以提高效率，而高机密性数据则使用多层加密方案如 IDEA 与混合密钥管理。完整性验证基于哈希函数分级实现，基础场景采用 MD 系列算法，关键业务则使用 SHA 系列以增强防篡改能力，尽管计算开销较大但安全性显著提升。私有云凭借物理隔离机制为高敏感任务提供最高防护等级。调度系统通过动态选择最优加密组合策略，在保障安全性的同时平衡任务处理效率，实现了混合云安全性与效率的协同优化。

近年来，随着密码学、可信硬件与分布式计算等新技术的融合，一些创新性隐私加密算法为混合云任务调度开辟了新路径，但目前尚未得到广泛应用。

基于密码学理论的多方安全计算协议通过分布式协作实现加密状态下的联合运算，适用于跨云数据融合场景，但其复杂计算特性难以满足实时性要求较高的任务。可信执行环境技术利用硬件层的隔离机制，在私有云专属区域保护敏感计算过程，并通过可控的数据流向公有云提升处理效率，但其依赖特定硬件架构限制了部署灵活性。数据脱敏技术通过对敏感数据进行变形处理，在基因分析等统计类任务中展现优势，但存在信息损失带来的计算偏差风险。分布式学习框架如联邦学习，通过本地模型训练与加密参数交互实现全局更新，但在新型攻击面前仍需结合其他加密手段增强防护。然而，这些算法的复杂加密流程、额外计算开销以及对特定数据类型的依赖，限制了其在现有隐私任务调度算法中的有效整合与应用。

2.2 多目标优化

2.2.1 多目标优化问题与评价指标

在现实工程与科学问题中，多目标优化是一个普遍存在的挑战，尤其是在资源分配、调度等场景中，通常需要同时优化多个相互冲突的目标，例如成本、效率与风险。这些目标往往无法通过单一方案同时达到最优，因此传统的单目标优化方法如加权求和法虽然简单，但其依赖于先验的权重设定，难以权衡安全性与效率之间的关系。为了解决这一问题，多目标优化问题通过生成 Pareto 前沿解集，为决策者提供了多样化的选择空间。以 Sun 等人[9]在混合云任务调度中的研究为例，一个典型的三目标优化问题可以形式化为以下数学模型：

$$\begin{aligned} \text{Minimize} \quad & f_1(X),\ f_2(Y),\ f_3(Z) \\ \text{subject to} \quad & g_i(X,Y,Z) \le 0, \quad i = 1,2,\ldots,m \\ & h_j(X,Y,Z) = 0, \quad j = 1,2,\ldots,p \\ & X \in \mathcal{X},\ Y \in \mathcal{Y},\ Z \in \mathcal{Z} \end{aligned} \tag{2-1}$$

其中，$f_1(X)$、$f_2(Y)$、$f_3(Z)$ 是需要最小化的三个冲突目标，$g_i(\cdot) \le 0$ 和 $h_j(\cdot) = 0$ 分别表示不等式约束和等式约束，而 X、Y、Z 为决策变量，分别属于可行域 $\mathcal{X}$、$\mathcal{Y}$、$\mathcal{Z}$。

在多目标优化问题中，绝对最优解通常不存在，因此引入 Pareto 支配关系与 Pareto 前沿的概念来描述解的优劣。具体而言，对于一个解 $\mathbf{u}$，若其在所有目标上均优于另一个解 $\mathbf{v}$，即满足 $\forall i, f_i(\mathbf{u}) \le f_i(\mathbf{v})$ 且存在至少一个目标 j 使得 $f_j(\mathbf{u}) < f_j(\mathbf{v})$，则称 $\mathbf{u}$ 支配 $\mathbf{v}$，记为 $\mathbf{u} \prec \mathbf{v}$。Pareto 最优解集则是指所有不被其他解支配的可行解的集合，其对目标空间的映射被称为 Pareto 前沿。这一理论为多目标优化问题的求解提供了理论基础，强调了在多个冲突目标之间寻找平衡解的必要性。

为了量化多目标优化算法的性能，常用的评价指标包括世代距离（Inverted Generational Distance, IGD）和超体积（Hypervolume, HV）。这两个指标在多目标优化中具有重要的评价意义，常用于衡量算法在收敛性与解集质量方面的表现。

反转世代距离（Inverted Generational Distance, IGD）是一种常用的评价指标，其通过计算真实 Pareto 最优解集到非支配解集的平均距离，提供了一个对算法收敛性与解集质量的综合评估。，其计算公式为：

$$\text{IGD}(\mathcal{P}, \mathcal{P}^*) = \frac{\sum_{i=1}^{F^*} d(v_i, \mathcal{P})}{F^*} \tag{2-2}$$

其中，$\mathcal{P}$ 为算法解集，$\mathcal{P}^*$ 为真实 Pareto 前沿，F^* 为真实 Pareto 前沿中解的个数，v_i 为真实 Pareto 前沿中的第 i 个解，$d(v_i, \mathcal{P})$ 表示真实 Pareto 前沿中第 i 个解到算法解集 $\mathcal{P}$ 的最近欧氏距离。IGD 指标通过计算真实 Pareto 前沿中每个解到算法解集的平

均最短距离，能够更全面地评估解集的收敛性和多样性。

超体积（HV）则是通过计算非支配解集与参考点在目标空间中所围成的超立方体体积来评估解集的质量，对于三目标优化问题其定义如下：

$$\mathrm{HV}(\mathcal{P}, r) = \lambda_3 \left(\bigcup_{x \in \mathcal{P}} \prod_{i=1}^{3} [f_i(x), r_i] \right) \tag{2-3}$$

其中，λ_3 表示三维空间中的勒贝格测度，即体积，$f_i(x)$ 表示第 i 个解在第 i 个目标上的归一化值，r 为参考点。HV 值越大，表明解集的覆盖范围与分布均匀性越好。虽然 HV 指标能够综合反映解集的收敛性与多样性，但其计算复杂度较高，且参考点的选择对结果准确性有一定影响。

此外，世代距离（GD）也可用于衡量算法解集与真实 Pareto 前沿之间的逼近程度，通过计算算法解集与真实 Pareto 前沿解集之间的最小欧氏距离来衡量解的收敛性，其值越小表明解的收敛性能越优。同时，GD 与 IGD 对真实 Pareto 最优解集的依赖性较强，若选取的最优解集不准确，可能会导致对算法性能的错误评估。

解决多目标优化问题的主要方法可分为精确算法与元启发式算法两类。精确算法基于严格的数学模型，如分支定界法试图求得理论最优解，但其计算复杂度往往随问题规模呈指数级增长，仅能有效处理小规模或特殊结构的优化问题。而实际工程、生物信息学、运筹学等领域的优化问题通常具有高维度、非线性、强约束等复杂特性，且大多数被归类为 NP-hard 问题，除非 NP 等于 P，否则无法在多项式时间内精确求解。因此，研究者转向近似方法，即元启发式算法，以在合理时间内找到近似最优解。

元启发式算法基于自然界或数学规律抽象出的搜索策略，主要分为两类：一类受现实世界启发（如粒子群优化模拟鸟群觅食、遗传算法模拟生物进化、模拟退火模拟冶金退火过程），另一类源于数学或抽象规则（如禁忌搜索通过禁忌表避免搜索重复、可变邻域搜索通过动态切换邻域结构平衡探索与开发）。这些算法通过在解空间中高效搜索权衡解，能够在复杂场景下逼近多目标 Pareto 前沿，尤其适用于大规模离散优化与动态优化问题[38]。这种通用性使得元启发式算法成为解决现代复杂多目标优化问题的核心工具。下一小节将介绍多目标元启发式算法的编码类型与离散优化编码方式，并以 NSGA-II 为例分析其框架设计，最后探讨算法的发展方向。

2.2.2 多目标元启发式算法

多目标优化领域的元启发式算法主要分为群智能算法与进化算法两类。群智能算法如多目标粒子群优化（MOPSO）和多目标灰狼优化（MOGWO），基于群体协作行为求解问题；进化算法则以多目标遗传算法（如 NSGA-II、MOEA/D、SPEA-II）为

代表，通过模拟生物进化机制实现全局搜索。其中，NSGA-II[39]采用快速非支配排序与拥挤距离度量解质量，并通过精英选择策略维持种群进化方向，其时间复杂度从 $O(MN^3)$ 降低至 $O(MN^2)$，显著提升了大规模问题的求解效率。在 4 目标以内优化问题上，NSGA-II 展现出逼近真实 Pareto 前沿的能力。本小节介绍了多目标元启发式算法支持的实数、离散与二进制等多种编码类型，并介绍离散优化问题在元启发式算法中的编码方式。并以 NSGA-II 为例，详细介绍了多目标遗传算法框架中适用于离散编码的经典遗传算子与精英选择策略设计，为后续引入改进措施提供了理论基础。最后，介绍多目标元启发式算法的发展方向。

多目标元启发式算法的编码方案主要分为三类：实数编码、整数编码和二进制编码。其中，实数编码作为一种通用方案，可通过编解码方式灵活表示多种优化问题，但其基于连续搜索空间的特性易忽略离散约束，导致局部最优问题[40]；整数编码专门适配离散优化问题，例如本文研究的虚拟机分配编码，但其需定制交叉和变异算子以保证解的合法性并提高优化速度；二进制编码则通过离散变量的直接映射，例如本文中的执行模式选择，支持特定问题建模，但在大规模场景下易因编码冗余降低效率。相较基于位置更新机制的群智能算法，例如 PSO 的粒子速度模型，NSGA-II 在离散编码的优化上更具灵活性，其染色体结构更易实现离散空间搜索。以组合优化为例，NSGA-II 通过可以离散编码表达任务序列，并设计交叉算子避免非法解产生，可以适应不同的编码方案。

任务调度等问题属于离散优化问题，其编码方法主要分为两类：第一类沿用实数编码方案，通过 Sigmoid 等传递函数将连续值映射为离散决策变量[41]。该方法保留了传统元启发式算法连续编码的结构特性，但存在编码效率低、难以精准描述离散目标函数特征的问题，且连续搜索区间易导致算法陷入局部最优；第二类则直接采用离散编码方案，虽通过减少决策空间加速收敛，但其固有的离散特性易导致个体状态信息丢失，且只能使用离散交叉与变异算子，从而影响搜索质量。为适应任务调度等离散优化问题的特殊需求，需结合定制化遗传算子以保持解的合法性与多样性，从而提高算法的搜索速度与质量。

多目标遗传算法的遗传算子主要包括变异与交叉算子，二者在遗传算法中发挥着不同的作用。变异算子通过引入新的编码方案，能够增加种群的多样性，从而增强全局搜索能力。然而，变异操作也可能破坏已有的优质编码，特别是在变异概率过高的情况下，遗传算法可能会退化为随机搜索，进而延长收敛时间。位翻转变异是一种典型的变异算子，它通过随机选择个体基因的某一位并将其值进行翻转，从而生成新的编码方案。然而，固定概率的位翻转可能导致收敛速度缓慢，因此可以通过动态调整变异概率来优化表现。例如，基于位翻转的非均匀变异算子能够结合进化代数和基因位状态进行变异概率的动态调整，在初期以较高的概率引入多样性以增强全局探

索能力，而在后期逐步降低变异强度，从而加速向高质量解的收敛。

交叉算子则主要负责在个体之间交换优秀的部分编码方案，以提高局部寻优能力。然而，交叉操作所能生成的编码范围相对有限，可能导致种群多样性下降，进而影响非支配解集的质量。多点交叉是一种广泛应用于多种编码类型的交叉算子，它通过在多个交叉点对父代基因进行分块重组，适用于解空间基因无明显位置依赖性且存在多峰局部最优的优化场景。然而，由于多点交叉随机选择交叉点，可能导致具有关联关系的部分编码难以同时被转移。因此，一些研究采用两点交叉代替多点交叉，这种方式既能充分交换编码信息，又能保留部分关联关系，从而在提升解集质量的同时维持种群的多样性。

针对离散优化问题的特性，NSGA-II 的精英选择策略通过非支配排序与拥挤距离机制（Crowding Distance）共同实现解的筛选与种群多样性维护。首先，算法对合并种群执行快速非支配排序，将种群划分为不同 Pareto 非支配等级的子集，按优先级顺序依次为 $F_1, F_2, F_3, \ldots, F_l$；随后，按照等级顺序将各子集依次保留并存入下一代种群中，直至种群规模超过设定值 N；最后，对存入种群的最后一等级子集，通过拥挤距离机制顺次择优保留，直至种群规模达到 N。拥挤距离机制通过公式

$$\text{Crowd}(s) = \sum_i \text{norm}(f_i(s)) \tag{2-4}$$

计算每个解的分布稀疏程度，表征其在各目标维度上的多样性。该机制优先保留拥挤距离较大的解，避免种群因趋同化导致早熟收敛。当高等级解的数目超过种群容量时，系统自动淘汰拥挤距离较小的个体，从而确保解集的多样性并提升算法在离散优化问题中的搜索性能。

元启发式算法的未来发展可从多个维度探索突破方向。首先，在高维目标空间的多样性维护方面，随着目标维度的增加，非支配解比例呈指数级增长，传统 NSGA-II 拥挤距离机制难以有效保持种群多样性。研究可通过融合参考点法（如 NSGA-III）与指标选择策略（如 IBEA 的指标化选择[42]），增强高维空间中的解集分布性与多样性。其次，根据无免费午餐定理，通用算法无法在所有问题上保持最优性能，针对特定场景定制元启发式优化算法可以提高收敛速度与非支配解集质量。例如，本文的混合云隐私任务调度中需定制遗传算子，以避免非法解生成并降低计算代价。此外，通过构建混合型算法框架，可在全局探索与精细开发之间实现平衡。例如，将模拟退火作为局部搜索算子嵌入 NSGA-II 架构，可有效改善算法在离散解空间中的收敛精度；或结合启发式算法，提升复杂约束场景下的求解效率。最后，动态自适应参数调控是元启发式算法未来发展的重要方向之一。以遗传算法为例，针对不同遗传算子的局部寻优与全局搜索能力特征，结合问题特性和算法运行阶段动态调整参数，能够显著提升算法性能。例如，在早期搜索阶段，通过提高变异概率增强全局探索能力并增加种群

多样性；在后期收敛阶段，降低变异强度并提高交叉概率，有助于加速收敛并提升解精度。

2.3 本章小结

本章介绍了多目标优化、混合云架构及其隐私保护技术的基础知识。首先，介绍了多目标优化问题的核心概念、评价指标及使用的元启发式算法。重点分析了 NSGA-II 的遗传算子与精英选择策略在离散优化问题中的应用，并介绍了算法未来发展方向。其次，阐述了混合云的基本概念、优势及部署模式，特别是在资源扩展、成本控制与合规性方面的价值。还针对混合云中的隐私保护问题，介绍了隐私加密算法的主要技术路线及其在数据处理与协同计算中的应用，为后续研究奠定了理论基础。

第三章　混合云中动态细粒度的隐私任务调度模型

混合云架构通过整合公有云的丰富资源与私有云的隐私保障能力，为处理隐私数据提供了高效平台。然而，现有任务调度方法在隐私管理中存在以下不足：缺乏安全性的量化评估指标，导致安全与效率难以平衡，还存在着静态的任务模型导致跨云调度效率低下，粗粒度隐私标签难以适配不同数据所有者差异化隐私需求的问题。针对上述挑战，本章提出一种混合云中动态细粒度的隐私任务调度模型，其主要特点为：首先，采用细粒度隐私标签机制，可根据不同类型数据的隐私标签自动调整选用的加密算法；其次，支持公私云协作的动态任务模型，能够调度时动态插入加密与验证子任务以提升协作能力；最后，提出量化的混合云安全性指标，关联隐私保护强度与执行效率，支撑多目标优化调度，从而平衡任务处理效率与隐私安全性。

本章首先在3.1节构建混合云资源模型，明确混合云架构与虚拟机的网络及计算资源特性；其次，在3.2节建模任务，提出可在调度中切换为独立执行模式或跨云协作模式的动态任务模型，并分别考虑任务计算网络与计算资源的占用，以精确描述跨云协作中的任务处理过程；在3.3节，定义混合云隐私安全性量化指标，结合隐私数据大小与隐私加密算法的安全性，构建系统安全性评估模型；最后，在3.4节建立以完工时间、安全性及成本为目标的优化问题，定义决策空间与约束体系，为后续多目标优化算法提供依据。

通过模型构建，本章为混合云环境下隐私任务的安全与效率的平衡调度问题提供了完整的数学框架，为后文多目标优化算法的设计与实验验证奠定了基础。

3.1 混合云资源模型

混合云协调私有云的隐私保护能力与公有云的丰富计算资源，形成了一个具备隐私敏感数据处理能力的异构计算环境。如图3.1所示，该系统由两个资源池组成:

私有云资源池 $\mathcal{S}=\{s_1,s_2,...,s_S\}$，由 S 台异构虚拟机组成，其中 $s_i\,(i\in\{1,...,S\})$ 表示第 i 台私有虚拟机，其计算能力独立配置；公有云资源池 $\mathcal{V}$ 提供 J 种虚拟机类型 $(\mathcal{J}=\{1,...,J\})$，$n\in\{1,2,\ldots,N_j\}$，表示类型 j 的第 n 个虚拟机，类型 j 的公有虚拟机实例记为 $v_{(j,n)}$ $(n\in\{1,...,N_j\})$，其全局索引由复合下标 (j,n) 唯一标识，复合下标计算方式见公式(3-1)。公有虚拟机的租用成本 R_j 与类型 j 的计算能力 B_j 正相关，由云服务商定价。混合云中的隐私数据存储在私有云虚拟机中，其具体定义详见3.3.1小节。

$$(j,n) = \sum_{j'<j} N_{j'} \cdot j' + n \tag{3-1}$$

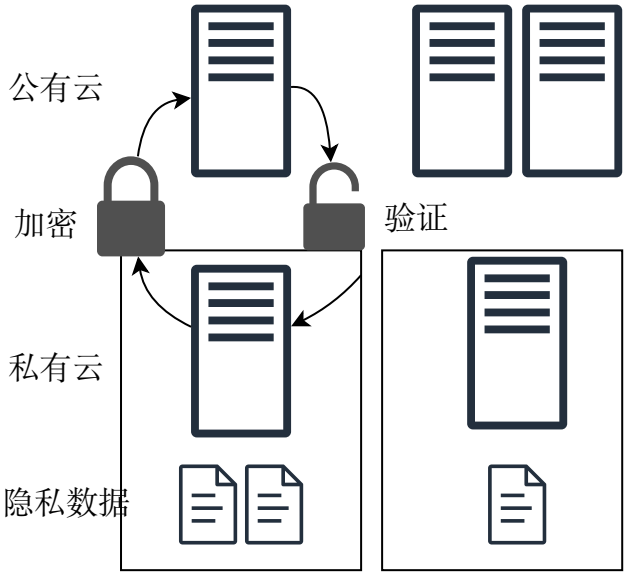

图 3.1 混合云系统架构示意图

在计算能力配置方面，私有云虚拟机 s_i 的计算能力为 A_i（MHz），即每秒可执行 A_i 兆次 CPU 运算。而公有云虚拟机 $v_{(j,n)}$ 继承其所属虚拟机类型的计算能力 B_j（MHz）。不失一般性的，本文认为假设 B_j 与其类型下标 j 满足单调递增关系 $B_1 < B_2 < \cdots < B_J$，即公有云虚拟机类型编号越大，其计算能力越强。在网络连接方面，私有云与公有云之间通过带宽固定的半双工通道直接通信，数据传输速率限定为 β_{net}（Mbps）。基于此设计，跨云传输一条隐私数据 d 的时延可以表示为：

$$T^{\text{trans}}(d) = \frac{\text{size}(d)}{\beta_{\text{net}}} + \tau_{\text{prop}} \tag{3-2}$$

其中 size(d) 表示隐私数据的大小，τ_{prop} 为固定时延。这一传输时间计算模型为后续任务调度中跨云协作的总时间评估提供了关键依据。

混合云系统运营成本主要由公有云资源租用开销组成。本文采用主流云服务商按秒计费的定价机制，虚拟机 $v_{(j,n)}$ 的计费时间 $T_{(j,n)}$ 定义为其实例内最后一个任务的完成时刻，具体表达式为：

$$T_{(j,n)} = \max_{m \in \mathcal{M}} \left(\text{FT}(e_m) \cdot \mathbf{Y}[m][(j,n)]\right) \tag{3-3}$$

其中 $\mathbf{Y}[m][(j,n)]$ 为二元决策变量，当任务 e_m 分配至虚拟机 $v_{(j,n)}$ 时取值为 1，否则

为 0。基于此，系统总运营成本可量化为所有公有云实例费用之和：

$$\text{Cost} = \sum_{j=1}^{J} \sum_{n=1}^{N_j} R_j \cdot T_{(j,n)} \tag{3-4}$$

其中 R_j 表示类型 j 虚拟机每秒租用成本。完成时间的缩短直接减少虚拟机租用时长 $T_{(j,n)}$，从而降低公有云租用成本；而安全性约束的增强可能限制可用虚拟机类型的选择范围，进而影响公有云租用成本。根据成本目标与完成时间和安全性之间的关联性，本文在第四章算法设计中采用一种优先级策略，优化成本目标，从而实现调度的综合性能提升。

本节从任务调度研究的角度出发，对混合云模型进行了适当简化。首先，不考虑私有云和公有云内部的多层次网络架构，将混合云网络抽象为私有云与公有云之间的直接连接结构；其次，任务调度过程采用非抢占式执行模式，即任务执行期间独占计算资源直至完成。最后，在成本建模中仅考虑公有云虚拟机的租用成本，因为私有云成本主要由建设、维护等固定费用组成，与调度无关，故忽略这一部分。这些简化符合混合云任务调度的经典建模惯例[43]，保留混合云异构资源核心特征，为后续章节构建细粒度隐私保护机制与跨云协作策略提供可靠的研究环境。

3.2 公私云协作的动态任务模型

本节提出一种考虑公私云协作的的动态任务建模方法，其动态特性主要体现在任务调度期间的可切换执行模式：根据调度器选择的隐私加密算法，任务有不同的处理流程。具体而言，当任务被调度至私有云执行时（后文称为独立模式），其处理流程为单一任务；若需通过公有云协作执行（后文称为协作模式），则系统会在任务处理前于私有云节点动态插入数据加密子任务，并在处理后追加数据完整性验证子任务。同时，将网络传输时间与计算时间分别独立建模，以精确描述公私云跨云协作中的资源竞争特征。

3.2.1 任务基本定义与任务的执行模式

混合云系统中待处理的任务集合定义为 $\mathcal{E} = \{e_1, e_2, \ldots, e_M\}$，其中 M 为任务总数。每个任务 e_m 在提交时关联唯一的隐私数据 $d_k \in \mathcal{D}$，且所有隐私数据仅存储于私有云以满足安全性要求。任务特征通过以下两个关键指标描述：（1）数据规模 $\text{size}(e_m) \triangleq \text{size}(d_k)$，单位为兆比特（Mbit），表示任务执行所需的隐私数据量；（2）计算强度 $o_{\text{proc}}(e_m)$（单位：CPU 周期/bit），定义为任务处理单位比特数据所需的平均 CPU 周期。其中，计算资源总需求可表示为 $o_{\text{proc}}(e_m) \times \text{size}(e_m)$，该公式仅涵盖任务处理过程中的计算开销，不包括加解密操作的开销。计算强度作为任务固有属性，根

据其算法类型、数据处理逻辑等特征预先配置[44]，为后续任务调度与资源分配提供了重要的量化依据。

任务的执行模式由安全策略矩阵 $\mathbf{Z}$ 动态确定：当调度策略选择 $\mathbf{Z}[m][0]=1$（最高安全等级策略）时，任务 e_m 以独立模式 $e_m^{(\mathrm{SA})}$ 在私有云内执行；若采用其他安全策略，则系统形成公私云协作的线性工作流（Linear Workflow, LW）[27] $e_m=\{e_m^{(\mathrm{E})},e_m^{(\mathrm{P})},e_m^{(\mathrm{V})}\}$，在预处理阶段插入加密子任务 $e_m^{(\mathrm{E})}$，并将计算子任务 $e_m^{(\mathrm{P})}$ 卸载至公有云执行，最终在私有云追加验证子任务 $e_m^{(\mathrm{V})}$。执行模式的决策可形式化为：

$$e_m \triangleq \begin{cases} e_m^{(\mathrm{SA})}, & \mathbf{Z}[m][0]=1 \\ \{e_m^{(\mathrm{E})},e_m^{(\mathrm{P})},e_m^{(\mathrm{V})}\}, & \text{其他} \end{cases} \tag{3-5}$$

其中，安全策略选择矩阵 $\mathbf{Z}$ 用于动态确定任务的执行模式：$\mathbf{Z}[m][0]=1$ 表示任务 e_m 采用私有云内独立执行的安全策略，其安全策略选择矩阵具体定义见第3.3节。

3.2.2 完成时间计算

本节设计的动态任务模型具有“云爆发”能力，即当混合云整体负载显著增加时，调度器可通过调整安全策略将部分任务的 $\mathbf{Z}[m][0]$ 设置为非 1 值，触发协作模式，将计算工作负载卸载至公有云，同时使私有云资源得以释放以处理其他任务。为量化不同模式的执行效能，需建立任务完成时间的数学模型。

当 $\mathbf{Z}[m][0]=1$ 时，任务 $e_m^{(\mathrm{SA})}$ 以独占模式运行，占用虚拟机资源直至处理完毕。其完成时间可建模为：

$$\mathrm{FT}(e_m^{(\mathrm{SA})})=\mathrm{ST}(e_m^{(\mathrm{SA})})+\frac{\mathrm{size}(e_m)\cdot o_{\mathrm{proc}}(e_m)}{\sum_{i=1}^{S}A_i\cdot\mathbf{X}[m][i]} \tag{3-6}$$

式中，$\mathbf{X}\in\{0,1\}^{S\times M}$ 为私有虚拟机分配矩阵，$\mathbf{X}[m][i]=1$ 表示任务分配至 s_i，$\mathrm{ST}(e_m^{(\mathrm{SA})})$ 表示任务开始时间，其受虚拟机中任务执行顺序 $\mathbf{\Pi}$ 影响；A_i 为虚拟机 s_i 的计算能力 CPU 频率。数据规模 $\mathrm{size}(e_m)$ 与计算强度 $o_{\mathrm{proc}}(e_m)$ 之积表示任务的计算量。

当 $\sum_{q=1}^{Q}\mathbf{Z}[m][q]=1$ 时，任务 e_m 在协作模式下执行，采用线性工作流建模。将被分解为三个任务，加密子任务 $e_m^{(\mathrm{E})}$、处理子任务 $e_m^{(\mathrm{P})}$ 和验证子任务 $e_m^{(\mathrm{V})}$，且须满足依赖关系 $e_m^{(\mathrm{E})}\prec e_m^{(\mathrm{P})}\prec e_m^{(\mathrm{V})}$，其中符号 $\prec$ 表示前驱子任务必须完成全部操作后才能触发后续子任务。图3.2展示了各阶段的资源占用时序。下面首先介绍各子任务的完成时间，随后通过分类讨论私有虚拟机中执行加密、验证与独立任务时的三种并行关系确定起始时间，最后基于上述计算给出混合云系统的完工时间公式。

加密子任务 $e_m^{(\mathrm{E})}$ 在私有云执行期间产生的时间开销包含加密计算和数据传输两

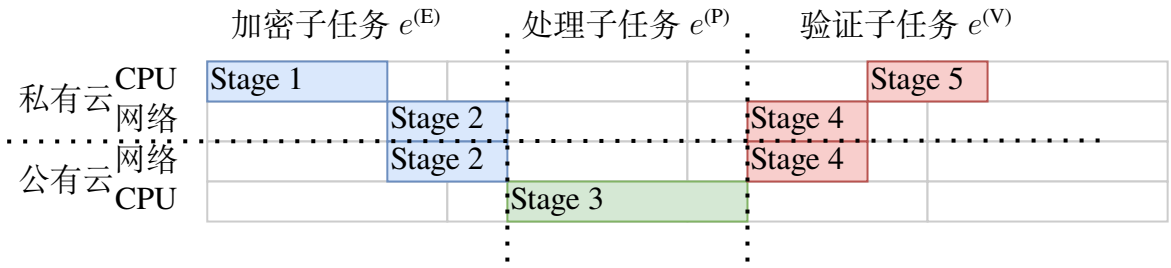

图 3.2　协作模式下的任务处理时序与资源占用

个阶段，其完成时间可表述为：

$$\mathrm{FT}(e_m^{(\mathrm{E})}) = \mathrm{ST}(e_m^{(\mathrm{E})}) + \frac{\mathrm{size}(e_m) \cdot o_{\mathrm{enc}}(e_m)}{A_i} + T_{\mathrm{trans}}(\mathrm{size}(e_m)) \tag{3-7}$$

其中 $\mathrm{ST}(e_m^{(\mathrm{E})})$ 表示子任务的开始时间，$o_{\mathrm{enc}}(e_m)$ 为加密计算强度系数（其计算规则详见第3.3节），A_i 为私有云虚拟机 s_i 的 CPU 频率，$T_{\mathrm{trans}}(\mathrm{size}(e_m))$ 为隐私数据后的传输时间（定义参见公式(3-2)）。本文假设加密过程保持数据体积不变，即 $\mathrm{size}(e_m^{(\mathrm{E})}) = \mathrm{size}(e_m)$，这是因为多数加密算法（如 AES）密文大小与原文基本保持一致，不会导致数据规模显著膨胀。

处理子任务 $e_m^{(\mathrm{P})}$ 的完成时间由以下三个阶段的时间开销组成：首先对输入的加密数据进行解密，随后进行处理，最后再对处理结果进行加密。其计算模型可表述为：

$$\mathrm{FT}(e_m^{(\mathrm{P})}) = \mathrm{ST}(e_m^{(\mathrm{P})}) + \frac{\mathrm{size}(e_m) \cdot (o_{\mathrm{dec}} + o_{\mathrm{proc}}(e_m) + 0.1 \times o_{\mathrm{enc}})}{B_j} \tag{3-8}$$

其中，$\mathrm{ST}(e_m^{(\mathrm{P})})$ 表示子任务的开始时间，$\mathrm{size}(e_m)$ 为输入数据规模，B_j 为公有云虚拟机的计算能力。计算强度系数由三部分构成：o_{dec} 表示解密计算开销，$o_{\mathrm{proc}}(e_m)$ 为数据处理开销，$0.1 \times o_{\mathrm{enc}}$ 则对应于结果数据的再加密开销。本文约定计算结果大小为输入数据规模的 10%，即 $\mathrm{size}(d_{\mathrm{r}}) = 0.1 \cdot \mathrm{size}(e_m)$，该设定基于典型数据处理场景中输出数据普遍远小于输入数据的特征。

验证子任务 $e_m^{(\mathrm{V})}$ 的完成时间由结果数据回传和验证解密两个阶段构成，其计算模型为：

$$\mathrm{FT}(e_m^{(\mathrm{V})}) = \mathrm{ST}(e_m^{(\mathrm{V})}) + T_{\mathrm{trans}}(e_m^{(\mathrm{V})}) + \frac{\mathrm{size}(e_m^{(\mathrm{V})}) \cdot o_{\mathrm{dec}}}{A_i} \tag{3-9}$$

式中 $\mathrm{ST}(e_m^{(\mathrm{V})})$ 表示子任务开始时间，$\mathrm{size}(d_{\mathrm{r}}) = 0.1 \cdot \mathrm{size}(e_m)$ 为处理结果的数据规模，$T_{\mathrm{trans}}(d_{\mathrm{r}})$ 表示加密结果的回传时延（计算方式与公式(3-2)一致）。o_{dec} 为解密计算强度系数，A_i 为私有云虚拟机的计算能力。

在任务独占虚拟机资源约束下，为优化公私云间的协作效能，本文针对混合云环境中虚拟机的网络与计算资源特性，提出了一种解耦网络与计算资源的任务执行方式。该方式允许同一任务在执行期间仅占用单一类型资源（网络或计算），而释放另

一种资源供其他任务使用，从而显著提高了系统的并行性与资源利用率。然而，这种资源解耦方式可能导致任务执行顺序的灵活性增加，进而导致任务开始时间无法确定。为解决这一问题，本文引入严格全序关系对虚拟机内部任务的执行顺序进行约束。为确保调度时序的可计算性与确定性，定义虚拟机 s_i 内部任务的顺序约束。记 s_i 包含的任务序列为 $\{e_1, e_2, ..., e_{N_i}\}$，其严格全序关系 $\mathbf{\Pi}_i \subset \mathcal{E} \times \mathcal{E}$ 由相邻任务的前驱-后继拓扑确定：

$$\mathbf{\Pi}_i = \{(e_m, e_{m+1}) \mid 1 \leq m \leq N_i - 1\} \tag{3-10}$$

该序关系是一个链式结构，确保任务执行顺序唯一。定义前驱映射函数 prev : $\mathcal{E} \to \mathcal{E} \cup \{\perp\}$：

$$\text{prev}(e_m) = \begin{cases} e_{m-1}, & \text{若存在 } m \in [2, N_i] \text{ 使得 } e_m = e_m \\ \perp, & \text{若 } e_m \text{ 是虚拟机中第一个任务} \end{cases} \tag{3-11}$$

其中 $\perp$ 表示空值。由此得到无前驱任务的开始时间：

$$\text{ST}(e_m^{(\text{ANY})}) = 0, \text{ 当} e_m = \perp \tag{3-12}$$

其中，上标 (ANY) $\in$ {E,SA,P} 表示子任务的类型，其含义为任意无前驱子任务的开始时间都为 0。

通过前驱映射函数 prev : $\mathcal{E} \to \mathcal{E} \cup \{\perp\}$ 的序关系定义（见公式(3-11)），当任务 $e_m^{(\text{ANY})}$ 存在前驱任务 prev$(e_m^{(\text{ANY})}) \neq \perp$ 时，其开始时间需根据前驱任务与当前任务的资源占用特征动态计算。为清晰刻画这一机制，首先对任务类别进行结构性划分：单阶段任务全程仅占用计算资源，包括处理任务 $e^{(\text{P})}$ 与独立任务 $e^{(\text{SA})}$；双阶段任务分两个阶段使用资源，例如加密任务 $e^{(\text{E})}$ 依次占用计算 → 网络资源完成加密与传输，而验证任务 $e^{(\text{V})}$ 依次占用网络 → 计算资源用于接收与验证。

根据任务组合的资源占用特征，前驱任务与当前任务的交互可归纳为三类并行模式，共包含 10 种组合（私有云内 9 种 $\{\text{E}, \text{V}, \text{SA}\} \times \{\text{E}, \text{V}, \text{SA}\}$，公有云 1 种 $\{\text{P}\} \to \{\text{P}\}$，其详细定义见表3.1）。第一类是顺序执行模式，适用于两任务存在资源冲突的场景，例如加密任务 $e^{(\text{E})}$ 与验证任务 $e^{(\text{V})}$ 分别在计算 → 网络阶段重叠，需要完全顺序执行。第二类是可中途并行模式，适用于前驱任务完成首阶段后释放部分资源的情况，例如加密任务 $e^{(\text{E})}$ 完成加密计算阶段后可执行独立任务 $e^{(\text{SA})}$。第三类是可在开始并行，适用于资源需求互补的场景，例如独立任务 $e^{(\text{SA})}$ 与验证任务 $e^{(\text{V})}$ 可分别占用计算与网络资源，实现即时并行执行。这些模式通过精确建模任务间的资源依赖关系，为调度算法的设计与分析提供了理论基础。

本文采用分步计算方法计算任务开始时间：首先基于虚拟机内部顺序执行的约束（即每个任务除第一个外都有一个直接前驱任务，且任务仅在前驱任务完成后才能

表 3.1　任务组合的并行性分类

前驱任务类型	当前任务类型	并行性
E	V	顺序执行
E	SA	
V	E	
V	SA	
SA	E	
SA	SA	
P	P	
E	E	可中途并行
E	SA	
V	V	
SA	V	可在开始并行

启动），计算前驱任务驱动的初始开始时间 $\mathrm{ST}_{\mathrm{local}}$，随后结合跨云协作的任务依赖关系，计算最终开始时间 ST。这一方法不仅确保了任务完成时间计算的确定性，避免了并行执行可能引发的资源竞争问题，同时也为调度算法的设计提供了理论基础。

虚拟机内部时序计算：基于虚拟机 s_i 内的任务执行序列 $\mathbf{\Pi}_i$，计算任务在虚拟机内部开始时间 $\mathrm{ST}_{\mathrm{local}}$，其表达式为：

$$\mathrm{ST}_{\mathrm{local}}(e_m) = \begin{cases} \mathrm{FT}(e_m^{(\mathrm{PRED})}) & \text{顺序执行} \\ \max\left\{ \begin{array}{l} \mathrm{ST}'(e_m^{(\mathrm{PRED})}) + T_{\mathrm{s1}}(e_m^{(\mathrm{PRED})}), \\ \mathrm{FT}(e_m^{(\mathrm{PRED})}) - T_{\mathrm{s1}}(e_m^{(\mathrm{NOW})}) \end{array} \right\} & \text{可中途并行} \\ \max\left\{ \begin{array}{l} \mathrm{ST}'(e_m^{(\mathrm{PRED})}), \\ \mathrm{FT}(e_m^{(\mathrm{PRED})}) - T_{\mathrm{s1}}(e_m^{(\mathrm{NOW})}) \end{array} \right\} & \text{可在开始并行} \end{cases} \tag{3-13}$$

其中，第一阶段的执行时间可以表示为：

$$T_{s1}(e_m) = \begin{cases} T_{\mathrm{comp}}(e_m^{(\mathrm{E})}) = \dfrac{\mathrm{size}(e_m) \cdot o_{\mathrm{enc}}(e_m)}{A_i}, & e_m \in \mathcal{E}_{\mathrm{E}} \\ T_{\mathrm{trans}}(e_m^{(\mathrm{V})}) = \dfrac{\mathrm{size}(e_m)}{B}, & e_m \in \mathcal{E}_{\mathrm{V}} \end{cases} \tag{3-14}$$

跨云协作的开始时间修正，考虑跨云协作任务之间的数据流依赖（加密任务 → 处理任务 → 验证任务顺序关系），得到最终的开始时间 ST：

$$\mathrm{ST}(e_m) = \begin{cases} \mathrm{ST}_{\mathrm{local}}(e_m), & e_m \in \mathcal{E}_{\mathrm{E}} \cup \mathcal{E}_{\mathrm{SA}} \\ \max\left\{\mathrm{ST}_{\mathrm{local}}(e_m^{(\mathrm{P})}),\, \mathrm{FT}(e_m^{(\mathrm{E})})\right\}, & e_m \in \mathcal{E}_{\mathrm{P}} \\ \max\left\{\mathrm{ST}_{\mathrm{local}}(e_m^{(\mathrm{V})}),\, \mathrm{FT}(e_m^{(\mathrm{P})})\right\}, & e_m \in \mathcal{E}_{\mathrm{V}} \end{cases} \tag{3-15}$$

图3.3直观展示了三类并行性任务的执行时序。以加密任务与独立任务组合为例，当加密任务完成计算阶段并释放计算资源后，系统可立即启动独立任务的计算过程，同时加密任务继续使用网络资源传输数据，形成网络-计算资源的并行利用。这种资源解耦调度策略通过对任务执行阶段的精细化分析，在维持资源独占性的前提下充分挖掘并行潜力，为优化系统完工时间提供理论依据。

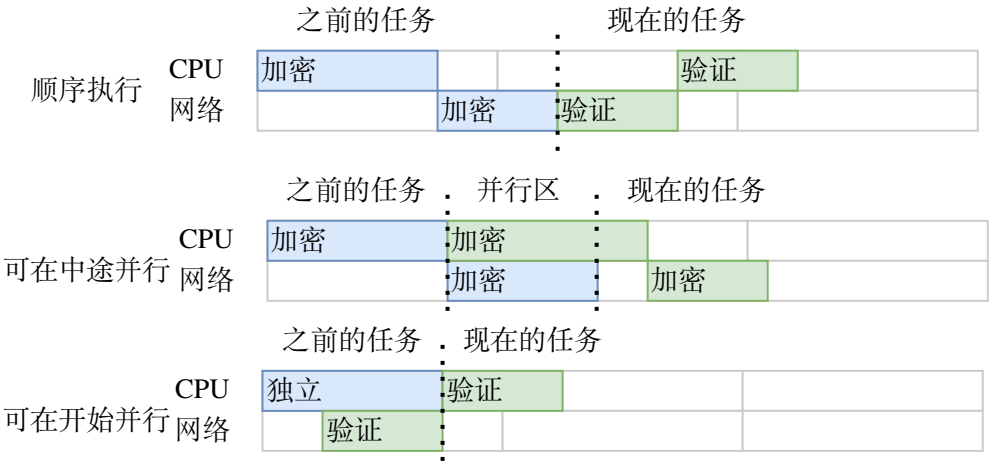

图 3.3　网络与计算资源互补的任务执行时序示意图

混合云系统的完工时间（Makespan）定义为系统处理完成当前任务集合全部任务的最晚完成时间，其数学表示为：

$$\text{Makespan} = \max_{e_m \in \mathcal{M}} \mathrm{FT}(e_m) \tag{3-16}$$

其中任务完成时间 $\mathrm{FT}(e_m)$ 由公式(3-6)(3-7)(3-8)和(3-9)定义。

3.3 细粒度的隐私任务安全策略

在混合云环境中，隐私任务调度需要同时兼顾数据处理效率和隐私安全性。然而，通过对现有隐私任务调度研究的分析发现，当前隐私任务安全策略面临两大挑战：一是不同数据拥有者的多样化隐私偏好导致任务复用性较低，由于数据主权或行业规范的差异性，同类计算任务往往需要重复设计专用隐私处理任务；二是安全性与效率难以动态协调，现有方法因缺乏对安全策略的量化建模能力，难以权衡效率和安

全性，在混合云负载提升时选择开销较低的隐私加密算法来提升任务处理效率。

针对上述问题，本文提出了一种细粒度的隐私任务安全策略。首先，基于细粒度隐私标签将数据拥有者的隐私需求映射为对不同隐私加密算法组的要求，调度算法通过灵活组合加密算法来适配多样化的隐私偏好，从而支持处理来自不同数据拥有者的计算任务。其次，通过量化安全等级与脆弱性评分，建立了混合云系统安全性的量化指标，使得调度算法在满足最低安全需求的同时，既可以选择高安全性加密算法以进一步提升系统安全性，又能够根据系统负载动态调整策略以提升处理效率。此外，该方法通过将安全性作为多目标优化的优化目标而非硬性约束，实现了安全性与效率的协同优化。这一细粒度的安全策略框架有效解决了现有方法的局限性，为混合云环境下的隐私任务调度提供了更为灵活和高效的解决方案。

3.3.1 隐私数据的基本属性

隐私数据集在混合云的私有部分中存储，其定义为 $\mathcal{D} = \{d_1, d_2, \ldots, d_K\}$，其中 K 为数据总数量。每个数据对象 d_k 包含四个基本属性：数据大小、存储位置、隐私标签和最低安全等级。数据大小 size(d_k) 表示隐私数据的数据量，单位为 Mbit。存储位置描述了隐私数据在私有云虚拟机中的分布，一条隐私数据可以存储在不同的私有云虚拟机中，本文采用矩阵 $\mathbf{L} \in \{0,1\}^{K \times S}$ 进行编码，$\mathbf{L}[k][i] = 1$ 表示数据 d_k 存储在虚拟机 s_i 中。例如，公式(3-17)所示的存储分布矩阵中，数据 d_1 存储在虚拟机 s_1 和 s_2 中，而数据 d_2 仅存储在虚拟机 s_3 中。

$$\mathbf{L} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3-17}$$

隐私数据 d_k 的隐私标签 Pref(d_k) 由混合云管理者根据数据所有者的合规性要求手动设置（例如，若数据所有者偏好商密系列算法，则 Pref$(d_k) = 0$），用于指导调度算法选择适配的安全策略。最低安全等级 $\alpha_{\min}(d_k) \in [0,1]$ 定义了调度算法可接受的最低安全阈值：在混合云资源充足时，调度算法优先选择最高安全等级的策略以提升系统整体安全性；在资源不足时，调度算法可在保证最低安全等级的前提下适度降低部分数据的安全等级，以提升任务处理效率。

3.3.2 隐私加密算法组

本文以混合云任务调度领域常用的三类隐私加密算法为例，定义了以下三类隐私标签：Pref $= 0$ 表示数据拥有者倾向于采用国家密码标准算法组合（如 SM4+SM3）；Pref $= 1$ 表示数据拥有者更倾向使用 NIST 推荐算法组合（如 AES-256+SHA-1）；Pref $= 2$ 表示数据拥有者更倾向采用欧盟 ECRYPT 密码算法征集计划的胜出算法（如

HC-128+SHA-1）。值得注意的是，隐私标签的构建不仅限于对不同组织推荐的隐私加密算法算法要求，还可扩展至数据类型与应用场景（如图像、文字、音视频等）的多样化需求，例如对于街景图片，可以局部模糊处理则可以有效遮挡人物图像、车牌路牌等敏感信息。

针对每类隐私标签，本文构建了独立的隐私加密算法组。每个策略库包含 Q_p 个策略，编号为 Pref$p$-q，其中 $q \in \{1, 2, ..., Q_p\}$。为简化调度设计，本文约定所有策略库的策略总数相同，即 $Q_1 = Q_2 = Q_3$。此外，每个策略库中固定包含编号为 $q = 0$ 的私有云内处理策略，且该策略不计入 Q_p。具体而言，$q \geq 1$ 为跨云协作策略，其安全等级 α(Pref$p$-q) 随编号增大单调递减，而加密与解密开销 o_{enc}(Pref$p$-q) 则递增，这种设计避免了低安全高开销策略的冗余使用。

任务调度需满足隐私合规性约束，即调度算法选择的安全策略必须满足最低安全需求。数学表达为：

$$\text{C1:}\alpha(e_m, q) \geq \alpha_{\min}(d_k) \quad \text{当 } \mathbf{Z}[m][q] = 1 \text{ 且 } \text{data}(e_m) = d_k \tag{3-18}$$

其中 $\alpha(\cdot)$ 为策略库定义的安全等级函数，将策略编号映射到具体安全值。此约束确保了调度决策在满足合规性要求的同时，优化资源分配与任务执行效率。

3.3.3 系统安全性优化目标设计

在混合云环境中，尽管私有云内计算可提供最高安全等级，但突发负载、大规模请求或数据共享需求往往需要将部分隐私数据交由公私云协作处理，从而导致潜在的数据泄露风险。为量化公私云协作可能带来的隐私风险，本文设计了安全性优化目标，旨在辅助调度算法在任务卸载时选择最优策略，以最大限度地降低系统潜在风险。本文的安全性优化目标与混合云中数据价值与其脆弱性评分有关，通过这一量化指标能够有效评估任务卸载过程中的风险水平，为调度决策提供科学依据。

首先，现有研究已通过数据相似性、分布距离等方法实现对数据价值的量化估计[45-46]。然而，本文侧重于任务调度场景，采用了一种简化的数据价值模型，将数据价值直接定义为数据大小 size(d)。这一模型基于一个直观假设：数据量越大，其价值越高。尽管这是一种较为粗略的评估方式，但其能够在混合云环境中较好地反映数据的相对价值，为后续混合云安全性执行提供了基础依据。其次，通过脆弱性评分 $V(d_k) \in [0, 1]$ 动态评估数据潜在风险，初始值为 $V(d_k) = 0$。当数据采用较低安全等级策略时，其脆弱性增加，具体更新规则为：其次，数据选用过较弱的安全策略会降低数据安全性，使数据受到潜在泄露风险。本文将数据潜在风险定义为数据的脆弱性 $V(d)$，数据初始的脆弱性为 0，表示无潜在泄露风险。若选用了较弱的安全策略，则

数据的脆弱性会增加。

$$V_{\text{new}}(d_k) = \begin{cases} V_{\text{prev}}(d_k), & \alpha_{\text{now}} \geq \alpha_{\text{hist}} \\ \min(V_{\text{prev}}(d_k) + \Delta V, 1), & \alpha_{\text{now}} < \alpha_{\text{hist}} \end{cases} \tag{3-19}$$

其中 $\Delta V = \alpha_{\text{hist}} - \alpha_{\text{now}}$ 表示安全降级幅度。例如，当医疗数据 d_3 的安全等级从 $\alpha = 0.95$ 降级至 $\alpha = 0.8$ 时，其脆弱性增加 0.15（从 $V = 0.05$ 更新为 $V = 0.20$）。最后，本文认为，若数据曾使用过较低的安全策略，再次使用同等级或更高的安全策略，数据的脆弱性不会增加。这是因为，数据的安全主要由隐私加密算法保护，若隐私加密算法未被攻破，数据无论被使用多少次均不会泄露。因此，在混合云高负载时，调度算法优先将已有潜在风险的数据卸载到公有云处理，从而避免其余数据安全性降低，提升系统整体安全性。

根据以上原则，本文引入安全性优化目标 Security，定义为：

$$\text{Security} = \sum_{k=1}^{K} \text{size}(d_k) \cdot (1 - V(d_k)) \tag{3-20}$$

该目标引导调度算法在系统整体安全性与任务处理效率之间达到平衡。

3.4 优化问题建立

基于混合云调度的多维度需求，本节建立三目标优化模型，协同优化完工时间、系统安全性和计算成本三个关键指标。首先，明确了决策变量空间，包括私有云分配矩阵 $\mathbf{X}$、公有云分配矩阵 $\mathbf{Y}$ 以及安全策略选择矩阵 $\mathbf{Z}$。其次，构建了完整的约束条件体系，涵盖安全等级合规约束、数据本地化存储约束、决策变量完整性约束以及跨云协作约束等，确保调度方案的可行性与合规性。最后，结合完工时间最小化、计算成本最小化与安全性最大化目标，形式化定义了多目标优化问题。该模型通过动态决策安全策略选择与资源分配，实现了性能、安全与成本的平衡，为后续调度算法的设计与分析提供了理论基础。

3.4.1 决策变量

混合云任务调度模型决策空间由三类变量构成。私有云分配矩阵 $\mathbf{X} \in \{0,1\}^{M \times S}$ 定义了任务与虚拟机 s_i 的映射关系，其中 $\mathbf{X}[m][i] = 1$ 表示任务 e_m 被分配至私有云虚拟机 s_i 执行。公有云分配矩阵 $\mathbf{Y} \in \{0,1\}^{M \times (J \times N)}$ 描述了跨云协作任务的处理子任务调度方案，其中复合下标 (j,n) 对应第 j 类公有云虚拟机的第 n 个虚拟机 $v_{(j,n)}$，$\mathbf{Y}[m][(j,n)] = 1$ 表示协作任务的处理子任务 $e_m^{(P)}$ 分配至虚拟机 $v_{(j,n)}$。安全策略选择矩阵 $\mathbf{Z} \in \{0,1\}^{M \times Q}$ 记录了每个任务的安全策略选择，$\mathbf{Z}[m][q] = 1$ 表示任务 e_m 采用

编号为 q 的安全策略，这一决策直接影响任务的加解密计算开销与系统整体安全等级。这三类决策变量共同构成了混合云调度问题的优化基础。

3.4.2 约束条件

为确保调度方案的可行性与合规性，本文定义了以下约束条件。首先，安全等级合规约束 (C1) 要求每个任务 e_m 所选安全策略的安全等级不低于其关联数据的最低安全阈值，具体定义见公式(3-18)。其次，数据本地化存储约束 (C2) 确保任务仅能在已存储其关联隐私数据的私有云虚拟机上执行，其形式化定义为：

$$\text{C2:} \sum_{i=1}^{S} \mathbf{L}[k][i] \cdot \mathbf{X}[m][i] = 1 \quad \text{当 data}(e_m) = d_k \tag{3-21}$$

该约束保证任务能够访问所需隐私数据，且每个任务都能分配到一台私有虚拟机。

决策变量完整性约束 (C3-C5) 规范了决策变量基本属性：C3 要求所有决策变量为二元变量；C4 强制每个任务被分配至唯一私有云虚拟机；C5 要求每个任务必须且仅能选择一个安全策略。其形式化定义如下：

$$\text{C3: } \mathbf{X}[m][i] \in \{0,1\},\ \mathbf{Y}[m][(j,n)] \in \{0,1\},\ \mathbf{Z}[m][q] \in \{0,1\} \tag{3-22}$$

$$\text{C4: } \sum_{i=1}^{S} \mathbf{X}[m][i] = 1 \quad \forall m \in \mathcal{M} \tag{3-23}$$

$$\text{C5: } \sum_{q=0}^{Q} \mathbf{Z}[m][q] = 1 \quad \forall m \in \mathcal{M} \tag{3-24}$$

公有云决策变量约束 (C6) 规定，当任务选择非私有云安全策略（$\mathbf{Z}[m][0] = 0$）时，必须分配一个公有云虚拟机处理协作子任务，其形式化为：

$$\text{C6: } \sum_{j=1}^{J} \sum_{n=1}^{N_j} \mathbf{Y}[m][(j,n)] = \sum_{q=1}^{Q} \mathbf{Z}[m][q] \quad \forall m \in \mathcal{M} \tag{3-25}$$

跨云任务时序约束 (C7) 为成本计算提供基础，定义公有云虚拟机 $v_{(j,n)}$ 的总运行时间（见公式(3-3)），确保成本计算基于虚拟机实际运行时长。

上述约束条件共同构建了混合云任务调度问题的完整约束体系。

3.4.3 优化问题

基于上述定义，混合云任务调度问题可形式化为三目标优化模型：

$$\begin{aligned}
\min_{\mathbf{X},\mathbf{Y},\mathbf{Z}} \quad & \text{Makespan} = \max_{m\in\mathcal{M}} \text{FT}(e_m) \\
\min_{\mathbf{Y},\mathbf{Z}} \quad & \text{Cost} = \sum_{j=1}^{J}\sum_{n=1}^{N_j} R_j \cdot T_{(j,n)} \\
\max_{\mathbf{Z}} \quad & \text{Security} = \sum_{k=1}^{K} \text{size}(d_k)(1 - V(d_k)) \\
\text{s.t.} \quad & \text{C1-C7}
\end{aligned} \tag{3-26}$$

其中，Makespan 表示系统完工时间，Cost 为混合云成本，Security 为系统安全性指标。其中 R_j 表示第 j 类公有云虚拟机的单位时间计费价格。

该模型属于多目标的大规模整数优化问题，其可行解空间随任务规模呈指数级增长，精确求解困难。因此，本研究后续章节将设计启发式算法，通过近似求解方法实现性能、安全与成本的权衡优化。

3.5 本章小结

本章提出了混合云中动态细粒度的隐私任务调度模型，以解决混合云环境下隐私任务调度面临的安全与效率难以平衡的问题，并满足不同数据所有者的差异化隐私需求，还提高了跨云协作效率。首先，建立了混合云资源模型，明确了私有云与公有云的资源特性及其协作机制，并确定了混合云的成本目标。其次，提出了一种公私云协作的动态任务模型，其支持任务在独立模式与跨云协作模式间动态切换，并精确描述了任务的计算与网络资源占用情况，为高效优化混合云处理效率提供依据。随后，在隐私安全层面，设计了细粒度的隐私标签机制，构建了隐私加密算法组与安全性量化评估模型，满足不同数据所有者的差异化隐私需求。并根据数据大小与隐私加密算法的安全系数，提出了混合云安全性的量化指标。最后，建立了以完工时间、安全性和成本为目标的多目标优化问题，明确了决策变量与约束条件，为混合云环境下隐私任务的调度平衡安全性与效率的矛盾奠定了坚实基础。

第四章　多目标隐私任务调度算法

在混合云环境下，隐私任务调度问题需要同时优化完工时间、安全性以及成本三个目标，然而这些目标之间往往存在冲突。例如，提升安全性可能会延长完工时间，而降低成本可能会影响安全性。传统的优化算法使用加权求和的方式进行多目标优化，但存在着难以确定加权权重的问题。此外，而跨云协作的任务执行顺序也会影响整体调度效果。针对上述问题，本章提出了一种基于卸载窗口检测与遗传优化的调度框架（NSGA-OW），旨在兼顾完工时间、安全性与成本的协同优化，同时提升私有云资源的利用效率。

首先，针对跨云协作过程中产生的虚拟机空闲时段问题，设计了卸载窗口首次适应填充算法（OW-FF）。该算法通过动态检测并利用协作任务卸载至公有云时的等待时段，插入后续任务，从而提升私有云资源利用率。其次，结合 OW-FF 算法提出了改进的遗传算法框架 NSGA-OW。设计混合编码策略并根据编码特点，设计了虚拟机分块多点交叉算子与负载感知变异算子，加快收敛速度并提高调度质量。还分析了用户对于三种优化目标的偏好，优先优化安全性与完工时间目标，提升调度速度。NSGA-OW 与 OW-FF 协同优化。NSGA-OW 输出的虚拟机分配与安全策略决策编码为 OW-FF 提供输入，OW-FF 则通过空闲时段检测进一步提升调度质量。

本章后续内容将详细描述算法的设计细节及其实现过程。第4.1节介绍 OW-FF 算法的设计与实现，第4.2节 NSGA-OW 的算法框架及其改进的遗传算子等算法，最后分析 NSGA-OW 算法的时间复杂度。

4.1 混合云中虚拟机资源空闲时段优化

在混合云调度场景中，当任务以协作模式执行时，其线性工作流包含加密、公有云处理及验证三个阶段。将处理任务卸载至公有云产生的等待时延会在私有云调度中形成资源空闲时段。传统调度方法（如 SPGA[8]）因静态任务排序策略无法有效利用此类空闲时段（如图4.1所示），导致跨云协作效率低下。针对该问题，本节提出卸载窗口首次适应填充算法（Offload Window FirstFit Fill，OW-FF），通过动态检测和填充协作任务执行过程中的空闲时段（即卸载窗口），优化私有云资源利用率。通过首次适应与限制测试卸载窗口次数的策略降低算法复杂度，使 OW-FF 在提升资源利用率的同时保持线性时间复杂度。

本文将虚拟机在公私云协作过程中由于等待公有云返回处理结果而产生的资源空闲时段定义为卸载窗口（Offload Window, OW）。具体而言 OW 中，$e_w^{(\text{prev})}$ 表示窗口的

算法 4.1 卸载窗口首次适应填充算法（OW-FF）

Input: $\mathcal{E}$: 任务集合, $\mathbf{X}$: 私有云分配矩阵, $\mathbf{Y}$: 公有云分配矩阵, $\mathbf{Z}$: 安全策略矩阵
Output: 任务执行序列 $\mathbf{\Pi} = \{\pi_1, \pi_2, ..., \pi_S\}$

for $i \in \{1, ..., S\}$ **do**
  $\mathcal{Q}_i^{\mathrm{OW}} \leftarrow \emptyset$; // 初始化卸载窗口队列
  $\pi_i \leftarrow \emptyset$; ; // 初始化执行队列
end
while $\mathcal{E} \neq \emptyset$ **do**
  取当前任务 $e_m \leftarrow \arg\min_{e_j \in \mathcal{E}} j$; // 按任务编号升序调度
  $\mathcal{E} \leftarrow \mathcal{E} \setminus \{e_m\}$; // 根据私有云分配矩阵选择当前任务所在私有云虚拟机
  s_i: $\exists i, \mathbf{X}[m][i] = 1$;
  if $\mathcal{Q}_i^{OW} = \emptyset$ **then**
    if e_m 为协作任务 **then**
      $w \leftarrow \{e_w^{(\mathrm{start})}, e_w^{(\mathrm{end})}, \Delta\tau_w, r_w\}$; // 创建新的卸载窗口 w
      $e_w^{\mathrm{start}} \leftarrow e_m^{(\mathrm{E})}$, $e_w^{\mathrm{end}} \leftarrow e_m^{(\mathrm{V})}$ 按式(4-1)计算 $\Delta\tau_w$; // 确定卸载窗口的长度
      $r_w \leftarrow 5$; // 初始化失败计数器
      $\mathcal{Q}_i^{\mathrm{OW}} \leftarrow \mathcal{Q}_i^{\mathrm{OW}} \cup \{w\}$
    else
      $\pi_i \leftarrow \pi_i \oplus e_m^{(\mathrm{SA})}$; // 独立任务直接加入队列
    end
  end
  for $w \in \mathcal{Q}_i^{OW}$ **do**
    if $FT(e_m) \leq ST(e_w^{end}) \wedge ST(e_m) \geq FT(e_w^{start})$ **then**
      $\pi_i \leftarrow \pi_i \oplus e_w^{\mathrm{start}}$; // 原开始任务加入调度队列
      $e_w^{\mathrm{start}} \leftarrow e_m$; // 更新开始任务
      break
    end
    else
      $r_w \leftarrow r_w - 1$; // 尝试插入窗口失败
      /* 删除 5 次尝试插入失败的窗口 */
      if $r_w = 0$ **then**
        $\mathcal{Q}_i^{\mathrm{OW}} \leftarrow \mathcal{Q}_i^{\mathrm{OW}} \setminus \{w\}$ $\pi_i \leftarrow \pi_i \oplus \{e_w^{\mathrm{start}}, e_w^{\mathrm{end}}\}$
      end
    end
  end
end
return $\mathbf{\Pi}$

起始任务，其完成时间 $\mathrm{FT}(e_w^{(\mathrm{prev})})$ 确定窗口的起始时刻；e_w^{end} 表示窗口的终止任务，其固定为验证阶段任务 $e_m^{(\mathrm{V})}$，其开始时间 $\mathrm{ST}(e_w^{\mathrm{end}})$ 作为窗口的终止时刻。窗口时长 $\Delta\tau_w$

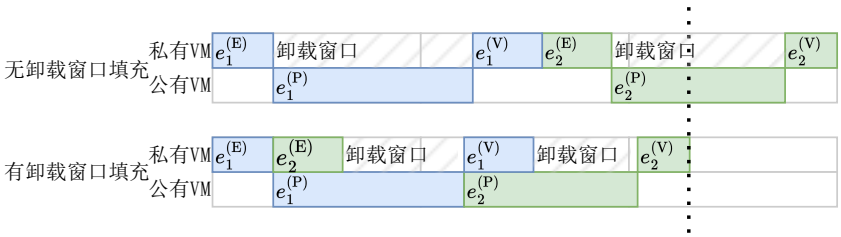

图 4.1　卸载窗口对完工时间的影响示意图

限制了新任务的执行时间上限，确保插入任务不会影响后续任务的执行。当新任务插入窗口时，起始任务 $e_w^{(\text{prev})}$ 更新为最后插入任务的完成时间，而终止任务 e_w^{end} 保持不变，从而维持协作任务的时序逻辑。窗口时长 $\Delta\tau_w$ 可通过公式(4-1)计算，即：

$$\Delta\tau_w = \text{ST}(e_w^{\text{end}}) - \text{FT}(e_w^{(\text{prev})}) \tag{4-1}$$

OW-FF 算法4.1通过任务集合 $\mathcal{E}$ 与决策矩阵 $\mathbf{X}$、$\mathbf{Y}$、$\mathbf{Z}$ 确定执行顺序 $\mathbf{\Pi}$，其核心逻辑包括：任务按编号升序调度（算法第 6 行），确保验证任务在加密任务后执行，维持跨云协作的执行顺序依赖；当检测到协作任务时，若不存在可用卸载窗口，则以加密任务 $e_m^{(\text{E})}$ 与验证任务 $e_m^{(\text{V})}$ 为边界创建新窗口（算法第 11-14 行）；通过顺序扫描窗口队列，仅允许执行时间满足 $\text{FT}(e_{\text{new}}) \leq \Delta\tau_w$ 的任务插入（算法第 20 行）；检测到窗口可插入任务时，通过首次适应策略更新窗口起始任务并调整窗口（算法第 21-22 行）；对于连续五次匹配失败的无效窗口，将其边界任务加入执行队列并移除窗口（算法第 26-29 行）。

本节提出的卸载窗口首次适应填充算法（OW-FF）通过动态检测并填充私有虚拟机因跨云协作产生的资源空闲时段，显著提升了跨云协作效率。作为 NSGA-OW 算法的核心组件，OW-FF 采用首次适应策略与最大重试次数限制，将算法时间复杂度严格控制在 $O(M)$，同时实现近似 HEFT 的任务插入效果，提高了了调度质量与私有虚拟机资源利用率。

4.2 多目标优化问题求解

针对混合云环境下隐私任务调度的多目标优化难题，现有方法常面临两个挑战：安全策略限制了虚拟机分配决策的可行位置导致搜索效率低下；基于遗传的元启发式算法计算速度慢。为此，本章基于 NSGA-II 框架提出改进型多目标优化算法 NSGA-OW，通过混合编码策略与动态偏好机制的创新实现混合云隐私任务调度中多目标联合优化。

NSGA-OW 的改进体现在三方面：首先，通过混合式离散编码结构解耦安全策略与虚拟机分配决策，将隐私安全约束嵌入遗传个体编解码过程，降低安全策略对搜索

效率影响；其次，设计虚拟机感知与成本优化的遗传算子，在维持种群多样性基础上强化关键目标的搜索效率；最后，引入考虑用户偏好的精英选择策略，突破传统非支配排序对多目标同等优化的限制，优先优化安全性与完工时间等用户更感兴趣的指标，加快收敛。该算法通过与上节提出的 OW-FF 卸载窗口检测结合，形成决策变量优化与时序编排的多目标元启发式算法，为混合云环境提供高效、安全的调度解决方案。

4.2.1 混合编码方案设计

针对混合云任务调度中复杂的虚拟机分配与安全策略的决策变量，本节设计了混合编码方案，以平衡搜索效率与解的质量。私有云分配编码采用整数编码策略，每个任务 e_m 的私有云分配决策编码为 $X'_m \in \{1, 2, \ldots, S_k\}$，其中 $S_k = \sum_{i=1}^{S} \mathbf{L}[k][i]$ 表示任务关联数据 d_k 在私有云中的可访问虚拟机数量。通过约束 $X'_m \leq S_k$，编码中内嵌了数据本地化约束（C2），例如当数据 d_1 存储在虚拟机 s_1 和 s_2 时，编码范围限定为 1-2，无需额外的约束校验。公有云分配编码采用唯一索引机制，将公有云虚拟机实例映射为连续整数值 $Y'_m \in \{1, 2, \ldots, V_{\text{total}}\}$，其中 $V_{\text{total}} = \sum_{j=1}^{J} N_j$ 表示公有云虚拟机总数。根据公式(3-1)将编码值转换为具体虚拟机标识符 $v_{(j,n)}$，例如当公有云提供两种类型虚拟机（类型 1 有 3 台，类型 2 有 2 台）时，编码 1-5 分别对应 $v_{(1,1)}, v_{(1,2)}, v_{(1,3)}, v_{(2,1)}, v_{(2,2)}$，这种线性唯一索引机制简化了公有云资源的表示。

在安全策略编码部分，针对同一隐私数据可能被多个任务共享的场景，本文提出数据安全策略编码 $Z_k^{\text{enc}} \in \{q \mid \alpha(d_k, q) \geq \alpha_{\min}(d_k)\}$，其中 $k = 1, 2, \ldots, K$。通过为每个数据 d_k 独立选择满足最低安全等级的策略，而非每个任务单独决策，该机制减少了决策空间。当多个任务共享相同数据时（即 $M > K$），决策变量减少量为 $(M-K)\times Q$，同时通过限制编码范围过滤不符合最低安全要求的选项，确保约束 C1 的成立。任务执行模式编码采用 0-1 编码 $Z_m^{\text{offload}} \in \{0, 1\}$，表示任务是否跨云协作，0 表示私有云独立执行，1 表示公有云协同处理。该编码独立于安全策略选择，允许在满足安全等级约束的前提下，根据负载动态选择执行模式以优化效率。

最终的个体编码由私有云分配、公有云分配、数据安全策略和任务协作模式四部分串联构成，表示为：

$$\begin{aligned}\text{Individual} = &\underbrace{X'_1 \oplus \cdots \oplus X'_M}_{\text{私有云分配}} \oplus \underbrace{Y'_1 \oplus \cdots \oplus Y'_M}_{\text{公有云分配}} \\ &\oplus \underbrace{Z_1^{\text{enc}} \oplus \cdots \oplus Z_K^{\text{enc}}}_{\text{数据安全}} \oplus \underbrace{Z_1^{\text{offload}} \oplus \cdots \oplus Z_M^{\text{offload}}}_{\text{执行模式}}\end{aligned} \tag{4-2}$$

其中“$\oplus$”表示编码串联操作。通过内嵌最低安全需求约束（C1）和隐私数据存储约束（C2），该编码机制避免了约束校验与修复，同时通过压缩安全策略维度，有效降

低了搜索空间复杂度，在多个任务共享同一数据的情况下可以获得更好的调度性能。

4.2.2 虚拟机分块多点交叉算子

算法 4.2　虚拟机分块多点交叉算子

Input: 父代个体 Ind_1, Ind_2; 交叉概率 p
Output: 子代个体 Ind'_1, Ind'_2

1 初始化 $\mathcal{S}_{\text{init}} \leftarrow \emptyset$;
2 **foreach** 私有云虚拟机 $s \in \mathcal{S}$ **do**
3 　**if** *rand*() $< p$ **then**
4 　　$\mathcal{S}_{\text{init}} \leftarrow \mathcal{S}_{\text{init}} \cup \{s\}$; // 筛选待交换的私有云虚拟机
5 　**end**
6 **end**
7 初始化 $\mathcal{E}'_1 \leftarrow \emptyset$, $\mathcal{E}'_2 \leftarrow \emptyset$;
8 **foreach** $s \in \mathcal{S}_{\text{init}}$ **do**
9 　$\mathcal{E}'_1 \leftarrow \mathcal{E}'_1 \cup \{e_m \mid Ind_1.X_m = s\}$; // 收集个体 1 中 s 上的任务
10 　$\mathcal{E}'_2 \leftarrow \mathcal{E}'_2 \cup \{e_m \mid Ind_2.X_m = s\}$; // 收集个体 2 中 s 上的任务
11 **end**
12 $Ind'_1 \leftarrow Ind_1$; $Ind'_2 \leftarrow Ind_2$; // 初始化子代个体
13 **foreach** $e_m \in \mathcal{E}'_1$ **do**
14 　$Ind'_2.X_m \leftarrow Ind_1.X_m$; // 交换私有云分配编码
15 　$Ind'_2.Y_m \leftarrow Ind_1.Y_m$; // 交换公有云实例编码
16 　$Ind'_2.Z_m^{\text{offload}} \leftarrow Ind_1.Z_m^{\text{offload}}$; // 交换卸载决策编码
17 **end**
18 **return** Ind'_1, Ind'_2

在混合云隐私任务调度场景中，任务分配的质量直接影响卸载时隙的利用率与虚拟机负载均衡状态，进而对完工时间等优化目标产生显著影响。传统多点交叉算子对任务编码序列进行独立交叉操作，难以保持虚拟机内部任务分配的执行顺序，容易破坏已优化的卸载窗口结构，导致调度性能下降。针对这一问题，本节提出了一种虚拟机分块多点交叉算子，其核心创新在于将交叉操作的粒度从单任务提升至虚拟机资源单元，通过以同一台私有云虚拟机作为任务分块单位进行交叉操作，从而提升 NSGA-OW 算法的收敛速度。

算法4.2详细描述了虚拟机分块多点交叉算子的实现过程。在算法的 3-5 行中，通过交叉概率参数 p 筛选私有云虚拟机集合 $\mathcal{S}_{\text{init}}$ 作为待交换的基因，这一步骤确保了交叉操作能够以虚拟机为单位交换两个个体中的任务。随后，在算法的 9-10 行中，分别在父代个体中收集与选中虚拟机关联的完整任务块 $\mathcal{E}'$，该任务块不仅包含初始选定任务的私有云分配信息，还涵盖了任务的公有云分配方案及安全策略，从而确保交

叉后的编码能够完整保留父代个体的优化信息。在算法的 14-16 行中，编码交换阶段将父代的私有云分配参数 X_m、公有云实例选择 Y_m 以及加密卸载决策 Z_m^{offload} 作为一个整体与子代个体进行合并，实现了对应虚拟机中任务的完整交换，从而保留了优秀个体的卸载窗口结构与负载均衡策略。

由于4.2.1小节设计的混合编码方案内嵌的隐私与数据约束机制，本文的交叉算子能够在不引入额外修复过程的前提下，确保交叉操作始终生成可行解。通过将交叉粒度提升至虚拟机资源单元，该算子有效继承了父代个体的优质分配模式，同时维持了虚拟机内部任务执行顺序，从而优化了卸载窗口和负载均衡状态。然而，完全采用虚拟机分块的交换策略在进化初期可能导致种群多样性过早丧失，进而引发局部收敛问题。因此，本文保留了传统多点交叉算子，保证种群多样性。

4.2.3 混合云负载感知变异算子

算法 4.3 混合云负载感知变异算子

Input: 父代个体 Ind, 变异概率 $p_0 = 0.05, p_1 = 0.2$

Output: 子代个体 Ind'

```
1  Ind' ← Ind ;                                              // 初始化子代个体
2  根据公式(4-4)计算负载均值 μ_L;
3  根据公式(4-5)计算负载标准差 σ_L;
   // 虚拟机分配决策变异
4  foreach 虚拟机 vm ∈ S ∪ V do
5      L(vm) ← 根据公式(4-3)计算;
6      p ← 根据公式(4-6)计算变异概率;
7      foreach 任务 e_m ∈ M_vm do
8          E ← Ind'.vm ;    // 提取子代个体中分配给虚拟机 vm 的任务对应的编码段
9          foreach 编码位 b ∈ E do
10             if rand() < p then
11                 b ← ¬b ;                                  // 执行位翻转
12             end
13         end
14     end
15 end
   // 安全决策变异
16 foreach 编码位 b ∈ {Z^enc, Z^offload} do
17     if rand() < p_0 then
18         b ← ¬b ;                                          // 执行位翻转
19     end
20 end
21 return Ind';
```

通过对混合云调度问题的分析发现，虚拟机负载过载会使完工时间变长，而公有云虚拟机负载过低导致的资源利用率低下则造成成本目标上升。为了解决负载不均衡对完工时间与成本的负面影响，本文设计了一种负载感知的变异算子。该算子根据虚拟机负载调整变异概率提升了算法的收敛速度，并降低完工时间且优化了成本。该变异算子，针对负载过大的虚拟机，系统增加其变异概率，促使任务向负载接近均值的虚拟机迁移，从而均衡整体负载分布；对于负载过低的虚拟机，通过提高变异概率来尝试关闭负载过低的公有云虚拟机，进而降低运营成本。负载感知变异算子的具体实现详见算法4.3。

为了准确评估混合云环境中的虚拟机负载状况，本节建立了负载计算模型，并定义了相关的统计指标。如公式(4-3)所示，虚拟机负载率可表示为：

$$L(vm)=\begin{cases}\dfrac{\sum_{e_m\in\mathcal{M}_i} o_{\text{proc}}\times \text{size}(e_m)}{A_i}, & vm\in\mathcal{S}\ \text{（私有云虚拟机）}\\ \dfrac{\sum_{e_m\in\mathcal{M}_i} o_{\text{proc}}\times \text{size}(e_m)}{B_j}, & vm\in\mathcal{V}\ \text{（公有云虚拟机）}\end{cases} \tag{4-3}$$

其中，$\mathcal{M}_i$ 表示分配到虚拟机 vm 的任务集合，A_i 为私有云虚拟机 s_i 的计算能力（单位：MHz），B_j 为公有云虚拟机类型 j 的计算能力（单位：MHz），o_{proc} 表示任务 e_m 的计算量（单位：CPU 周期）。。

在排除没有任务的虚拟机后，本文对虚拟机的负载情况进行统计分析，分别计算其负载均值与标准差，如公式(4-4)和(4-5)所示。负载均值 μ_L 表示所有活跃虚拟机负载的平均值，计算公式为：

$$\mu_L=\frac{1}{N}\left(\sum_{s_i\in\mathcal{S}}L(s_i)+\sum_{v_{(j,n)}\in\mathcal{V}}L(v_{(j,n)})\right) \tag{4-4}$$

负载标准差 σ_L 则用于衡量虚拟机负载与平均值的平均距离，计算公式为：

$$\sigma_L^2=\frac{1}{N}\left(\sum_{s_i\in\mathcal{S}}(L(s_i)-\mu_L)^2+\sum_{v_{(j,n)}\in\mathcal{V}}(L(v_{(j,n)})-\mu_L)^2\right) \tag{4-5}$$

其中 N 为活跃虚拟机总数。基于上述统计指标，本文设计了根据虚拟机负载调整变异概率公式：

$$p=\begin{cases}p_0+\dfrac{|L_v-\mu_L|}{3\sigma_L}\cdot(p_1-p_0), & |L_v-\mu_L|\le 3\sigma_L\\ p_1, & |L_v-\mu_L|>3\sigma_L\end{cases} \tag{4-6}$$

该公式通过变异最低概率 p_0 与最大概率 p_1，调整变异算子的位翻转变异概率。对于负载偏离平均值的虚拟机，其变异概率会相应增加，从而促进混合云负载均衡，并尝试关闭关闭负载过低的公有云节点。

4.2.4 考虑用户偏好的精英保留策略

NSGA-II 算法作为一种基于遗传算法的多目标优化方法，虽然具备全局搜索能力强、调度质量高以及提供多样化调度方案的优点，但其收敛速度较慢。在混合云隐私任务调度场景中，完工时间的优化直接影响用户体验，而安全性的提升可以降低隐私数据潜在风险。相比之下，成本目标由于公有云资源的规模效应，其对调度质量的影响相对较小。因此，本文设计了考虑用户偏好的精英保留策略，优先优化安全性与完工时间，以加快调度速度。该策略保留了原始精英选择机制的核心，即通过非支配排序保留精英解以加速收敛，同时利用拥挤度排序维持种群多样性。与原始策略不同，本文通过动态生成权重对拥挤度函数进行加权，引导算法优先优化用户关注的关键目标。同时，参考随机分配权重策略的研究[35]，每轮迭代中动态调整权重，既引导算法优先优化关键目标，又通过多样化的搜索方向避免静态加权导致的过早收敛问题，从而降低丢失全局最优解的风险。

算法 4.4 考虑用户偏好的精英保留策略

Input: 非支配排序后的当前种群 $P = \{F_1, F_2, \ldots, F_k\}$；子代规模 N

Output: 子代种群 P'

$P' \leftarrow \emptyset$；// 初始化子代种群

$i \leftarrow 1$

while $|P'| + |F_i| \leq N$ **do**

  $P' \leftarrow P' \cup F_i$；// 保留精英个体

  $i \leftarrow i + 1$

end

repeat

  随机生成偏好权重 $w_1, w_2, w_3 \sim$ rand()

until 满足 $w_1 > w_2 \wedge w_3 > w_2$;

根据公式(4-7)改进的拥挤度 Crowd$'(s)$ 降序排列 F_i

选取前 $N - |P'|$ 个个体加入 P' **while** $|P'| < N$ **do**

  $P' \leftarrow P' \cup \{F_i[1]\}$；// 选择拥挤度最高的非精英个体

  $F_i \leftarrow F_i[1:]$

end

return P'

具体来说，算法在每次拥挤度计算时随机生成三个权重，并约束安全性与完工时间的权重大于成本目标，确保优化资源向关键目标集中。这允许个体在这两个方向上分布更为密集，以便实现更深入的优化；而在优化成本的方向，减少个体分布密度，从而将更多资源用于优化用户更关注的目标。本文将动态权重在拥挤度计算中考虑

用户偏好，从而使优化算法优先优化关键目标。改进的拥挤度计算如公式(4-7)所示：

$$\text{Crowd}'(s) = w_1 \cdot (1 - \text{norm}(\text{Makespan})) + w_2 \cdot \text{norm}(\text{Security}) + w_3 \cdot (1 - \text{norm}(\text{Cost})) \quad (4\text{-}7)$$

其中 norm$(\cdot)$ 表示目标值的归一化处理函数。对于需要最小化的完工时间与成本目标，采用 $1 - \text{norm}(\cdot)$ 的形式进行处理，使得目标值越小拥挤度越大；对于需要最大化的安全性目标，直接使用 norm$(\cdot)$ 进行转换，确保安全性越高拥挤度越大。算法的具体流程如算法4.4所示，其核心在于动态调整权重以引导优化方向。每次执行精英选择时，随机生成权重 $w_1, w_2, w_3 \in [0, 1]$，并满足约束条件 $w_1 > w_2$ 且 $w_3 > w_2$，以确保完工时间与安全性的权重始终高于成本目标。

4.2.5 考虑卸载窗口的非支配排序遗传算法流程

在本章的前面章节中，我们已逐步实现了 NSGA-OW 算法框架的各个组件：首先，针对混合云资源空闲时段的优化问题，设计了卸载窗口首次适应填充算法（OW-FF），用于动态检测与填充私有虚拟机因跨云协作产生的空闲时段；其次，设计了虚拟机分块多点交叉算子与负载感知变异算子，交叉算子可以交换个体的编码提升收敛速度，变异算子可以为种群引入新编码提升全局搜索性能；最后，引入考虑用户偏好的精英选择策略，在非支配排序过程中优先优化安全性与完工时间目标以加快算法执行速度。在此基础上，本节将上述组件整合为完整的 NSGA-OW 算法框架，通过协同优化机制实现混合云隐私任务的多目标调度优化。NSGA-OW 算法流程如图4.2所示。

种群初始化：为了产生良好的初始种群，本文采用随机初始化策略生成规模为 N 的初始种群。对于每个个体，依据公式(4-8)随机生成遗传编码，具体包括虚拟机分配 X'_m、公有云映射 Y'_m、数据加密策略 Z_d^{enc} 及安全方法选择 Z_m^{method}。其中，虚拟机分配 X'_m 在可用虚拟机范围内随机选择，公有云映射 Y'_m 在公有云的虚拟机范围内内生成，数据加密策略 Z_d^{enc} 从满足最低安全需求的策略集合中随机选取，而安全方法选择 Z_m^{method} 是随机的 0-1 编码。该初始化策略确保了种群的多样性，为后续遗传操作提供了充分的搜索空间。

空闲时段检测与目标函数计算：对当前种群中的每个个体，调用算法4.1进行空闲时段检测与任务排序，根据遗传编码确定的私有虚拟机分配矩阵 $\mathbf{X}$、公有云分配矩阵 $\mathbf{Y}$ 及安全策略 $\mathbf{Z}$，该阶段利用首次适应规则检测并尝试将任务插入空闲时段，同时通过重试限制策略将算法复杂度控制在 $O(M)$。完成调度编排后，分别计算个体的三个目标函数值：完工时间 Makespan、安全性 Security 及成本 Cost。

$$
\begin{aligned}
X'_m &= \text{round}\left(\text{rand}(0,1)\cdot\sum_{i\in S}\mathbf{L}[d][i]\right)\\
Y'_m &= \text{round}\left(\text{rand}(0,1)\cdot\sum_{j\in J}N_j\right)\\
Z_d^{\text{enc}} &= \text{round}\left(\text{rand}(0,1)\cdot\sum_{k\in K}\mathbb{I}(\alpha(d_k,q)\geq\alpha_{\min}(d_k))\right)\\
Z_d^{\text{method}} &= \text{round}\left(\text{rand}(0,1)\right)
\end{aligned}
\tag{4-8}
$$

其中，$\sum_{i\in S}\mathbf{L}[d][i]$ 表示存储任务隐私数据的可用虚拟机数，$\sum_{j\in J}N_j$ 为公有云虚拟机总数，$\sum_{k\in K}\mathbb{I}(\alpha(d_k,q)\geq\alpha_{\min}(d_k))$ 统计符合最低安全需求的策略数。

交叉与变异遗传操作生成新种群：对经过非支配排序与精英选择后的个体执行遗传进化操作。在交叉操作中，采用混合普通多点交叉与虚拟机分块多点交叉（算法4.2）的策略，其中后者以概率 p 选取私有云虚拟机作为交换单元，整体迁移其关联任务的分配方案，保留优质调度模式的同时加速种群收敛。在变异操作中，执行负载感知变异（算法4.3），根据虚拟机负载偏离均值的程度调整位翻转概率（公式(4-6)），从而使虚拟机间负载均衡，并提升种群的多样性。这种混合遗传操作的设计不仅继承了传统 NSGA-II 算法的探索能力，还通过虚拟机分块与负载感知机制强化了局部搜索效率，有效提升了调度质量与算法的收敛速度。

非支配排序与考虑用户偏好的精英选择：将交叉与变异操作产生子代种群与负载种群合并后，采用快速非支配排序算法分层，形成前沿层级集合 $\{F_1,F_2,...,F_k\}$。随后，应用考虑用户偏好的精英选择策略（算法4.4）进行筛选优秀个体进入下一轮迭代。改进的精英选择策略不仅保留了传统 NSGA-II 算法的多样性维持能力，还通过偏好引导机制提升了关键目标方向的搜索效率，从而加快调度算法速度。

迭代终止判断：重复执行空闲时段检测至用户偏好的精英保留步骤，直至达到预设迭代次数 T。最终输出包含非支配解集的 Pareto 前沿解集，各解在安全、时效与成本分布均匀，用户可根据实际需求选择最优调度方案。

4.2.6 时间复杂度分析

本小节分析 NSGA-OW 的时间复杂度。原始 NSGA-II 算法的时间复杂度主要取决于种群规模 L、迭代次数 T、交叉变异算子以及非支配排序的计算开销。在 NSGA-OW 框架中，单个个体的编码由任务分配、虚拟机选择和安全策略三部分构成，编码总长度为 $3M+K$，其中 M 表示任务总数，K 表示隐私数据数量，且满足 $K\leq M$，S 表示私有虚拟机数量。

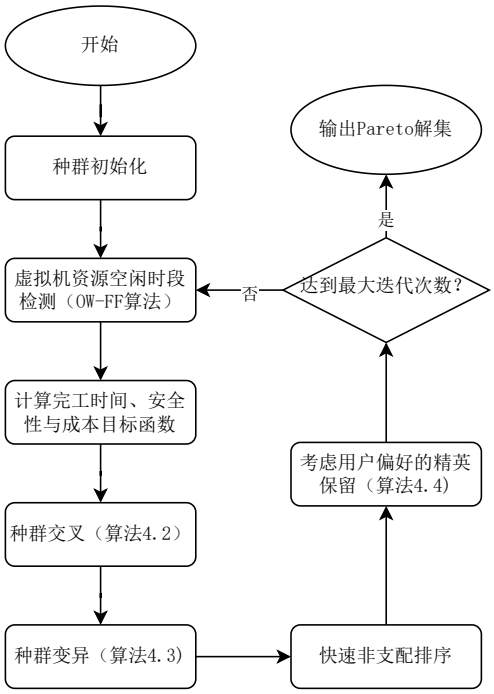

图 4.2　NSGA-OW 算法流程示意图

首先，分析 NSGA-OW 对每个个体的处理开销。原始的多点交叉与位翻转变异算子的时间复杂度为线性，由于隐私数据数量 K 少于等于任务数量 M，其时间复杂度为 $O(M)$。虚拟机分块多点交叉算子在基础交叉操作上增加了私有云虚拟机的筛选与关联任务块的查找操作，其时间复杂度分别为 $O(S)$ 和 $O(M)$，通常情况下任务的数量 M 多于私有云虚拟机的数量 S，因此整体复杂度仍为 $O(M)$。混合云负载感知变异算子增加了计算虚拟机负载均值与标准差步骤，其时间复杂度保持不变，为 $O(M)$。此外，我们卸载窗口首次适应填充算法（OW-FF）的时间复杂度，考虑最复杂的优化场景，即每个任务均在协作处理模式下运行。定义任务的分配与比较为一次单位操作。在此场景下，每个任务包含 3 个子任务，算法按任务下标顺序分配任务，其中公有云任务仅需进行一次分配操作。私有云中维护的卸载窗口队列，在每次窗口匹配失败的最差情况下，每个任务需要分配一个卸载窗口，且每个窗口由于最大重试次数限制最多比较 5 次。综上所述，每个任务最多产生 7 次单位操作，作为常数项时间复杂度可以忽略。因此，OW-FF 算法的时间复杂度为 $O(M)$。因此，NSGA-OW 对每个个体的处理时间复杂度为 $O(M)$。最后，本文为三目标优化算法，快速非支配排序需要每个个体与其他个体进行比较，其时间复杂度为 $O(L^2)$,。因此，NSGA-OW 每次迭代的时间复杂度为 $O(L^2)$。

综上所述，NSGA-OW 的总时间复杂度为 $O(TL^2 + TLM)$，其时间复杂度与迭代次数 T, 种群数量 L 以及任务数量 M 有关，与原始 NSGA-II 保持一致。上述分析表明，本文设计的变异、交叉算子以及 OW-FF 算法并未增加原算法的时间复杂度，在保证优化性能的同时，维持了 NSGA-II 快速非支配排序算法在时间复杂度上的优势。

4.3 本章小结

本章针对混合云环境下隐私任务多目标调度优化问题展开研究，提出了一种考虑卸载窗口的非支配排序遗传算法 NSGA-OW。首先，针对跨云协作过程中虚拟机空闲时段导致资源利用率低下的问题，设计了卸载窗口首次适应填充算法（OW-FF）。该算法通过动态检测空闲时段并引入重试限制策略，将时间复杂度控制在 $O(M)$ 内，可以与遗传算法框架整合。其次，根据多目标优化问题的特点，设计混合编码方案，并通过内嵌约束规则减少不可行解的出现次数，从而提高搜索效率。在此基础上，设计了遗传算子组合：虚拟机分块多点交叉算子以虚拟机为单位交换任务分配方案，确保优质分配模式的完整继承；负载动态变异算子则根据虚拟机负载动态调整变异概率，优化完工时间与运营成本。最后，设计带偏好的精英选择策略，通过随机加权方式引导算法优先优化安全性与完工时间目标，加快关键目标的搜索速度。算法性能方面，改进的 NSGA-OW 的时间复杂度与 NSGA-II 算法相同，确保其在混合云大规模任务调度场景中的高效适用性。

第五章　实验结果与分析

本章通过实验验证所提出的混合云隐私任务调度模型与算法的有效性。实验通过与多目标优化算法对比，测试本文提出的算法的收敛性与解集质量，还通过与最近的混合云隐私任务调度算法对比，分析不同任务规模、数据隐私需求和加密计算开销对调度结果的影响。研究结果表明，本文方法在提升任务执行效率、数据安全性保障方面均展现出优势，同时成本不劣于其他算法。

5.1 实验设置

为了验证所本文提出的模型和算法的有效性。采用 Python 3.13 编程语言，基于 SimPy 4.1 离散事件框架开发混合云仿真系统，运行环境为 Windows 10 操作系统。为降低随机性对实验结果的影响，通过设置随机数种子控制随机过程，最终结果取 10 次实验的平均值。本文实现的混合云离散仿真系统，可以反映本文模型中混合云处理隐私数据的特点与流程：其中私有云作为隐私数据处理核心节点，负责敏感数据存储与任务调度决策；而公有云作为弹性计算资源池，通过安全链路与私有云协同工作。混合云采用隐私数据优先的任务调度策略，具体而言，私有云接收任务请求后，根据数据隐私标签等信息，动态决策任务在本地独立处理或跨云协作执行。

网络拓扑设计参考[2] 中提出的混合云网络模型，公私云虚拟机间直接连接，固定带宽 100Mbps。混合云中跨云传输时延为 75ms（来自阿里云北京-新加坡地域 2025 年 2 月实测）。混合云虚拟机配置参考[47-48]，私有云部署 10 台异构虚拟机，CPU 主频均匀分布于 1000-3000 MHz 区间。公有云资源配置使用 Microsoft Azure Bs v2 系列虚拟机，的 5 种实例类型（3500-56000 MHz），按秒计费，单价区间 0.0104-1.1790 $/h。混合云完整资源配置见表5.1。

表 5.1　混合云虚拟机资源配置

类型	计算能力 (MHz)	费用 ($/h)
私有云	1000-3000*	-
公有云 B2ts v2	3,500	0.0104
公有云 B4ls v2	7,000	0.1470
公有云 B8ls v2	14,000	0.2950
公有云 B16ls v2	28,000	0.5890
公有云 B32ls v2	56,000	1.1790

* 私有云虚拟机的计算能力均匀分布

混合云仿真系统包括一个可配置任务生成器，每次生成 200 个任务。参考[49] 的研究，任务与隐私数据之间存在一一对应关系，即每个任务需要处理一条隐私数据。隐私数据的规模在 10–100 MB 之间均匀分布随机生成，并存储在 10 台私有云虚拟机中的 5 台上。只有存储了对应隐私数据的私有云虚拟机才能处理相关任务。任务的数据量直接由隐私数据的大小决定，而任务的计算结果大小固定为隐私数据大小的 10%，这反映了实际场景中计算结果通常小于输入数据的特点[2]。每处理 1bit 隐私数据所需的 CPU 周期在 600 至 1200 之间随机分配，数据加解密阶段所需的 CPU 周期则取决于所选的隐私加密算法。任务具有不可抢占性，一旦分配给某台虚拟机，便独占资源直至完成。

参考[19] 的研究，每条隐私数据具有最低隐私等级，限定了必须选择不低于该等级的隐私加密算法，以满足合规性要求。为模拟数据多样性，本文设置了三种最低隐私等级：低等级隐私数据范围为-0.1 至 0.1，中等级为 0.2 至 0.9，高等级为 0.9 至 1.0，低、中、高三类隐私数据的比例为 1:8:1。此外，本文基于数据治理的典型场景设计了三种地域化隐私标签，分别对应不同的安全策略：中国标签满足网络安全法合规要求，美国标签符合 NIST 标准，欧洲标签适配 GDPR 合规场景。私有数据的隐私标签比例为 1:1:1，以反映多样化的数据治理需求。

为确保混合云环境下隐私数据的安全性，并满足不同数据所有者的细粒度隐私偏好，本研究构建了一个包含三类加密算法的隐私加密算法组。通过对文献[2, 50-52] 中各类加密方案的加解密开销与安全评估指标的综合分析，并结合中、美、欧三国密码标准算法征集活动的成果，归纳出了三类隐私加密算法组。具体而言，数据所有者 1 的隐私标签采用了中国商密标准的 SM4、SM2 及其配套的 SM3 算法；数据所有者 2 的隐私标签部署了美国 NIST 推荐的 AES、SHA 和 ChaCha20 等加密方案；数据所有者 3 的隐私标签则选用了欧洲 ENCRYPT 密码学竞赛中胜出的 HC-128 等算法。表5.2、表5.3与表5.4分别展示了各隐私加密算法组的加解密开销及安全系数指标。

5.2 评价方案与目标函数介绍

本文选择了两组对比算法，第一组是多目标元启发式优化算法，第二组是隐私任务调度算法。

本文选择了任务调度中常见的多目标元启发式优化算法作为第一组对比算法，以评估 NSGA-OW 算法的多目标优化性能。所选算法包括：

- NSGA-II[39]：经典的多目标进化算法，采用快速非支配排序与精英保留机制，在多种场景中展现了均衡的解集分布和良好的收敛性，常被用于任务调度领域[53]。
- SPEA2[47]：一种改进的多目标进化算法，引入外部存档机制防止边界解丢失，

表 5.2　数据所有者 1 使用的隐私加密算法组

加密方案	加密 (CPU 周期/bit)	解密 (CPU 周期/bit)	安全系数
SM4+SM3[51]	599.53	605.63	0.90
AES+SHA-1[2]	112.41	112.41	0.20

表 5.3　数据所有者 2 使用的隐私加密算法组

加密方案	加密 (CPU 周期/bit)	解密 (CPU 周期/bit)	安全系数
AES+ECC+SHA256[50]	175.83	255.40	0.90
RC4+MD5[2]	66.15	66.15	0.26

表 5.4　数据所有者 3 使用的隐私加密算法组

加密方案	加密 (CPU 周期/bit)	解密 (CPU 周期/bit)	安全系数
HC-128+SHA-1[52]	195.75	183.20	0.60
RC4+MD5[2]	66.15	66.15	0.26

采用支配强度与 k 邻近距离分别优化排序精度和密度评估，其 ZDT 基准测试性能优于传统算法。研究表明，SPEA2 在 ZDT 等基准测试问题上相较于 NSGA-II 展现出更高的性能。

- SMPSO[54]：多目标粒子群优化算法，在单目标 PSO 的基础上引入了外部档案机制，缓解了因粒子移动导致的关键边界解丢失问题。相较于传统进化算法，SMPSO 具有更快的收敛速度。

通过对比上述算法的反世代距离（IGD）以及超体积（HV）指标，可以评估 NSGA-OW 算法的收敛速度与解集质量。

本文选择了混合云隐私任务调度算法作为第二组对比算法，以评估 NSGA-OW 算法的任务调度性能。对比算法包括：

- SPGA[8]：通过结合遗传算法和粒子群优化元启发式算法，该算法优化混合云中的任务调度，同时满足隐私需求，提高私有云资源利用率，并确保任务在截止时间前完成。

- Min-CAMin[11]：Stavrinides 提出了一种根据标准差调整完成时间约束以降低混合云成本的隐私任务调度策略，优先降低混合云的开销同时兼顾优化完工时间，并设计了 Min-CAMin 和 Max-CAMin 两种启发式算法。经过预实验分析，本文选用 Min-CAMin 作为对比算法。该算法优先调度最短完成时间的任务，与 Max-CAMin 相比，能更有效缩短完工时间。

通过比较完工时间、安全性和成本指标，可以评估 NSGA-OW 算法任务调度的性能。

本文提出的 NSGA-OW 算法的参数设定如下：种群规模为 100，迭代次数为 1000。在每次迭代中，种群中每个个体会与其编号相邻的个体进行一次交叉操作；交叉操作采用混合交叉算子，其中普通多点交叉算子和虚拟机分块多点交叉算子的使用概率均为 50%。同时，每个个体可能发生变异，变异概率通过公式(4-6)动态计算，其最低概率为 $p_0 = 0.02$，最高概率为 $p_1 = 0.2$。

由于其他隐私调度算法未考虑多目标优化，本文采用一种简单的多标准决策（Multi-Criteria Decision Making, MCDM）方法，即伪权重距离方法[55]，从 Pareto 前沿中自动提取代表性折衷解，从而实现算法间的性能对比。具体步骤如下：

首先，通过 NSGA-OW 算法生成非支配解集，每个解的三维目标向量（完工时间、安全性、成本）依据式(5-1)进行归一化处理：

$$\mathbf{p} = \left(\mathrm{norm}\left(\mathrm{Makespan}^{(p)}\right), \mathrm{norm}\left(\mathrm{Security}^{(p)}\right), \mathrm{norm}\left(\mathrm{Cost}^{(p)}\right)\right) \tag{5-1}$$

其中，归一化函数定义为 $\mathrm{norm}(x) = (x - x^{\min})/(x^{\max} - x^{\min})$，$x^{\max}$ 和 $x^{\min}$ 分别为目标 $\mathrm{Makespan}^{(p)}, \mathrm{Security}^{(p)}, \mathrm{Cost}^{(p)}$ 在非支配解集中的最大值和最小值。归一化后，目标值被映射到区间 $[0, 1]$，得到伪权重向量 $\mathbf{p} \in [0, 1]^3$。

接着，根据用户偏好定义目标权重参数（本文采用两种权重，分别考虑效率（完工时间）优先 $[0.8, 0.2, 0.0]$ 和安全性优先 $[0.2, 0.8, 0.0]$），并计算所有非支配解的伪权重与目标权重间的欧氏距离。最终选取距离最小的解作为折衷解，分别将其命名为 NSGAOW-1（效率优先）和 NSGAOW-2（安全优先）。

本节设计了算法性能评价方案。首先，通过对比 NSGA-OW 与三种经典多目标优化算法（NSGA-II、SPEA2、SMPSO），验证其在多目标优化方面的性能；其次，对比两种混合云隐私任务调度算法，以完工时间（式(3-16)）、安全性（式(3-20)）及成本（式(3-4)）为评价指标，评估隐私任务调度性能。采用伪权重法从 Pareto 前沿解集中选取效率和安全性两种偏好的折衷方案，分别命名为 NSGAOW-1 和 NSGAOW-2，为混合云任务调度提供多样化解决方案。

5.3 多目标优化的收敛性与非支配解集质量分析

为评估多目标优化算法的性能，本文选取反转世代距离（Inverted Generational Distance, IGD）和超体积（Hypervolume, HV）指标对解集的收敛性及多样性性能进行分析。统一采用的 OW-FF 以充分利用虚拟机的资源空闲时段，通过对比 NSGA-OW 与 NSGA-II、SPEA2 和 SMPSO 三种元启发式算法的实验结果，可验证改进算法的多

目标优化性能。实验结果如图 5.1和图 5.2所示。

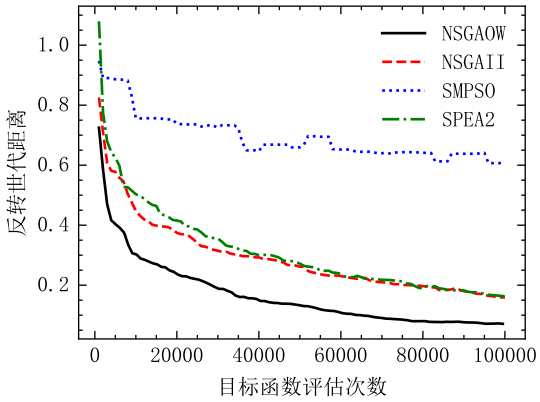

图 5.1　各算法收敛速度比较

实验结果表明，NSGA-OW 在收敛性能和全局解集质量上均显著优于对比算法。在收敛速度方面，NSGA-OW 仅需 30000 次函数评估即可达到 IGD 值 0.19 的水平，而 NSGA-II 与 SPEA2 需 83000 次和 79000 次评估才能达到同等精度；在解集质量方面，NSGA-OW 的 HV 指标为 0.53 分别较 SPEA2 和 NSGA-II 提升 35.9% 与 43.2%。这种性能提升主要得益于针对混合云任务调度问题改进的遗传算子设计。虚拟机分块多点交叉算子以虚拟机为单位交换任务分配方案，保留了优质分配模式的完整性，而负载动态变异算子通过动态调整变异概率进一步提升了搜索效率和精度。值得注意的是，SMPSO 在离散调度任务场景中表现不佳。尽管其采用了外部存档和拥挤机制来优化多目标搜索性能，但 SMPSO 属于群优化算法，基于连续空间的速度更新规则难以有效匹配虚拟机分配问题的离散决策特征。实验结果验证了 NSGA-OW 多目标优化算法在求解本文多目标优化问题上相比其他算法具有良好的收敛速度与解集质量。

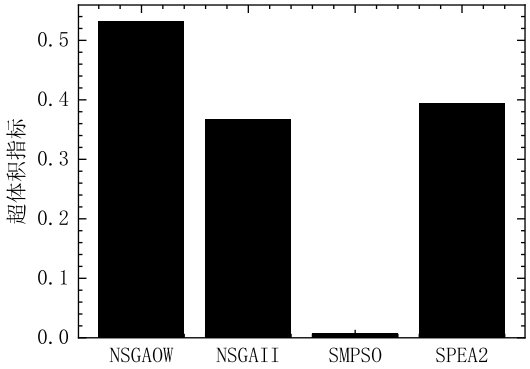

图 5.2　超体积指标

5.4 任务量和数据规模对算法的影响

本节实验分为任务量和数据规模两个维度进行性能评估。在任务量对比实验中，隐私数据量服从 [10-100] Mbit 的均匀分布，任务总数分别设置为 100、200、300、400 和 500 个任务，重点考察任务数量维度变化对调度性能的影响规律。在数据规模对比分析中，保持任务总数为 200 个恒定，通过设定隐私数据量的均值分别为 10、30、50、70 和 90Mbit 构建不同测试场景，其中每个场景的数据量以对应的均值中心 ±5 Mbit 均匀分布展开（例如均值 30 Mbit 时数据量分布在 [25,35] Mbit 范围内均匀分布），旨在量化分析单任务数据规模变化对调度目标的影响特征。

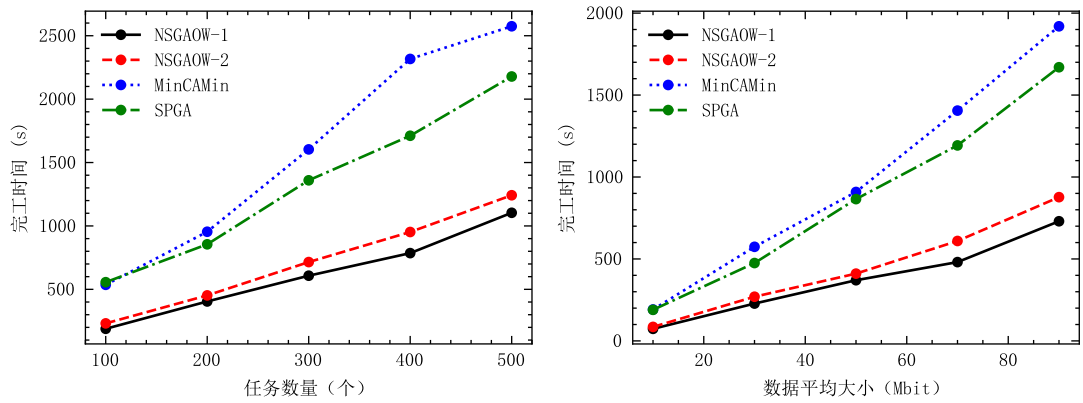

图 5.3 任务数量与数据规模对混合云完工时间的影响分析

首先，本文分析任务数量与数据规模对完工时间的影响。从实验结果可以看出，无论是任务数量还是数据规模的增加，完工时间均呈现上升趋势，这符合一般认知。值得注意的是，当任务数量为 200（数据量均值为 55 Mbit）时，完工时间为 452.18 秒，与数据规模实验中的 50 Mbit 场景 410.47 秒（NSGAOW-2）比较接近。这一结果验证了实验环境的可复现性。

接下来，本文观察到 NSGAOW-1 和 NSGAOW-2 在两种实验环境下均表现出最佳的完工时间性能。例如，在任务数量为 500 时，NSGAOW-1 的完工时间为 1103.69 秒，显著低于 MinCAMin 的 2574.63 秒和 SPGA 的 2178.51 秒。这一优势主要源于 NSGAOW 算法在设计中对公私云协作和虚拟机空闲时段的优化，显著提高了资源利用率。相比之下，对比算法 SPGA 和 MinCAMin 均由于采用单目标优化策略，未能达到完工时间的最优化。SPGA 以私有云利用率为首要优化目标，MinCAMin 以混合云成本为优化核心，其完工时间在 500 任务时分别为 2178.51 秒和 2574.63 秒，显著高于 NSGAOW 系列算法。这表明，仅通过任务完成时间约束间接优化完工时间的效果，不如直接优化多目标的 NSGAOW 算法。

此外，本文发现 NSGAOW-2 的完工时间普遍高于 NSGAOW-1，例如在任务数量为 500 时，NSGAOW-2 的完工时间为 1242.01 秒，而 NSGAOW-1 为 1103.69 秒。这

是由于 NSGAOW-2 优先考虑安全性优化，在保证高安全性的同时，牺牲了部分完工时间。然而，得益于本文改进的公私云协作模型和多目标优化特性，NSGAOW-2 相较于其他算法（如 MinCAMin 和 SPGA）仍表现出显著的完工时间优势。例如，在数据规模为 90 Mbit 时，NSGAOW-2 的完工时间为 877.04 秒，远低于 SPGA 的 1669.69 秒。

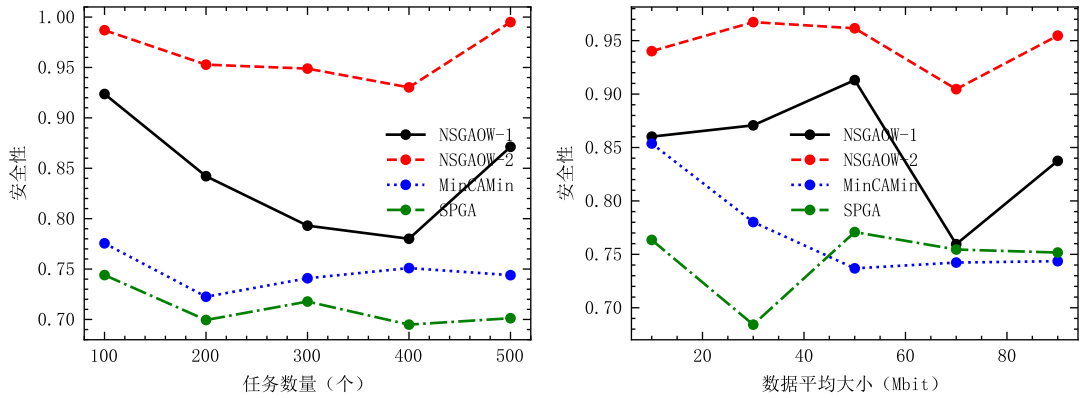

图 5.4　任务数量与数据规模对混合云安全性的影响分析

本文分析任务数量与数据规模对安全性的影响，在这里，本文发现 NSGAOW-2 的效果优于其余 3 种策略，而 NSGAOW-1 的安全性不劣于对比算法。例如：在任务数量为 500 时，NSGAOW-2 的安全性达 1.00，而 MinCAMin 和 SPGA 分别为 0.74 和 0.70；应了 NSGAOW 算法在优化本文提出量化的安全性指标中可以获得较好的结果，降低了隐私数据的泄露机会，能更好保护隐私安全。

同时，尽管 NSGAOW-1 更倾向于通过公私云协作优化完工时间，但其安全性仍不劣于对比算法。例如：在任务数量为 100 时，NSGAOW-1 的安全性为 0.92，高于 MinCAMin 的 0.78 和 SPGA 的 0.74；这种表现得益于多目标优化方法的特性：只有安全性、完工时间与成本至少有一方占优的非支配解才有可能成为备选调度方案。此外，本文采用伪权重法（权重为 [0.8, 0.2, 0.0]）选择调度方案，尽管偏向完工时间优化，但仍兼顾了安全性，因此 NSGAOW-1 在安全性上仍有一定优势。

同时本文发现，当任务数量增加时，NSGAOW-1 与 NSGAOW-2 的安全性差距显著扩大。例如：在任务数量为 100 时，两者安全性分别为 0.92 和 0.99，差值仅为 0.07；在任务数量为 500 时，两者安全性分别为 0.87 和 1.00，差值扩大至 0.13。这种现象的原因是：随着任务数量的增加，私有云负载显著上升，NSGAOW-1 倾向于通过更多的公私云协作降低完工时间，从而牺牲了部分安全性以提升任务执行效率。这反映了 NSGAOW 算法能够为用户提供多样化的调度方案，满足不同的优化需求。

在数据量较大时，NSGAOW-1 的安全性表现有所下降。例如，在数据量为 70 Mbit 时，NSGAOW-1 的安全性为 0.76，与 MinCAMin 的 0.74 和 SPGA 的 0.75 接近。这一现象的主要原因是：高数据量场景下，高开销加密算法的处理时间显著增加，导致私

有云资源的空闲时段无法被充分利用，因此 NSGAOW-1 倾向于选择开销较低但安全性较弱的加密算法。这也体现了数据量对算法策略的动态影响。

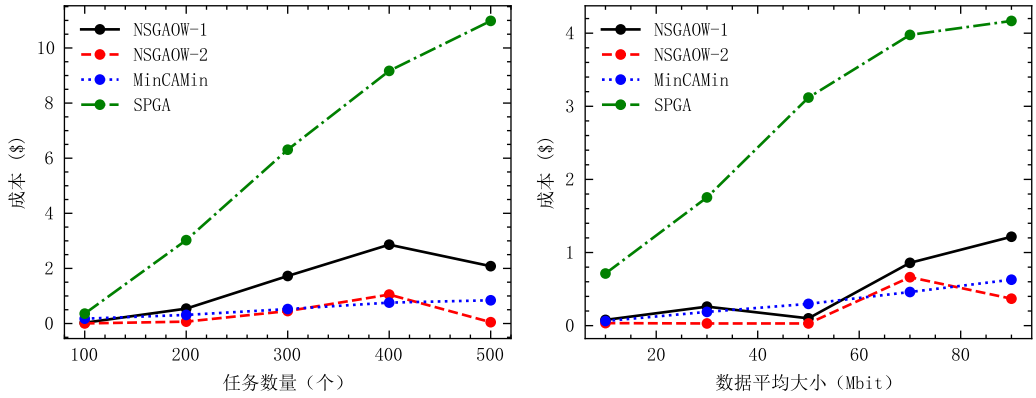

图 5.5　任务数量与数据规模对混合云成本的影响分析

在成本方面，本文从实验数据中发现 NSGAOW-2 和 MinCAMin 表现最优，在数据量为 90Mbit 是分别为 0.37 元和 0.63 元。这是因为 MinCAMin 的主要优化目标就是成本，它会通过第一轮循环筛选完成时间满足要求的虚拟机，第二轮循环中从这些虚拟机中选择成本最低的虚拟机。而 NSGAOW-1 同时优化安全性与成本，且选择安全性偏高的调度方案，而在私有云独立处理的安全性最高，而本文认为私有云成本固定，选择高安全性也就使成本下降。而 SPGA 虽然考虑最大化私有云资源利用率，理论上可以降低成本，却未考虑优化公有云成本。导致 SPGA 会倾向租用更多公有云虚拟机以满足截止时间需求。导致成本最高，如 90 Mbit 时达 4.17 元 NSGAOW-1 优先考虑完工时间，也就会租用更多公有云虚拟机，增加成本。如 90 Mbit 时成本为 1.22 元，显著低于 SPGA 的 4.17 元

本节实验表明，NSGA-OW 算法具有良好性能。其中，优先效率优化的 NSGAOW-1 策略在任务量为 200 数据规模为 [10, 100] Mbit 的均匀分布时，完工时间为 404.43 秒，相较于对比算法 SPGA（855.07 秒）降低了 52.7%。而优先安全性优化的 NSGAOW-2 策略在安全性指标上同样表现优异，其安全性为 0.95，相较于对比算法 MinCAMin（0.72）提升了 31.9%。同时 NSGA-OW 在成本上不劣于对比算法。

5.5 数据隐私需求对算法的影响

本小节研究隐私数据特性对调度算法性能的影响，分析数据的最低隐私需求、数据热点及存储分布对完工时间的影响。在最低隐私需求的影响实验中，本文固定数据的范围是 5-10Mbit，本文通过调整任务的最低隐私需求范围，设计了三种不同的场景：场景 2 为低隐私需求占 10%、中隐私需求占 80%、高隐私需求占 10%（原始配置）；场景 1 为低隐私需求占 70%、中隐私需求占 30%（整体隐私需求较低）；场景 3

为中隐私需求占 20%、高隐私需求占 80%（整体隐私需求较高）。

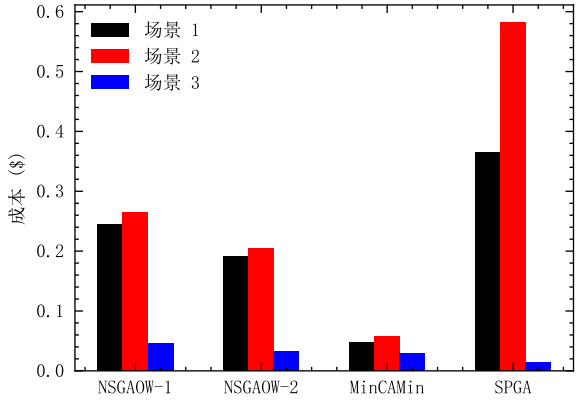

图 5.6　不同隐私需求场景对成本的影响

通过对不同安全性场景下算法成本表现的分析，本文总结出以下关键发现：

在场景 2 中，四类调度策略均表现出最高的成本，如 NSGAOW-1 成本为 0.26 元，SPGA 成本为 0.46 元。这是因为场景 2 包含了最多中等隐私需求的数据，这类数据可以通过跨云协作处理，但必须使用高等级的隐私加密算法，导致加解密开销显著增加，进而消耗更多公有云资源，推高了整体成本。

从场景 1 到场景 2，NSGAOW-1 的成本从 0.24 元增至 0.26 元，上升了 8.3%，而 NSGAOW-2 的成本上升了 5.3%，从 0.19 元增至 0.20 元，相比其他算法的上升幅度最少。这种小额上升反映了 NSGAOW 系列在跨云协作中的优势，其通过动态调度策略优化资源分配，降低了加解密开销对成本的影响。而 SPGA 的成本从 0.36 元增至 0.46 元，上升了 27.8%，在四类策略中上升最严重。这是因为 SPGA 仅考虑满足隐私安全约束，未对安全性进行量化，导致在需要高等级安全时，难以权衡在低成本私有云或高成本公有云之间的选择，造成资源浪费与成本上升。

最后，在三类场景中，MinCAMin 始终保持了最优的成本性能，场景 1 为 0.05 元，场景 2 为 0.05 元，场景 3 为 0.03 元，这是由于其设计目标集中于成本优化。然而，在高隐私需求的场景 3 中，MinCAMin 的成本优势不再明显，其成本为 0.03 元，与 NSGAOW-2 和 NSGAOW-1 的 0.03 和 0.05 元接近，这是因为高隐私数据强制使用私有云资源处理任务，限制了成本优化空间。

在完工时间方面，本文从实验数据中发现不同算法在不同场景下的完工时间表现出显著差异，体现了其优化目标与策略的根本区别。具体分析如下：整体来说，本文发现更关注成本与安全性的调度策略，NSGAOW-2 与 MinCAMin 这两个算法的完工时间在场景 3（高安全性需求）中低于场景 1（低安全性需求）。例如，NSGAOW-2 在场景 3 中的完工时间为 102.62 秒，低于场景 1 的 116.98 秒；MinCAMin 在场景 3 中的完工时间为 132.72 秒，也低于场景 1 的 146.41 秒。这种现象的原因是：这两种

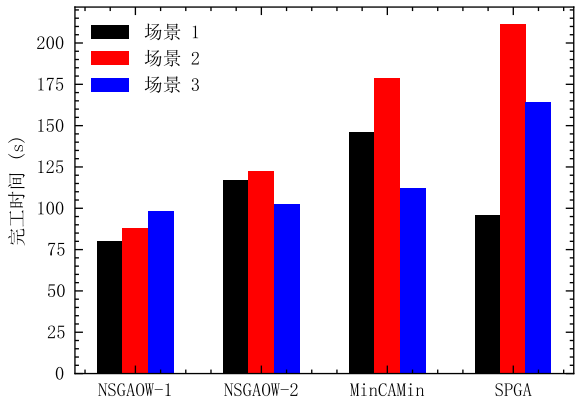

图 5.7　不同隐私需求场景对完工时间的影响

算法优先使用私有云资源处理任务以提高安全性并降低成本。在场景 1 中，由于部分隐私数据可进行公私云协作，而非完全独立处理，这种混合模式可能导致私有云资源利用率降低，进而影响公私云协作效率，导致完工时间延长。而在场景 3 中，所有隐私数据均被强制调度至私有云处理，私有云资源的集中利用反而提高了整体效率。

而而优先优化完工时间的的调度策略 NSGAOW-1、SPGA 这两种算法的完工时间在场景 1 中低于场景 3。例如，NSGAOW-1 在场景 1 中的完工时间为 80.15 秒，低于场景 3 的 98.17 秒；SPGA 在场景 1 中的完工时间为 96.19 秒，也低于场景 3 的 129.77 秒。这与它们的优化目标密切相关：NSGAOW-1 与 SPGA 优先优化完工时间，因此在场景 1 中尽可能利用公私云协作来加速任务处理，而在场景 3 中，由于高安全性需求限制了公私云协作的使用，完工时间相对增加。

而完工时间方面，本文发现不同算法在不同场景下的完工时间有着不同的特性。整体来说，本文发现更关注成本与安全性的调度策略，如 NSGAOW-2 与 MinCAMin，场景 3 的完工时间会小于场景 1 的完工时间。这是因为他们优先使用私有云处理任务以提高安全性并降低成本，然而场景 1 中隐私数据可公私协作又可独立处理的，而独立处理的任务会大量占用私有云资源，导致公私云协作效率下降，进而造成完工时间延长。而优先优化完工时间的的调度策略（NSGAOW-1、SPGA）会尽量将任务优先公私协作，因此场景 1 的完工时间会大于场景 3 的完工时间。

5.6 加密计算开销对调度算法的影响

在加密开销对调度算法的影响研究中，为了控制变量，实验固定隐私加密算法组中仅包含两条策略：一条是协作安全策略，即隐私数据经过加密后由公有云和私有云协作处理；另一条是私有云独立处理策略。实验在固定任务数量、数据大小等条件下，通过调整协作安全策略的加密开销来观察调度性能的变化。根据前文隐私加密算法组的加密开销范围，调整加解密开销的范围确定为 50,200,350,500,650 CPU 周

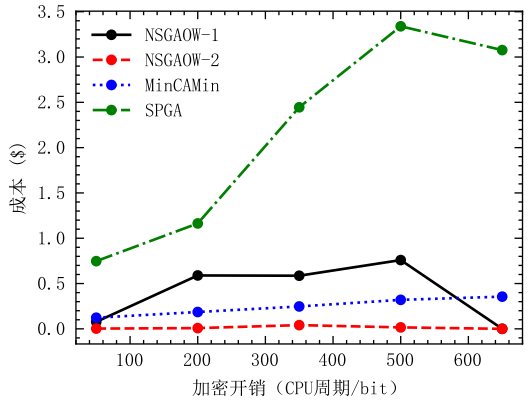

图 5.8　不同加密开销对成本的影响

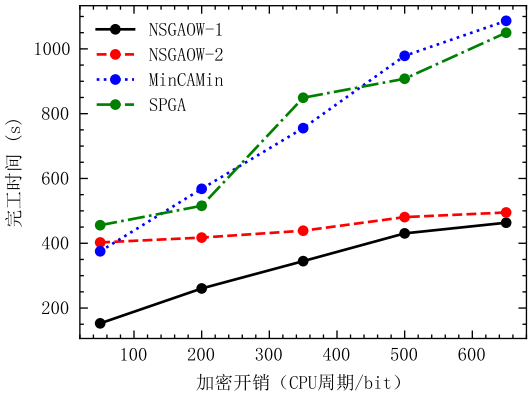

图 5.9　不同加密开销对完工时间的影响

期/bit。

本文对分析加密开销对成本的影响发现：SPGA 在成本方面表现最差，其成本在所有加密开销场景下均显著高于其他算法，如开销为 500 时，SPGA 成本为 3.34 元，而 NSGAOW-2 仅为 0.02 元。这主要源于 SPGA 仅考虑提高私有云利用率，未优化公有云租用，导致公有云资源利用率低下。其次，NSGAOW-1 在多数场景下成本高于 NSGAOW-2，这是由于 NSGAOW-1 优先通过公私云协作降低完工时间，增加了公有云租用开销。然而，当加密开销达到 650 时，NSGAOW-1 与 NSGAOW-2 的成本均趋近于零（分别为 0.003 和 0.001 元），表明此时公私云协作对效率的贡献几乎消失。这种现象是因为：当加密开销接近或超过在私有云直接处理的开销时，公私云协作无法进一步优化完工时间或成本，因此算法选择私有云处理隐私数据。基于这一发现，本文初步提出了一个可用于判定隐私加密算法是否适用于混合云公私云协作的指标：加密开销小于直接计算的开销。此外，NSGAOW-2 和 MinCAMin 在成本方面表现优异，这得益于二者均优先将任务调度至私有云处理，从而减少了公有云租用开销。

本文分析了加密开销对完工时间的影响，发现随着加密开销的增加，所有算法的完工时间普遍呈现上升趋势。这种增长主要由加密计算的复杂性和资源消耗增加所

致。

其次，NSGAOW-2 的完工时间增长幅度最低，这得益于其优先利用私有云空闲资源处理任务，从而减少了对加密开销的敏感性。例如，当加密开销从 50 增加到 650 时，NSGAOW-2 的完工时间仅从 402.52 秒增至 495.06 秒，增幅为 23%，显著低于其他算法。

最后，随着加密开销的增加，NSGAOW-1 与 NSGAOW-2 的完工时间差值逐渐缩小。例如：在开销为 50 时，差值为 249.74 秒，NSGAOW-1：152.78 秒，NSGAOW-2：402.52 秒；在开销为 650 时，差值缩小至 31.63 秒，NSGAOW-1：463.43 秒，NSGAOW-2：495.06 秒。

这种变化的原因与加密开销对成本的影响一致：加密开销的增加削弱了公私云协作对完工时间的优化贡献。当加密开销接近或超过私有云直接处理的开销时，公私云协作的收益显著降低，因此 NSGAOW-1 的完工时间逐步趋近于 NSGAOW-2。这一现象进一步验证了公私云协作的适用性受加密开销与直接计算开销比值的约束。

5.7 本章小结

本章通过实验验证了所提出的混合云动态细粒度隐私任务调度模型与 NSGA-OW 多目标调度算法在任务调度中的性能表现。实验分为多目标优化性能、任务量与数据规模影响、隐私需求影响以及加密开销影响四个部分。首先，通过 IGD 和 HV 指标评估发现，NSGA-OW 在收敛速度和解集质量上均优于对比算法，这一优势主要得益于针对混合云隐私任务调度问题所改进的遗传算法框架。其次，任务量与数据规模实验 NSGA-OW 算法在完工时间上降低了 52.7%，安全性提升了 31.9%，同时在成本上不劣于对比算法。随后，隐私需求影响实验显示，NSGA-OW 在不同隐私需求场景下均能有效平衡效率与安全，其在处理高隐私需求任务时表现尤为优异。最后，加密开销实验中观察到一个指标用于判定隐私加密算法是否适用于混合云公私云协作的指标，隐私加密算法开销小于直接计算的开销。本章实验结果表明，所提出的模型在多目标优化、任务调度效率及安全性方面均有一定优势，为混合云环境下隐私任务的高效调度提供了有效解决方案。

第六章　总结与展望

6.1 总结

混合云架构结合了私有云的隐私安全能力与公有云的丰富资源，实现了对隐私数据的高效处理。随着数据隐私安全法规的不断完善以及隐私保护需求的日益提升，如何在隐私安全要求下实现高效的任务调度成为关键挑战。现有研究中，完工时间等效率要求与隐私安全性之间也难以平衡，缺乏公私云协同机制降低了处理速度，粗粒度的隐私安全约束难以适应多样化的隐私需求。针对上述挑战，本文构建了动态细粒度的隐私任务调度模型，并设计了多目标隐私任务调度算法，以优化完工时间、安全性与成本三个目标。

本文工作总结如下：

1. 分析了混合云隐私任务调度领域的研究现状，发现现有调度方法在效率与安全性权衡方面存在不足，同时在细粒度隐私保护、公私云间协作方面需要进一步加强。基于此，本文制定了建立动态细粒度的隐私任务调度模型并设计多目标隐私任务调度算法的研究方案。

2. 构建了动态细粒度的隐私任务调度模型。结合隐私数据所采用的隐私保护等级，制定了混合云系统级安全评估指标，为安全与效率的权衡提供了依据。此外，通过设计细粒度的隐私标签，确定合适的隐私隐私保护等级并确定加密开销，同时使用线性工作流模型描述公私云协同处理任务的模式，以提升混合云安全性和协作效率。

3. 设计了 NSGA-OW 多目标隐私任务调度算法，结合遗传算法与虚拟机空闲时段优化启发式规则，同时优化完工时间、安全性与成本指标，实现了效率与安全性的权衡。

4. 通过实验验证了模型与算法的有效性。对实验结果的分析表明，由于本文考虑了细粒度的隐私标签以及公私有云间精确的依赖协同，相较于传统的单目标隐私调度算法，本文方法在完工时间、安全性与成本方面具有一定优势。同时，本文设计的 NSGA-OW 算法相比传统的多目标元启发式算法，在收敛速度与 Pareto 前沿解集质量上表现出更好的性能，验证了模型与算法的有效性。

6.2 展望未来

混合云作为隐私数据处理的理想平台，在政务处理等领域具有重要价值。虽然本文提出的动态细粒度的隐私任务调度模型与多目标调度算法解决了混合云中效率与安全的权衡等问题，但由于作者水平与时间限制，仍有一些问题需要未来进一步研究：

1. 考虑具有复杂依赖关系的任务场景。本文考虑任务间无依赖的任务调度，并分析了公私云间协同处理时任务内部的依赖关系。但对于具有任务间依赖关系的科学工作流，或者具有分支与重复交换的更复杂的隐私保护算法，本文的模型并不适用。未来可以研究可动态调整拓扑结构的 DAG 工作流[56]等方法，以应对这些复杂场景。

2. 结合数据动态放置与任务调度，进行联合优化。为了简化模型，本文假设隐私数据存储在固定的私有虚拟机中，但这难以应对热点数据引发的私有虚拟机负载不均衡问题。后续研究可考虑数据的动态放置与缓存策略[57-58]，优化数据的放置与调度，以增强混合云处理热点数据的能力。

3. 进一步加强算法的实时性，提高调度算法的实用价值。尽管 NSGA-OW 算法通过改进遗传算子等技巧提高了收敛速度，但由于其多目标遗传算法的本质，优化速度相比启发式算法依然不足。因此，未来可以通过对优化问题进行进一步分析与分解，将一部分优化目标转化为近似凸目标，从而快速求解。此外，也可以考虑采用机器学习以及启发式等方法提升算法的运行速度。