1、背景

随着科技的飞速发展和市场经济的快速变革，信息技术与物理设备的结合已经成为制造业发展的新动力。物联网（IoT）技术作为其中的核心部分，正在深刻改变传统制造业的面貌。物联网通过将传感器、执行器和计算设备等连接到互联网，构成了一个能够实时感知、传输和处理信息的智能网络。这一技术使得传统制造业能够实现数据驱动的智能化管理和自动化生产，例如在汽车制造中，通过物联网传感器实时监控生产线上的每一步，数据被用来优化装配过程，减少错误和停机时间，实现了更短的生产周期和更高的产品质量。近年来，随着物联网设备的普及和计算能力的提升，3C 制造业中的智能化水平不断提高。特别是在手机制造等领域，物联网设备通过与生产机器的连接，不仅能实时收集生产数据，还能优化生产过程中的各个环节，提高生产效率。例如，通过实时监测设备运行的状态和车间生产的流程，IoT 系统可以及时发现生产中的瓶颈和问题，优化生产调度，减少设备停机时间。然而，智能化生产的过程中产生了大量复杂的计算任务。由于本地的工业生产设备如机械臂和机床等的计算能力有限，本地设备执行计算任务难度较大，花费的时间也很长。

计算卸载技术为该问题提供了可行解决方案。工业生产中通过在车间部署边缘服务器，同时与云端建立长期外部连接，可将生产过程中产生的计算任务卸载至算力更强的边缘或云端服务器处理。但任务卸载过程会因数据传输产生额外时延，因此如何平衡计算时间与传输

时延以最大化生产效率成为重要考量。

在此基础上，建立了一个边缘云架构，辅助工业车间处理复杂的计算任务，如图 1 所示。

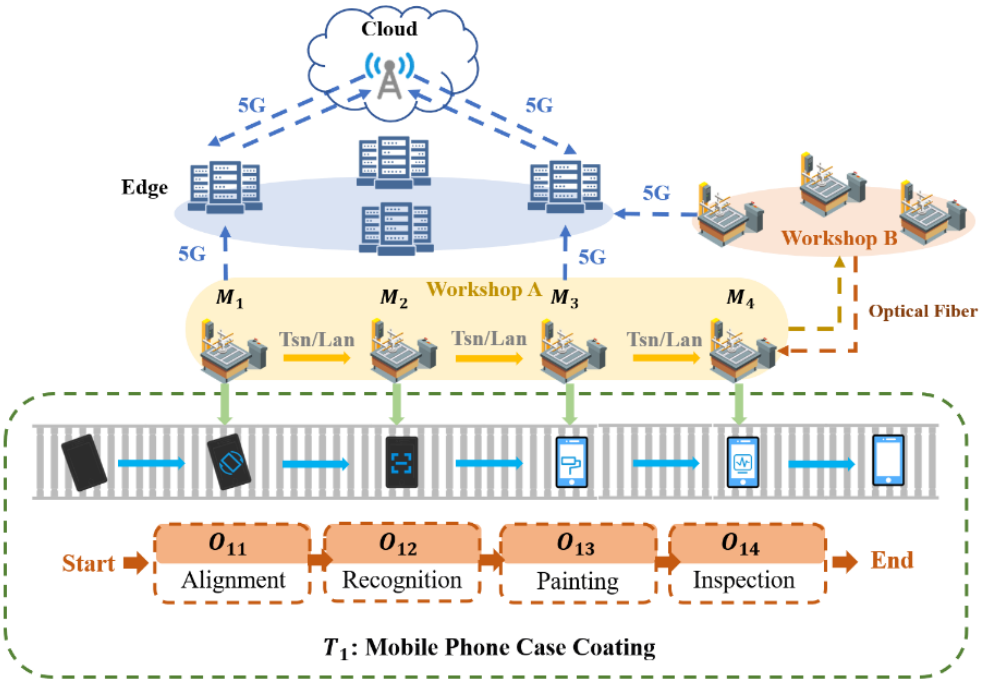

图 1 云辅助移动边缘计算架构

然而，仍有三个问题需要解决。第一个是通信延迟问题。机器、边缘和云端之间的计算任务卸载过程不可避免地会造成额外的通信延迟，然而在实际工业生产过程中，有些计算任务是延迟敏感性，因此需要尽可能**降低系统的总时间延迟**。其次是资源分配问题，由于计算任务会被分配至机器、边缘或者云端，计算任务所需的计算资源大小是不同的，如何合理的将计算任务进行**分配实现资源的有效配置**也是一个重要的问题。该架构的成本还包括整个系统中的能耗成本。计算任务在本地、边缘和云端会产生不同的能耗，如何在调配过程中**尽可能最小化能耗**也是一个很重要的角度。现有的计算卸载策略很难在该架构中**平衡资源分配和成本，同时满足延迟要求**。

具体三个阶段的执行过程如下：

(1) 本地设备将计算任务上传至边缘/云端或者保留在本地

(2) 在本地\边缘\云端执行计算任务

(3) 边缘和云端将计算后的结果传输回本地设备

目标：希望通过确定系统内决策变量的值优化系统时延

2、计算卸载时延最优化问题模型构建

系统内符号及解释：

MAIN PARAMETERS

Notation	Definition
T_i	Production task i
M_j	Machine j
O_{ik}	Operation k of T_i
t_i^c	Completion time of T_i
t_i^w	Waiting time for T_i
t_i^p	Processing time for T_i (including all operations)
t_{ik}^o	Processing time for operation k in task i
T^α	Execution time of all production tasks
t_i^d	Data transmission latency for T_i
t_i^e	Computation time for T_i
d_i	Data size required by T_i
s^e	Data transfer speed of the edge device
s^c	Data transfer speed of the cloud
c_i	Computational load of T_i
f_j^m	Computational capacity of M_j
f^e	Computational capacity of the edge device
f^c	Computational capacity of the cloud
γ_i	Importance score of T_i
h_i^m	Importance weight of T_i
δ_i	Urgency score of T_i
z_i	Deadline of T_i
ϵ_i	Resource demand index of T_i
$\check{t}^c$	Minimum computational resource
$\hat{t}^c$	Maximum computational resource
S^p	Reverse priority score

系统时间总花费包括执行生产任务的时间和完成计算任务的时间（数据传输时延+执行计算任务），此外，系统兼顾考虑了任务优先级，分别从任务重要性、任务紧迫性和资源花费三个角度考虑。

生产任务花费时间：

$$T^{\alpha}=\max(t_1^c,...t_i^c)$$

$$t_i^c=t_i^w+t_i^p$$

$$t_i^p=\sum_{k=1}^{K}t_k^o$$

计算任务花费时间：

$$T^{\beta}=\sum_{i=1}^{I}t_i^d+\sum_{i=1}^{I}t_i^e$$

任务重要性指数

$$\gamma_i=t_i^w\times h_i^m$$

任务紧迫性指数

$$\delta_i=\exp\left(-(z_i-t_i^w)\times h_i^m\right)$$

资源需求系数

$$\epsilon_i=\frac{t_i^c-\check{t}^c}{\hat{t}^c-t_i^c}$$

反优先级指数

$$S_i^p=\gamma_i\times\delta_i\times\epsilon_i$$

系统时延及优先级：

$$R=T^{\alpha}+T^{\beta}+S^p$$

优化目标

$$\min R$$

决策变量：

DECISION PARAMETERS

Notation	Definition
T^{α}	Total execution time of production
T^{β}	Total computation time of compution and transmission
γ_i	Importance of task T_i
δ_i	Urgency of task T_i
ϵ_i	CPU resources required by task T_i
S^p	Reverse priority score of the task

3、样例结果

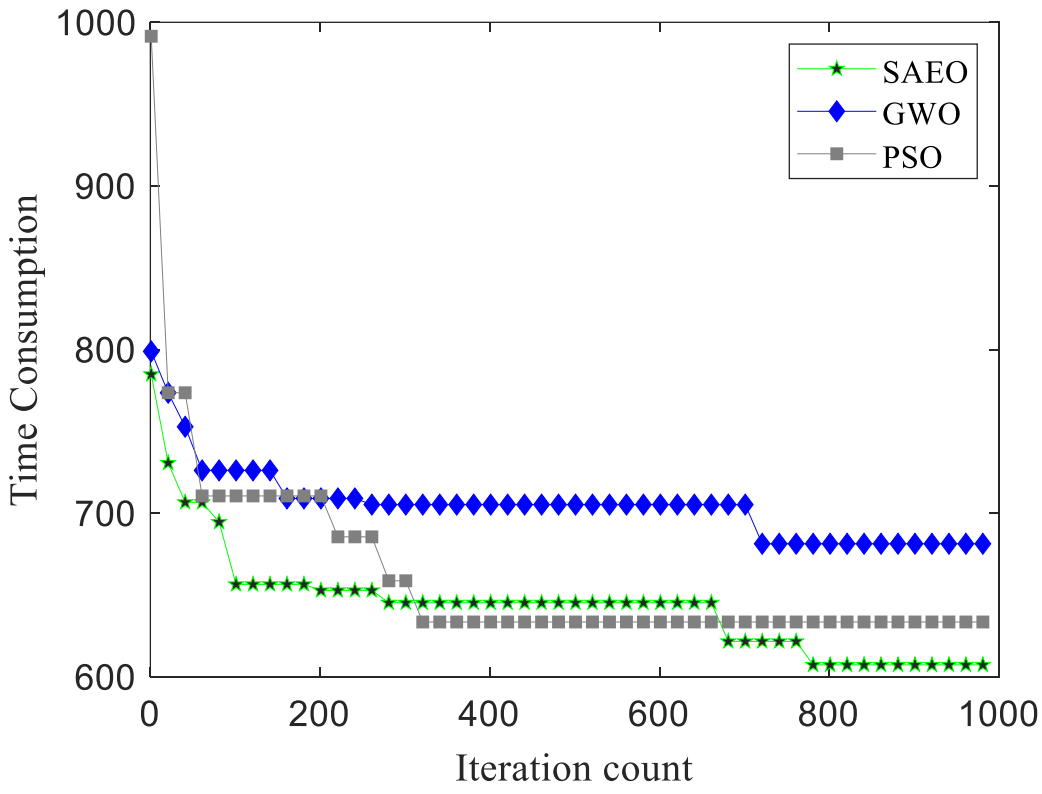

图 2 不同算法优化系统时延迭代曲线

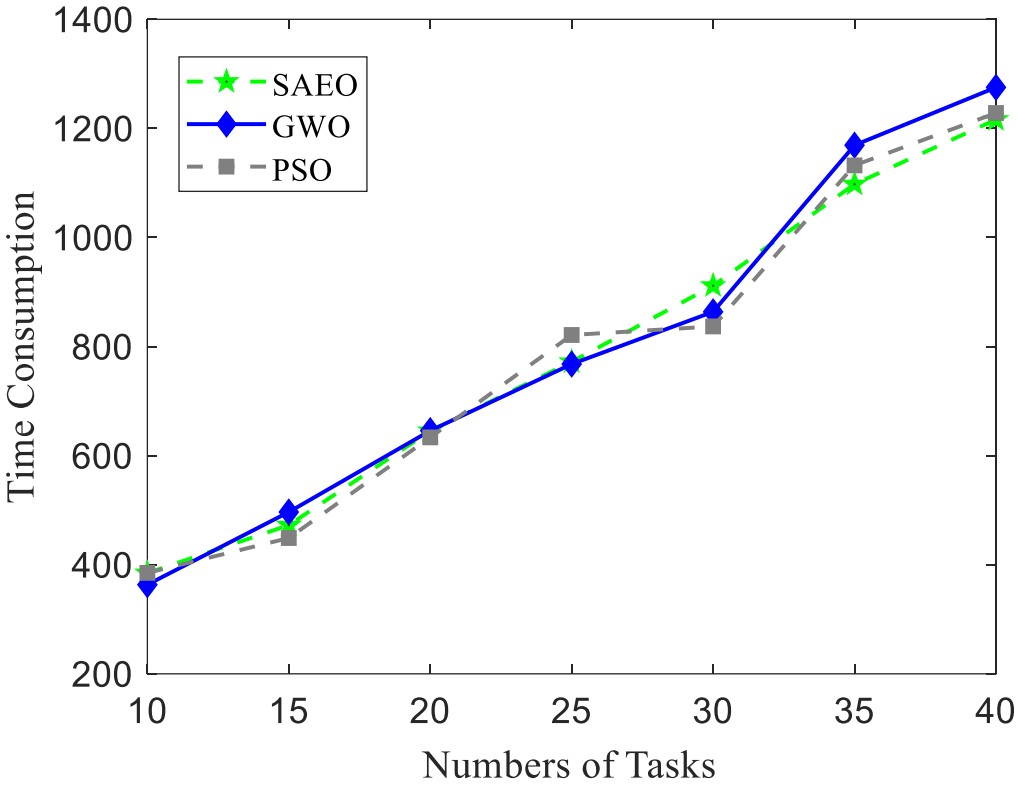

图 3 不同算法随不同任务数量的时延曲线

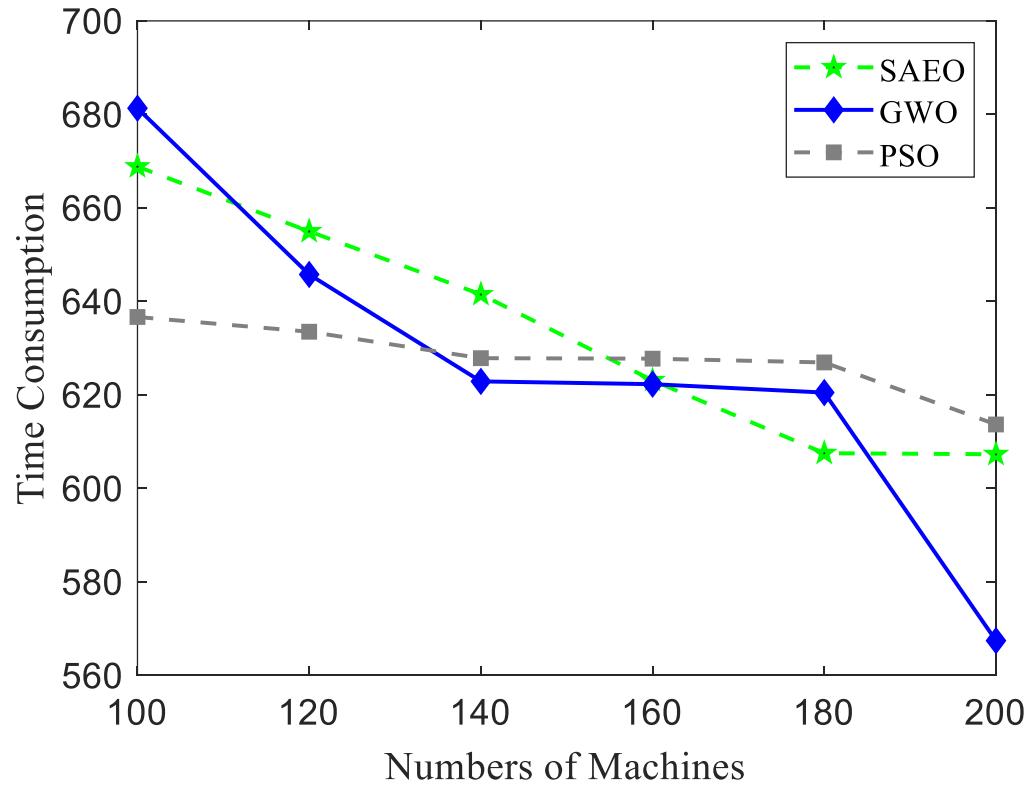

图 4 不同算法随不同机器数量的时延曲线

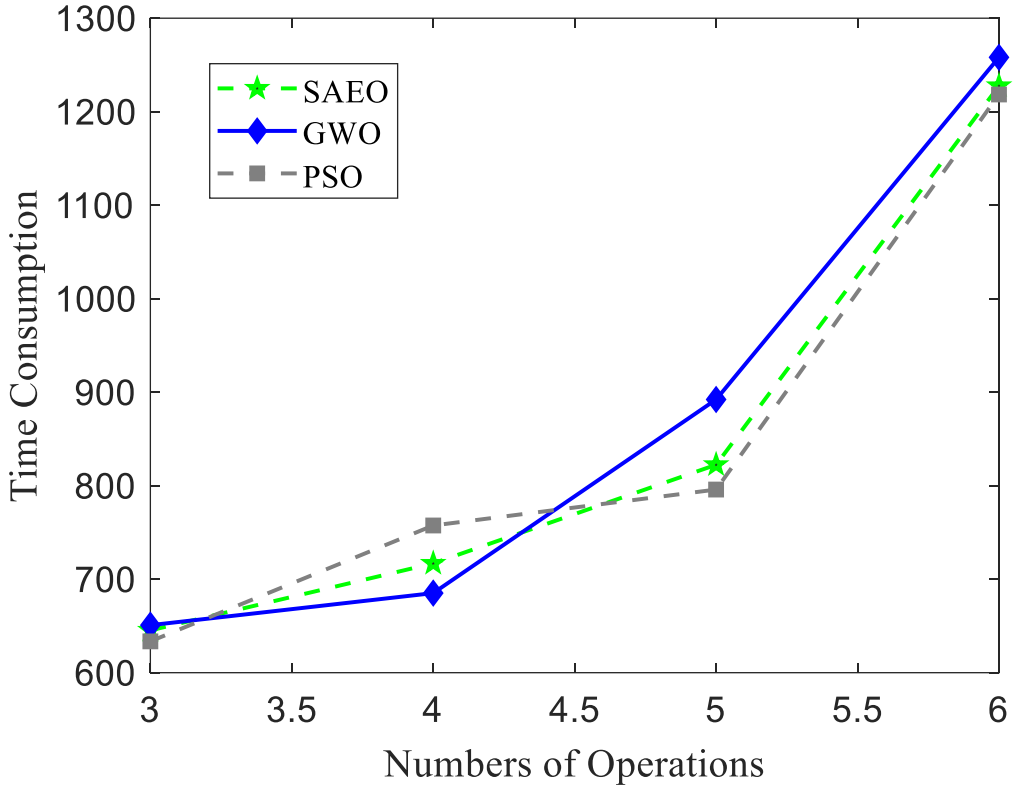

图 5 不同算法随不同操作数量的时延曲线

4、思路提示

(1) 可以结合不同优化算法的优点形成融合算法；

(2) 可以结合一些策略如莱维飞行，混沌映射等，**不要抄袭现有算法思路**；

(3) 可以结合代理模型以节省计算资源；

(4) 可以结合降维方法以更好地处理高维度问题。

5、具体要求

(1) 通过改进的智能优化算法来求解提出的优化问题，其中**至少要将提出的算法与三种不同的**启发式智能优化算法对比，如果优化了

该问题，增加了新的约束条件或者策略将获得额外分数；

(2) 优化算法可采用 Python 或 Matlab 进行编程实现；

(3) 6 月 10 日下午 1:30 分组进行实验结果的 Presentation 展示，以 PPT 的形式呈现各组的实验结果或效果，详细解释如何改进算法，优化结果等；

(4) 通过报告的方式整理大作业的内容：具体包括所提出的算法细节、最终实验结果、关于算法优缺点的评论等部分内容；

(5) 最终提交报告(**大标题三号宋体加粗；Word 正文小四号宋体、单倍行距；各小节标题用黑体四号字**)双栏排版 6-8 页，报告正文大标题下面添加每组组员姓名+学号；

(6) 最终提交分组报告+结果展示 PPT，连同报告对应的实验源码一起打包提交，压缩包命名的方式为“**组号-姓名+学号**”；

(7) 提交时间在 2025 年 6 月 20 日前，提交方式邮件：ziqi_wang@emails.bjut.edu.cn。