

4-19-2022

Design and Development of a Simulation System for Scheduling in Cloud Manufacturing Based on Microservice Architecture

Yongkui Liu

1.School of Mechano-Electronic Engineering, Xidian University, Xi'an 710071, China;; yongkuiliu@163.com

Ming Zeng

2.Industrial Cloud Manufacturing (Sichuan) Innovation Center Co. Ltd, Chengdu 610000, China;

Lin Zhang

3.School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China;

Jinwei Guo

1.School of Mechano-Electronic Engineering, Xidian University, Xi'an 710071, China;

See next page for additional authors

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Invited Papers & Special Columns is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Design and Development of a Simulation System for Scheduling in Cloud Manufacturing Based on Microservice Architecture

Abstract

Abstract: In view of the lack of the low coupling and highly extensible cloud manufacturing scheduling simulation system, that leads to inconvenience of the performance test on scheduling algorithms, *a scalable cloud manufacturing scheduling model with the scheduling goal of the minimization of cost and completion time and the maximization of quality of service is proposed. A micro-service architecture based cloud manufacturing scheduling simulation system is designed and developed, which realizes the functions of system management, resource management, task management and simulation management, and has the characteristics of flexible autonomy, stability and scalability.* The simulation system facilitates the performance simulation, test and optimization of the scheduling algorithms, and provides a simulation platform for the performance test and parameter optimization of cloud manufacturing scheduling algorithms.

Keywords

cloud manufacturing, scheduling, simulation system, microservice, design and development

Authors

Yongkui Liu, Ming Zeng, Lin Zhang, Jinwei Guo, Siyang Yuan, and Yaoyao Ping

Recommended Citation

Yongkui Liu, Ming Zeng, Lin Zhang, Jinwei Guo, Siyang Yuan, Yaoyao Ping. Design and Development of a Simulation System for Scheduling in Cloud Manufacturing Based on Microservice Architecture[J]. Journal of System Simulation, 2022, 34(4): 700-711.

基于微服务架构的云制造调度仿真系统设计与开发

刘永奎¹, 曾鸣², 张霖³, 郭金维¹, 原思阳¹, 平焱焱¹(1. 西安电子科技大学 机电工程学院, 陕西 西安 710071; 2. 工业云制造(四川)创新中心有限公司, 四川 成都 610000;
3. 北京航空航天大学 自动化科学与电气工程学院, 北京 100191)

摘要: 针对当前缺乏低耦合和高扩展的云制造调度仿真系统, 无法方便地对调度算法性能进行测试的问题, 以成本和完工时间最小化以及服务质量最大化为调度目标, 提出一个可扩展的云制造调度模型, 并在此基础上设计和开发了一个基于微服务架构的云制造调度仿真系统, 实现了系统管理、资源管理、任务管理和仿真管理等功能, 具有灵活自治、稳定、可扩展等特征。该仿真系统能够对调度算法的性能进行仿真、测试和优化, 从而为云制造调度算法性能测试和参数优化提供了一个仿真平台。

关键词: 云制造; 调度; 仿真系统; 微服务; 设计与开发

中图分类号: TP311 文献标志码: A 文章编号: 1004-731X(2022)04-0700-12

DOI: 10.16182/j.issn1004731x.joss.21-1017

Design and Development of a Simulation System for Scheduling in Cloud Manufacturing Based on Microservice Architecture

Liu Yongkui¹, Zeng Ming², Zhang Lin³, Guo Jinwei¹, Yuan Siyang¹, Ping Yaoyao¹

(1. School of Mechano-Electronic Engineering, Xidian University, Xi'an 710071, China; 2. Industrial Cloud Manufacturing (Sichuan) Innovation Center Co. Ltd, Chengdu 610000, China; 3. School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China)

Abstract: In view of the lack of the low coupling and highly extensible cloud manufacturing scheduling simulation system, that leads to inconvenience of the performance test on scheduling algorithms, a scalable cloud manufacturing scheduling model with the scheduling goal of the minimization of cost and completion time and the maximization of quality of service is proposed. A micro-service architecture based cloud manufacturing scheduling simulation system is designed and developed, which realizes the functions of system management, resource management, task management and simulation management, and has the characteristics of flexible autonomy, stability and scalability. The simulation system facilitates the performance simulation, test and optimization of the scheduling algorithms, and provides a simulation platform for the performance test and parameter optimization of cloud manufacturing scheduling algorithms.

Keywords: cloud manufacturing; scheduling; simulation system; microservice; design and development

引言

2009年, 李伯虎等^[1]提出了云制造概念, 并在后续工作中不断对该概念进行扩展和完善^[2-3]。云制

造的核心思想是通过构建云制造平台, 将分散在不同企业的制造资源集中起来, 形成逻辑上统一的虚拟制造资源, 然后通过中心化、智能化的管理向参与企业和终端消费者提供按需制造服务, 从而达到

收稿日期: 2021-09-30 修回日期: 2022-02-25

基金项目: 国家自然科学基金(61973243, 61873014)

第一作者: 刘永奎(1981-), 男, 博士, 副教授, 研究方向为云制造、制造系统智能调度与优化、工业机器人学习及智能、工业大数据分析

与数字孪生。E-mail: yongkui@163.com

社会制造资源广泛共享和优化配置的目的。云制造已被列入《中国制造2025》规划^[4]。

研究者从各个角度对云制造进行了研究, 包括服务模式^[5]、设备接入^[6]、资源存储^[7]、交易安全^[8], 以及研究趋势^[9]等。其中, 调度作为实现制造服务高效管理的一项关键技术, 也引起了相关领域研究者的广泛关注。模型方面, 研究者从服务和任务的角度出发考虑了工作量、企业效率、物流^[10-11]、企业间信任问题^[12]等因素; 算法方面, 除传统元启发式算法^[13]外, 深度强化学习^[14-15]等新兴人工智能算法也被用来解决云制造调度问题。对云制造调度过程进行仿真是当前研究的迫切需求。然而, 由于云制造系统的高度复杂性, 很少有人去研究并开发一个对云制造调度算法和过程进行仿真验证的云制造调度系统。构建云制造调度仿真系统是一个可行的途径。然而, 目前研究者研究重点^[16-18]不同, 构建的调度模型各异, 提出的调度算法也采用不同的技术和语言。此外, 现有仿真系统^[19-20]几乎都采用单体式架构, 且部署在本地, 扩展性较差。在拓展性方面, 单体式架构和分布式架构并无明显的优劣之分, 只是方式不同。单体式需要拓展时, 选择从外部提供的资源和能力, 注入到体系的内部, 不断的从内护自己; 分布式则采用以自己的核心能力为内核, 持续吸引外部合作者, 形成一个不断成长的生态体系。但是在可靠性、维护成本和吞吐量等方面, 分布式具有明显的优势。

微服务将一个系统拆分成多个互相独立的服务, 每个服务相互独立, 可以分别测试、部署, 便于云制造调度模型的开发。由于一个微服务只负责一类事务, 便于云制造调度模型的技术栈升级和后期维护。且当一个微服务发生问题时, 如内存泄漏, 不会影响到其他的微服务, 能提高云制造调度模型的容错性。

综上, 本文基于微服务架构设计并开发一个云制造调度仿真系统, 旨在为云制造调度研究提供一个仿真环境和验证平台。

1 云制造调度模型

本节的主要任务是构建一个可扩展的云制造调度模型。云制造调度问题建模是云制造调度仿真系统设计和开发的前提和基础。模型的内容直接关系到系统的功能模块划分与数据库表的设计, 模型的好坏直接影响到系统功能的合理与否和适用与否。

1.1 企业、服务与资源

假设云制造平台中有 $I(I \geq 1)$ 家制造企业, 记为 $Ent = \{E_i | 1 \leq i \leq I\}$, 共提供 $J(J \geq 1)$ 种服务, 记为 $Type = \{y_j | 1 \leq j \leq J\}$ 。每家企业 E_i 提供 $K_i(1 \leq K_i \leq J)$ 种不同类型的服务, 记 $S_{i,k}$ 为企业 E_i 的第 $k(1 \leq k \leq K_i)$ 种服务。假设服务 $S_{i,k}$ 具有服务类型、服务数量、服务单价、服务质量、服务效率、服务可靠性等属性, 并分别记为 $y_{i,k}$ 、 $a_{i,k}$ 、 $c_{i,k}$ 、 $q_{i,k}$ 、 $eff_{i,k}$ 、 $rel_{i,k}$ 。

1.2 任务与子任务

云制造平台不断收到用户提交的任務。假设云平台在某一时刻收到 M 个任务, 记为 $Task = \{T_m | 1 \leq m \leq M\}$, 其中 T_m 表示第 m 个任务。假设每个任务都有 N 个子任务, 记 $T_m = \{sT_{m,n} | 1 \leq n \leq N\}$, 其中 $sT_{m,n}$ 表示任务 T_m 的第 n 个子任务。假设 $sT_{m,n}$ 具有任务类型、任务工作量属性并分别记为 $ry_{m,n}$ 、 $wl_{m,n}$ 。

任务层面, 主要考虑最大成本以及最大加工时间等总体约束; 子任务层面, 主要考虑最低服务质量和最低可靠性等约束属性。不同任务的多个子任务具有不同的执行顺序, 比如顺序结构、选择结构、并行结构、循环结构或者以上的混合等。

1.3 物流

假设云制造平台的物流企业分布在 $O(O \geq 1)$ 个不同的地方, 记为 $A = \{A_o | 1 \leq o \leq O\}$ 。假设任意两地之间都可直接或间接连通, 且 $A_{o_1} \sim A_{o_2}$ 与 $A_{o_2} \sim A_{o_1}$ 距离相同, A_{o_1} 和 A_{o_2} 两地之间的距离记为

$d(A_{o1} \leftrightarrow A_{o2})$ 。特别的, $d(A_{o1} \leftrightarrow A_{o1})=0$, 即忽略企业内物流, 且假设同一地点的企业之间的距离为 0。

其中, 物流单价和物流速度是主要考虑的 2 个基本属性。物流单价为任务被运送单位距离的价格, 记为 $lup(lup > 0)$, 比如 $A_{o1} \sim A_{o2}$ 的物流成本为 $lc_{o1,o2}$, 则 $lc_{o1,o2} = d(A_{o1} \leftrightarrow A_{o2})lup$ 。物流速度主要与运输货物的方式有关, 典型的运输方式有陆运、空运、水运, 本模型主要考虑陆运的方式, 运输工具以货车为主, 采用运输过程中货车的平均速度并记为 $ls(ls > 0)$, 记 $A_{o1} \sim A_{o2}$ 的物流时间为 $lt_{o1,o2}$, 则 $lt_{o1,o2} = d(A_{o1} \leftrightarrow A_{o2})/ls$ 。物流成本和物流时间是主要考虑的 2 个方面。物流发生在子任务之间, 记从子任务 $sT_{m,n}$ 到子任务 $sT_{m,n+1}$ 的物流成本为 $lc_{o1,o2} = lc_m^{n,n+1}$, 记从子任务 $sT_{m,n}$ 到子任务 $sT_{m,n+1}$ 的物流时间为 $lt_{o1,o2} = lt_m^{n,n+1}$ 。

1.4 目标函数

目标分析是在 $y_{i,k} = ry_{m,n}$ 也就是 $sT_{m,n}$ 能被 $S_{i,k}$ 选择去执行的情况下进行的, 考虑多目标调度, 即最小化成本、最小化完成时间和最大化服务质量。记总成本为 $C = \sum_{m=1}^M c_m$, $c_m = sc_m + lc_m$, $sc_m = \sum_{n=1}^N (c_{i,k} \cdot wl_{m,n})$ 和 $lc_m = \sum_{n=1}^{N-1} lc_m^{n,n+1}$ 分别表示任务 T_m 的制造成本和物流成本。完成时间为 $CT = \sum_{m=1}^M ct_m$, $ct_m = st_m + lt_m + wt_m$, $st_m = \sum_{n=1}^N wl_{m,n}/eff_{i,k}$ 为任务 T_m 的执行时间; $lt_m = \sum_{n=1}^{N-1} lt_m^{n,n+1}$ 为其物流时间; $wt_m = \sum_{n=1}^{N-1} wt_m^{n,n+1}$ 为从子任务 $sT_{m,n}$ 到子任务 $sT_{m,n+1}$ 的等待时间。记服务质量(用产品合格率表示)为 $Q = \sum_{m=1}^M \sum_{n=1}^N q_{m,n}/M$, 其中 $q_{m,n}$ 为 $sT_{m,n}$ 所选择的服务质量。则调度目标为

$$\min z = \omega_c \cdot C + \omega_{CT} \cdot CT + \omega_Q \cdot (1 - Q) \quad (1)$$

式中: z 是关于 C 、 CT 、 Q 的一个多目标函数;

ω_c 、 ω_{CT} 和 ω_Q 分别为总成本、总完成时间和总服务质量(不合格率)的权重, 并且满足 $\omega_c + \omega_{CT} + \omega_Q = 1$ 。

1.5 约束条件

约束条件主要分为 2 种: ①服务使用者对任务执行提出的人为约束; ②为实际生产需要满足的固有约束。人为约束主要包括成本、时间、服务质量、可靠性等; 固有约束主要为匹配约束。

(1) 成本约束

任务 T_m 的成本不得超出服务使用者给定的最大成本, 即 $c_m = sc_m + lc_m \leq c_m^{\max}$ 。

(2) 时间约束

任务 T_m 的开始加工时间不得早于服务使用者要求的开始加工时间 t_m^{start} , 记任务 T_m 实际开始加工时间为 t_m^{start} , 则 $t_m^{\text{start}} \geq t_m^{\text{start}}$ 。

同理, 任务 T_m 的加工完成时间不得大于服务使用者要求的结束时间 t_m^{end} , 记任务 T_m 实际加工完成时间为 t_m^{end} , 即 $t_m^{\text{end}} \leq t_m^{\text{end}}$ 。

(3) 服务质量约束

子任务 $sT_{m,n}$ 匹配到服务 $S_{i,k}$ 的服务质量不得低于服务使用者要求的最低服务质量, 即 $q_{i,k} \geq q_{m,n}^{\min}$ 。

(4) 可靠性约束

子任务 $sT_{m,n}$ 匹配到服务 $S_{i,k}$ 的服务可靠性不得低于服务使用者要求的最小服务可靠性, 即 $rel_{i,k} \geq sreI_{m,n}^{\min}$ 。

(5) 匹配约束

子任务 $sT_{m,n}$ 只能与一个服务 $S_{i,k}$ 匹配, 并且该子任务的全部加工过程均由与之匹配的服务完成。为了描述和计算方便, 引入时间基本单位 $p(p \geq 1)$, 表示任务加工过程中不可再分割的基本时间段, 并采用布尔变量表示匹配结果, 即

$$\sum_{i=1}^M \sum_{k=1}^K \sum_{p=1}^P f(i, k, m, n, p) = 1 \quad (2)$$

1.6 云制造调度

在以上研究和定义的基础上, 基于特定的调

度算法,开展云制造调度,考虑服务状态随时间动态变化,面向一个或多个需求任务,实现云平台中海量的服务和用户需求任务的最优匹配,从而实现特定的调度目标(如实现任务总体完成情况)^[13]。

2 系统需求分析

2.1 系统概述

2.1.1 实现目标

目前,云制造调度仿真系统开发主要存在以下问题:

(1) 算法中制造设备等资源服务信息以及提交的任务信息往往以矩阵的形式表示,不熟悉算法模型的人对这些矩阵难以理解,即使明白矩阵的含义,在算法中针对某一资源或任务做属性修改也很不方便;

(2) 算法的结果也是以矩阵来表示,但由于结果不够直观,往往需要研究者将结果绘制成甘特图。尤其是数据量巨大时,结果的处理就会变得非常困难;

(3) 一般采用单体式架构。在模型做较大改变或者在需要添加某些功能时,需要对系统整体做较大修改;

(4) 传统研究者通常使用C/C++或者Java编写算法。随着技术的发展,Python成为深度强化学习框架的主要语言。然而,不同编程语言的算法需要使用相应语言开发的系统与之配合,给算法研究者带来极大的不便。

针对以上问题,本系统拟实现以下目标:

(1) 实现对资源信息、任务信息和物流信息的有效管理,在页面中可以详细展示所有属性信息,且可对目标信息进行增加、删除和修改等操作;

(2) 能直接对调度算法生成的结果进行处理,并以甘特图的形式在页面中显示;

(3) 采用微服务架构,保证系统的低耦合性和高扩展性;

(4) 对开发语言友好,能调用由不同语言开发的算法,获取算法结果并将其转换为甘特图。

2.1.2 角色分析

云制造调度仿真系统主要包括4种角色:资源提供者、服务使用者、平台运营者,以及超级管理员。在系统中的权限如下:

资源提供者:能够管理和访问所提供的资源,并能够查看调度结果;

服务使用者:能够管理和访问自身的需求任务,并能够查看调度结果;

平台运营者:具有系统管理的部分权限,包括用户管理、资源管理、任务管理,以及仿真管理部分访问和管理权限;

超级管理员:具有系统的全部权限。

2.2 功能性需求分析

功能性需求分析是软件需求分析的主体。云制造调度仿真系统主要实现API网关、系统管理、资源管理、任务管理,以及仿真管理等功能。

2.2.1 资源管理

资源管理模块所管理的对象包括三大类:①企业管理;②企业内的资源和服务(如机械臂、机床等制造设备)管理;③物流信息管理。企业管理主要包括对企业名称、企业地址、企业可靠性等属性的管理和企业的增加、删除、修改与查询等功能的实现;资源/服务管理则主要包括对名称、类型、效率、单价等属性的管理和资源/服务的增加、删除、修改与查询等功能的实现;物流管理主要包括对企业地址、企业间距离和物流成本和物流速度等物流信息的管理。

2.2.2 任务管理

任务管理模块主要对资源需求者发布的任务以及任务分解后的子任务进行管理。任务管理主要包括对任务描述、任务数量、任务QoS要求等属性的管理和任务的增加、删除、修改与查询等功能的实现;而子任务管理主要包括对子任务描

述、子任务数量、子任务要求的服务类型等属性的管理以及任务的增加、删除、修改与查询等功能的实现。

2.2.3 仿真管理

仿真管理模块主要基于调度算法一方面要对调度算法所需参数进行修改、查询，另一方面则要实现由不同语言编写的调度算法的调用，获取相应的调度结果，并将调度结果转换成甘特图展示。

3 系统设计与实现

云制造调度仿真系统的逻辑架构如图 1 所示。系统采用 B/S(浏览器/服务器)开发模式，分为用户接口层、业务层、数据访问层和数据存储层。

用户接口层提供系统的视图界面，实现用户与

系统的交互。用户可以通过浏览器对系统进行访问。

业务层是实现系统业务应用的主体。通过用户接口层发送的请求会首先发往网关微服务，然后将请求路由转发到相应的微服务。各个微服务负责处理具体的业务、实现不同的功能，其中，系统管理微服务除用户管理、登录等必要功能外，还要有系统安全设计。

数据访问层实现数据的读写和持久化。数据访问层是连接业务层与数据库的媒介，经业务层处理过的数据只有通过数据访问层才能间接对数据库中的数据进行各种 SQL 操作。

数据存储层实现系统数据的存储。由于本系统采取微服务架构，所以除网关微服务外的所有微服务都拥有各自的数据库，以独立存储每个微服务的数据。

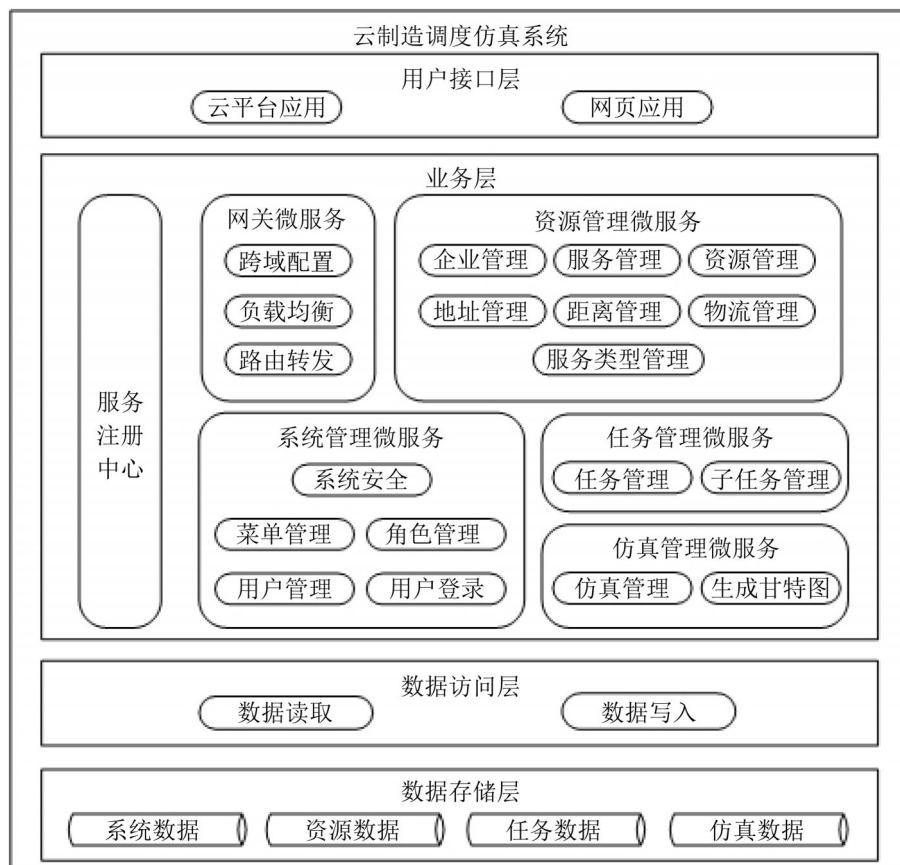


图 1 云制造调度仿真系统架构

Fig. 1 System architecture of cloud manufacturing scheduling simulation system

3.1 资源管理微服务

资源管理微服务负责的功能包括企业管理、服务管理、资源管理、服务类型管理、地址管理、距离管理,以及物流参数管理等。

企业、服务和资源三者存在从属关系:服务属于企业,资源属于服务。因此,在进行删除操作时需要考虑待删除项是否存在所属服务或资源。以删除服务为例说明三者的从属关系,如图2所示。首先获取待删除服务的ID,然后通过getSerStatus()方法获取服务状态以判断服务是否被占用,如被占用,则删除失败;如未被占用,则通过getResource()方法查询该服务是否还有所属资源。如存在所属资源则继续判断资源是否可以删除,如资源同样无法删除则删除失败,资源可删除则执行删除操作后同服务不存在所属资源的情况一起执行服务删除操作,至此服务删除流程执行完毕。

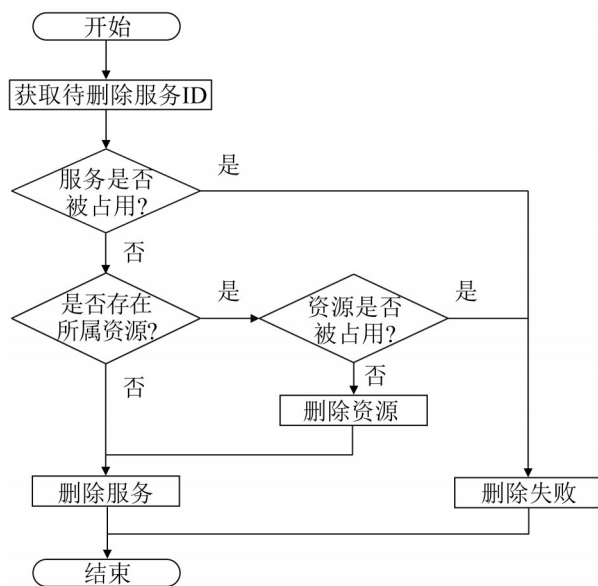


图2 删除服务流程图

Fig. 2 Flowchart of deleting services

地址管理、距离管理和物流参数管理都是为物流管理服务的。物流是在子任务由不同企业的服务承担的情况下产生的,企业有地址属性,而服务是属于企业的,承担相邻2个子任务的服务

所在企业的地址之间的距离即物流距离。在此基础上,结合物流参数管理中的物流速度和物流成本等参数,即可对物流时间、物流成本等进行计算。因此,地址管理、距离管理和物流参数管理三者具有逻辑上的不可分割性。如图3所示为企业管理界面的企业列表。

企业id	企业名称	企业地址	上架时间	操作
1	企业一	甲	2021-01-01 15:15:00	上架 删除
2	企业二	乙	2021-01-02 15:15:00	上架 删除
3	企业三	丙	2021-01-03 15:15:00	上架 删除
4	企业四	丁	2021-01-04 15:15:00	上架 删除
5	企业五	戊	2021-01-05 15:15:00	上架 删除
6	企业六	己	2021-01-06 15:15:00	上架 删除
7	企业七	庚	2021-01-07 15:15:00	上架 删除
8	企业八	辛	2021-01-08 15:15:00	上架 删除
9	企业九	壬	2021-01-09 15:15:00	上架 删除
10	企业十	癸	2021-01-10 15:15:00	上架 删除

图3 企业列表

Fig. 3 List of enterprises

3.2 任务管理微服务

任务管理微服务负责管理发布在云制造平台的任务,包括任务管理和子任务管理2个模块。

依据第1节提出的模型,子任务是由任务分解而来的,且子任务才是与服务匹配的主体,因此,任务的添加必定伴随着子任务的添加。任务添加流程如图4所示。首先,在前端输入任务的属性信息,并添加子任务。添加子任务时要判断子任务所需类型云制造平台是否提供,如提供,则须请求云制造平台添加所需服务类型;如不提供,则可以选择所需类型并输入子任务属性信息,完成该子任务的添加。然后,循环判断子任务是否添加完毕直至完成全部子任务的添加。最后,提交任务即可完成任务的添加操作。发布任务的系统界面如图5所示。

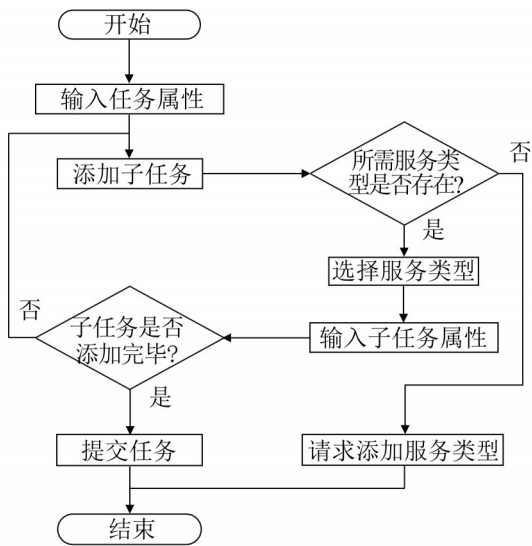


图 4 任务添加流程图
Fig. 4 Flowchart of adding tasks

发布 ×

- * 任务编号
- * 任务描述
- * 任务数量
- * 开始时间
- * 结束时间
- * 最大成本
- * 提交时间

图 5 任务发布
Fig. 5 Task submission

3.3 仿真管理微服务

仿真管理微服务负责管理算法仿真参数，并实现对算法的仿真以及算法仿真结果的图形化展示，主要包括参数管理和算法仿真 2 个模块。

算法调用是仿真管理微服务的核心功能，图 6 即为算法调用时序图。首先，前端发起算法调用的 GET(解释)请求。经 API 网关微服务路由到仿真管理微服务后，后端算法调用类 CallAndPut 通过

results()方法向下逐层调用，并从数据库中获取参数信息。然后，类 CallAndPut 通过 getArray()方法采用 JNI(Java 本地接口)的方式调用由 C++编写的算法，向算法传递参数、获取算法结果矩阵并对算法结果进行适当处理。接着，类 CallAndPut 响应前端请求并将算法结果以 JSON(解释)格式发往前端。最后，前端接收算法结果并将其解构为矩阵，利用 canvas.js 对算法结果进行处理并生成甘特图。甘特图的生成情况以及调度算法的测试在最后的系统测试部分里已作出详细的说明。

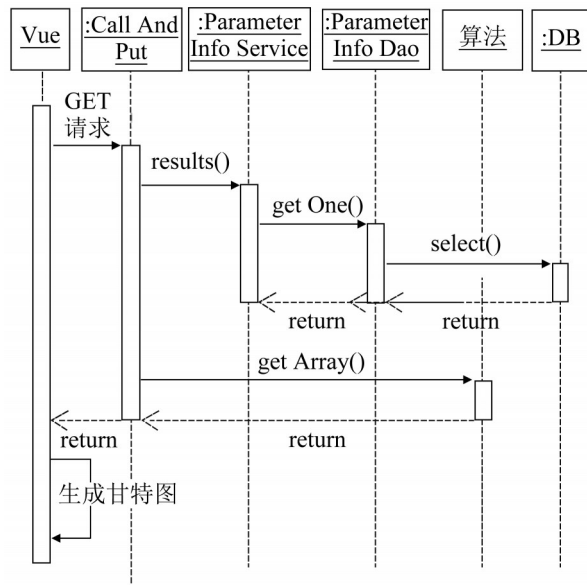


图 6 算法调用时序图
Fig. 6 Diagram of algorithm invocation sequence

不同的调度算法通过 C++代码实现，并编译生成动态链接库文件(.dll)，在本系统中使用 Java 调用该文件，因不同的调度算法生成的动态链接库文件不同，所以本系统以此适应不同的调度算法。同时算法的开发有以下要求：输入参数应包含任务需要使用的资源、任务使用资源的顺序、任务使用每个资源的时间；输出结果包含各个任务使用某资源的时长、开始时间和结束时间，导出信息的格式为[任务序号，任务使用资源 1 的开始时间，任务使用资源 1 的结束时间，任务使用资源 2 的开始时间，任务使用资源 2 的结束时间，……]。

4 系统部署与测试

4.1 系统部署

本文开发的云制造调度仿真系统在完成开发后要部署在航天云网 INDICS(industrial intelligent cloud system)平台, 部署环境如表1所示。

表1 部署环境
Table 1 Deployment environment

序号	名称	技术
1	部署平台	INDICS 平台
2	平台技术	Kubernetes+Docker
3	操作系统	Linux
4	数据库	云 MySQL 5.7
5	注册中心	Eureka

该平台是基于 Kubernetes+Docker 容器技术搭建的, 能够实现应用的可视化部署。此外, 该平台分别创建镜像并上传到 INDICS 平台, 并对系统进行数据库、文件存储以及注册中心等相关配置, 系统即可成功运行。前端运行状态如图7所示, 后端全部微服务在注册中心的状态如图8所示。

4.2 系统测试

在第3节中已通过系统截图证明了系统功能均可实现, 本节主要对有关的调度算法测试, 以及从性能、安全性等方面对系统进行非功能测试。

4.2.1 调度算法测试

在调度算法测试中, 模拟实验的默认参数如表2所示, 并针对不同的任务数量, 即分别对任务数量为10、30、50进行测试, 并生成相应的甘特图, 如图9所示。

图7 前端运行状态

Fig. 7 Front end running state

服务名称	状态	服务地址	创建时间	资源配置	所属应用	所属应用组	注册中心名称	操作
CPSSMGATEWAY	开启	172.20.139.178:88	2020-10-29 16:33:45	1核/2G	cpssm-gateway	cpssm	default	在线测试 应用详情
CPSSMSIMULATION	开启	172.20.34.68:9000	2021-03-12 10:05:50	1核/2G	cpssm-simulation	cpssm	default	在线测试 应用详情
CPSSMADMIN	开启	172.20.6.71:8080	2020-10-29 16:32:41	1核/2G	cpssm-admin	cpssm	default	在线测试 应用详情
CPSSMRESOURCE	开启	172.20.139.177:7000	2020-10-27 12:08:32	1核/2G	cpssm-resource	cpssm	default	在线测试 应用详情
CPSSMTASK	开启	172.20.50.160:8000	2020-10-28 20:45:17	1核/2G	cpssm-task	cpssm	default	在线测试 应用详情

图8 注册中心微服务列表

Fig. 8 List of registry microservices

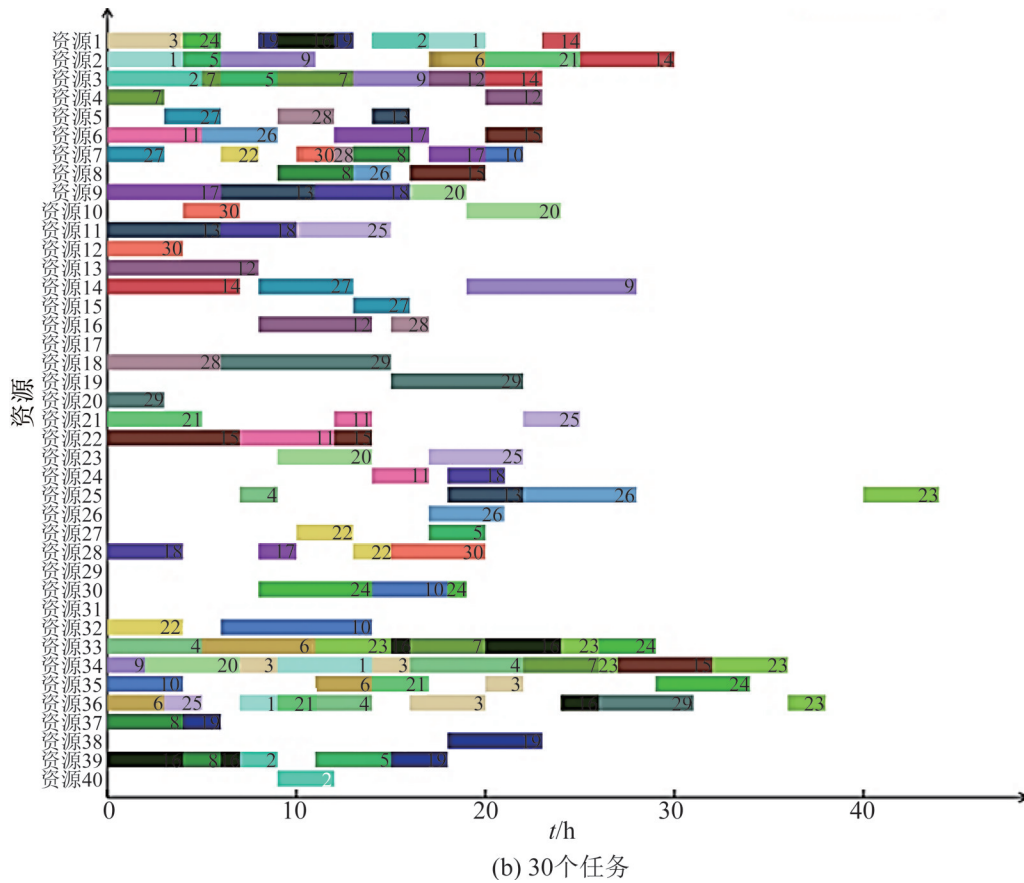
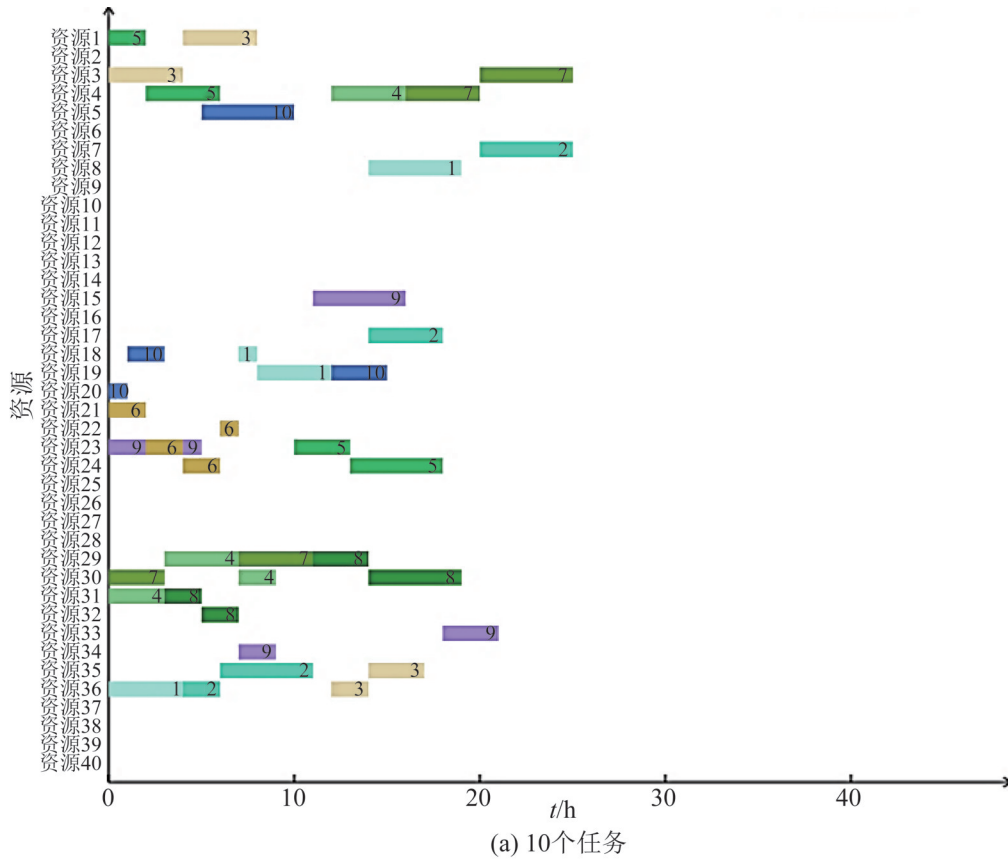
表2 模拟实验的默认参数

Table 2 Default parameters of simulation experiments

变量	值	变量	值	变量	值	变量	值
I	10	$eff_{i,k}$	[0.5,1]	$d_{o1,o2}/km$	[0,100]	w_{CT}	0.3
J	4	M	10/30/50	$lup/(km/h)$	[0,100]	w_Q	0.3
$c_{i,k}(\text{元})$	[0,100]	N	4	$ls(\text{元})$	80	$c_m^{\max}(\text{元})$	5000
$q_{i,k}$	[0,1]	$wl_{m,n}$	[1,10]	w_C	0.4	$t_m^{\text{end}} - t_m^{\text{star}}/\text{月}$	2
						$q_{m,n}^{\min}$	0.8
						$streJ_{m,n}^{\min}$	0.6

<http://www.china-simulation.com>

• 707 •



<http://www.china-simulation.com>

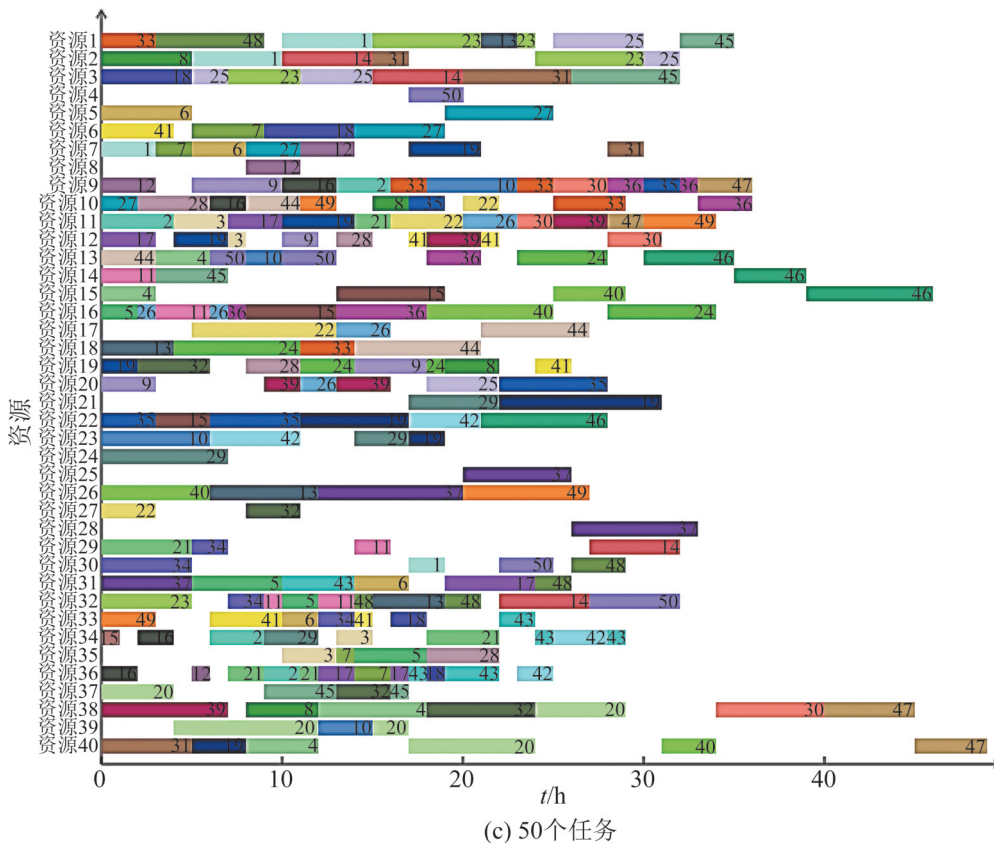


图9 甘特图
Fig. 9 Gantt chart

图9通过采取不同数量的任务对调度算法进行测试, 由生成的调度图可以看出, 任务数量越多, 调度的时间会变长, 模拟实验的结果如表3所示。

表3 模拟实验的结果
Table 3 Results of simulation experiments

任务数量	总时间/h	总成本(元)	总质量(合格率)
10	25	1 230	0.85
30	43	2 965	0.93
50	52	4 968	0.92

4.2.2 安全性测试

本文从系统安全和数据安全两方面对系统进行安全性测试。

系统安全测试用例如表4所示, 针对表中各测试用例, 执行相应的测试操作, 均可达到预期结果, 系统安全测试通过。

表4 系统安全测试用例表
Table 4 System security test case

用例描述	测试操作	预期结果
用户未登录是否可以访问系统微服务	在未登录的情况下, 在浏览器中输入微服务地址并访问	访问失败, 跳转到登录界面
用户是否可以访问无权限的微服务	用户已登录, 在浏览器中输入该用户无访问权限的微服务地址并访问	访问失败
同一用户是否可以同时在两个设备登录	用户已在 Chrome 中登录, 在 Microsoft Edge 中登录该用户	用户在 Microsoft Edge 中登录成功, Chrome 中该用户退出登录
用户登录时间超过 token 有效期后是否需要重新登录	将 token 设为 1 min, 在 1 min 后对系统进行任意操作	操作失败, 跳转到登录界面

数据安全在系统实现的表单填写中已进行了验证,如图5所示,在表单中输入非法字符,表单无法提交并给出错误提示。

综上所述,系统安全和数据安全均通过测试,本系统具有良好的安全性。

4.2.3 稳定性测试

稳定性保证系统在一定时期内能保持一定的技术优势且在一定时期内能稳定运行。本文采用基于Java开发的压力测试工具Apache JMeter对系统进行性能测试。通过Apache JMeter模拟10 s内1 000个用户对系统主页进行访问的场景,以获取请求的成功、失败数,响应时间等数据以对系统性能进行判断。

对系统响应时间的评价可以依据如表5所示。如图10所示为请求的响应时间汇总报告表,由图10可知,请求的平均响应时间为33.15 ms,最大响应时间为515 ms,其中,99%的请求响应时间都在300 ms以内。

由上述测试数据可知,本系统响应速度很快,满足要求。

表5 “2-5-8”原则
Table 5 "2-5-8" principle

响应时间/s	评价
$t \leq 2$	响应很快
$2 < t \leq 5$	响应较快
$5 < t \leq 8$	响应较慢
$t > 8$	响应很慢

Requests	Executions			Response Times / (ms)							Throughput	Network (KB/sec)	
	Label ^	#Samples ^	FAIL ^	Error % ^	Average ^	Min ^	Max ^	Median ^	90th pct ^	95th pct ^		99th pct ^	Transactions/s ^
Total	1000	0	0.00%	33.15	24	515	26.00	49.00	53.00	264.98	98.51	60.99	13.37
HTTP请求	1000	0	0.00%	33.15	24	515	26.00	49.00	53.00	264.98	98.51	60.99	13.37

图10 请求响应时间
Fig. 10 Request response time

5 结论

本文提出一个可扩展的云制造调度模型,并基于微服务架构完成了仿真系统的设计与开发工作。模型方面,本文提出的模型较为全面地考虑了云制造调度的要点,并对其中一些因素和指标进行适当地增强和减弱,使得该模型更突出调度这个要点的同时,还能较好地适应不同的调度算法;系统方面,与传统单体式架构系统相比,本文开发的系统扩展性更好,稳定性更高,灵活自治且能面向更多场景。

本文开发的云制造调度仿真系统在许多方面还可以进一步完善,如还需要结合实际应用构建更加符合实际的云制造调度模型,并进行系统设计和开发。在算法方面,还需要结合新兴人工智能技术,如深度强化学习、多智能体深度强化学习和迁移学习等进一步开发相应的调度算法,并

进行算法的性能测试和评估。此外,还需要针对云制造调度的支持功能进行进一步开发,如服务搜索与匹配、企业交易协商等。

参考文献:

- [1] 李伯虎,张霖,王时龙,等. 云制造——面向服务的网络化制造新模式[J]. 计算机集成制造系统, 2010, 16(1): 1-8.
Li Bohu, Zhang Lin, Wang Shilong, et al. Cloud Manufacturing: a New Service-Oriented Networked Manufacturing Model[J]. Computer Integrated Manufacturing Systems, 2010, 16(1): 1-8.
- [2] 李伯虎,柴旭东,张霖. 智慧云制造——一种互联网与制造业深度融合的新模式、新手段和新业态[J]. 中兴通讯技术, 2016, 22(5): 2-6.
Li Bohu, Chai Xudong, Zhang Lin. Smart Cloud Manufacturing: a New Kind of Manufacturing Paradigm, Approach and Ecosystem of Deep Integration of the Internet and the Manufacturing Industry[J]. ZTE Technology Journal, 2016, 22(5): 2-6.
- [3] 李伯虎,柴旭东,侯宝存,等. 云制造系统3.0——一种“智

- 能+"时代的新智能制造系统[J]. 计算机集成制造系统, 2019, 25(12): 2997-3012.
- Li Bohu, Chai Xudong, Hou Baocun, et al. Cloud Manufacturing System 3.0—New Intelligent Manufacturing System in Era of "Intelligence+"[J]. Computer Integrated Manufacturing Systems, 2019, 25(12): 2997-3012.
- [4] 国务院. 国务院关于印发《中国制造2025》的通知[EB/OL]. [2021-06-30]. http://www.gov.cn/zhengce/content/2015/05/19/content_9784.htm.
- The State Council. Notice of the State Council on Printing and Distributing "Made in China 2025"[EB/OL]. [2021-06-30]. http://www.gov.cn/zhengce/content/2015/05/19/content_9784.htm.
- [5] 刘永奎, 王力羣, 王曦, 等. 云制造再探讨[J]. 中国机械工程, 2018, 29(18): 2226-2237.
- Liu Yongkui, Wang Lihui, Wang Xi, et al. A Revisit to Cloud Manufacturing[J]. China Mechanical Engineering, 2018, 29(18): 2226-2237.
- [6] 张映锋, 张耿, 杨腾, 等. 云制造加工设备服务化封装与云端接入方法[J]. 计算机集成制造系统, 2014, 20(8): 2029-2037.
- Zhang Yingfeng, Zhang Geng, Yang Teng, et al. Service Encapsulation and Virtualization Access Method for Cloud Manufacturing Machine[J]. Computer Integrated Manufacturing Systems, 2014, 20(8): 2029-2037.
- [7] 贵州航天云网科技有限公司. 基于边缘计算的云制造数据高效处理系统: CN2020106735-66.6[P]. 2020-12-01. Guizhou CASICloud Technology Co., Ltd. Cloud Manufacturing Data Efficient Processing System Based on Edge Computing: CN2020106735-66.6[P]. 2020-12-01.
- [8] Zhi Li, Ali Vatankhah Barenji, George Q Huang. Toward a Blockchain Cloud Manufacturing System as a Peer to Peer Distributed Network Platform[J]. Robotics and Computer-Integrated Manufacturing(S0736-5845), 2018, 54: 133-144.
- [9] Göran Adamson, Lihui Wang, Magnus Holm, et al. Cloud Manufacturing: a Critical Review of Recent Development and Future Trends[J]. International Journal of Computer Integrated Manufacturing(S0951-192X), 2017, 30(4/5): 347-380.
- [10] Liu Yongkui, Xu Xun, Zhang Lin, et al. An Extensible Model for Multitask-Oriented Service Composition and Scheduling in Cloud Manufacturing[J]. Journal of Computing and Information Science in Engineering (S1530-9827), 2016, 16(4): 1-11.
- [11] Liu Yongkui, Xu Xun, Zhang Lin, et al. Workload-Based Multi-Task Scheduling in Cloud Manufacturing[J]. Robotics and Computer-Integrated Manufacturing(S0736-5845), 2017, 45: 3-20.
- [12] 曹文颖, 贾国柱, 孔继利, 等. 基于企业间信任的云制造资源调度[J]. 工业工程与管理, 2020, 25(4): 32-40.
- Cao Wenyong, Jia Guozhu, Kong Jili, et al. Cloud Manufacturing Resource Scheduling Based on Trust between Enterprises[J]. Industrial Engineering and Management, 2020, 25(4): 32-40.
- [13] Liu Yongkui, Wang Lihui, Wang Xi Vincent, et al. Scheduling in Cloud Manufacturing: State-of-the-Art and Research Challenges[J]. International Journal of Production Research(S0020-7543), 2019, 57(15/16): 4854-4879.
- [14] Liang Huagang, Wen Xiaoqian, Liu Yongkui, et al. Logistics-Involved QoS-Aware Service Composition in Cloud Manufacturing with Deep Reinforcement Learning [J]. Robotics and Computer Integrated Manufacturing (S0736-5845), 2021, 67: 101991.
- [15] Chen Shengkai, Fang Shuilang, Tang Renzhong. A Reinforcement Learning Based Approach for Multi-Projects Scheduling in Cloud Manufacturing[J]. International Journal of Production Research(S0020-7543), 2019, 57(10): 3080-3098.
- [16] 赵淳, 张霖, 陶飞, 等. 基于云制造平台企业合作演化仿真系统的设计与实现[C]// 第九届中国多智能体系统与控制 (MASC2013). 焦作: 中国人工智能学会, 2014: 1-5.
- Zhao Chun, Zhang Lin, Tao Fei, et al. Design and Implement of Enterprise Cooperation Evolution Simulation System Base on Cloud Manufacturing Platform[C]// The 9th China Conference on Multi-agent Systems and Control (MASC2013). Jiaozuo: Chinese Association for Artificial Intelligence, 2014: 1-5.
- [17] 北京航空航天大学. 基于多智体的云制造仿真系统和方法: CN201610997856. X[P]. 2017-01-31.
- BeiHang University. Cloud Manufacturing Simulation System and Method Based on Multi-Agent: CN201610997856.X[P]. 2017-01-31.
- [18] Zhou Longfei, Zhang Lin, Ren Lei, et al. Real-Time Scheduling of Cloud Manufacturing Services Based on Dynamic Data-Driven Simulation[J]. IEEE Transactions on Industrial Informatics(S1551-3203), 2019, 15(9): 5042-5051.
- [19] 潘俊浩. 面向工业机器人的云制造仿真服务系统的研究与开发[D]. 厦门: 厦门大学, 2017.
- Pan Junhao. Research and Development of Cloud Manufacturing Simulation Service System for Industrial Robots[D]. Xiamen: Xiamen University, 2017.
- [20] Toly Chen, Min-Chi Chiu. Development of a Cloud-Based Factory Simulation System for Enabling Ubiquitous Factory Simulation[J]. Robotics and Computer-Integrated Manufacturing(S0736-5845), 2017, 45: 133-143.