

# 基于演化博弈的边缘资源服务合作机制运行研究

乔 晗<sup>1</sup>, 程良伦<sup>1,2</sup>

<sup>1</sup>广东工业大学自动化学院, 广东 广州

<sup>2</sup>广东工业大学计算机学院, 广东 广州

收稿日期: 2023年1月22日; 录用日期: 2023年2月17日; 发布日期: 2023年2月28日

## 摘 要

本文介绍一种基于演化博弈理论的服务组合收益优化算法。在传统的边缘过程自动化控制系统中, 常见一些需要集中寻优、灵活调整输出的算法决策, 通常这些算法的决策过程需要在特定的软件系统模块中进行单独配置和求解, 其现场的可操作性对于工程师来说并不友好, 并且在过程控制过程中, 依赖算法的计算模块通常配置于云中央计算单元中, 其结果通常需要现场通信方式传递给边缘控制设备, 算法处理的实时可靠性受限于通信系统的稳定性, 对于边缘过程自动化控制系统的准确性和稳定性都存在潜在的威胁。本文基于对上述问题的考虑, 提出一种能够运行在低算力运行时中的组件化资源服务的自组织智能算法, 通过对演化稳定性策略的分析, 探讨了边缘资源之间合作行为的进化路径。通过符合IEC 61499标准的功能块结构对演化博弈进行组件化封装, 通过组件的方式提高在边缘过程控制系统中算法计算的复用性与实时操作性。

## 关键词

演化博弈, 服务组合, 边缘过程自动化, 算法组件化封装

# Research on the Operation of Edge Resource Service Cooperation Mechanism Based on Evolutionary Game

Han Qiao<sup>1</sup>, Lianglun Cheng<sup>1,2</sup>

<sup>1</sup>School of Automation, Guangdong University of Technology, Guangzhou Guangdong

<sup>2</sup>School of Computer Science and Technology, Guangdong University of Technology, Guangzhou Guangdong

Received: Jan. 22<sup>nd</sup>, 2023; accepted: Feb. 17<sup>th</sup>, 2023; published: Feb. 28<sup>th</sup>, 2023

文章引用: 乔晗, 程良伦. 基于演化博弈的边缘资源服务合作机制运行研究[J]. 计算机科学与应用, 2023, 13(2): 281-300. DOI: [10.12677/csa.2023.132028](https://doi.org/10.12677/csa.2023.132028)

## Abstract

This paper introduces a service portfolio revenue optimization algorithm based on evolutionary game theory. In the traditional edge process automation control system, it is common to make algorithm decisions that require centralized optimization and flexible adjustment of output. Usually, the decision-making process of these algorithms needs to be separately configured and solved in specific software system modules. Its on-site operability is not friendly to engineers, and in the process control process, the calculation modules that rely on algorithms are usually configured in the cloud central computing unit. The results usually need to be transmitted to the edge control equipment by field communication. The real-time reliability of the algorithm processing is limited by the stability of the communication system, which has potential threats to the accuracy and stability of the edge process automation control system. Based on the consideration of the above problems, this paper proposes a self-organizing intelligent algorithm for component-based resource services that can run in low computational power runtime. Through the analysis of evolutionary stability strategy, the evolutionary path of cooperative behavior between edge resources is discussed. The evolutionary game is encapsulated by components through the function block structure conforming to IEC 61499 standard, and the reusability and real-time operability of algorithm calculation in the edge process control system are improved by components.

## Keywords

Evolutionary Game, Service Composition, Edge Process Automation, Algorithm Componentization Packaging

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 简介

近年来, 随着云制造等先进制造理念的推行与发展, 服务于过程控制、分布式控制等相关领域的工业标准也得到人们的重点关注。在传统分布式过程控制中, IEC 61131 标准经过多年的应用与检验, 占据过程自动化控制标准的重要位置, 但随着边缘计算等分布式计算的兴起, 其在边缘过程自动化的应用不能很好的适配。

在 IEC 61131 基础上, IEC 61499 定义了通过功能块用于开发分布式工业控制解决方案的领域特定建模语言。IEC 61499 通过改进软件组件的封装以提高可重用性、提供独立于供应商的格式以及简化对控制器到控制器通信的支持, 从而扩展了 IEC 61131-1。对于 IEC 61131-3 标准提供更多的灵活性与可靠性。它的分发功能和对动态重新配置的内在支持为工业 4.0 和工业物联网应用提供了所需的基础设施。

边缘过程自动化控制系统是相对于中央控制系统的概念, 通常来说, 其运行在靠近过程自动化控制设备中, 将一部分过程控制中的计算任务实现本地计算、本地控制, 有效提升控制系统整体的有效性、稳定性与实时性。但是受限于边缘计算设备硬件配置、工作环境等问题, 其不能很好地支持一些复杂计算任务, 通常这些复杂计算需要云中央控制系统承担, 然后通过现场总线设备将计算结果回传至边缘控制节点中, 这一过程依赖稳定传输, 对于过程自动化的稳定性构成一定的挑战。

IEC 61499 支持分布式的功能块组件, 通过对于复杂计算的组件化封装, 使其能够在边缘设备中运行, 功能块组件的高度灵活性、可重构性以及可复用性对于边缘控制设备的性能提升有显著效果。

在研究关注边缘资源的控制同时, 边缘资源之间的制造服务组合关系同样值得关注。其中, 关注制造资源之间的合作机制, 而不是制造资源之间的横向合作。确切地说, 加强制造资源之间的合作是发展过程制造最重要的方法之一。其优点包括及时合作解决制造任务和通过规模收益获得的利润空间的成本回报。制造资源之间的合作机制提升整体生产过程生产效益, 为工业化提供了坚实的基础。

本文的其余部分组织如下。第二节简要介绍相关工作。本文解决问题的描述在第三节中说明。在第四节中详细描述基于演化博弈理论的边缘资源合作模型。在第五节中介绍基于演化博弈论的边缘资源合作算法设计。在第六节中, 通过实验仿真, 验证算法有效性。最后, 第七节总结本文工作以及未来可以进行的工作。

## 2. 相关工作

以 CPS 为支撑的工业自动化成为科研人员与业界相关从业人员的共识, CPS 是由计算和物理组件集成的柔性交互系统[1]。CPS 对于边缘过程自动化控制系统支撑其实现自主计算、信息 - 环境交互模式都有很好地概念支撑。IEC 61499 如何与分布式智能自动化的实现技术集成在一起是我们需要重点研究关注的问题。接下来我们将从两个视角观察这个问题: 1) 基于组件的设计模式, 2) 分布式智能与边缘自主计算。

### 1) 基于组件的设计模式

目前提出的基于组件的分布式自动化系统建模框架主要是基于自动化对象概念, 即可兼容自动化的接口的组件对象模型, 自动化对象用来表示自动化和控制功能性能的硬件和软件建模单元[2]。IEC 61499 标准定义了三种类型的功能块组件: BFB、CFB、SIFB, 其中, 基础功能块由于定义了良好的事件与参数接口, 能够实现对于自动化对象的良好封装, 一般来说可以将其视为软件组件[3]。Wiesmayr B, Zoitl A [4]认为函数块是完全封装的软件组件。对于函数块定义, 需要一个静态接口描述和一个内部实现。服务序列可以描述成为组件接口上的事件流。Black G 和 Vyatkin V 提出了一种基于 IEC 61499 功能块的行李处理系统多智能体控制方法。侧重于演示一个分散的控制系统, 具有可重构和容错。该设计遵循自动化对象的方法, 生成一个表示单个部分输送机的功能块组件[5]。

Zhabelova G, Yang C W 等人提出了一种实际的信息物理组件体系结构的实现, 使用 IEC 61499 标准与外部仿真环境(如 Matlab)连接。演示了该体系结构的一个应用程序, 用于开发和验证分布式网络智能应用程序。该方法为多智能体控制技术在电力系统领域的应用奠定了基础。使用 IEC 61499 作为底层技术, 可以在工业级硬件平台上直接执行此类应用程序[6]。Dai W, Vyatkin V [7]提出了一种基于构件的分布式自动化程序设计模式。此设计模式旨在提高程序的可重用性并减少重新开发时间。分析了 IEC 61131-3 和 IEC 61499 平台中基于组件的设计问题。并在[8]中讨论了 PLC 控制向 IEC 61499 功能块的事件驱动和基于组件的体系结构的迁移问题。Zhou N, Li D 等人提出了一种形式化的语义集成模型驱动设计方法作为一种整体解决方案。根据 IEC 61499 体系结构的调整, 指定了一种特定于领域的建模语言(DSML), 并扩展了任务模型、任务到资源的分配和非功能规范。然后明确定义提出的 DSML 的形式结构语义和行为语义。设计时形式验证也可以通过自动的模型转换来实现。采用元可编程环境, 便于灵活的建模、验证和代码生成[9]。Dai W, Vyatkin V [10]提出了一种将 soa 应用于分布式自动化领域的形式化模型。讨论了在工业自动化中应用 SOA 的实际场景。为了支持 SOA, IEC 61499 模型提出了一种基于服务的执行环境体系结构。

## 2) 分布式智能与边缘自主计算

博弈论作为数学学科的一个分支, 在优化问题领域有着广泛的应用, 近年来, 许多人将博弈论与各种技术相结合, 以充分发挥博弈论的优势。

Feng T 等使用进化博弈理论分析了双方的行为进化趋势。在原有模型的基础上, 解释了在惩罚和激励条件下, 装配式生产商之间的协调与合作机制[11]。Z. Haifeng 等人运用博弈论, 分析了由制造商和供应商组成并以前者为主导的供应链利益分配模型。讨论了模型的前提条件, 并给出了一种简单实用的分配方法, 不仅可以根据贡献获得收益, 还可以根据链状态获得收益[12]。

杨等[13]通过进化博弈理论分析了影响供应链中生产者和供应商之间合作或竞争机制的动态进化因素, 包括博弈主体的初始比例、利益分配、初始投资成本和折扣因子。Su Y 等[14]在供应链网络中, 基于集群企业合作行为的强联系和弱联系, 提出了横向对称和纵向不对称两种演化博弈模型, 研究了合作行为的演化路径和影响因素。

Xing X H [15]使用进化博弈论对冷链商品制造商和物流服务提供商的行为策略进行了测试, 结果表明, 补贴和惩罚是双方采取信息共享策略的必要条件。Tan J 等[16]利用演化博弈论分析了不同惩罚和信息共享风险成本的供应链网络企业之间信息共享行为的演化过程。分析和基于代理的仿真结果表明, 当供应链网络中企业之间的信息量很大时, 很难形成共享合作; 加大处罚力度, 控制成本共享风险, 可以增加供应链信息共享网络的概率, 缩短信息共享时间。

Chouinard J, Brennan R [17]认为分布式智能控制涉及到控制模型与物理系统的更紧密匹配。过程控制是通过许多简单的、自治的和合作的实体(即代理)的突发性来实现的, 这些实体不仅在局部决定如何行动(如子例程所做的)和采取什么行动(如对象所做的), 而且还决定何时启动它们自己的活动。Olsen S, Wang J [18]等人认为实现边缘控制设备的动态可重构, 是解决可用于分布式工业过程控制的模块化软件的需求的重要点之一。

## 3. 问题描述

在云制造环境下, 一个复杂的工艺制造过程需要运行在网络边缘端的制造资源提供大量且不同制造服务。制造资源与资源之间、运行在制造资源中的组件与组件之间都存在着大量的合作关系, 如何实现资源与资源、组件与组件中的快速服务组合, 是满足实现复杂工艺制造过程的关键问题。

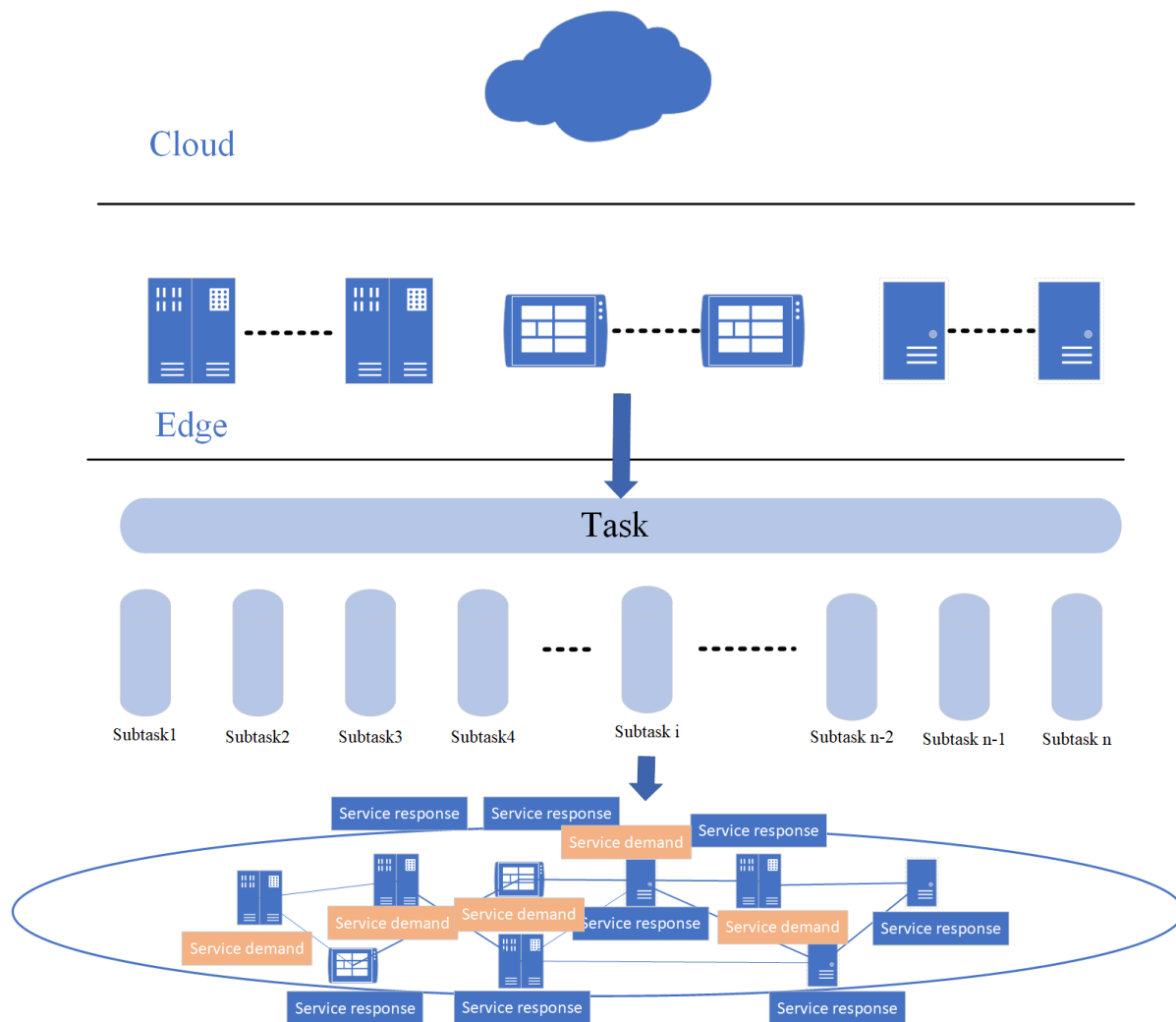
通常来说, 一个复杂的制造过程可以解耦为多级子过程处理, 每一级子过程中都存在着大量的制造资源间的服务合作过程, 具体的服务组合过程可以如下描述。一, 实现对于复杂工艺过程的任务解耦, 制造过程可以分解为  $n$  级子过程  $\{p_1, p_2, \dots, p_n\}$ , 其中  $p_i$  表示第  $i$  个子过程, 每个子过程都需要大量的制造资源之间实现协同合作, 完成子过程的工艺制造需求。针对子过程  $p_i$ , 在边缘端存在大量的备选制造资源响应制造需求以实现本级制造任务, 这些备选的制造资源之间需要协同合作才能完成制造任务, 所以每一级子过程  $p_i$  中备选的制造资源天然的就可以分为两类: 制造任务资源服务需求方和制造任务资源服务响应方。

制造任务资源服务需求方表示为  $\{d_i^{sd1}, d_i^{sd2}, \dots, d_i^{sdm}\}$ , 其中  $d_i^{sdj}$  表示第  $i$  个子过程中第  $j$  个制造服务需求。制造任务资源服务响应方表示为  $\{p_i^{sp1}, p_i^{sp2}, \dots, p_i^{spq}\}$ , 其中  $p_i^{sdq}$  表示第  $i$  个子过程中第  $q$  个响应的候选服务。在云制造环境下, 如何实现边缘制造资源间的合作组合优化过程如下图 1 所示。

云制造中, 针对过程制造的边缘资源组件合作运行, 首先是对过程制造实现子过程分级分解。其次是针对每一级制造过程的制造需求寻找边缘制造资源。最后在边缘制造资源中, 找到合适的制造任务资源服务响应方和需求方, 实现资源合作。因此, 本研究主要是解决受限边缘资源合作的问题, 常见解决手段有: 群体智能算法寻优、利用深度学习/强化学习等机器学习算法。但它们都有一定的局限性, 群



体智能算法容易陷入局部最优, 需要复杂调参, 不能很好地适应生产变化。机器学习的深度学习和强化学习对硬件配置要求较高, 算法部署不轻量, 无法运行在算力受限的边缘资源中, 且需要数据集支持, 需要反复训练, 算法迁移实时性不好。



**Figure 1.** Cloud platform edge resource combination process

**图 1.** 云平台边缘资源组合流程

本文采用演化博弈理论, 计算模型简单, 算法轻量, 博弈决策可以部署到边缘设备中, 不用调参, 博弈模型直接分析整个过程制造的经济收益, 利用演化博弈论中的进化动态模型来分析边缘资源的合作与不合作行为, 通过组件化封装演化博弈算法, 实现在算力受限的边缘资源中实现资源与资源之间、资源内部的博弈收益最大化。首先研究了基于边缘资源服务组合的收益模型, 然后提出了一种适用于该模型的最大收益选择算法。

#### 4. 边缘资源合作的演化博弈模型

合作行为是最普遍的群体行为之一, 其目的是通过合作实现整个群体获得更大的竞争优势或更大的共同利益。不同个体间的合作存在基本的博弈关系, 传统博弈论在研究博弈关系时, 主要基于三个元素:

博弈参与者、博弈策略集和博弈收益。在传统博弈中, 参与者都被假定为完全理性的, 即不管是服务需求方还是服务响应方, 其总是选择优势策略去实现自身收益, 最终判定是否收益最大化是根据博弈系统是否达到纳什均衡来判定博弈各方是否达到均衡态, 纳什均衡即在该状态下, 参与博弈的各方没有任何一个个体可以通过改变自身策略而增加自己的博弈收益。

在工业生产中, 制造任务可以是一个完全的经济收益考量问题。对于一个复杂的制造过程可以解耦为多级子过程处理, 每一级子过程的完成度都可以量化为一个收益值, 当每一级子过程都获得最大收益时, 对于复杂制造过程来说即获得最大制造收益。本文对于收益模型的考量主要基于以下方面: 基础收益  $B$ : 当有制造资源参与解决每级制造任务时, 都会对与解决制造任务提供基础收益。合作收益  $C$ : 每级制造任务中, 存在服务需求方和响应方协作解决制造任务, 提供基础收益额外的合作收益。协作收益  $P$ : 制造资源之间协同合作有助于加快完成制造任务, 提高生产效率和收益, 鼓励协作完成制造任务。惩罚收益  $N$ : 制造资源之间消极协作, 会降低生产效率, 对于不合作的制造资源, 降低其制造收益。能耗  $E$ : 制造资源都有基本的生产消耗, 所有的制造过程都应在基本的能耗约束下完成。

$$\max \left( \sum_{i=1}^n R_i \right) = \left( \sum_i^n i * \cos((i+1) * q_1 + i) * \sum_i^n i * \cos((i+1) * q_2 + i) + cons_i \right) \quad (1)$$

Subject to

$$q_i \in \{B, C\} \quad (2)$$

$$q_i = B_i + C_i \quad (3)$$

$$cons_i = P_i - N_i + \max \left( 0, \frac{E_i - E_{\max}}{E_{\max}} \right) \quad (4)$$

$$0 \leq B_i \leq B_{\max} \quad (5)$$

$$0 \leq C_i \leq C_{\max} \quad (6)$$

$$P_i \geq 0 \quad (7)$$

$$N_i \geq 0 \quad (8)$$

其中,  $n$  是子过程个数。在本文中, 对于每个子过程的收益都认为是有相同的需求。  $E_{\max}$  表示所有资源所能消耗的最大单体能耗, 也可以理解为最大生产成本,  $B_{\max}$  表示所有资源的最大单体基础收益,  $C_{\max}$  表示所有资源最大单体合作收益。

在传统博弈模型中, 通常是研究两个个体之间的博弈交互作用。演化博弈则在传统博弈基础上, 基于群体关系考虑博弈收益, 研究群体中的相互交互作用, 通过解释有限理性的博弈冲突, 随着事件变化, 不断更新自己的选择策略, 向着系统最优目标靠拢, 从而得到群体中收益最大的演化稳定策略(ESS)。基于上述考量, 本文提出基于演化博弈算法边缘资源合作博弈模型, 在解决子过程的制造过程中, 将符合本级的边缘资源分为制造任务资源服务需求方和制造任务资源服务响应方, 它们都可以根据自身情况选择合作和不合作两种策略, 制造任务资源服务需求方和合作方会互相影响。

### 1) 模型构建

构建基于演化博弈的边缘制造资源模型, 做出如下定义:

a) 演化博弈双方为制造任务资源服务需求方  $d_i^{sdj}$ , 制造任务资源服务响应方  $p_i^{sdq}$ , 他们具有相同的策略空间  $RS = \{合作, 不合作\}$ 。

- b) 制造资源在生产过程中会产生基本的开销, 即生产成本记为  $RCC$ 。
- c) 制造资源服务需求方群体在博弈收益矩阵中记为  $A$ , 其合作概率记为  $y$ , 不合作概率记为  $(1-y)$ 。制造资源服务响应方群体在博弈收益矩阵中记为  $B$ , 其合作概率记为  $x$ , 不合作概率记为  $(1-x)$ 。
- d) 参与制造的资源都会产生收益, 这个收益记为  $RBI$ 。
- e) 博弈过程鼓励所有制造资源选择合作策略, 从而向解决子过程制造需求进化。合作产生合作收益, 记为  $RCI$ 。
- f) 博弈收益鼓励合作, 调节边缘资源合作博弈演化方向激励函数  $EC_i = R_iL \times RC_i$ , 其中  $R_iL$  ( $i = A, B$ ) 为资源本身可合作协同系数,  $RC_i$  ( $i = A, B$ ) 为合作成本系数。

g) 博弈收益惩罚投机策略, 调节边缘资源合作博弈演化方向惩罚函数  $PF_i$ , 其中  $PF_i$  ( $i = A, B$ ) 为惩罚因子。

根据上述定义, 基于演化博弈理论的边缘资源合作模型可得如下博弈收益矩阵。其中资源  $A$  中选择合作的群体比例为  $y$ , 选择不合作的群体比例为  $1-y$ , 资源  $B$  中选择合作的群体比例为  $x$ , 选择不合作的群体比例为  $1-x$  (表 1)。

**Table 1.** Revenue matrix between resource services  
**表 1.** 资源服务之间的收益矩阵

不同资源群体间博弈策略		资源 B	
		合作	不合作
资源 A	合作	$R_A BI + R_A CI + EC_A - R_A CC,$ $R_B BI + R_B CI + EC_B - R_B CC$	$R_A BI + EC_A - R_A CC,$ $R_B BI - PF_B$
	不合作	$R_A BI - PF_A,$ $R_B BI + EC_B - R_B CC$	$R_A BI,$ $R_B BI$

表 1 中,  $EC_A = R_A L \times RC_A$ ,  $EC_B = R_B L \times RC_B$ ,  $PF_A = R_B L \times R_A CI$ ,  $PF_B = R_A L \times R_B CI$ ,  $RC_A = R_A CI / (R_A CI + R_A BI)$ ,  $RC_B = R_B CI / (R_B CI + R_B BI)$ 。

## 2) 演化稳定性策略分析

资源 A 在合作策略下的期望收益为

$$ER_A^c = x \cdot (R_A BI + R_A CI + EC_A - R_A CC) + (1-x) \cdot (R_A BI + EC_A - R_A CC) \quad (10)$$

资源 A 在不合作策略下的期望收益为

$$ER_A^{NC} = x \cdot (R_A BI - PF_A) + (1-x) \cdot R_A BI \quad (11)$$

资源 A 的综合期望收益为

$$ER_A = y \cdot ER_A^c + (1-y) \cdot ER_A^{NC} \quad (12)$$

则资源 A 随时间变化的复制动力学方程为:

$$dy/dt = y \cdot (ER_A^c - ER_A) \quad (13)$$

同理资源 B 的合作策略下的期望收益为

$$ER_B^c = y \cdot (R_B BI + R_B CI + EC_B - R_B CC) + (1 - y) \cdot (R_B BI + EC_B - R_B CC) \quad (14)$$

资源 B 在不合作策略下的期望收益为

$$ER_B^{NC} = y \cdot (R_B BI - PF_B) + (1 - y) \cdot R_B BI \quad (15)$$

资源 B 的综合期望收益为

$$ER_B = x \cdot ER_B^c + (1 - x) \cdot ER_B^{NC} \quad (16)$$

则资源 B 随时间变化的复制动力学方程为:

$$dx/dt = x \cdot (ER_B^c - ER_B) \quad (17)$$

综合式(10)、(11)、(12)、(13)、(14)、(15)、(16)、(17), 可得制造任务资源服务需求方和制造任务资源服务响应方两个群体选择合作的复制动力学方程组为:

$$\begin{cases} \dot{x} = x \cdot (ER_B^c - ER_B) = x \cdot (1 - x) \cdot [y \cdot (R_B CI + PF_B) + EC_B - R_B CC] \\ \dot{y} = y \cdot (ER_A^c - ER_A) = y \cdot (1 - y) \cdot [x \cdot (R_A CI + PF_A) + EC_A - R_A CC] \end{cases} \quad (18)$$

根据李雅普诺夫第一法对复制动力学方程组求合作策略下和混合策略下的稳定均衡点进行稳定性分析, 上述微分方程组的解曲线即是策略动态演化过程, 其解曲线的稳定解即均衡点, 当均衡点满足李雅普诺夫第一法时(10), 改稳定点即策略演化稳定均衡点。

$$\begin{cases} \det(J) > 0 \\ \text{tr}(J) < 0 \end{cases} \quad (19)$$

其中,  $J$  为复制动力学方程组对应的雅可比矩阵,  $\det(J)$  为该雅可比矩阵的行列式,  $\text{tr}(J)$  是该雅可比矩阵的迹。即该复制动力学方程组的局部稳定解为(0, 0), (0, 1), (1, 0), (1, 1),  $(p_1^*, p_2^*)$ , 其中:

$$p_1^* = \frac{R_B CC - R_B L \cdot R_B CI / (R_B CI + R_B BI)}{R_B CI + PF_B},$$

$$p_2^* = \frac{R_A CC - R_A L \cdot R_A CI / (R_A CI + R_A BI)}{R_A CI + PF_A}.$$

由此可得(10)的复制动力学方程组的雅可比矩阵为:

$$J = \begin{bmatrix} \frac{dx}{dx} & \frac{dx}{dy} \\ \frac{dy}{dx} & \frac{dy}{dy} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \quad (20)$$

其中,  $J_{11} = \frac{dx}{dx} = (1 - 2x) \cdot [y \cdot (R_B CI + PF_B) + EC_B - R_B CC]$ ,  $J_{12} = \frac{dx}{dy} = x \cdot (1 - x) \cdot (R_B CI + PF_B)$ ,  $J_{21} = \frac{dy}{dx} = y \cdot (1 - y) \cdot (R_A CI + PF_A)$ ,  $J_{22} = \frac{dy}{dy} = (1 - 2y) \cdot [x \cdot (R_A CI + PF_A) + EC_A - R_A CC]$ 。则



$$\det(J) = J_{11} \cdot J_{22} - J_{12} \cdot J_{21} \quad (21)$$

$$\text{tr}(J) = J_{11} + J_{22} \quad (22)$$

## 5. 基于演化博弈论的边缘资源合作算法设计

本节根据所设计的资源合作收益矩阵, 基于所建立的收益模型, 设计使用基于演化博弈论的边缘资源合作方算法。

### 1) 演化博弈算法原理

演化博弈理论中有一个演化稳定策略的概念。该策略描述的是当种群中所有个体都采取该策略的时候, 一小部分突变策略个体无法改变整个种群的演化方向。演化稳定策略是演化博弈论领域非常重要的概念, 其核心思想就是适应度更大的个体有更大的繁殖率, 因此在种群中的占比会增加。

与经典博弈论不同, 经典的博弈论关注的是两个个体之间的交互作用。所研究的问题是个体在不清楚博弈对象做法的前提下, 如何做才能最大化自己的利益。演化博弈论将种群的思维模式引入到博弈论中。研究的对象是许多个体组成的群体。个体会和许多其他个体交互, 在群体的概念下, 找寻整个群体的最大收益状态。

### 2) 演化博弈算法设计

根据演化博弈论的基本原理, 目的就是在所有群体中, 尽可能增加有收益优势的资源的比例, 从而增加最终整个群体选择策略的总体收益。根据这个思想, 在每一级制造任务中, 在能进行合作协同的资源之间, 找出能够实现收益最大化的两类资源, 使其组合完成制造任务, 最终实现整体制造任务的收益最大化。

如上所述, 在云制造中, 存在大量服务于制造任务的边缘制造资源, 它们之间的服务组合不是固定的两两组合, 而是可以根据收益最大化原则选择最适合的制造服务需求方和制造服务响应方两类资源实现协同合作, 从而最优解决每级制造子过程。本文通过上述指导思想, 借鉴演化博弈理论, 通过扩展经典演化博弈策略算法, 提出一种激励不同边缘制造资源之间合作收益的服务组合算法。

1) 初始化备选资源: 在每级子过程之间, 将参与工艺制造的候选资源按照服务需求和服务响应分为可合作的两类资源候选集分别记为群体  $A = \{d_i^{sd1}, d_i^{sd2}, \dots, d_i^{sdm}\}$ , 其中  $d_i^{sdj}$  表示第  $i$  个子过程中第  $j$  个制造服务需求。群体  $B = \{p_i^{sp1}, p_i^{sp2}, \dots, p_i^{spm}\}$ , 其中  $p_i^{sdq}$  表示第  $i$  个子过程中第  $q$  个制造服务响应。在两类服务候选集中, 所有资源都会先将有限的算力投入到根据第四节定义的收益模型计算自身的收益函数, 并根据自身收益情况选择是否合作和不合作。

2) 计算两类备选资源的制造收益: 在群体 A、B 之间, 所有参与解决某一子过程的制造资源都可选择合作策略或不合作策略, 根据资源服务收益矩阵可得当前选择策略的收益值, 并根据选择收益最高的两个资源进行组合, 完成本级制造任务, 具体伪代码如表 2 所示。

a) 全局参数初始化, 读入所有可能参与本级子过程制造任务的所有候选资源集的基础参数。

b) 根据当前博弈阶段, 计算两类资源集中所有资源的博弈阈值即鞍点  $(p_1^*, p_2^*)$ 。当收益小于鞍点, 即绝对不合作资源。

c) 计算参与合作的资源的比例, 并计算其期望收益, 当其收益超过鞍点时, 根据其所属资源集, 根据收益值加入资源服务需求方队列和资源响应方队列。

d) 在两类资源队列中, 根据收益值排序, 分别校验两类资源队列中的策略集是否为合作, 否则依次顺延查找。因为非合作资源会在博弈轮次中因为惩罚收益降低其在后续博弈轮次中的比例, 容易陷入局部最优, 所以查找策略集都为合作的, 收益最大的两类资源即为最优生产资源。

**Table 2.** System resulting data of standard experiment  
**表 2.** 标准试验系统结果数据

基于演化博弈的资源服务组合

```

Input:
服务需求方参与合作的资源比例 y
服务需求方不参与合作的资源比例 1-y
服务响应方参与合作的资源比例 x
服务响应方不参与合作的资源比例 1-x
服务需求方基础收益 BaCI
服务需求方合作收益 RaCI
服务需求方协作收益 ECa
服务需求方惩罚收益 PFa
服务需求方基础收益 BbCI
服务需求方合作收益 RbCI
服务需求方协作收益 ECb
服务需求方惩罚收益 PFb
博弈次数 m
Output 最大收益 res[2][n]
for auto i : m
  for auto i : n
    for auto j:n
      d(i) = y · [ x · (RiBI + RiCI + ECi - RiCC) + (1-x) · (RiBI + ECi - RiCC) ]
          + (1-y) · [ x · (RiBI - PFi) + (1-x) · RiBI ]
      p(j) = x · [ y · (RiBI + RiCI + ECi - RiCC) + (1-y) · (RiBI + ECi - RiCC) ]
          + (1-x) · [ y · (RiBI - PFi) + (1-y) · RiBI ]
      serviceRequest(i) = d(i);
      serviceResponse(j) = p(j);
    end
  end
end
if d(i) <  $\frac{R_B CC - R_B L \cdot R_B CI}{R_B CI + PF_B} / (R_B CI + R_B BI)$  && p(i) <  $\frac{R_A CC - R_A L \cdot R_A CI}{R_A CI + PF_A} / (R_A CI + R_A BI)$ 
  res[0][i] = RaBI
  res[1][j] = RbBI
return max(res[0][i]+res[1][j])
sort(serviceRequest(n))
sort(serviceResponse(n))
bool cooperationStrategy = Game(serviceRequest(i) , serviceResponse(j))
if(cooperationStrategy)
  output i,j
else
  i = i + 1 or j = j + 1
return max(res[0][i]+res[1][j])

```

算法解释说明：所有边缘资源进入生产状态时，会首先将有限的算力资源用于计算上述博弈收益，调用其方法。首先计算参与合作的资源比例，通过数组寻找在当前博弈阶段下的成本并计算在该成本下的阈值。如果参与合作的资源收益小于阈值，则返回相应的收益。

3) 基于演化博弈算法的系统稳定性分析：根据演化博弈收益矩阵求出的复制动力学方程可求出系统局部稳定点，将局部稳定点带入到式(21)、(22)中，可求得如下表 3 所示在各种收益情况下的各均衡点稳定情况。

**Table 3.** System stability analysis under different income conditions  
**表 3.** 不同收益情况下系统稳定性分析

情况	参数条件	均衡点	稳定点	不稳定点	鞍点
1	资源 A: 合作收益 > 不合作收益, 主动合作收益 < 独立收益 资源 B: 合作收益 > 不合作收益, 主动合作收益 < 独立收益	(0, 0) (0, 1) (1, 0) (1, 1) $(p_1^*, p_2^*)$	(1, 1) (0, 0)	(0, 1) (1, 0)	$(p_1^*, p_2^*)$
2	资源 A: 合作收益 > 不合作收益, 主动合作收益 > 独立收益 资源 B: 合作收益 > 不合作收益, 主动合作收益 < 独立收益	(0, 0) (0, 1) (1, 0) (1, 1) $(p_1^*, p_2^*)$	(1, 1)	(1, 0)	(0, 0) (0, 1)
3	资源 A: 合作收益 > 不合作收益, 主动合作收益 < 独立收益 资源 B: 合作收益 > 不合作收益, 主动合作收益 > 独立收益	(0, 0) (0, 1) (1, 0) (1, 1) $(p_1^*, p_2^*)$	(1, 1)	(0, 1)	(1, 0) (0, 0)
4	资源 A: 合作收益 > 不合作收益, 主动合作收益 > 独立收益 资源 B: 合作收益 < 不合作收益, 主动合作收益 > 独立收益	(0, 0) (0, 1) (1, 0) (1, 1) $(p_1^*, p_2^*)$	(0, 1)	(0, 0)	(0, 0)
5	资源 A: 合作收益 > 不合作收益, 主动合作收益 < 独立收益 资源 B: 合作收益 < 不合作收益, 主动合作收益 < 独立收益	(0, 0) (0, 1) (1, 0) (1, 1) $(p_1^*, p_2^*)$	(0, 0)	(1, 0)	(1, 1) (0, 1)
6	资源 A: 合作收益 > 不合作收益, 主动合作收益 > 独立收益 资源 B: 合作收益 < 不合作收益, 主动合作收益 < 独立收益	(0, 0) (0, 1) (1, 0) (1, 1) $(p_1^*, p_2^*)$	(0, 1)	(1, 0)	(0, 0) (1, 1)
7	资源 A: 合作收益 > 不合作收益, 主动合作收益 > 独立收益 资源 B: 合作收益 > 不合作收益, 主动合作收益 > 独立收益	(0, 0) (0, 1) (1, 0) (1, 1) $(p_1^*, p_2^*)$	(1, 1)	(0, 0)	(0, 1) (1, 0)
8	资源 A: 合作收益 > 不合作收益, 主动合作收益 < 独立收益 资源 B: 合作收益 < 不合作收益, 主动合作收益 > 独立收益	(0, 0) (0, 1) (1, 0) (1, 1) $(p_1^*, p_2^*)$	$(p_1^*, p_2^*)$ 非渐近稳定点		(0, 0) (0, 1) (1, 0) (1, 1)
9	资源 A: 合作收益 < 不合作收益, 主动合作收益 < 独立收益 资源 B: 合作收益 < 不合作收益, 主动合作收益 > 独立收益	(0, 0) (0, 1) (1, 0) (1, 1) $(p_1^*, p_2^*)$	(1, 0)	(1, 1)	(0, 0) (0, 1)

Continued

10	资源 A: 合作收益 < 不合作收益, 主动合作收益 > 独立收益	(0, 0) (0, 1) (1, 0)	(0, 1)	(1, 1)	(0, 0) (1, 0)
	资源 B: 合作收益 < 不合作收益, 主动合作收益 < 独立收益	(1, 1) $(p_1^*, p_2^*)$			
11	资源 A: 合作收益 < 不合作收益, 主动合作收益 < 独立收益	(0, 0) (0, 1) (1, 0)	(0, 0)	(1, 1)	(0, 1) (1, 0)
	资源 B: 合作收益 > 不合作收益, 主动合作收益 < 独立收益	(1, 1) $(p_1^*, p_2^*)$			
12	资源 A: 合作收益 < 不合作收益, 主动合作收益 > 独立收益	(0, 0) (0, 1) (1, 0)	(1, 0)	(0, 0)	(0, 1) (1, 1)
	资源 B: 合作收益 > 不合作收益, 主动合作收益 > 独立收益	(1, 1) $(p_1^*, p_2^*)$			
13	资源 A: 合作收益 < 不合作收益, 主动合作收益 < 独立收益	(0, 0) (0, 1) (1, 0)	(1, 0)	(0, 1)	(0, 0) (1, 1)
	资源 B: 合作收益 > 不合作收益, 主动合作收益 > 独立收益	(1, 1) $(p_1^*, p_2^*)$			
14	资源 A: 合作收益 < 不合作收益, 主动合作收益 < 独立收益	(0, 0) (0, 1) (1, 0)	(0, 0)	(1, 1)	(0, 1) (1, 0)
	资源 B: 合作收益 < 不合作收益, 主动合作收益 < 独立收益	(1, 1) $(p_1^*, p_2^*)$			
15	资源 A: 合作收益 < 不合作收益, 主动合作收益 > 独立收益	(0, 0) (0, 1) (1, 0)	(1, 0) (0, 1)	(1, 1) (0, 0)	$(p_1^*, p_2^*)$
	资源 B: 合作收益 < 不合作收益, 主动合作收益 > 独立收益	(1, 1) $(p_1^*, p_2^*)$			
16	资源 A: 合作收益 < 不合作收益, 主动合作收益 > 独立收益	(0, 0) (0, 1) (1, 0)	$(p_1^*, p_2^*)$ 非渐近稳定点		(0, 0) (0, 1) (1, 0)
	资源 B: 合作收益 > 不合作收益, 主动合作收益 < 独立收益	(1, 1) $(p_1^*, p_2^*)$			

上述表 3 对演化系统在所有资源收益条件不同的情况下的稳定性进行了完整分析。由系统稳定性分析表可知, 所有渐进稳定点皆在系统均衡点附近, 不存在(0, 1)之间的渐近稳定点, 且在收益情况 15、16 的状态下, 系统没有渐进稳定点, 处于持续震荡过程。

## 6. 实验分析

本节我们将根据实验, 讨论基于演化博弈算法的服务组件合作有效性, 测试在相同约束条件下, 对比不同规模的子过程。



1) 案例研究

假设存在如下生产过程产线:

a) 产线中每个工位均存在相当数量的候选服务组件。

b) 每个工位完成当前制造子过程都需从其大量的候选服务集中选取至少两个制造服务完成当前制造任务。

c) 制造服务之间存在合作博弈关系

d) 每个工位都需挑选收益最高的制造服务组合, 从而保证整体生产过程获得最大生产收益。

如下生产注塑制品需要下料、运输、合并、测漏、外观检测、重量检测、分料、装箱、封箱以及贴标 10 个生产工艺单元。场景如图 2 所示。

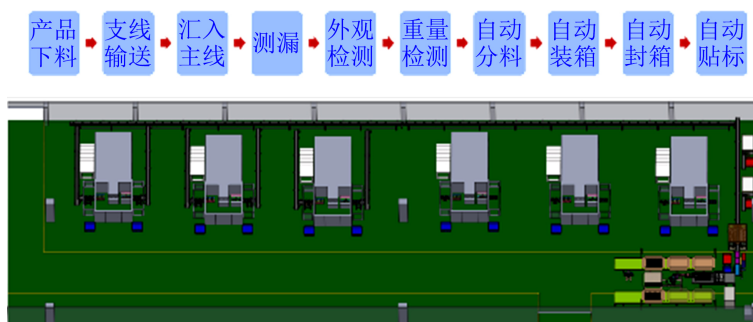


Figure 2. Injection molding product packaging production line  
图 2. 注塑制品包装生产线

2) 实验、结果和分析

为了证明我们提出的方法在解决边缘资源合作问题时的有效性, 考虑到生产的总能耗, 进行了以下比较实验。通过基于演化博弈理论的资源合作算法通经典启发式算法(包括遗传算法、模拟退火算法)进行对比分析。考虑了在不同规模候选服务下具有相同约束的服务资源组件组合场景。基于 IEC 61499 标准的功能块组件设计, 实现资源边缘端算法封装, 从而验证算法在实际应用中的有效性, 如图 3, 图 4 所示。

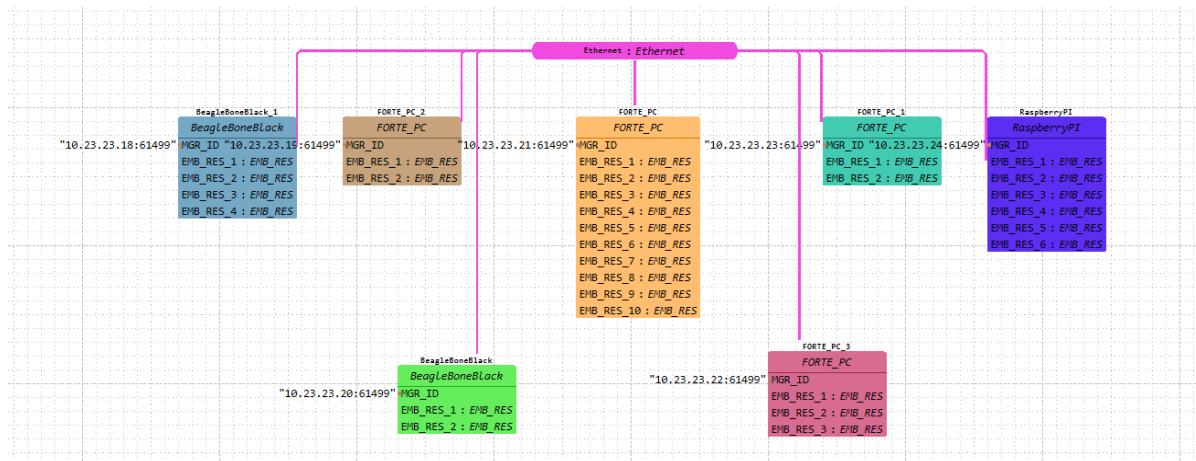
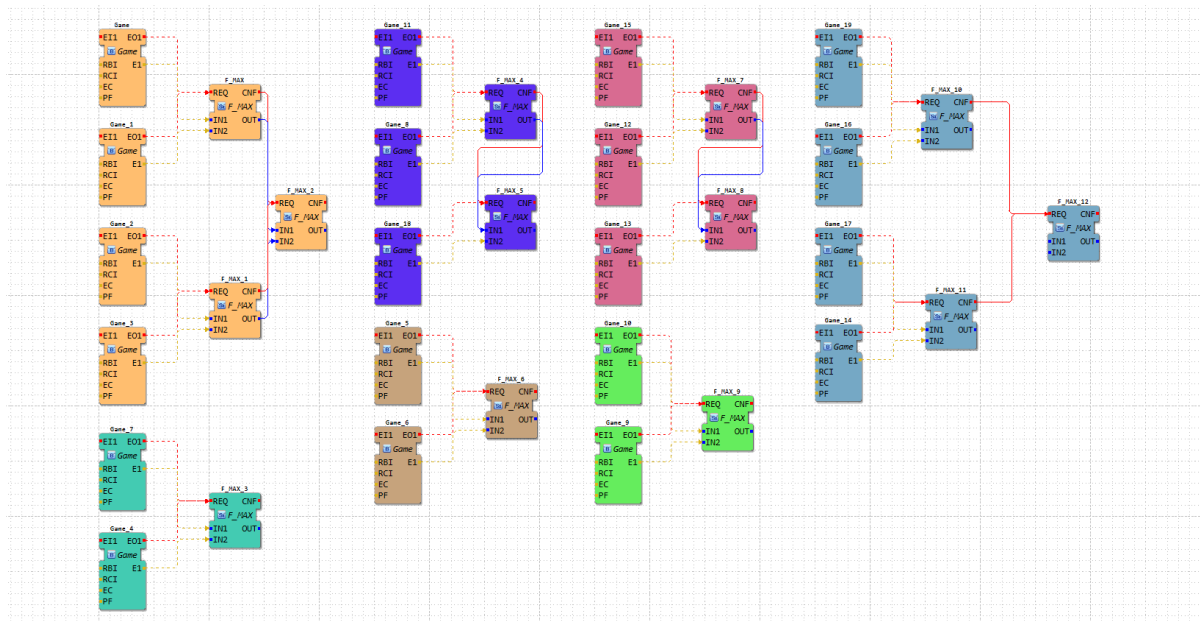


Figure 3. Edge resource model  
图 3. 边缘资源模型



**Figure 4.** Evolutionary game algorithm component  
**图 4.** 演化博弈算法组件

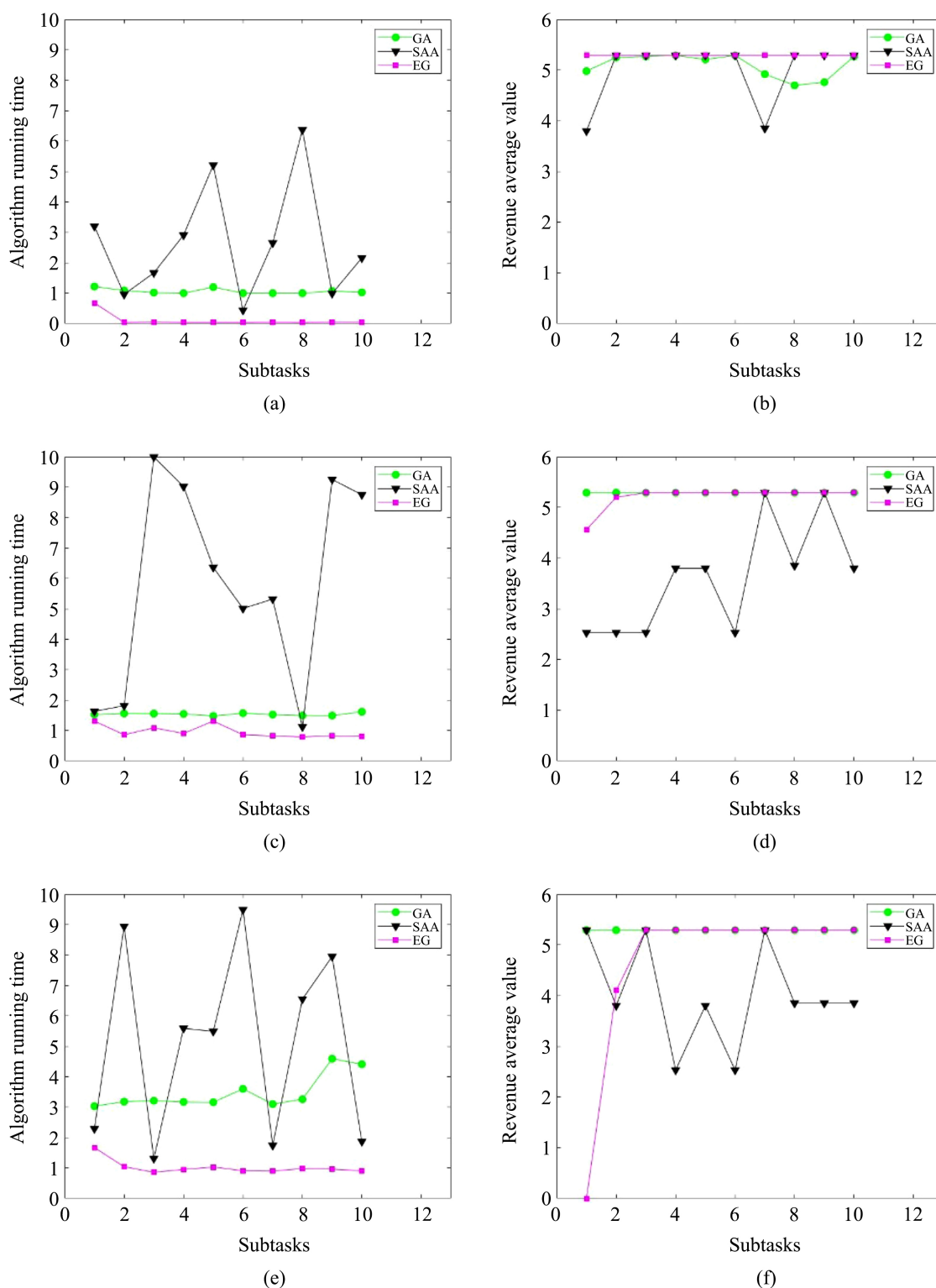
在本节中, 考虑一个生产制造过程由 10 个子过程  $\{p_1, p_2, \dots, p_{10}\}$  组成, 每个子过程都需要至少两个边缘资源服务合作完成。通过实验, 控制每个子过程的候选服务集规模, 依次为 10, 30, 50。通过计算如第四节中构建的(1)收益模型, 确定每级子过程中最符合生产收益的资源服务。

每个算法的其他参数设置如表 4 所示。

**Table 4.** Algorithm parameter setting  
**表 4.** 算法参数设置

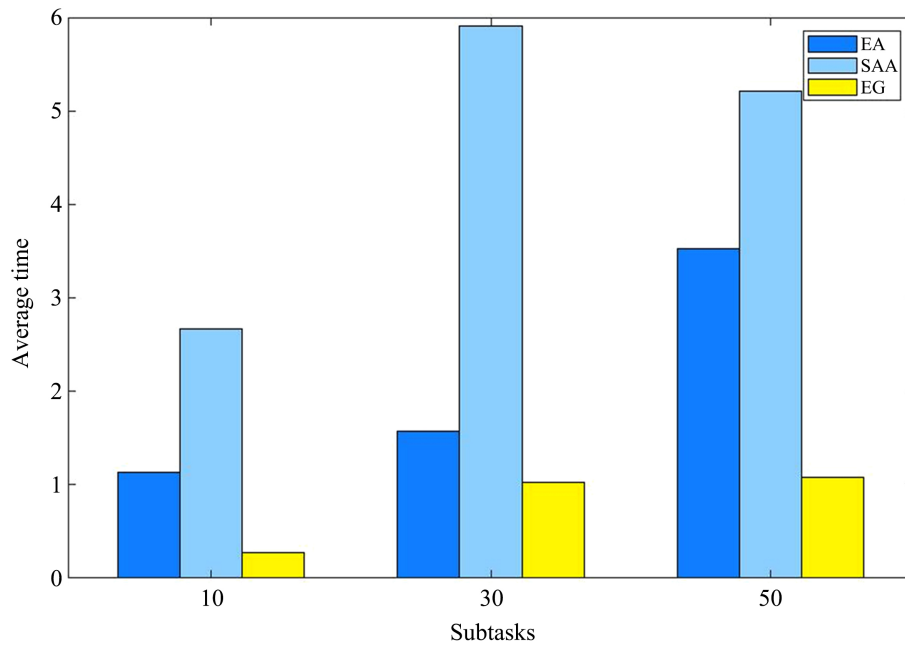
Algorithm	parameter	Setting value
GA	Crossing value	0.9
	Mutation value	0.05
SAA	Temperature	30
	Attenuation velocity	0.95
	step	0.05
EG	Encourage cooperation	0.6
	Penalty factor	0.8

根据上述参数设置, 从实验仿真图 5(b), 图 5(d), 图 5(f)可以看出, 在相同能耗约束条件下, 演化博弈算法和遗传算法能够取得较为接近的收益, 而模拟退火算法由于步长设置以及衰减速率的影响则相对无法稳定求解。图 5(a), 图 5(c), 图 5(e)可以看出, 在算法收敛速度方面, 随着候选服务集规模增加, 由于演化博弈算法直接通过复制动态方程求解每级子过程的收益, 在收敛速度上较为稳定, 而遗传算法的收敛速度随着服务规模增加而显著增加, 模拟退火算法在寻优过程中, 由于服务规模增加时常会陷入局部最优而无法稳定算法收敛速度。

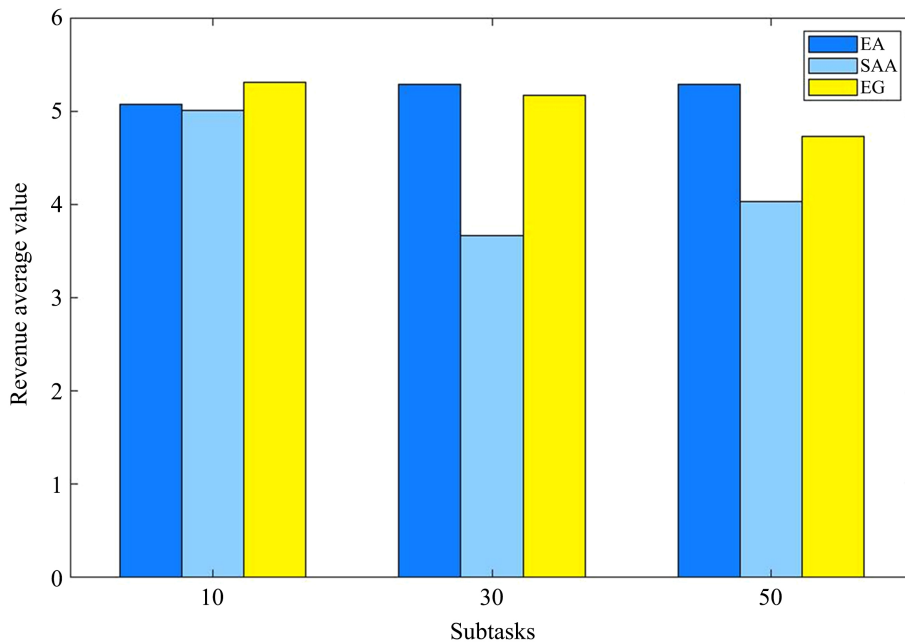


**Figure 5.** Under the sub-process (ST), service (S) and global cost constraint (GC) of different scales, the average revenue value and algorithm running time of ten runs are compared. (a), (b) Candidate service set of each subprocess = 10. (c), (d) Candidate service set of each subprocess=30. (e), (f) Candidate service set of each subprocess = 50

**图 5.** 在不同规模的子过程(ST)、服务(S)和全局成本约束(GC)下, 十次运行的平均收益值和算法运行时间对比。(a)、(b) 每个子过程的候选服务集 = 10。(c)、(d) 每个子过程的候选服务集 = 30。(e)、(f) 每个子过程的候选服务集 = 50



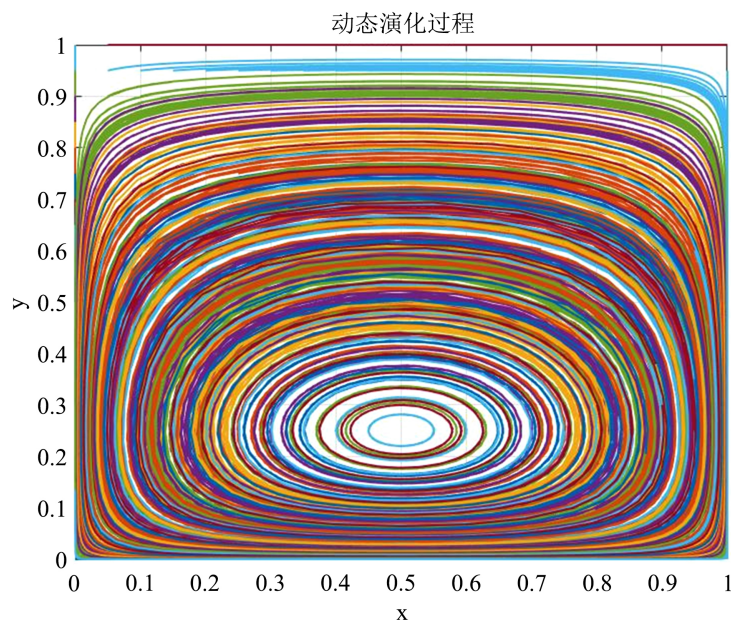
**Figure 6.** Average convergence rate of algorithm under different service scales  
**图 6.** 不同服务规模下算法收敛速度均值



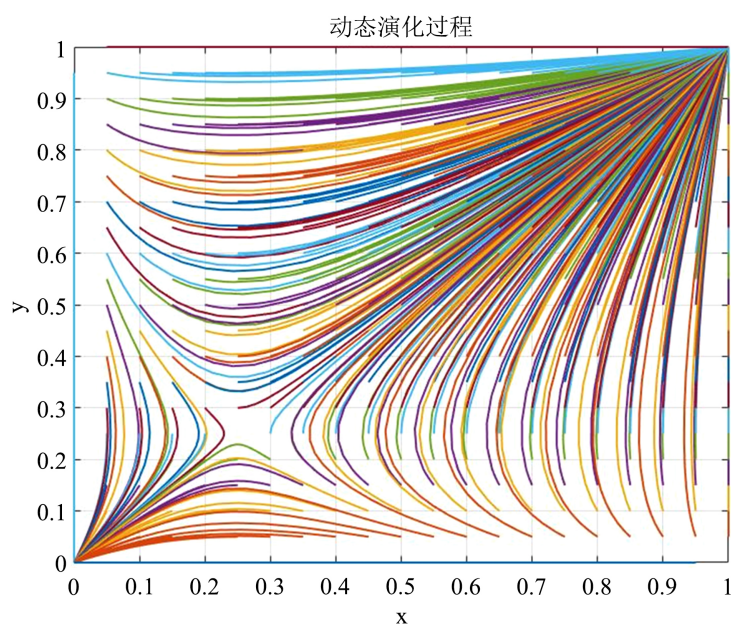
**Figure 7.** Comparison of average revenue under different service scales  
**图 7.** 不同服务规模下收益均值对比

综合来看, 如图 6、图 7 所示, 随着服务规模增加, 由于演化博弈算法通过微分方程求解复制动态方程在算法收敛速度上有较为明显的优势, 在大规模候选服务集的应用中, 由于资源服务的增加, 在计算复制动态方程时可能会出现鞍点附近从而导致算法无法收敛的情况, 如图 8 所示。此时, 根据表 2 设计算法及时调整服务奖励与惩罚因子, 动态调解算法收敛情况。如图 9 所示。



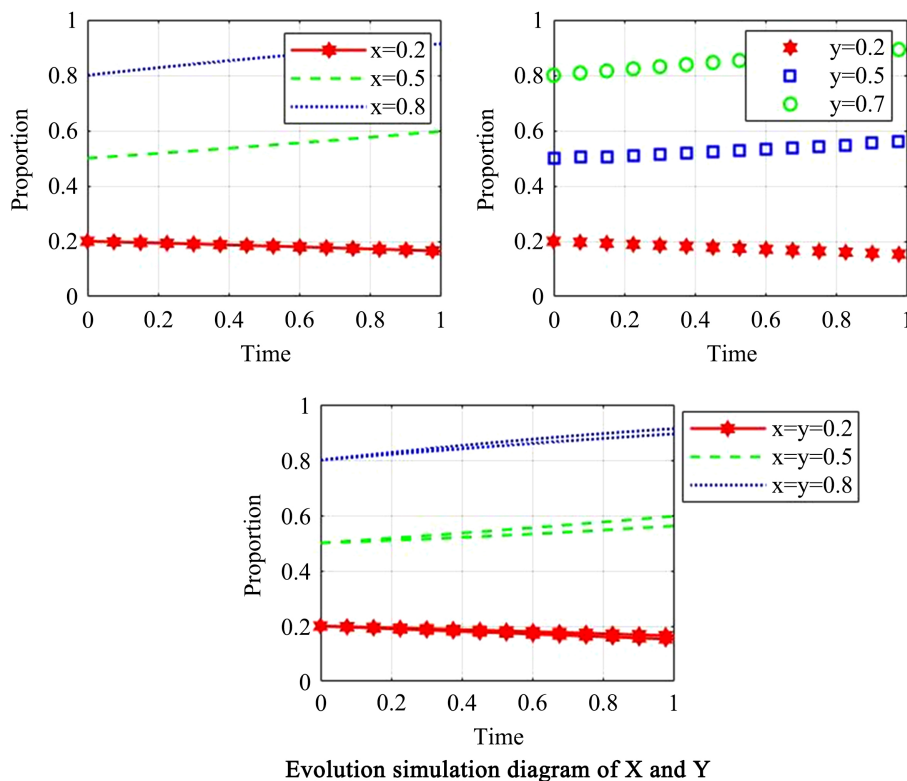


**Figure 8.** The copied dynamic equation cannot converge at the saddle point  
**图 8.** 复制动态方程在鞍点处无法收敛



**Figure 9.** Dynamic evolution process of system convergence  
**图 9.** 系统收敛的动态演化过程

在上述演化过程中, 随着激励函数和惩罚因子的调节, 围绕系统鞍点 $(p_1^*, p_2^*)$ , 每个子过程的候选服务集都在合作与不合作之间形成渐进稳定点。在这种情况下, 激励合作因子较高, 这意味着参与合作的服务获得了奖励。因此, 即使这些候选资源参与服务的意愿较弱, 但如果及时调节资源服务的收益状态, 每个子过程中的候选服务最终策略是参与服务获得最大收益, 通过遍历所有候选服务能够使得算法收敛于一个较为稳定的最佳收益状态。如图 10 所示。



**Figure 10.** Evolution state of candidate resources under different incentives

**图 10.** 不同激励下的候选资源演化状态

通过上述实验表明, 应用演化博弈的生产制造过程中, 每个子过程的候选服务根据不同的设备工况可能存在初始参与意愿具有阈值。随着系统动态调解奖励以及资源服务调整状态初值, 当服务初始参与意愿大于阈值时, 那么系统将会及时收敛到一个比较合理的收益区间, 加速算法收敛。

## 7. 结论

在工业生产过程中, 受限于硬件配置、生产环境、网络延迟、设备异构等问题, 在云制造网络中存在大量的边缘资源需要在不依靠云的情况下实现及时的就地自组织智能运行。并且传统的关注云制造服务调度以及服务组合的领域, 研究者主要关注基于服务质量(QoS)等传统制造指标, 这些传统关注点聚焦于局部服务质量, 容易忽略整个生产制造过程的全局经济收益, 而在工业生产中, 更高的经济收益才是工业制造中追求的最终目标。

本文提出基于演化博弈理论的边缘资源组件化算法, 可以基于组件的方式运行在轻量级的运行时环境中, 在一定权限内实现自组织智能化调配边缘资源, 并且其工作目标服务于整体系统经济收益。为此, 本文建立基于合作关系的边缘资源收益模型。然后, 基于所建立的模型, 将不同资源分为两类集合, 提出了一种以资源组合的演化博弈匹配算法, 以解决资源组合和收益最大化问题。通过实验, 我们发现: 1) 当一方资源选择合作(或不合作)策略时, 另一方也会选择相同的策略作为其最终策略结果; 2) 适当的激励合作奖励和不合作惩罚将影响资源是否选择合作, 以及选择策略趋于稳定的速度; 3) 在一些情况下所有资源的初始合作意愿有一个阈值。当资源的初始合作意愿大于阈值时, 即使激励合作成本较高, 仍有大量的资源选择合作解决制造过程, 以获得更高的制造收益; 4) 需要注意的是, 当激励调节小于成本时, 系统会陷入循环震荡状态, 无法收敛在稳定点。基于上述实验结果, 可以调整基于演化博弈模

型的收益优势资源比例, 根据合作的强度及时调整对资源的合作奖励。

## 基金项目

工业软件组件通用模型、理论及其应用方法研究。国家自然科学基金联合基金项目, 资助号 U20A600004。

## 参考文献

- [1] Lyu, G. and Brennan, R.W. (2020) Towards IEC 61499-Based Distributed Intelligent Automation: A Literature Review. *IEEE Transactions on Industrial Informatics*, **17**, 2295-2306.
- [2] Vyatkin, V.V., Christensen, J.H. and Lastra, J.L.M. (2005) OONEIDA: An Open, Object-Oriented Knowledge Economy for Intelligent Industrial Automation. *IEEE Transactions on Industrial Informatics*, **1**, 4-17. <https://doi.org/10.1109/TII.2005.843829>
- [3] Sunder, C., Zoitl, A., Christensen, J.H., Steininger, H. and Ritsche, J. (2008) Considering IEC 61131-3 and IEC 61499 in the Context of Component Frameworks. 2008 6th IEEE International Conference on Industrial Informatics, Daejeon, 13-16 July 2008, 277-282. <https://doi.org/10.1109/INDIN.2008.4618109>
- [4] Wiesmayr, B. and Zoitl, A. (2020) Requirements for a Dynamic Interface Model of IEC 61499 Function Blocks. 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, 8-11 September 2020, 1069-1072. <https://doi.org/10.1109/ETFA46521.2020.9212107>
- [5] Black, G. and Vyatkin, V. (2009) Intelligent Component-Based Automation of Baggage Handling Systems with IEC 61499. *IEEE Transactions on Automation Science and Engineering*, **7**, 337-351. <https://doi.org/10.1109/TASE.2008.2007216>
- [6] Zhabelova, G., Yang, C.-W., Patil, S., et al. (2014) Cyber-Physical Components for Heterogeneous Modelling, Validation and Implementation of Smart Grid Intelligence. 2014 12th IEEE International Conference on Industrial Informatics (INDIN), Porto Alegre, 27-30 July 2014, 411-417. <https://doi.org/10.1109/INDIN.2014.6945548>
- [7] Dai, W. and Vyatkin, V. (2013) A Component-Based Design Pattern for Improving Reusability of Automation Programs. *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*, Vienna, 10-13 November 2013, 4328-4333. <https://doi.org/10.1109/IECON.2013.6699831>
- [8] Dai, W. and Vyatkin, V. (2012) Redesign Distributed PLC Control Systems Using IEC 61499 Function Blocks. *IEEE Transactions on Automation Science and Engineering*, **9**, 390-401. <https://doi.org/10.1109/TASE.2012.2188794>
- [9] Zhou, N., Li, D., Vyatkin, V., Dubinin, V. and Liu, C. (2020) Toward Dependable Model-Driven Design of Low-Level Industrial Automation Control Systems. *IEEE Transactions on Automation Science and Engineering*, **19**, 425-440. <https://doi.org/10.1109/TASE.2020.3038034>
- [10] Dai, W., Vyatkin, V., Christensen, J.H. and Dubinin, V.N. (2015) Bridging Service-Oriented Architecture and IEC 61499 for Flexibility and Interoperability. *IEEE Transactions on Industrial Informatics*, **11**, 771-781. <https://doi.org/10.1109/TII.2015.2423495>
- [11] Feng, T., Tai, S., Sun, C. and Man, Q. (2017) Study on Cooperative Mechanism of Prefabricated Producers Based on Evolutionary Game Theory. *Mathematical Problems in Engineering*, **2017**, Article ID: 1676045. <https://doi.org/10.1155/2017/1676045>
- [12] Zhao, H., Wang, X. and Su, Y. (2010) Research on Benefit Allocation of the Supply Chain Dominated by Manufacture. 2010 International Conference on Computational and Information Sciences, Chengdu, 17-19 December 2010, 17-20. <https://doi.org/10.1109/ICCIS.2010.12>
- [13] Yang, P. and Pei-Hua, F. (2012) The Cooperation and Competition Mechanism of Supply Chain Based on Evolutionary Game Theory. In: He, X., Hua, E., Lin, Y. and Liu, X., Eds., *Computer, Informatics, Cybernetics and Applications. Lecture Notes in Electrical Engineering*, Vol. 107, Springer, Dordrecht, 1259-1265. [https://doi.org/10.1007/978-94-007-1839-5\\_136](https://doi.org/10.1007/978-94-007-1839-5_136)
- [14] Su, Y., Chai, G., Gong, L. and Li, Z. (2011) Evolutionary Analysis on Cooperative Behavior of Cluster Enterprises in Supply Chain Network. 2011 4th International Conference on Business Intelligence and Financial Engineering, Wuhan, 17-18 October 2011, 575-579. <https://doi.org/10.1109/BIFE.2011.60>
- [15] Xing, X.-H., Hu, Z.-H., Wang, S.-W and Luo, W.P. et al. (2020) An Evolutionary Game Model to Study Manufacturers and Logistics Companies' Behavior Strategies for Information Transparency in Cold Chains. *Mathematical Problems in Engineering*, **2020**, Article ID: 7989386. <https://doi.org/10.1155/2020/7989386>
- [16] Tan, J., Jiang, G. and Wang, Z. (2019) Evolutionary Game Model of Information Sharing Behavior in Supply Chain

- Network with Agent-Based Simulation. *International Journal of Intelligent Information Technologies*, **15**, 54-68.  
<https://doi.org/10.4018/IJIT.2019040104>
- [17] Chouinard, J. and Brennan, R. (2006) Software for Next Generation Automation and Control. 2006 4th *IEEE International Conference on Industrial Informatics*, Singapore, 16-18 August 2006, 886-891.  
<https://doi.org/10.1109/INDIN.2006.275694>
- [18] Olsen, S., Wang, J., Ramirez-Serran, A. and Brennan, R.W. (2005) Contingencies-Based Reconfiguration of Distributed Factory Automation. *Robotics and Computer-Integrated Manufacturing*, **21**, 379-390.  
<https://doi.org/10.1016/j.rcim.2004.11.011>