

Article

Semantic-Based Multi-Objective Optimization for QoS and Energy Efficiency in IoT, Fog, and Cloud ERP Using Dynamic Cooperative NSGA-II

Hamza Reffad ¹  and Adel Alti ^{1,2,*} 

¹ LRSD, Faculty of Sciences, Computer Science Department, University Ferhat Abbas Sétif 1, Sétif 19000, Algeria; hamza.reffad@univ-setif.dz

² Department of Management Information Systems, College of Business & Economics, Qassim University, Buraidah 51452, Saudi Arabia

* Correspondence: a.alti@qu.edu.sa; Tel.: +966-(0)55-990-1163

Abstract: Regarding enterprise service management, optimizing business processes must achieve a balance between several service quality factors such as speed, flexibility, and cost. Recent advances in industrial wireless technology and the Internet of Things (IoT) have brought about a paradigm shift in smart applications, such as manufacturing, predictive maintenance, smart logistics, and energy networks. This has been assisted by smart devices and intelligent machines that aim to leverage flexible smart Enterprise Resource Planning (ERP) regarding all the needs of the company. Many emerging research approaches are still in progress with the view to composing IoT and Cloud services for meeting the expectation of companies. Many of these approaches use ontologies and metaheuristics to optimize service quality of composite IoT and Cloud services. These approaches lack responsiveness to changing customer needs as well as changes in the power capacity of IoT devices. This means that optimization approaches need an effective adaptive strategy that replaces one or more services with another at runtime, which improves system performance and reduces energy consumption. The idea is to have a system that optimizes the selection and composition of services to meet both service quality and energy saving by constantly reacting to context changes. In this paper, we present a semantic dynamic cooperative service selection and composition approach while maximizing customer non-functional needs and quickly selecting the relevant service drive with energy saving. Particularly, we introduce a new QoS energy violation degree with a cooperative energy-saving mechanism to ensure application durability while different IoT devices are run-out of energy. We conduct experiments on a real business process of the company SETIF IRIS using different cooperative strategies. Experimental results showed that the smart ERP system with the proposed approach achieved optimized ERP performance in terms of average service quality and average energy consumption ratio equal to 0.985 and 0.057, respectively, in all simulated configurations compared to ring and maser/slave methods.

Keywords: ERP; services composition; QoS optimization; context-aware; IoT-Fog cloud service ontology; multi-agent strategies; improved NSGA-II algorithm



Citation: Reffad, H.; Alti, A. Semantic-Based Multi-Objective Optimization for QoS and Energy Efficiency in IoT, Fog, and Cloud ERP Using Dynamic Cooperative NSGA-II. *Appl. Sci.* **2023**, *13*, 5218. <https://doi.org/10.3390/app13085218>

Academic Editor: Vincent A. Cicirello

Received: 1 April 2023

Revised: 18 April 2023

Accepted: 19 April 2023

Published: 21 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Internet of Things (IoT) is a technology recently introduced in the industrial field to provide many services facilities, including data collection for service monitoring and analysis, control, and maintenance. Thus, this technology is revolutionizing the ways factories and industrial organizations improve their tasks. With the technological development in the IoT industrial world, emphasis has been placed on designing advanced industrial applications, including manufacturing, predictive maintenance, smart logistics, and energy networks.

For Enterprise Resource Planning (ERP) [1], the Industrial Internet of Things (IIoT) is crucial as it can provide support for widespread business tasks, including smart tracking items, a smart Cloud ERP system using warehouse robotics and drone deliveries, and so on. With the fast advances of Cloud computing and IIoT, we have witnessed many paradigms to assure the quality and safety of many business tasks. Those solutions provide essential advantages for significant quality enhancement of business outputs and services. Moreover, Cloud and IIoT can provide smart and flexible ERP deliveries, which have become a great shift for Small and Medium Enterprises (SMEs). Nevertheless, the difficulty lies in mastering the evolution of customer needs as well as the considerable growth in the number of IoT devices that include heterogeneous smart sensors. These devices significantly increase the amount of data exposed to processing. Therefore, it is crucial for ERP when managing an increased number of IoT devices, including CPU load, latency, bandwidth, and energy.

On the other hand, a great deal of network latency and congestion is managed through Fog computing. This is a new computing model, which is placed between the Cloud and IoT devices, and is a very important solution for facilitating the deployment of relevant IoT services. Furthermore, it is essential to know that the selection of relevant services is not a standardized solution. It can vary according to specific enterprise needs [2]. To reach the goal, more advanced and intelligent techniques might be used via the exploitation of additional semantic information to effectively select and compose services among a large number of service candidates.

To this end, researchers shift out to support intelligent QoS-aware service selection and composition approaches as a solution that could make a significant contribution to deal with a wide range of IoT, Fog, and Cloud services. Furthermore, selecting and composing services can be achieved in three main ways: (1) ontologies [3–9], (2) metaheuristic [10–21], and (3) IoT-Fog enabled services [22,23]. Each one of them has its advantages as well as its drawbacks. Recently, Polyakov et al. [24] proposed an approach to estimate the QoS of composite services based on intuitionistic fuzzy sets. The importance of the proposed approach is due to the fact that it can be applied to all types of service systems. What can be deduced from techniques mentioned is that none of them can achieve an optimum consensus between QoS, energy, and execution time. These techniques are not efficient and flexible enough to support service failures, and none of them can achieve an optimum consensus between service quality, service speed, service cost, continuous changes of customer's needs, limited device energy, etc. In fact, they consume a large amount of processing time to intelligently find optimal composite services. Furthermore, existing works lack a parallel strategy that effectively finds relevant services, customizes ERP business processes, and improves the quality of the solution. A hybrid dynamic multi-agent system that combines both genetic and cooperative agents can solve the optimization of the service composition problem in an acceptable amount of time. A multi-agent system can discover and select relevant services by launching many intelligent agents to potential sites based on the ontology of the IoT-Fog-Cloud semantic service model in order to optimize when composing business processes both in terms of QoS and energy. In addition, smart and mobile devices with limited energy for the deployment of services should not be neglected. The most striking case is the exponential increase of heterogeneous IoT-based services and the continuous evolution of various business requirements based on both users' requirements and preferences. So, it is difficult to obtain ideal customization results of ERP systems and realize an effective service selection and composition method for ERP applications. There is thus a requirement to explore the relative importance of a semantic-based constraint violation strategy to ensure energy saving in IoT devices.

We focus on semantic web technology, smart devices, and, especially, the fragmented system process of enterprise departments using cooperative intelligent agents, which are widely used despite the fact that the advantages of these technologies can help us to build an effective and efficient business process management model.

1.1. Research Gaps

Despite the significance of the IIoT for the flexibility of smart ERP, many such technologies frequently do not include both QoS and energy attributes. This is because both customer needs and the energy capacity of IoT devices are frequently not considered in existing smart ERP management approaches. As business tasks are influenced by their quality energy contexts, it is important to manage such services and energy contexts during the selection and composition of appropriate services at the runtime.

1.2. Motivations and Contributions

This paper presents an efficient and effective cooperative method for smart ERP IoT-Fog-Cloud service selection and composition. It extends the CCS-2S component of the global framework architecture introduced in [25] to enhance the precision and optimize the execution time of IoT with Fog and Cloud services composition in order to offer useful and optimal business processes depending on the user's QoS constraints and contextual preferences. In particular, we introduce a new QoS energy violation with a leverage energy-efficient cooperative strategy. Since there exist different types of cooperative strategies for service composition, we propose a QoS-Energy aware multi-agent strategy, which leverages relevant services in the selection phase and optimizes energy consumption. The proposed study is very important because it is carried out in one of the new research fields, especially in light of the improvement of distributed business processes in Algerian companies, since semantic-based context-aware strategies can help to filter business services that violate both energy and customer constraints. An evaluation was undertaken of the proposed method using several kinds of agents through the IoT business process of the IRIS company in Setif city in Algeria. The study achieved promising ERP performance in terms of QoS and saving energy. The contributions to this dynamic and adaptive smart ERP are outlined as follows:

- We extend the previous model of CxQSCloudSERP ontology [25] with a variety of industrial IoT concepts such as robotics, drones, I-IoT, etc. for describing rich information of ERP services to ensure shifting and business agility. It enables the classification of services by category, role, QoS, and energy consumption.
- We propose a Semantic Context-aware Agent-based Concrete Composite Service with three stages (SCwA-CCS-3S) for smart ERP to accelerate the service composition task and offer customers a flexible personalized smart ERP.
- We propose an improved SCwA-CCS-3S, which provides a three-stage composition and optimization process. First, we introduced a novel dominance relationship with a new QoS energy violation degree through NSGA-II (Non-Dominated Sorting Genetic Algorithm II) [17]. Intelligent cooperative agents aim to quickly discover available composite services that meet customer's constraints. We then selected a composite service (concrete Business Process) among the set of all generated composite services, according to customer's QoS preferences. Finally, we adapted a composite service for managing unpredicted events, such as changes of customers' QoS needs, critical sensor events, and low latency that have stopped working.
- The proposed approach targeted the problem of agility and dynamicity of ERP in IRIS Setif Company of Algeria. It is based on different cooperative multi-agent strategies to compose business services quickly and inexpensively.

1.3. Organization of the Paper

The organization of this paper is as follows. Section 2 presents a motivating case study. Section 3 provides a review of different service composition solutions. An extended CxQSCloudSERP ontology model is introduced in Section 4. Section 5 describes the problem description. The proposed framework is detailed in Section 6. Section 7 presents experiments and the interpretation of results. Finally, the last section concludes this work.

2. Motivating Case Study: IRIS SETIF Company

In order to create a clear scientific vision of the research problem, this section analyses the IoT services and fog computing to illustrate their utility and effectiveness in distributed business processes using a real-life scenario. This scenario is based on the production and transport business processes of IRIS Company of SETIF city, Algeria. This company manufactures electronic products and provides wheels and mobile IT services to clients. These services are accessible through the IRIS Web portal. The IRIS services are hosted on a Cloud provider using the SaaS service type to provide additional support for services that are not in-house. This is illustrated in Figure 1, which covers the Cloud and the user application layers. A centralized Cloud ERP proxy selects and composes quality Cloud services based on IRIS company's needs and execution context. The Cloud ERP proxy, however, is in charge of satisfying the company's needs and preferences, deploying and maintaining ERP business processes. When the number of requests becomes enormous, it slows down the execution speed of the Cloud ERP proxy with considerable latency and can thus influence the QoS.

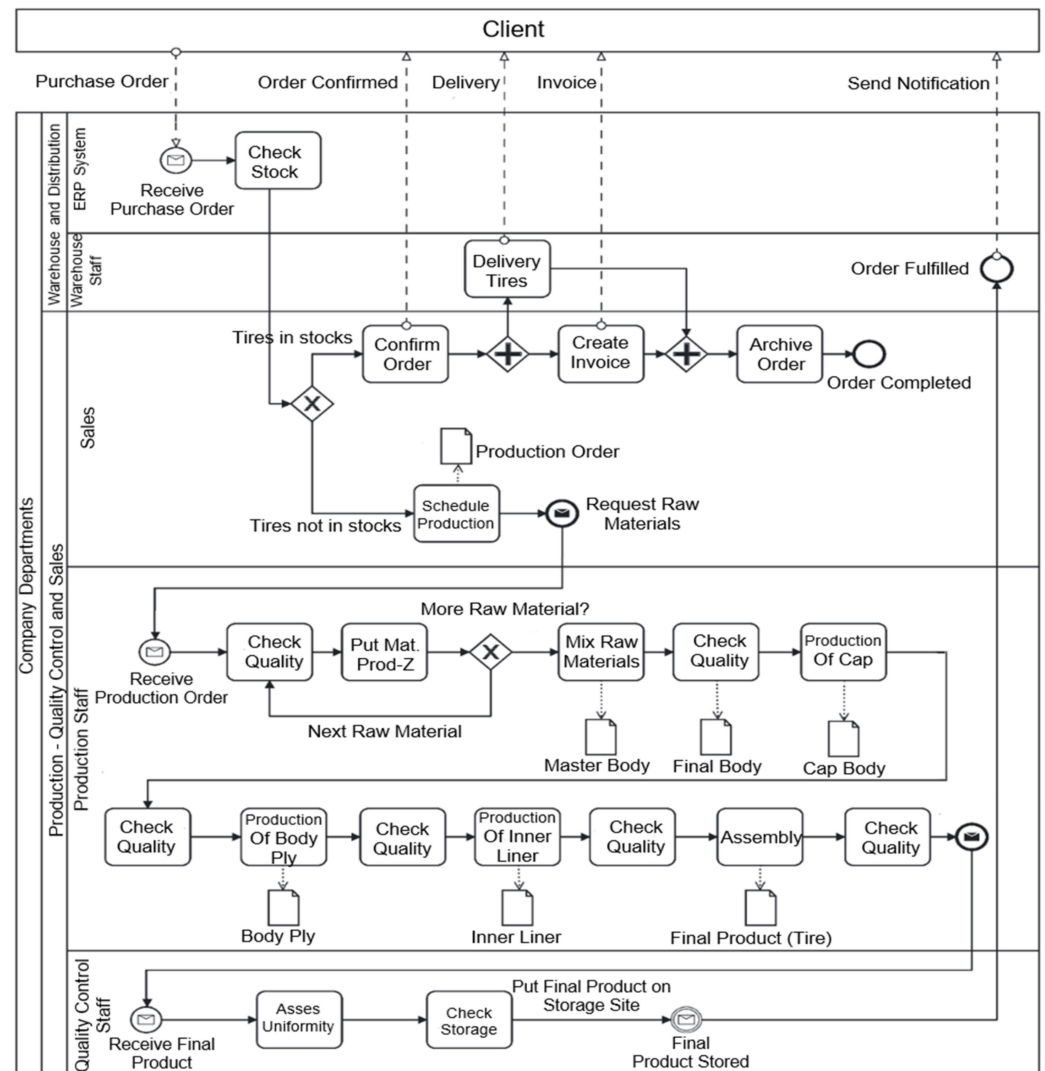


Figure 1. Cloud-based ERP production business process for IRIS.

Besides, IRIS is facing a continuous increase in business and suffering from resource limitations that can lead to production delays and significant economic losses to control raw materials and products in many different sites with different storage conditions of its growth cycle. Thus, its current traditional Cloud ERP software has become inefficient

and lacks flexibility and efficiency in production and transport management services. To overcome these constraints, it is necessary to modernize current traditional Cloud ERP services and introduce modern technologies, such as industrial IoT services, in this company to improve production and adopt effective transport. Many of benefits of industrial IoT that have been identified include: (1) robots to optimize the transport of raw materials and products on different locations, (2) smart sites equipped with sensors and actuators to ensure remote control of suitable storage conditions for growing products, (3) production time and energy cost optimization to effectively transport the appropriate quantity of raw materials, and (4) synchronization and monitoring of different robots with ERP services. We aim to dynamically integrate industrial IoT with existing production and transport management processes. All the challenges faced by integrating industrial IoT with existing Cloud ERP are as follows:

- How could we integrate different mobile robotics and monitoring services to ERP?
- How to choose reliable IoT services to be used in Cloud business processes to get better QoS at optimized cost?
- How can we manage the increase in IoT services while increasing the amount of data?
- How would manage data among IoT services and Cloud ERP affect latency and QoS?

As services can be heterogeneous, there will be different data formats. Furthermore, different types of protocols might be used. Thus, it will be difficult to integrate such robotics and compose them with existing business processes. To handle these problems, a semantic ontology model is proposed. The ontology undertakes this problem by hiding a large amount of heterogeneous IoT data and their different communication protocols. However, the selection and integration of quality IoT services are still difficult and differ according to the context of ERP application, time constraints, and evolution of the QoS needs and preferences of the company (non-functional needs and preferences). For example, an IoT service that detects incorrect data in real-time can cause a service failure or synchronization problem detection. Thus, it increases the complexity of IoT integration to the business processes. If the complexity of the ERP business processes augments using the semantic model, the most efficient way to resolve it is to combine IoT-Fog-Cloud services with the semantic web and multi-agent approach to make the business processes dynamic and optimal for every QoS and context change. Furthermore, Cloud-based ERP requires a high execution time to integrate and select quality services from a large candidate service and further requires both high IoT data processing and transmission time. A slight execution time and transmission delay mismatches the SLA agreement. To handle such a scenario, fog computing provides better QoS for production and transports IoT Cloud-based business processes. Finally, the estimation of the QoS of different services should meet comprehensive model normalization for appropriate prediction and management of QoS in ERP.

Figure 2 shows the use of an intelligent agents-based ontology to discover and select relevant services according to customer's needs, as well as fog components to improve QoS and transmission delay and reduce Cloud access by delocalizing business services at decentralized fogs. The proposed research work needs to provide:

1. A description of IoT services with richer semantic information for seamless integration with ERP business processes. Moreover, they need to integrate mobility-related features to represent mobile devices and monitor context changes in their locations.
2. The management of business processes efficiently and the optimization of QoS needs and preferences (i.e., the parallel discovery of services and effectively optimizing business processes based on context-aware multi-objective QoS attributes using multi-agent strategies).
3. Context monitoring (e.g., customer's needs, enterprise, and services) and the management of various adaptation tasks on multiple business process parts using a dynamic parallel strategy (e.g., easy replacement of a service with another equivalent, fast integration of new composite IoT service, and continuous discovery of available enhanced services to ensure the quality of ERP).

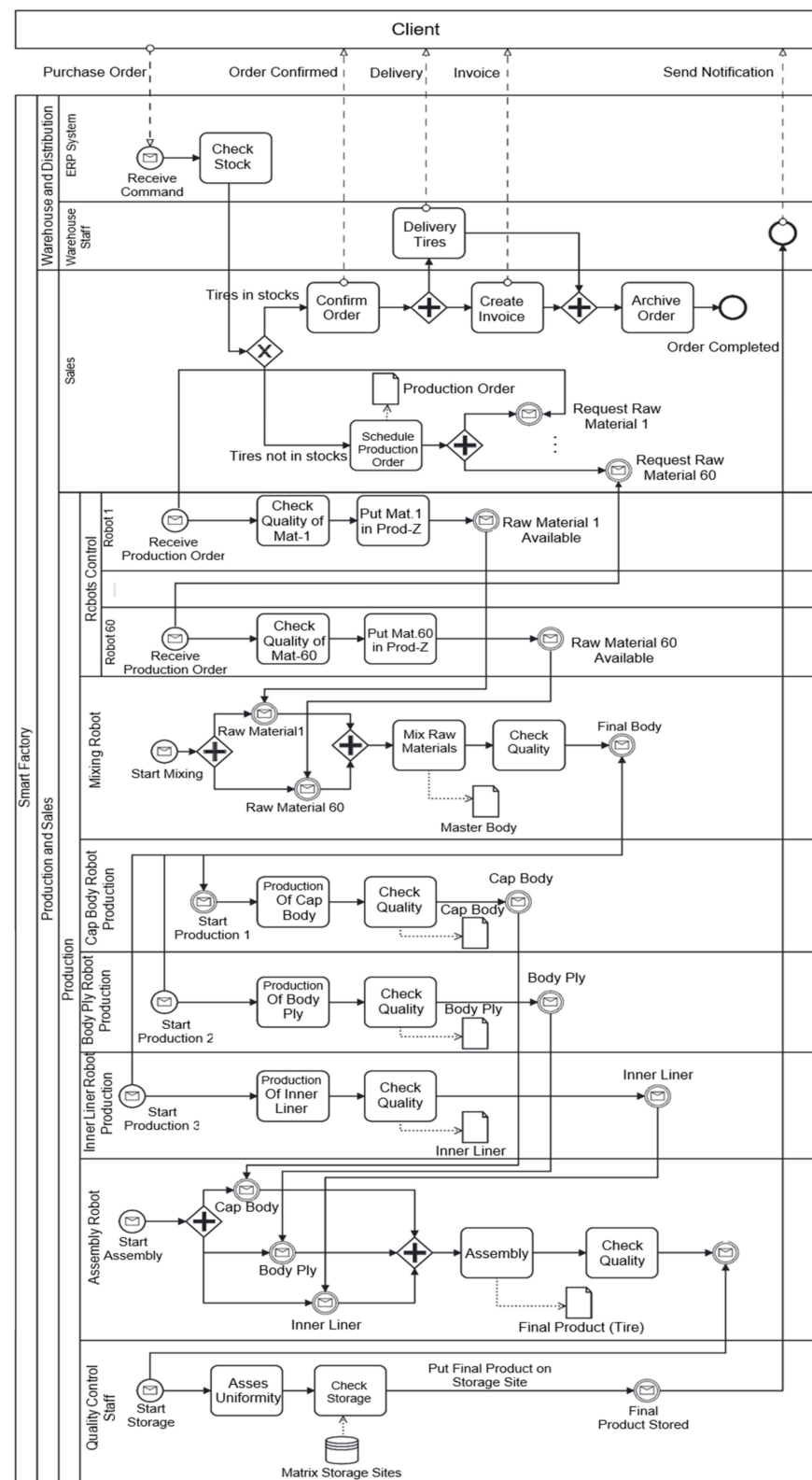


Figure 2. Proposed IoT Cloud-based ERP production business process for IRIS.

3. Related Work

In this section, we will review recent ontology models and optimization approaches related to IoT and Cloud ERPs. Most recent multi-objective optimization approaches can ensure Cloud ERP services description in general and integrate IoT services in particular. However, related works suffer from various limitations, including the energy of IoT device

management and control in the context of business processes. Our work is based on semantic multi-objective optimization for QoS and energy efficiency for IoT-Fog-Cloud ERP using NSGA-II multi-agent cooperation.

3.1. *Ontology Models for Representation of IoT and Cloud ERP Services*

Our work deals with a semantic dynamic approach that produces and optimizes context-aware IoT business processes for industry 4.0 based on an ontology model and multi-agent strategies. Many IoT and Cloud ERP ontology models have been proposed in recent decades [3–9,25,26]. They have been used for different business domains but do not have enough concepts to manage context-aware constraints, describe all business changes, and manage quality semantically equivalent IoT services. However, they all share the following issues: (1) low coverage of business domain-independent semantic service information that describes the Cloud, Fog, and IoT services; (2) lack of consideration of flow sharing (e.g., data flow sharing, event flow sharing), sensor mobility, and usage constraints (e.g., service dependency, energy consumption, and time-space service accessibility), which helps to reduce the impact of context changes on service composition. We distinguish three main categories: IoT-based ontology model, Cloud ERP services ontology model, and combine both ontology models.

Andročec et al. [3] proposed an ontology model for digitalizing different business activities, including IoT services and Cloud ERP APIs. They aim to integrate IoT with various Cloud ERP APIs that can be used to add automated functionalities to ERP systems. They connect various resources with IoT devices and integrate them with Cloud ERP APIs in one module. However, this model neither includes the semantic concepts of contextual management constraints and preferences nor the execution context. Moreover, this work does not consider the classification of Cloud ERP APIs and neglects the semantic grouping of IoT services by functionality, category, role, and QoS, which is considered very important in services selection.

In [4], the authors present a new ontology model called OntoSLAM based on autonomous robots for solving simultaneous localization and mapping (SLAM). It described all aspects related to autonomous robots as well as temporal–spatial movement regions. They integrated it into the Robot Operating System (ROS) to increase the application flexibility. The ontology model relies on many ontology models, including ISRO ontology [5], POS ontology [6], CORA ontology [7], and KnowRob ontology [8], which leads to standard robotics metadata. However, the authors wasted most of their work in the classification of robots and physical things, and hence it neglected significant contextual and QoS constraints for efficiently managing robotic services.

Authors in [9] used semantic Web and ontologies (OWL-S and SAWSDL) for describing OPC UA industrial applications to ensure semantic service annotation and interoperability. The approach builds a common ontology that integrates the production abstract services with concrete IoT-based services. Concrete services will be used on the shop floor. This common ontology makes the approach more interoperable in all levels of automation and infers the potential information. It provides relevant information accurately that helps to describe business processes. Unfortunately, it suffers from contextual information and custom conditions at each stage of the production business process.

3.2. *Service Cloud Composition and Optimization*

The first research domain is related to techniques for the dynamic composition of Cloud services. Three main categories of approaches are distinguished: constraints-based [26,27], semantic-based [28,29], and metaheuristic-based [30,31]. Rosenberg et al. [26] explored constraint specification language to describe rich constraints. The authors conclude that service composition is related to explicit user constraints. However, this approach lacks semantic descriptions of heterogeneous services provided by different service providers. This heterogeneity raises the problem of the right decision making during the services selection [32]. In [28], Alti et al. automated the generation of quality composite

service through context-aware service ontologies and semantic forward chaining. They grouped equivalent services by category, context, and QoS. However, this work lacks efficient and consistent quality Cloud services cooperation. Therefore, in such cases, reducing research time requires an intelligent cooperative strategy to obtain timely and quality solutions while the constraints are dealt at the right time.

From the above discussions, constraints with semantic models do not provide priority to the development and implementation of optimized management ERP systems as they are not aware of the importance of time in decision making. Other researchers recommend metaheuristic-based approaches as iterative methods to solve NP-hard optimization problems. Usually, two optimization approaches are used, which are outlined below.

3.2.1. Mono-Objective Approaches

Many QoS (time, cost, availability, etc.) are provided by Cloud services. The optimal composite Cloud service considered multiple QoS based on mono-objective techniques. It calculates the fitness function of each solution based on certain utilities to choose the optimal solution among feasible ones [33]. The mono-objective function based on the weighted sum [34] is used for optimization of multiple QoS with real cloud services using taboo search [10], GA [11], PSO [12], GWO [13], ACO [14], ICA [15], CHHO [16], and SMO [35]. However, results showed that the mono-objective function is effective with respect to the power loss reduction, but penalizes some QoS attributes.

3.2.2. Multi-Objective Approaches

Yao et al. [17] presented NSGA-II to enhance diversity preservation and convergence. Here, parent and offspring populations were generated and performed non-dominated sorting, and crowding operators were there to find out valuable non-dominated front. The Pareto-optimal front is reached with minimal iterations [36]. Wada et al. [19] modeled the multi-objective optimization problem with E^3 for ensuring service quality as well as effectiveness by employing multiple SLAs. Sadeghiram et al. [21] developed a distributed knowledge-based repair model to increase the efficiency and QoS satisfiability of NSGA-II. It considered QoS constraints for composition process and distributed nature of services, but failed to include industrial IoT and energy model. Peng et al. [37] proposed an improved multi-objective algorithm to seek a representative set of solutions with a dual strategy to adjust the usage of different creation operators using neighbor search in order to achieve a fine-grained search. In our work, we have included both QoS and energy aspects to enhance system performance.

3.3. IoT and Service-Based ERP

In [22], authors developed a CloudERP platform on which the client company customizes an entire ERP system to match their needs. CloudERP provides corporate customers with personalized ERP composite services from multiple providers. This work applied a genetic algorithm with a “rough set theory” and extracted web services from a web service platform. This purpose proposal stands out from the rest due to several elements, including the adaptive way that it reacts to changes in the customer context (customer-evolved QoS constraints and semantic preferences) and the Cloud service context (QoS) at any time.

Recently, in [23], Bolu et al. developed a task-planning approach for efficient management of IoT-based smart warehouses. This approach defines a heuristic model to select order tasks that are assigned to robots. It used a mathematical model to describe the spatial and temporal priority level of tasks, allowing multi-robots to perform tasks in less completion time. However, we noticed that scheduling a large number of tasks increased costs. Therefore, they can achieve effective results with other techniques such as early services selection.

3.4. Discussions

Table 1 presents a comparison between existing ontologies and optimization techniques for ERP semantic description and services composition in terms of (1) reusability, (2) extensibility, (3) dynamicity, (4) adaptability, (5) usage constraints (time-space constraints, services dependencies, energy), and (6) information coverage of thing diversity, thing mobility, service diversity, and events sharing.

Table 1. Review of existing ontologies, service composition, and IoT Cloud ERPs.

Related Works	Criteria	Reusability	Extensibility	Dynamicity	Adaptability	Usage Constraints				Information Modeling					
						Time Constraints	Spatial Constraints	QoS	Energy	Things Diversity	Things Mobility	Services Diversity	Service Dependency	Event Sharing Flow	Data Sharing Flow
Ontologies	Androćec et al. [3]	✓	✓	×	×	×	×	×	×	×	×	✓	×	×	×
	OntoSLAM [4]	✓	✓	✓	×	✓	✓	×	×	✓	✓	×	×	×	×
	ISRO [5]	✓	✓	✓	×	✓	✓	×	×	✓	✓	×	×	×	×
	POS [6]	✓	✓	✓	×	✓	✓	×	×	✓	✓	×	×	×	×
	CORA [7]	×	✓	×	×	✓	✓	×	✓	×	×	×	×	×	×
	KnowRob [8]	✓	✓	×	×	✓	✓	×	✓	✓	✓	×	×	×	×
	SAWSDL [9]	✓	✓	✓	×	✓	✓	✓	×	✓	✓	✓	✓	×	✓
	SSN [7]	✓	✓	×	×	×	×	✓	×	✓	✓	✓	×	×	×
Optimization approaches	Scalarizations	Latency [32]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		Taboo [10]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		GA [12]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		PSO [12]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		ACO [14]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		ICA [15]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		CHHO [16]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
	Multi-Object.	NSGA-II [17]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		SPEA2 [18]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		E3-MOGA [19]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		MOMS-GA [20]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		DWSC [21]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		QoS-Aware [33]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		SMO-MSF [35]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
		Diff-Evolution [37]	×	×	✓	×	✓	✓	×	×	×	×	×	×	×
IoT Service based ERP	CloudERP [22]	✓	✓	✓	×	✓	✓	✓	×	✓	✓	✓	×	×	×
	Bolu et al. [23]	✓	×	✓	×	✓	✓	✓	×	✓	✓	✓	×	×	×
	Fuzzy-Estimation [24]	×	×	✓	×	✓	✓	✓	✓	✓	✓	✓	×	×	×
Proposed Approach		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

These criteria were chosen for several reasons. Reusability refers to how existing concepts can be reused across multiple applications, as well as the choice of extensibility criteria that determine whether or not the model is extensible for domain-specific business models. The choice of dynamicity refers to how existing models monitor context changes and adapt business processes dynamically. The choice of adaptability refers to how existing techniques adapt business processes and improves their quality. Usage constraints refers to how existing models describe spatiotemporal constraints. Limited resources and device constraints are parameters needed in managing sensor mobility in terms of energy consumption. We have identified the common limitations of the studied approaches, particularly for integrating hybrid heterogeneous services providers (IoT, fog, and Cloud) with a heuristic-based optimization strategy to dynamically compose rich services for ERP in smart factories.

Most existing works have neglected both energy and QoS optimization when monitoring one or more products to ensure their safety. We need to integrate mobility-related features to represent mobile devices and monitor changes in their locations and cover-

age area. Moreover, when examining service optimization approaches, we notice that most works in [10–22,34,36] use bio-inspired and meta-heuristics approaches, enabling the optimization of QoS criteria for composite Cloud service regardless of their semantics, such as category, role, energy, spatial temporal constraints, etc. Furthermore, other related works [3–9,23,25,26,38,39] integrate IoT and Cloud services-related semantic features to handle service heterogeneity but without optimizing QoS and energy for IoT-Fog-Cloud ERP. We notice that no approach covers semantic service description for IoT, fog, and Cloud ERP with QoS and energy optimization under sensor mobility and context changes.

Current works have neglected the management and control of energy of IoT devices when composing one or more services and do not propose an energy-aware strategy. In this work, efficient selection of multiple local solutions with a multi-agent approach based on heuristics that considers QoS and energy is defined. It is used to select IoT, Fog, and Cloud services in intelligent and dynamic ways.

4. CxQSIoT-Fog-CloudSERP: An Extension of CxQSCloudSERP Ontology Model

We extend Context-aware Quality Cloud ERP (CxQSCloudSERP) [25] to the limit of its centralized composition of ERP services, even though Cloud ERP services are associated with great IoT and fog services such as warehouse robotics and drone delivery services. The extension of CxQSIoT-Fog-CloudSERP aims to dynamically adapt smart ERP systems to usage contexts of IoT, Fog, and Cloud. The proposed ontology is built based on three semantic services categories, IoT, Fog, and Cloud services, which are used to describe business processes. It classifies context-aware IoT, fog, and Cloud services based on business category, business role, QoS, location, and time.

4.1. Abbreviations

Table 2 summarizes the different abbreviations used in this paper.

Table 2. Abbreviations used in this work.

Abbreviations	Description
ACO	Ant Colony Optimization
BP	Business Process
CCS	Concrete Composite Service
CCS-2S	Concrete Composition Service with 2 Stages
CHHO	Chaotic Harris Hawks Optimization
CxQSCloudSERP	Context-aware Quality of Semantic Cloud Service ERP
ERP	Enterprise Resource Planning
GA	Genetic Algorithm
GWO	Grey Wolf Optimization
ICA	Imperialist Competitive Algorithm
IIoT	Industrial Internet of Things
IoT	Internet of things
IoT-BP	IoT-based business process
MOGA	Multiple Objective Genetic Algorithm
MOMS-GA	Multi-Objective Multi-State Genetic Algorithm
NSGA-II	Non-dominated Sorting Genetic Algorithm
OWL-S	Semantic Markup for Web Services
SCwA-IFCCS-3S	Semantic Context-aWare Agents on IoT-Fog Concrete Composition Service with three Stages
SMA	Small and Medium Enterprises
PSO	Particle Swarm Optimization
QoS	Quality of Service
ROS	Robot Operating System
SAWSDL	Semantic Annotations for WSDL and XML Schema
SCwA-CCS	Semantic-Based Context-aWare Agents for the Composition of Cloud Services
SCwA-CCS-3S	Semantic-Based Context-aWare Agents for Composition of Services
SLA	Service Level Agreement
SMO	Sequential Minimal Optimization
SPEA2	Strength Pareto Evolutionary Algorithm
SWRL	Semantic Web Rule Language
VCS	Virtual Composite Service
VS	Virtual Service

4.2. An Overview of CxQSCloudSERP

CxQSCloudSERP defines Cloud ERP in terms of virtual and concrete services-based business processes [25]. Figure 3 shows the CxQSCloudSERP ontology model of the context-aware services responsible for ERP management. The ontology is designed using the following main classes:

- **Customer:** Manages requirements and preferences of customers. It is specified with GUI and stored as ontology individuals. The customer's requirements are classified into functional needs and QoS constraints. Each functional needs to have a specific language based on logical operators (AND, OR, and Not). It is employed to infer virtual BP. The QoS constraint is split into regular intervals for each QoS attribute based on three semantic values:
 - High: quality values in $[qmin_i^{High}, qmax_i^{High}]$.
 - Medium: quality values in $[qmin_i^{Medium}, qmax_i^{Medium}]$.
 - Low: quality values in $[qmin_i^{Low}, qmax_i^{Low}]$.

The constraint (C_i) of the quality attribute q_i is defined as follows:

$$C_i = qmin_i^{semantic_value} \quad (1)$$

In CxQSCloudSERP, preferences are defined as an ordered list of QoS attributes that are assigned with semantic values. Based on this semantic value, automatically generate the weight w_i by using Equation (2):

$$w_i = \frac{e^{P_i}}{\sum_{j=1}^n e^{P_j}} \quad (2)$$

The computed priority P_i of the QoS q_i is:

$$P_i = explicitPriority_i + constraintRank_i \times n \quad (3)$$

where:

- n : Number of QoS attributes.
- *ExplicitPriority*: A priority level of QoS. It is defined by a value between 1 and n . The *explicitPriority* is null when the customer does not specify any preference.
- *ConstraintRank*: The importance of QoS constraint. There are four priority levels: high, medium, low, and none.
- **Service:** The ontology presents unified business concrete services, such as buy, sale, finance, etc. Each service uses resources and they are linked to other services that provide or require these resources.
- **Business Process (BP):** This BP includes a set of business tasks (*parallel, iterative, choice, repeat*) delivering the same features with varied QoS.
- **Context:** Circumstances are defined as execution contexts. Each execution context responds to a functional need that has a specific language. Execution contexts provide an efficient solution to filter different services of a business process. In our case, we divide the context into three categories: service context, environment context, and customer context.
- **QoS:** A variety of quality attributes are defined in our ontology during ERP specification, design, and execution. QoS attributes are designed and built in such a way that every service is evaluated in perfect conditions and in a real environment.

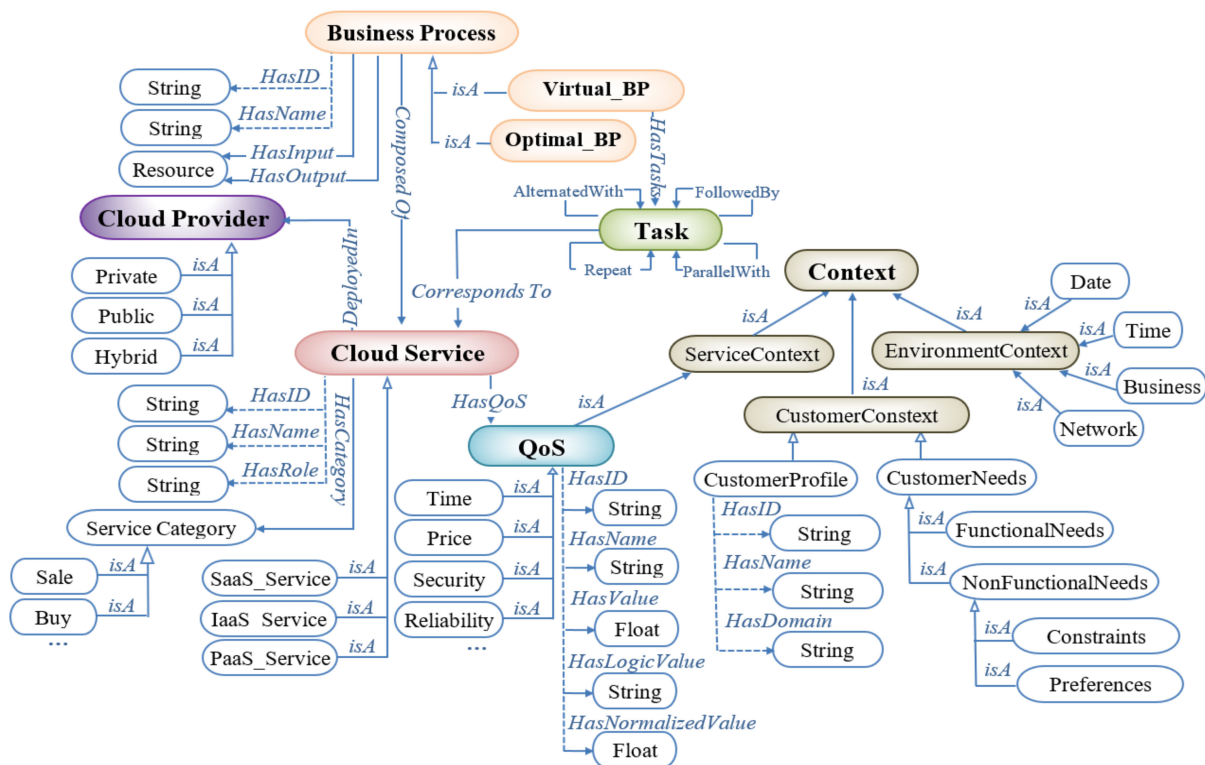


Figure 3. CxQSCloudSERP Ontology model.

4.3. CxQSIoT-Fog-CloudSERP

One of the main benefits of extending CxQSCloudSERP is that the business process can be made more flexible and interoperable through richer semantic service information of IoT field and Fog capabilities. It therefore becomes necessary to define entities related to IoT-based, Fog, and Cloud business processes, as illustrated in Figure 4.

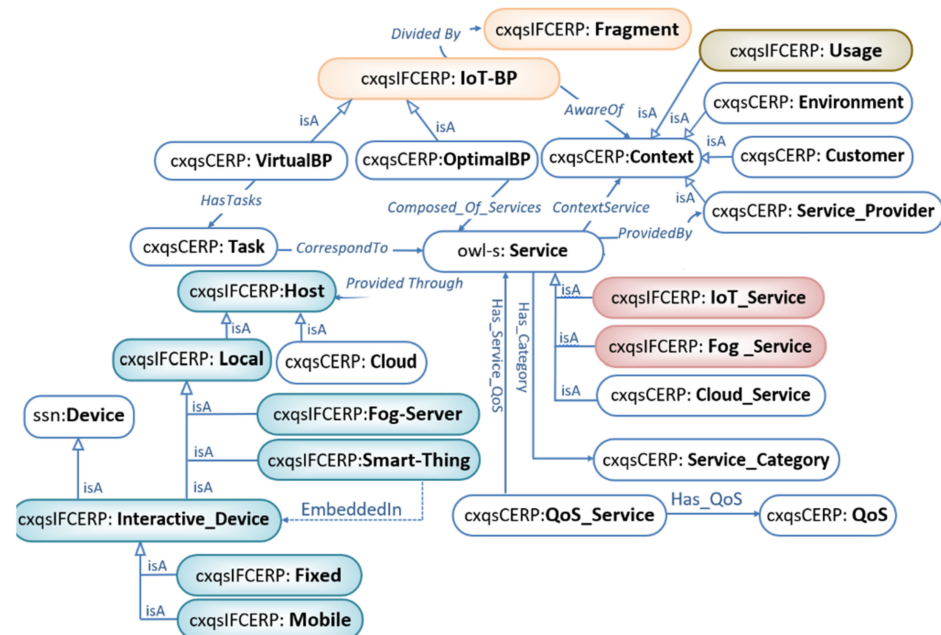


Figure 4. CxQSIoT-Fog-CloudSERP ontology model.

As illustrated in Figure 5, the service is deployed on top of the smart devices (sensors or actuators) or Fog servers as local hosts or the Cloud. Quality services are selected and

In this model, a *Service* is divided into three types: context-aware IoT services (CxIS-service), Fog, and Cloud services which have many inputs and outputs. Inputs/Outputs can be further divided into two types, data flows defined by a “specific data type and data format”, and an event flow handled by the action of a service. The service is described by an identifier, name, Uri, role, and category, and it is deployed on various hosts (IoT device, Fog server, or Cloud).

A service is defined by QoS properties such as reliability, availability, and security and a set of context quality properties such as energy. *IoT-Service* may be either *CxIEventListenerService* and *CxIActuatingService* to define the data collector and event listener and service’s actions, respectively.

4.3.2. Modeling IoT-Based Business Process

Figure 7 presents a semantic model of context-aware intelligent services. In this model, an IoT-based business process (*IoT-BP*) may be modeled to provide automation facilities using sensors and actuators. Several types of sensors and actuators with different sensing and actuating capabilities may be included in IoT-based business processes. We define a new concept of *IoT-BP* as an extension of business processes as shown in Figure 7. At a minimum, *IoT-BP* consists of tasks that may be either *Cloud Service*, *IoT Service*, or *Fog Service*. A customer (enterprise) has some needs and preferences, such as high availability, low latency, better automation, and good flexibility. To do this, the Service provider and the customer agree on high-quality constraints and resource energy saving that define the services provided to the customer under rigorous conditions within the service and business process provided. After that, quality IoT-based services are selected, integrated into a business process, and deployed in IoT devices or Fog while reducing energy consumption.

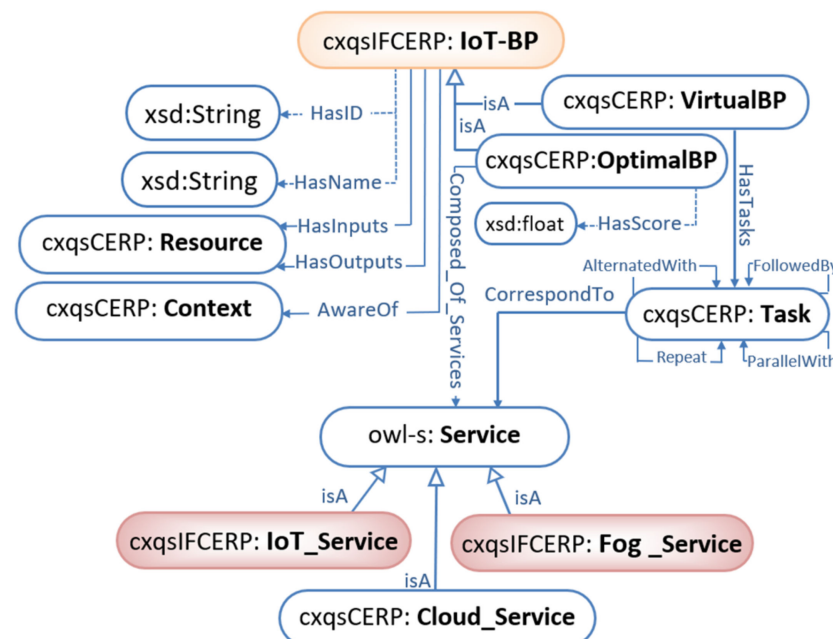


Figure 7. A semantic model of context-aware IoT business process.

4.3.3. Modeling Context

We extend the *Context* concept of CxQSCloudSERP with usage context to manage and save energy by better selecting the most appropriate services when producing and adapting concrete business processes. As illustrated in Figure 8, we introduce new concepts to *Usage context*: *service dependencies*, *service accessibility*, and *energy consumption*. We define accessibility through time constraints, space constraints, and space-time constraints. High levels of accessibility are a result of excellent services, which considers reliable and high

service qualities. Energy consumption describes the average energy rate consumed when deploying services on top of IoT devices, Fog servers, or Cloud.



Figure 8. Usage context for smart and Cloud services.

5. Problem Formulation

Our work aims to come up with an intelligent multi-agent cooperative strategy to generate and manage IoT business processes. It aims to improve system performance and optimize energy when using limited smart devices. The proposed approach must describe the required and consumed energy of IoT services during the decision-making process. That will help the system to measure the energy consumption at the run-time and thus react rapidly to these changes.

5.1. Description of Physical Servers

- We consider three levels of hierarchical infrastructure $\mathcal{I}$, consisting of $\mathcal{N}$ physical nodes. The IoT layer (*level 0*) regroups a set of $\mathcal{T}$ smart devices, sensors, actuators, and mobile devices. The fog computing layer (*level 1*) is a set of $\mathcal{F}$ Fog servers and gateways.
- The Cloud layer (*level 2*) consists of a set of $\mathcal{C}$ large-scale data centers.
- The physical infrastructure consists of a set of nodes (or servers) $\mathcal{N}$. Each node $N \in [1..\mathcal{T}]$ has the following features:
 - Each node N has a current workload $W_N^{current}$.
 - $Capacity_N$ denotes the energy capacity of the node N .
 - If N is a fixed node and continuously powered then its $Capacity_N = \infty$.

5.2. Description of Business Process

- We consider *VBP* a virtual business process consisting of m tasks $\mathcal{T} = \{t_1, t_2, \dots, t_m\}$ described using the proposed ontology.
- Each concrete composite service *CCS* has m services S . Each *CCS* is composition of atomic or composite services from all providers (IoT, Fog, and Cloud) that we have to select and compose according to a customer's constraints and preferences.
- Each service S_i has a set of QoS attributes qos_i^j .
- Each service S_i is defined by consumed energy S_i^{Energy} .

5.3. Problem Description

One problem is finding out an optimal composite service of ERP on IoT-Fog-Cloud infrastructure for minimizing energy consumption while optimizing customer QoS (i.e., response time, reliability, and security). To solve the problem, we use a three stages approach. First, we generate the virtual composite service (VCS), which includes all needed functional business tasks. Secondly, a multi-agent intelligence research algorithm gives all the possible composite service chains providing VCS. The optimal composite service solution is based on multi-objective QoS and energy-saving functions. Finally, we adapt the current composite service for incoming new customer's needs and/or context changes, including device power consumption. The problem is formalized as follows:

- The required workload $W_N^{required}$ is the total energy to be consumed while S_i is invoked on node N .

$$W_N^{required} = W_N^{current} + S_i^{Energy} \quad (4)$$

- The consumption rate R_N^i of service S_i on node N is defined as follows:

$$R_N^i = \begin{cases} \frac{Capacity_N}{Capacity_N - W_N^{required}} & \text{if } Capacity_N > W_N^{required} \\ \infty & \text{otherwise} \end{cases} \quad (5)$$

where f_{energy}^i is considered a negative attribute that will be minimized and the service S_i on a node, N is considered infeasible when $R_N^i = \infty$.

- The energy function of the CCS: $f_{energy}(CCS) = agg(f_{energy}^i)$, is calculated according to Table 3 where i varies from 1 to m .

$$f_{energy}^i = S_i^{Energy} + R_N^i \quad (6)$$

- Each *CCS* is evaluated by combining the normalized QoS values of IoT, Fog, and Cloud services via Table 3. The total QoS are normalized by the following equations:

$$q_j'(CCS) = \begin{cases} \frac{agg(q_j) - agg(q_j^{\min})}{agg(q_j^{\max}) - agg(q_j^{\min})} & \text{If } agg(q_j^{\max}) \neq agg(q_j^{\min}) \\ 1 & \text{Otherwise} \end{cases} \quad (7)$$

$$q_j'(CCS) = \begin{cases} \frac{agg(q_j^{\max}) - agg(q_j)}{agg(q_j^{\max}) - agg(q_j^{\min})} & \text{If } agg(q_j^{\max}) \neq agg(q_j^{\min}) \\ 1 & \text{Otherwise} \end{cases} \quad (8)$$

$agg(q_j^{\max})$: The highest combined score of the j^{th} QoS criterion of CCS,

$agg(q_j^{\min})$: The lowest combined score of the j^{th} QoS criterion of CCS.

$agg(q_j)$: The combined score of the j^{th} QoS criterion of CCS.

Table 3. QoS and Energy Process Metrics.

QoS Dimensions	Iterative Flow	Parallel Flow	Choice Flow	Repeat Flow
Time	$\sum_{j=1}^n Time_j$	$Max_{j=1}^n Time_j$	$Max_{j=1}^n Time_j$	$Time_j * k$
Security	$Min_{j=1}^n Sec_j$	$Min_{j=1}^n Sec_j$	$Min_{j=1}^n Sec_j$	Sec_k
Availability	$\prod_{j=1}^n Av_j$	$\prod_{j=1}^n Av_j$	$Min_{j=1}^n Av_j$	$(Av_j)^k$
Cost	$\sum_{j=1}^n Cost_j$	$\sum_{j=1}^n Cost_j$	$\sum_{j=1}^n Cost_j$	$Cost_j \times k$
Energy	$\sum_{j=1}^n Energy_j$	$\sum_{j=1}^n Energy_j$	$\sum_{j=1}^n Energy_j$	$Energy_j \times k$

We define multi-objective functions (f_{energy}, f_{QoS}) to minimize the total energy and maximize the total quality of service of the CSS as follows:

$$\text{Find CCS with } \begin{cases} \min f_{energy}(CCS) \\ \max f_{QoS}(CCS) \end{cases} \quad (9)$$

where $f_{energy}(CCS)$ and $f_{QoS}(CCS)$ are the energy aggregated function of the CCS and QoS aggregated function respectively.

And $f_{QoS}(CCS) = \sum_{j=1}^{nb} w_i \times qos_{ccs}^j$ where: the dimension of QoS.

Equation (10) defines the final score by combining the energy and QoS of each CCS:

$$score(CCS) = \sum_{j=1}^{nb} w_i \times qos_{ccs}^j + w_{energy} \times energy_{ccs} \quad (10)$$

6. The Framework

To develop an efficient service selection and composition approach that can provide minimum energy costs and optimal QoS for customers, we propose a new adaptive and dynamic system based on NSGA-II for multi-objective optimization and cooperative agents with more capacities like intelligence, scalability, and efficiency. Our goal is to ensure coordination between different agents of the IoT-Fog-Cloud ERP system. Such a method discovers relevant services based on customer needs while selecting services with high QoS and optimal energy cost, and provides a continuously optimized composite service based on context changes and/or customer preference changes. To achieve this goal, we use CxQSIoTFogCloudSERP ontology, the multi-agent approach, and multi-objective optimization NSGA-II. Using both QoS and energy saving and modified NSGA-II multi-agents improves the solution accuracy, minimizing the execution time and energy consumption, and it is effectively used in remote controlling smart services ERP appliances, monitoring the state of the products with an exact angle measurement.

We present a new multi-objective function based on both QoS and energy consumption for selecting the optimal personalized composite service and saving energy with satisfactory accuracy and minimal execution time. To exploit this new multi-objective function, we propose two modes: dynamic and adaptive. The dynamic mode consists of three service management layers, IoT, Fog, and Cloud. The adaptive mode allows adaption of some business process parts according to different context changes and customer preferences to integrate IoT and Fog services and compose them with Cloud services. Such dynamic and adaptive modes are believed to efficient and economical, to be (potentially) utilized in several virtual services and their corresponding concrete services. They are stored, respectively, in distributed virtual service registries and concrete service registries of smart interactive devices, Fog servers, and Cloud. We leveraged the advantages of ontology-based agents by forming a well-hierarchical service management model to reduce the discovery time of appropriate services and to target potential service providers.

The work presented here focuses on a typical scenario, production, and transport process in a wheel production domain, as described in Section 2, and on different agents

to generate an optimized smart ERP business process, as illustrated in Figures 1 and 2. In this work, we have used the multi-agent platform Jade [40] for cooperative IoT, Fog, and Cloud services composition. The CxQSIoTFogCloudSERP ontology is implemented using Protégé [41] to match customer needs to virtual services using inference rules.

6.1. General Architecture

Our system architecture of the SCwA-CCS (Semantic-Based Context-aWare Agents for the Composition of IoT with Fog and Cloud Services) is a semantic multi-agent architecture that is composed of the following agents (as illustrated in Figure 9): Customer, Semantic Context Broker, Context Supervisor, Knowledge, and SCwA-CCS Agent.

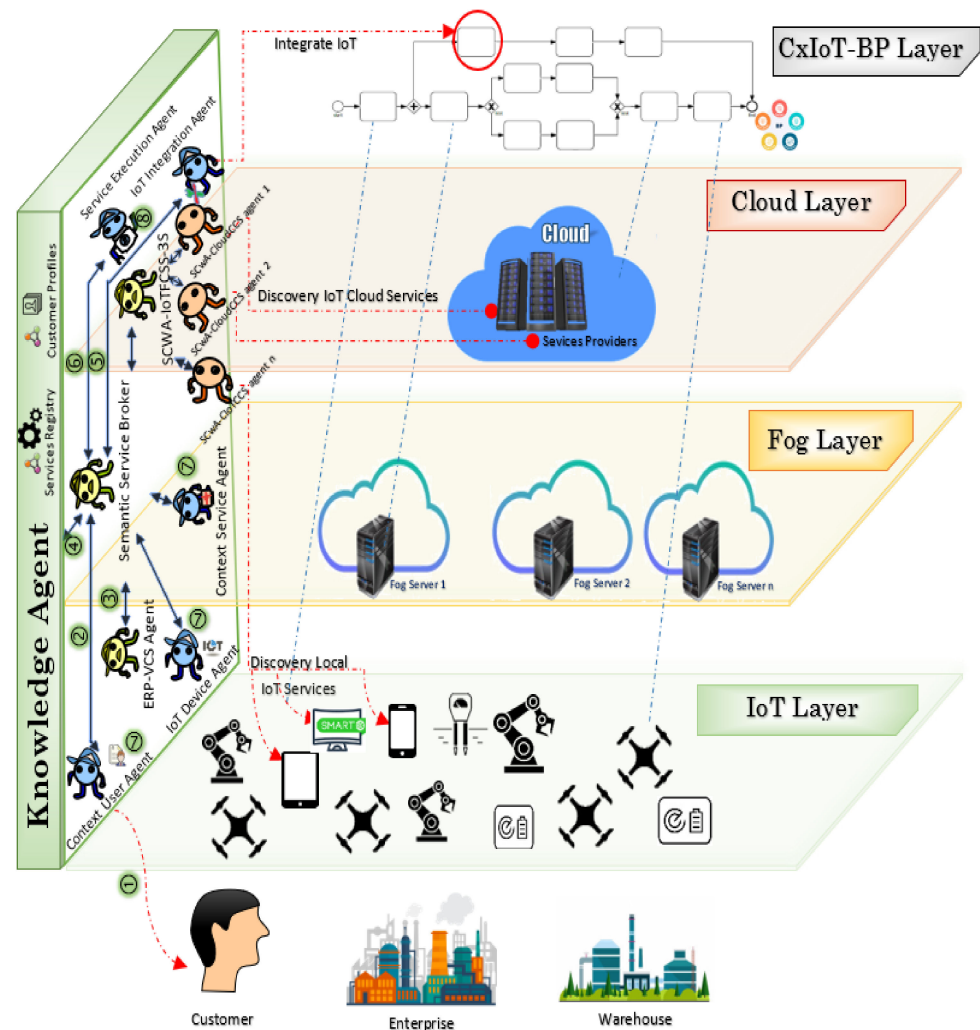


Figure 9. An overview of the proposed approach.

- **Customer:** The enterprise that wants a personalized IoT-ERP, which communicates with the system through GUI developed in Java.
- **Semantic Context Broker:** Acts as a semantic mediator between a customer and different service providers, with six kinds of agents interacting and collaborating to realize and manage concrete composition processes. The context broker is composed of the following four main agents:
 - a. **SCwA-CCS Agent:** Allows the customer to specify their requirements and receive their responses. This agent calls the following agents:
 - *Virtual Business Process Planner:* Agent generates virtual business process. It is implemented using JESS as an inference engine and SWRL as an inference rule.

- *SCwA-IFCCS-3S*: Agent discovers IoT, fog, and Cloud services in terms of customer preferences and energy constraints.
 - *ERP-CCS-3S Agent*: Finds the optimal concrete composite service.
 - *Context Supervisor Agent*: Monitors context changes (*customer context, service context, IoT devices context, environment context, and usage context*).
- b. **ERP-CCS-3S Agent**: Generates an optimal business process in the form of a Concrete Composite Service (CCS) based on QoS and energy constraints, costumer's preferences, and description of IoT, Fog, and Cloud services.
- c. **SCwA-IFCCS-3S Agent**: Discovers local and Cloud services with respect to a customer's needs and energy constraints, and generates possible composite services. It calls *ERP-ICS* and *ERP-FCS Agent* for discovering IoT Fog services, respectively, and calls *ERP-CCS Agent* for discovering Cloud services.
- **Context Supervisor Agent**: Monitors customer's context in real-time by reading the data from the sensors. It also enables service context monitoring (network connection, Cloud server load rate, available service provider, QoS, etc.) and the IoT context (availability, energy capacity, etc.) to provide the desired services for the customer.
- **Knowledge Agent**: Manages Cloud services and customer profiles that are defined by the user. It also ensures continuous updates of the ontology model.
- **Service Executor Agent**: Calls composite services and execution context management. It also takes over the management of IoT, Fog, and Cloud services.
- **The Services and Client Repository**: Made up of five sets of data descriptions: data from the sensor network, definition of IoT services for all smart interactive devices, definition of Fog service providers, and Cloud service providers, and a description of customer profiles using our ontology.

6.2. Functional Model

Initially, a customer sends their request, which consists of a customized ERP business process in terms of functional and QoS needs to a *Semantic Context Broker Agent* who plays the role of a mediator. Here, the *Semantic Context Broker Agent* is a core agent that calls the *SCwA-IFCCS-3S Agent* that is responsible for launching several *ERP-ICS* and *ERP-FCS* agents, which can migrate to IoT devices and Fog servers to look for the requested services. These agents search for local IoT and Fog services with the help of a knowledge-agent able to make partial or full services compositions of requested ERP business processes.

The *ERP-CCS-3S Agent*, from all the possible composite services, selects the best solution (single composite service or a set of composite services) that optimizes service qualities and energy cost. The *ERP-CCS-3S Agent* uses a customer's preferences to select an optimal composite service. Once the optimal composite service has been selected and the ERP business process has been deployed, the deployment results are sent back to the customer. If the *SCwA-IFCCS Agent* does not find any local services that match the customer's needs, it transmits the customer's query to the *ERP-CCS Agent* to discover and compose services involved in the business process on Cloud servers.

The Context Supervisor Agent is responsible for handling the problems encountered during the execution of the various services (such as changes of user's needs and/or customer preferences) and incoming new customer's needs in order to continuously provide an optimal composite service.

- If it is a simple service update (replacing Cloud service by another IoT service), a *SCwA-IFCCS-3S Agent* will replace it with another equivalent IoT/Fog service based on the Knowledge agent.
- If it is about more than one Cloud service, he calls *ERP-ICS* and *ERP-FCS* agents. It moves towards IoT devices and Fog servers, and then sends discovered services to ensure adaptation of business processes. The discovered services will be transmitted to the *ERP-CCS-3S Agent*. The *ERP-CCS-3S Agent* composes these services and selects the optimal composite service.

6.3. Detailed Algorithms

The proposed SCwA-IFCCS-3S agent consists in dynamic and adaptive modes. The flowchart of SCwA-IFCCS-3S is illustrated in Figure 10. The algorithm for establishing an optimal composite service for IoT, Fog, and Cloud environments changes over time due to context changes (processing load, change of customer's needs, low latency, insufficient energy of devices, and other factors). As a result, it is critical to adapt the actual composite service while taking into account a variety of management elements. The algorithm consists mainly of three main stages: *virtual business process generating*, *parallel multi-objective concrete composite service optimization*, and *adaptation of concrete composite service*. The stages involved in the implementation of SCwA-IFCCS-3S is given below:

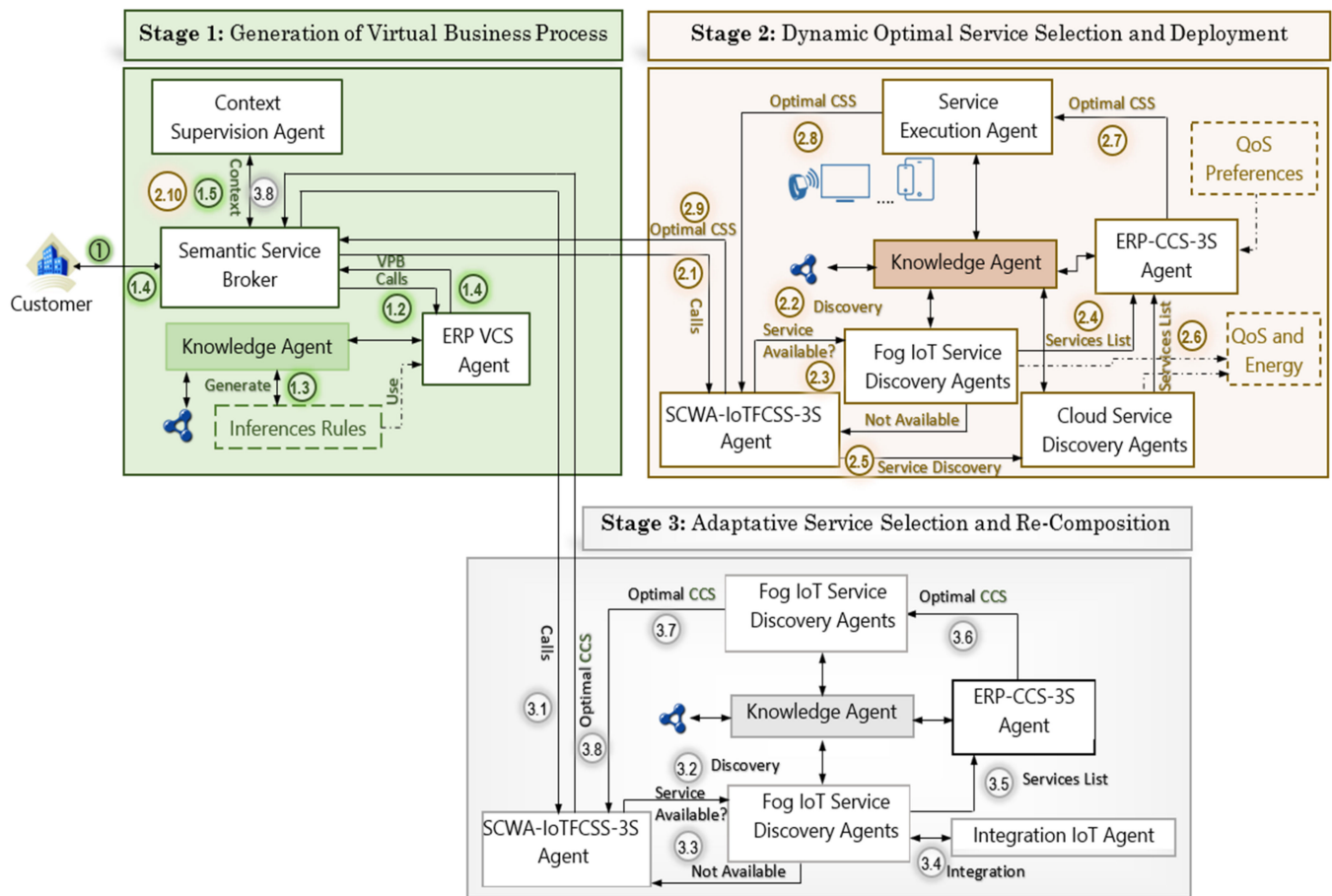


Figure 10. SCwA-IFCCS-3S functional model.

Stage 1. Generate virtual business process: In this stage, the research before the algorithm is used to systematically generate virtual IoT-BP according to the client's functional needs. Virtual services (e.g., tasks) are generated and optimized based on semantic relationships between VSs [28] and the reputation of each virtual service [42]. We automate this process to ensure dynamic changes in functional needs of the customer and his context.

Stage 2. Find local optimal solutions: We focus on dynamic multi-objective service selection and composition problems, we target each service provider, and the discovery agent closes in towards it. Each service provider, whether IoT, Fog, or Cloud, is modeled by ERP-ICS, ERP-FCS, and ERP-CCS agents, respectively, that cooperate to share local services. Therefore, each service provider has a set of services S_i which is a subset of the set S . The service provider agents create a set of NSGA-II agents to select the best Concrete Composite Service (CCS). It initializes them by a subset S_i and the energy cost and aggregated QoS to its composite service. Simultaneous use of penalties for energy consumption and penalty for QoS violation may hamper the reliability of service selection as well. The NSGA-II agent

finishes when finding the local optimal composite service for a set of service providers, thus *ERP-ICS* and *ERP-FCS* end by sending a set of optimal composite services to *ERP-CCS-3S* Agent. The collaborative mechanism between different agents via messages allows us to find the best optimal composite service.

- **Step 1.** Each individual (composite service) is encoded by an array of n concrete services, called chromosome.
- **Step 2. Evaluation:** Each individual is evaluated using the aggregated fitness function.
- **Step 3. Selection:** To pick the optimal individual, tournament selection is used. In the proposed approach, the relationship of non-domination has been adapted by adding a constraint violation criterion of selection to respect the customer's constraints and energy consumption. The new CSS violation degree (deg_{v_CCS}) is evaluated to penalize IoT, Fog, and Cloud services that do not agree with both QoS and energy constraints. This degree is calculated as follows:

$$deg_{v_CCS} = w_{QoS} \times deg_{v_CCS}^{QoS} + w_{energy} \times deg_{v_CCS}^{energy} \quad (11)$$

$$\text{With } deg_{v_CCS}^{QoS} = \begin{cases} C_{qos} - qos & \text{If } C_{qos} > qos \\ 0 & \text{Otherwise} \end{cases} \text{ and,} \quad (12)$$

$$deg_{v_CCS}^{energy} = \begin{cases} C_{energy} - energy & \text{If } C_{energy} > energy \\ 0 & \text{Otherwise} \end{cases}.$$

- Here w_{QoS} and w_{energy} denote weights of the QoS and energy, respectively, which is increased when the use of IoT services is greater than the Fog and Cloud services;
- qos : Aggregated value of QoS.
- $energy$: Aggregated value of energy.
- C_{QoS} : The QoS constraint.
- C_{energy} : The energy constraint.
- For every CSS and its new dominance relation, we select the CSS based on the lowest dominance value.
- **Step 4. Merge and evaluate local optimal business process.** Merging and comparing the fitness value of local BPs of each agent with the other BPs of another agent. Based on largest fitness value, the global composite service is selected.

Stage 3. Adaptive Strategy for re-composition of services: Generally, the proposed approach is designed to be able to perform in two modes: Dynamic or Adaptive cases when context changes are present or not. We compute the adaptation cost in terms of energy and QoS difference between the current optimal CCS_t and new CCS_{t+1} , with desired services to be added or deleted in order to manage context changes. More precisely, the list of services to be added is S_{add} and the list of services to be deleted is S_{del} . The next CCS_{t+1} contains $\overline{CCS_t \cup S_{add}} \cap S_{del}$. So, the multi-objective function of Equation (13) will include the third objective, which is the adaptation cost of newly added services and/or the cost of removed existing services in terms of energy and QoS. The new multi-objective function is defined in Equation (13).

$$\text{Find } CCS \text{ with } \begin{cases} \min f_{energy}(CCS_{t+1}) \\ \max f_{QoS}(CCS_{t+1}) \\ \min f_{cost}(CCS_{t+1} - CCS_t) \end{cases} \quad (13)$$

where $CCS_{t+1} - CCS_t$ is the difference between the current optimal CCS and new CCS according to the same VBP, since the existing optimal VBP does not change.

The detailed process is described as Algorithm 1 and illustrated in Figure 10.

Algorithm 1: Parallel NSGAI strategy for dynamic Cloud services composition**Inputs:** agent_pop_size, max_itr, populations**Outputs:** P_OP: Pareto optimal solutions**Begin**

```

1 : Generate initial agent populations as a set of CCS;
2 : P_OP  $\leftarrow \emptyset$ 
3 : Iteration  $\leftarrow 0$ ;
4 : while (Iteration  $\leq$  max_itr ) do
5 :   Fork (agent using agent_pop in populations)
6 :     Evaluate the objective functions (QoS and Energy) for each individual;
7 :     Local Pareto optimal  $\leftarrow$  Select best individuals by agent_pop;
8 :     P_OP  $\leftarrow$  P_OP  $\cup$  Local Pareto optimal
9 :     s  $\leftarrow 0$ 
10 :    while (s  $\leq$  agent_pop/2) do
11 :      (P1,P2)  $\leftarrow$  select two individuals parents of current agent_pop;
12 :      (C1,C2)  $\leftarrow$  Crossing both parents (P1,P2) to obtain two children;
13 :      agent_pop.add(C1); agent_pop.add(C2);
14 :      s  $\leftarrow$  s + 1;
15 :    end
16 :    s  $\leftarrow 0$ 
17 :    while (s  $\leq$  agent_pop/2) do
18 :      new_ind  $\leftarrow$  mutate children(ind);
19 :      agent_pop.add(new_ind);
20 :      s  $\leftarrow$  s + 1;
21 :    end
22 :  end
23 :  P_OP  $\leftarrow$  Selection of optimal solution from all local Pareto-Optimal;
24 :  Update new global Pareto optimal solutions for all agents;
25 :  Iteration++;
26 : end

```

End**Return** P_{OPTIMAL};

The pseudo-code of the semantic context broker algorithm in adaptive intelligent strategy is illustrated in Algorithm 2. At first, the semantic context broker keeps up with the context changes. It then calls the *SCwA-IFCCS* Agent, which in turn calls *ERP-ICS* and *ERP-FCS* agents for discovering services. It ends with sending the services list to the *ERP-CCS-3S* Agent. The *ERP-CCS-3S* Agent selects the optimal composite service as the optimal fragment according to the customer's preferences, and then recomposes it with the current composite service, subsequently sending it to the service deployer. Finally, the optimal composite service fragment is deployed and results are returned to customer.

Considering an existing production CCS while monitoring services as a fragment of BP (see Figure 2) and the new monitoring IoT-BP generated by the adaptive multi-agent and NSGA-II algorithm (Figure 11), the adaptation cost is the cost of adapting existing BP to IoT-BP while considering only such a fragment. In this case, we need to add three services: new sensed temperature service (S1), receive temperature service (S1), and control temperature service (S3) of final products, and remove service monitoring (S'3). The adaptation cost will then equal the energy and time cost of deploying S1 and S2.

Algorithm 2: Adaptive composite service algorithm based on Multi-agent and NSGAI**Inputs:** IoT-Business-Process, context, preferences, energy evolution**Outputs:** Optimal adaptative IoT-Business-Process**Begin**

1 : Acquisition of context features

2 : Calculate new constraints parameters

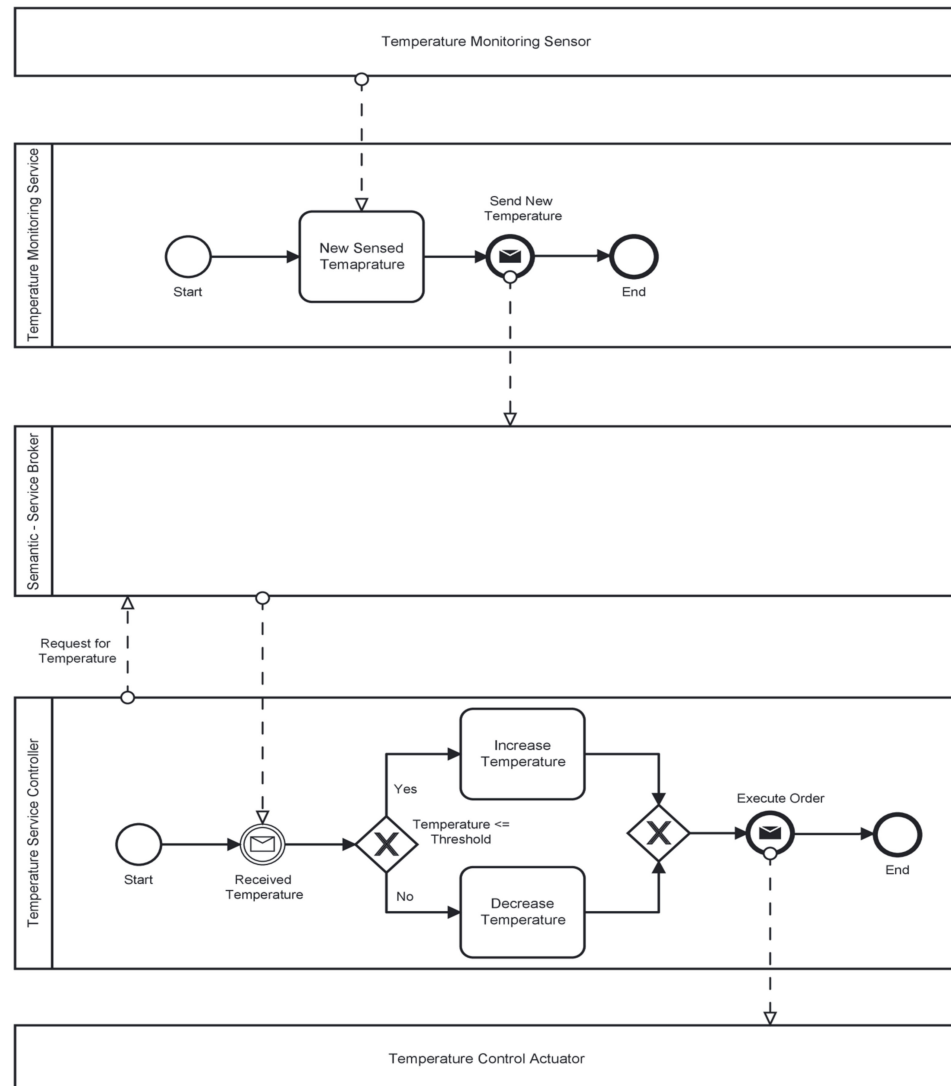
3 : Recalculate the energy capacity of IoT-Devices

4 : **For each** service s_{ij} in IoT-Devices **Do**5 : **If** energy capacity of device $d_{ij} \leq \text{energy of service } s_{ij}$ **Then**6 : Delete service s_{ij} from the available services7 : **Foreach** service s_{ik} in $\mathcal{N}$ **Do**8 : select services s_{ik} from $\mathcal{N}$ with minimal $f_{cost}(CCS_{t+1} - CCS_t)$ 9 : **End**10 : **Endif**11 : **If** number of services = 1 **Then**

12 : Use ontology to find next equivalent service

11 : **Else**

12 : Find the optimal IoT-BP with new changes according to the algorithm 1;

13 : **Endif****End****Return** Optimal IoT-BP;**Figure 11.** Dynamic semantic-based multi-agent flowchart for services composition.

7. Result and Discussion

This proposed work is implemented in Eclipse and compared with different cooperative strategies (Master/Slave, Ring, and Hybrid). The aim is to determine which strategy performed best in terms of QoS and energy for dynamic composition and adaptation of composite services. All experiments are conducted on a personal laptop that has the following configurations: 6th Generation Intel® Core™ i5-8250U processor, 128 SSD with 12GB of RAM with Eclipse, and Matlab 2020 running on Windows 10 OS.

7.1. Dataset

To explore the strategy of parallel dynamic IoT Fog and Cloud services composition for ERP via intelligent agents, a random dataset has been used to record the QoS attributes and devices configuration that are depicted in Figure 11. The dataset comprises 10 different virtual services performed by 100 different concrete IoT, Fog, and Cloud services. We consider 333 IoT services, and 667 Fog/Cloud services. Each service contains QoS values for five attributes. Each QoS is scored by the weighted sum of their normalized values, and the energy of each service is scored by its normalized value. QoS values are generated randomly to ensure that the results of our experiments are not biased by a specific dataset. The comparisons are carried out between different strategies (NSGA-II Standard, NSGA-II Ring, NSGA-II Master/Slave, and hybrid strategy) based on Equation (10) of each evaluated solution. It should be noted that each experiment is evaluated through the following parameters: Max-Generation, Population-Size, Mutation-Ratio fixed to 0.3, and Agent-Number.

The effectiveness and energy saving of our proposal has been validated and evaluated through multiple experiments on random datasets. To show how our validation works, we detail its validation procedure as follows:

1. The user specifies their preferences in terms of QoS and required functionalities.
2. All performance metrics are evaluated and recorded by different concrete configurations on three multi-agent models: ring, master/slave, and hybrid using simulations of the different number of services.
3. These simulations include results obtained from four performance metrics (violation QoS degree, normalized average execution time, average energy consumption rate, and cost) with different multi-agent models through several simulated configurations.

7.2. The Accuracy Solution's Comparison

The contribution to smart ERP is evaluated within the NSGA-II algorithm without agents and the NSGA-II algorithm using different multi-agent strategies (Ring, Master/Slave, and Hybrid). These strategies are described as follows:

- **Master/Slave strategy:** The first strategy is focused on determining optimized global solutions from local best solutions. This strategy involves the services' local discovery and solution local evaluation using slaves, while the master agent chooses the best global solution from the selected best local solutions.
- **Ring strategy:** The second strategy explores some sub-populations in parallel to find the best solution. Each agent shares its optimal solution with its neighbor.
- **Hybrid Strategy:** To explore the advantages of both previous strategies and ensure further improvement, each slave sends its optimal solution to the master agent. Each slave shares his optimal solution with his neighbor. Finally, the master agent returns the global solution.

All three strategies are involved in determining quality solutions in an acceptable execution time. The results of Equation (10) are crucial in determining the most efficient solution in terms of QoS and energy. The obtained results on NSGA-II with/without agents are tabulated in Tables 4 and 5. It is quite remarkable that the inclusion of the multi-agent approach performed by both QoS and energy violation degree reached about 97% accuracy, which is perfect compared to the strategy without agents.

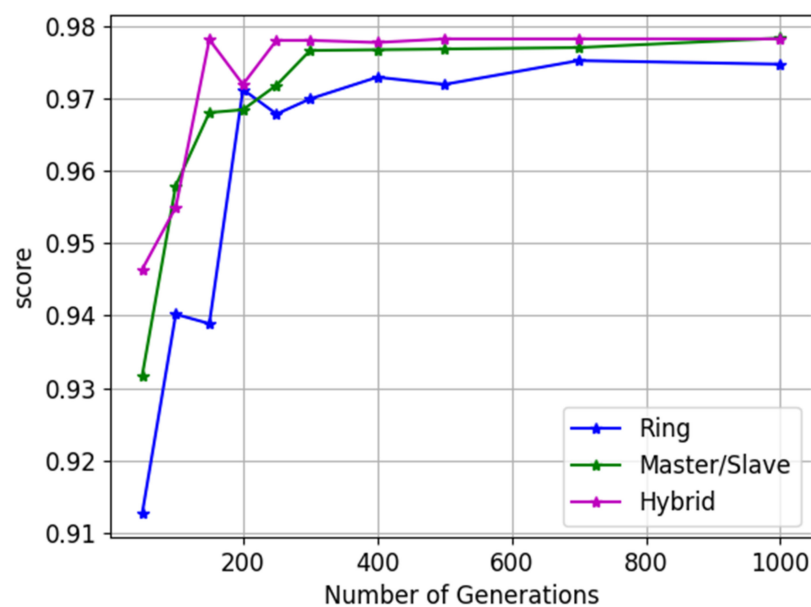
Table 4. NSGA-II without Agents in different Population Sizes and Generation Sizes.

Agents Number	Number of Generations									
	50	100	150	200	250	300	400	500	700	1000
20	0.799	0.774	0.761	0.746	0.792	0.905	0.766	0.771	0.859	0.779
40	0.774	0.811	0.849	0.834	0.845	0.820	0.851	0.880	0.847	0.782
60	0.881	0.934	0.876	0.895	0.874	0.872	0.910	0.869	0.849	0.905
80	0.931	0.875	0.915	0.91	0.977	0.941	0.912	0.941	0.977	0.905
100	0.808	0.909	0.939	0.939	0.972	0.925	0.977	0.888	0.946	0.974

Table 5. NSGA-II using different strategies Agents (Ring, Master/Slave, and Hybrid).

Agents Number	Number of Generations									
	50	100	150	200	250	300	400	500	700	1000
Ring	2	0.78	0.884	0.895	0.911	0.921	0.943	0.971	0.973	0.975
	4	0.913	0.94	0.939	0.971	0.967	0.97	0.973	0.972	0.975
	6	0.912	0.941	0.949	0.969	0.971	0.972	0.972	0.973	0.975
M/S	2	0.796	0.821	0.811	0.856	0.861	0.849	0.851	0.859	0.964
	4	0.931	0.957	0.968	0.969	0.972	0.974	0.976	0.976	0.977
	6	0.937	0.945	0.966	0.973	0.975	0.977	0.977	0.977	0.977
Hybrid	2	0.856	0.895	0.932	0.868	0.948	0.975	0.977	0.977	0.974
	4	0.946	0.957	0.977	0.972	0.977	0.977	0.976	0.977	0.977
	6	0.961	0.945	0.977	0.977	0.975	0.977	0.977	0.977	0.977

Figure 12 depicts the scores of different strategies. It was evaluated using Equation (10) to measure how accurately the solution was made by each strategy, with a higher score indicating better performance. It appears that in general, the hybrid strategy has the highest scores across most of the cases, indicating that is the most accurate model. However, it is important to note that the results of the hybrid strategy vary depending on the number of generations. Some solutions are better determined by different strategies.

**Figure 12.** Scores of different strategy agents (Ring, Master/Slave, and Hybrid).

7.3. Performance Comparison

Figure 13 shows the execution time comparison for different strategies and different numbers of agents. Execution time is the measure of how the strategy performs the service dataset, where a minimum execution time means a perfect performance. From the results,

it appears that the hybrid strategy has the minimum execution time. A practical increase of agents increased the volume of exchanged messages.

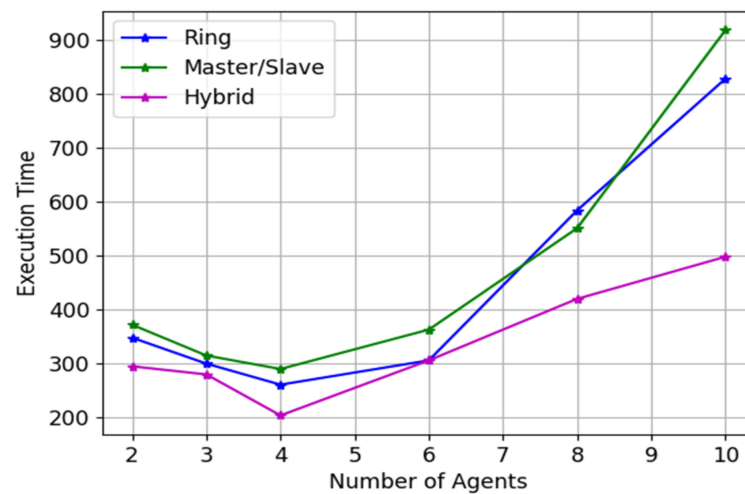


Figure 13. Execution time of different strategy agents (Ring, Master/Slave, and Hybrid).

7.4. The Energy Consumption in Dynamic and Adaptive Cases

For the dynamic case, we compared the proposed semantic-based parallel NSGA-II on three different multi-agent strategies (Ring, Master/Slave, and Hybrid) in terms of energy consumption and QoS. Figure 14 shows the results obtained by the two objective functions f_{energy} and f_{QoS} using the three agents' strategies. From the obtained results, we notice that the proposed dynamic agent-based NSGAII with hybrid strategy is better than other strategies in terms of optimized energy consumption and QoS scores.

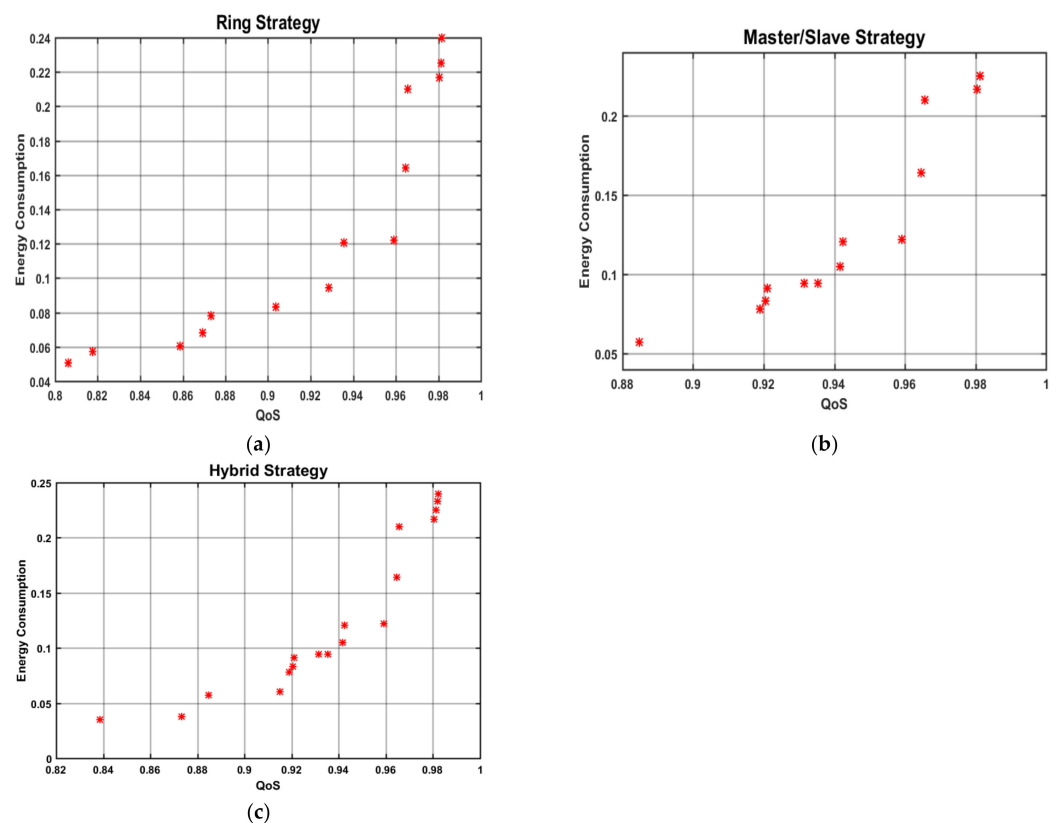


Figure 14. Pareto Optimal for Qos score and energy consumption in different strategies agents (a) Ring; (b) Master/Slave; (c) Hybrid.

For the adaptive case, we use three objective functions f_{energy} , f_{QoS} and f_{cost} on incremental changes of workload, energy device capacity, and changes in customer needs/preferences. The strategy adopted is hybrid because the results in Figures 12–14 show that it is the best strategy compared with Ring and Master/Slave strategies. From the obtained results of Table 6, we can observe that the adaptation cost is correlated with the number of rejected services (infeasible services). This is due to the number of deleted services from the current CCS that should be replaced with other services having enough energy capacity. On the other hand, when the needs/preferences of the customer change, the new CCS replaces the majority or the totality of its services, so the adaptation cost increment changes considerably.

Table 6. Different adaptations of CCS with context changes.

CCS	f_{energy}		f_{QoS}		f_{cost}	Number of Unfeasible
	Min	Max	Min	Max		
CCS_t	0.0473	0.2481	0.8386	0.9827	0	6
CCS_{t+1}	0.0521	0.2672	0.8290	0.9844	0.22	29
CCS_{t+2}	0.0677	0.2581	0.8306	0.9879	0.37	41
CCS_{t+3}	0.0572	0.2509	0.8424	0.9883	1	73
CCS_{t+4}	0.0653	0.2618	0.8532	0.9811	1	112

Table 6 shows the different adaptations of CCS with context changes. The CCS_{t+1} is the composite service when the global workload of nodes changes. The CCS_{t+2} is the composite service adapted to new device's capacity after a lapse of execution time. CCS_{t+3} is the adapted composite service to the new needs of the customer. CCS_{t+4} is the new composite service when the needs of the customer change and the capacity of many devices are incapable of executing their elementary service (infeasible services). In the case of CCS_{t+3} and CCS_{t+4} , f_{cost} equal to 1 because all services of the current composite service are replaced.

7.5. Discussion

The above results show that the hybrid strategy is faster and more precise than Master/Slave and Ring strategy. This is due to the benefit of the advantages of both strategies while avoiding some of their disadvantages. In the Master-Slave strategy, it is hard to predict the behavior of the agents or the waiting time by the master to get the Pareto from his slave. In the ring strategy, the failure of one agent will break the chain and does not invoke the other Agent, which means breaking the whole topology. In the adaptive mode, practical increase of replaced services increased the cost function. In addition, simultaneous inclusion of agent-based approaches along with the ontology approaches and inheritance context domain adaptation techniques can be a promising remedy to the changes in the energy consumption problem.

8. Conclusions

This paper presents an adaptive and dynamic solution in IoT, fog, and Cloud environments. Semantic multi-agent intelligence features have been exploited to quickly compose IoT-Fog-Cloud services concerned by QoS and energy constraints, which help us to obtain a personalized ERP and highlight advantages regarding related works. The approach consists of guiding the dynamic services composition process by generating IoT virtual business processes that meet the customer's functional needs based on the ontology model and inference rules. A multi-agent approach using a three-stage algorithm has been proposed for the optimization of the IoT business process (IoT-BP) in terms of QoS and energy. Simultaneous inclusion of agents with QoS constraints' violation and energy device capacity can be a promising remedy to changes in customer needs and preferences. Another promising factor of this proposed article is the adapted IoT-BP to context changes and energy degradation on IoT devices. The Fog is a layer used in our work to select local services for reducing communications to the Cloud and performing

global IoT-BP. An executable prototype is developed using the Protégé and Jade tools, subsequently displaying the overall optimal smart business process. Finally, we conclude that the proposed approach gives satisfactory performance on a random dataset involving multiple context change scenarios. It shows remarkable effectiveness through an intelligent adaptation mechanism for energy saving on IoT devices. The works presented use the deterministic measurement of QoS. In future work, an indeterministic measurement of QoS would be considered, as it has significant potential to improve the accuracy of smart factories, smart transportation systems, and smart production systems.

Author Contributions: Conceptualization, H.R. and A.A.; methodology, H.R. and A.A.; software, H.R. validation, H.R. and A.A.; formal analysis, H.R. and A.A.; investigation, H.R. and A.A.; resources, H.R. and A.A.; data curation, H.R. and A.A.; writing—original draft preparation, H.R.; writing—review and editing, A.A.; visualization, H.R.; supervision, A.A.; project administration, H.R. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is available on request due to restrictions.

Acknowledgments: The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Olsen, K.A.; Sætre, P. ERP for SMEs—is proprietary software an alternative? *Bus. Process Manag. J.* **2017**, *13*, 379–389. [\[CrossRef\]](#)
- Yousefpour, A.; Fung, C.; Nguyen, T.; Kadiyala, K.; Jalali, F.; Niakanlahiji, A.; Jue, J.P. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *J. Syst. Archit.* **2019**, *98*, 289–330. [\[CrossRef\]](#)
- Andročec, D.; Picek, R. Cloud ERP API Ontology. In Proceedings of the IEEE International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 20 July 2022; pp. 1–5.
- Cornejo-Lupa, M.A.; Cardinale, Y.; Ticona-Herrera, R.; Barrios-Aranibar, D.; Andrade, M.; Diaz-Amado, J. OntoSLAM: An Ontology for Representing Location and Simultaneous Mapping Information for Autonomous Robots. *Robotics* **2021**, *10*, 125. [\[CrossRef\]](#)
- Cornejo-Lupa, M.A.; Ticona-Herrera, R.P.; Cardinale, Y.; Barrios-Aranibar, D. A survey of ontologies for simultaneous localization and mapping in mobile robots. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–26. [\[CrossRef\]](#)
- Fiorini, S.R.; Carbonera, J.L.; Gonçalves, P.; Jorge, V.A.; Rey, V.F.; Heidegger, T.; Prestes, E. Extensions to the core ontology for robotics and automation. *Robot. Comput.-Integr. Manuf.* **2015**, *33*, 3–11. [\[CrossRef\]](#)
- Olszewska, J.I.; Barreto, M.; Bermejo-Alonso, J.; Carbonera, J.; Chibani, A.; Fiorini, S.; Li, H. Ontology for autonomous robotics. In Proceedings of the 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Lisbon, Portugal, 28–31 August 2017; pp. 189–194.
- Beetz, M.; Beßler, D.; Haidu, A.; Pomarlan, M.; Bozcuoğlu, A.K.; Bartels, G. Know rob 2.0—A 2nd generation knowledge processing framework for cognition-enabled robotic agents. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 20–25 May 2018; pp. 512–519.
- Katti, B.; Plociennik, C.; Schweitzer, M. A jumpstart framework for semantically enhanced opc-ua. *KI Künstl. Intell.* **2019**, *33*, 131–140. [\[CrossRef\]](#)
- Pop, C.B.; Vlad, M.; Chifu, V.R.; Salomie, I.; Dinsoreanu, M. A tabu search optimization approach for semantic web service composition. In Proceedings of the 10th International Symposium on Parallel and Distributed Computing, Cluj-Napoca, Romania, 6–8 July 2011; pp. 274–277.
- Sasikaladevi, N.; Arockiam, L. Genetic approach for service selection problem in composite web service. *Int. J. Comput. Appl.* **2012**, *44*, 22–29. [\[CrossRef\]](#)
- Long, J.; Gui, W. An environment-aware particle swarm optimization algorithm for services composition. In Proceedings of the International Conference on Computational Intelligence and Software Engineering, Wuhan, China, 11–13 December 2009; pp. 1–4.
- Bouzary, H.; Frank Chen, F. A hybrid grey wolf optimizer algorithm with evolutionary operators for optimal QoS-aware service composition and optimal selection in cloud manufacturing. *Int. J. Adv. Manuf. Technol.* **2019**, *101*, 2771–2784. [\[CrossRef\]](#)
- Yu, Q.; Chen, L.; Li, B. Ant colony optimization applied to web service compositions in cloud computing. *Comput. Electr. Eng.* **2015**, *41*, 18–27. [\[CrossRef\]](#)

15. Jula, A.; Othman, Z.; Sundararajan, E. A hybrid imperialist competitive-gravitational attraction search algorithm to optimize cloud service composition. In Proceedings of the 2013 IEEE Workshop on Memetic Computing (MC), Singapore, 16–19 April 2013; pp. 37–43.
16. Li, C.; Li, J.; Chen, H. A meta-heuristic-based approach for QoS-aware service composition. *IEEE Access* **2020**, *8*, 69579–69592. [\[CrossRef\]](#)
17. Yao, Y.; Chen, H. Qos-aware service composition using NSGA-II. In Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, Seoul, Republic of Korea, 24–26 November 2009; pp. 358–366.
18. Li, L.; Cheng, P.; Ou, L.; Zhang, Z. Applying multi-objective evolutionary algorithms to QoS-aware web service composition. In Proceedings of the International Conference on Advanced Data Mining and Applications, Chongqing, China, 19–21 November 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 270–281.
19. Wada, H.; Champrasert, P.; Suzuki, J.; Oba, K. Multi-objective optimization of SLA-aware service composition. In Proceedings of the 2008 IEEE Congress on Services-Part I, Honolulu, HI, USA, 6–11 July 2008; pp. 368–375.
20. Taboada, H.A.; Espiritu, J.F.; Coit, D.W. MOMS-GA: A multi-objective multi-state genetic algorithm for system reliability optimization design problems. *IEEE Trans. Reliab.* **2008**, *57*, 182–191. [\[CrossRef\]](#)
21. Sadeghram, S.; Ma, H.; Chen, G. A Novel Repair-Based Multi-objective Algorithm for QoS-Constrained Distributed Data-Intensive Web Service Composition. In Proceedings of the International Conference on Web Information Systems Engineering, Amsterdam, The Netherlands, 20–24 October 2020; Springer: Cham, Switzerland, 2020; pp. 489–502.
22. Chen, C.S.; Liang, W.Y.; Hsu, H.Y. A cloud computing platform for ERP applications. *Appl. Soft Comput.* **2015**, *27*, 127–136. [\[CrossRef\]](#)
23. Bolu, A.; Korçak, Ö. Adaptive task planning for multi-robot smart warehouse. *IEEE Access* **2021**, *9*, 27346–27358. [\[CrossRef\]](#)
24. Poryazov, S.; Andonov, V.; Saranova, E.; Atanassov, K. Two Approaches to the Traffic Quality Intuitionistic Fuzzy Estimation of Service Compositions. *Mathematics* **2022**, *10*, 4439. [\[CrossRef\]](#)
25. Reffad, H.; Alti, A. New approach for optimal semantic-based context-aware cloud service composition for ERP. *New Gener. Comput.* **2018**, *36*, 307–347. [\[CrossRef\]](#)
26. Rosenberg, F.; Leitner, P.; Michlmayr, A.; Celikovic, P.; Dustdar, S. Towards composition as a service-a quality of service driven approach. In Proceedings of the IEEE 25th International Conference on Data Engineering, Shanghai, China, 29 March–2 April 2009; pp. 1733–1740.
27. Deng, S.; Huang, L.; Wu, H.; Wu, Z. Constraints-driven service composition in mobile cloud computing. In Proceedings of the 2016 IEEE International Conference on Web Services (ICWS), San Francisco, CA, USA, 27 June–2 July 2016; pp. 228–235.
28. Alti, A.; Laborie, S.; Phillippe, R. Dynamic semantic-based adaptation of multimedia documents. *Trans. Emerg. Telecommun. Technol.* **2014**, *25*, 239–258. [\[CrossRef\]](#)
29. Reffad, H.; Alti, A.; Roose, P. Cloud-based Semantic Platform for Dynamic Management of Context-aware mobile ERP applications. In Proceedings of the 8th International Conference on Management of Digital Ecosystems, Biarritz, France, 1–4 November 2016; pp. 181–188.
30. Asghari, S.; Navimipour, N.J. Review and comparison of meta-heuristic algorithms for service composition in cloud computing. *Majlesi J. Multimed. Process.* **2015**, *4*, 239–258.
31. Safaei, A.; Nassiri, R.; Rahmani, A.M. Enterprise service composition models in IoT context: Solutions comparison. *J. Supercomput.* **2022**, *78*, 2015–2042. [\[CrossRef\]](#)
32. Chang, H.; Liu, H.; Leung, Y.W.; Chu, X. Minimum latency server selection for heterogeneous cloud services. In Proceedings of the 2014 IEEE Global Communications Conference, Austin, TX, USA, 8–12 December 2014; pp. 2276–2282.
33. Strunk, A. QoS-aware service composition: A survey. In Proceedings of the 2010 Eighth IEEE European Conference on Web Services, Ayia Napa, Cyprus, 1–3 December 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 67–74.
34. Rowley, H.V.; Peters, G.M.; Lundie, S.; Moore, S.J. Aggregating sustainability indicators: Beyond the weighted sum. *J. Environ. Manag.* **2012**, *111*, 24–33. [\[CrossRef\]](#)
35. Tarawneh, H.; Alhadid, I.; Khwaldeh, S.; Afaneh, S. An intelligent cloud service composition optimization using spider monkey and multistage forward search algorithms. *Symmetry* **2022**, *14*, 82. [\[CrossRef\]](#)
36. Huang, X.; Lei, X.; Jiang, Y. Comparison of three multi-objective optimization algorithms for hydrological model. In Proceedings of the International Symposium on Intelligence Computation and Applications, Wuhan, China, 27–28 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 209–216.
37. Peng, S.; Guo, T. Multi-Objective Service Composition Using Enhanced Multi-Objective Differential Evolution Algorithm. *Comput. Intell. Neurosci.* **2023**, *2023*, 8184367. [\[CrossRef\]](#)
38. Tiwary, M.; Kumar, S.; Agrawal, P.K.; Puthal, D.; Rodrigues, J.J.; Sahoo, K.S.; Sahoo, B. Introducing network multi-tenancy for cloud-based enterprise resource planning: An IoT application. In Proceedings of the 2018 IEEE 27th International Symposium on Industrial Electronics (ISIE), Banja Luka, Bosnia and Herzegovina, 1–3 November 2018; pp. 1263–1269.
39. Tavana, M.; Hajipour, V.; Oveisi, S. IoT-based enterprise resource planning: Challenges, open issues, applications, architecture, and future research directions. *Internet Things* **2020**, *11*, 100262. [\[CrossRef\]](#)
40. Bellifemine, F.; Caire, G.; Poggi, A.; Rimassa, G. JADE: A software framework for developing multi-agent applications. Lessons learned. *Inf. Softw. Technol.* **2008**, *50*, 10–21. [\[CrossRef\]](#)

41. Musen, M.A. The protégé project: A look back and a look forward. *AI Matters* **2015**, *1*, 4–12. [[CrossRef](#)]
42. Tari, K.; Amirat, Y.; Chibani, A.; Yachir, A.; Mellouk, A. Context-aware dynamic service composition in ubiquitous environment. In Proceedings of the 2010 IEEE International Conference on Communications, Cape Town, South Africa, 23–27 May 2010; pp. 1–6.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.