doi:10.3772/j.issn.1006-6748.2017.01.014

Research on logistics domain-oriented cloud resource management model and architecture①

Zhang Xiaodong (张小东)②* **, Zhan Dechen*, Chu Dianhui**
(* School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, P. R. China)
(** School of Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai 264209, P. R. China)

Abstract

To address the challenges posed by resource shortage or surplus to enterprises productivity, Internet platforms have been widely used, which can balance shortage and surplus in broader environments. However, the existing resource management models lack openness, sharing ability and scalability, which make it difficult for many heterogeneous resources to co-exist in the same system. It is also difficult to resolve the conflicts between distributed self-management and centralized scheduling in the system. This paper analyzes the characteristics of resources in the distributed environment and proposes a new resource management architecture by considering the resource aggregation capacity of cloud computing. The architecture includes a universal resource scheduling optimization model which has been applied successfully in double-district multi-ship-scheduling multi-container-yard empty containers transporting of international shipping logistics. Applications in all these domains prove that this new resource management architecture is feasible and can achieve the expected effect.

Key words: resource attribute, resource service model, resource calendar, resource management architecture, resource service optimized scheduling

0 Introduction

Cloud computing is a large shared resource pool, where tenants can choose service resources dynamically on demand[1,2]. Cloud computing can provide computing resource, including CPUs, hard disks, memories, networks, software, data and so on. Its service attributes are simple. Time or cost is usually defined as the optimization objective to construct a model based on capability or finance[3]. Besides computing resources, the application system based on cloud computing needs lots of shared domain-oriented resources, e. g. hard logistics resources including conveyers, transportation vehicles and labors, etc. Soft logistics resources includes demonstration, delivery, scheduling, integration, etc. Enterprises can introduce shareable resources to the Cloud in order to balance resource between surplus and shortage in a wider area. However, it brings great challenge to manage the resources owing to many varieties and large amounts of resources in the application system and different definitions and managing methods of the resources by resource service provider (RSP) and resource service demander (RSD). For example, it usually happens that the resources of RSP are abundant, but at the same time, RSD cannot find appropriate resources in the transportation service. The causes of this predicament are: 1) Information communication is poor. isolated information exists. 2) There are too many resources and tasks to make it difficult to find the best assignment for both RSP and RSD. The key point of the problem is that both resource management and service mode are imperfect. Therefore, this paper focuses on resource management in cloud computing, and resource management architecture is built including resource model, resource service mode, resource management architecture and an appropriate scheduling algorithm, which can be widely used for varieties of resources management and they are pretty scalable.

---

① Support by the National Key Technology Research and Development Program of China (No. 2012BAA13B01, 2014BAF07B02), the National Natural Science Foundation of China (No. 61273038), Natural Science Foundation of Shandong Province (No. ZR2015FM006), Science and Technology Major Project of the Ministry of Science and Technology of Shandong Province (No. 2015ZDXX0201B02).

② To whom correspondence should be addressed. E-mail: z_xiaodong7134@163.com

Received on Nov. 2, 2015

1 Related work

There has been a lot of related researches about resource management, as a hot issue it is, such as resource description frame (RDF)[4,5] and web ontology language (OWL)[6]. RDF describes resource by a triple consisting of resource, attribute type and attribute value. RDF schema (RDFS) is a defining language which is used to define the attribute element of metadata. People can describe any type of resources in their own words by RDFS. OWL, in contrast, defines resources with ontology which is more suitable in Internet. It can also describe the relation of different resources and the meaning of resource in certain field and has the ability of logic description and reasoning. However, if each user defines resource by RDF and OWL in their own way, there will be the problems mentioned above. Therefore, it is essential to unify specification and description.

Peng, et al.[7] found similarities of telecommunication resources after abstracting and objectifying them. They presented a concept of virtual resources and built a resource management model which could mask the difference among heterogeneous resources. Wang, et al.[8] introduced a concept of logical resource tree about heterogeneous resources in WAN and proposed a global unified hierarchical model of network resource by abstracting resource parameters. Comparing with the resource pool[9] and the global-local two-layer model[10], to some degree, the model masks the difference among heterogeneous resources and the model is extensible in some level. Keahey, et al.[11] proposed a multilevel resource management model and analyzed the heterogeneity of resource and scalability of the model. Li, et al.[12] built a meta-model of the resource and presented a unified frame of manufacturing resource management on the meta-model. The methods above can effectively mask the difference of heterogeneous resources and reduce the difficulty caused by resource diversity. It is convenient for the unified mode of expression to put the resource into the system and manage them in centralized way. However, it does not mention that it is likely that various resources need to cooperate to complete a task, that is, cooperation among different resources.

Li, et al.[13,14] proposed a cloud manufacturing framework, which combined cloud computing with network manufacturing. The virtual resource layer, in which resources are aggregated, is added into the framework. A variety of resources is accessed into it by cloud client technology where resources are also virtualized and served. It transforms the usage of resource from many-for-one service mode distributing resources in centralized way into many for many service mode. Casati, et al.[15] built a resource framework based on task demand, and presented an intuitionistic fuzzy and resource optimize algorithm to match and search resources. The above method considered user emotion, trading experience, decaying with time and other non-functional service attributes when it estimated resources. The frameworks mentioned above not only solve the problem caused by resource heterogeneity and technology variety, but also solve the cooperation of resources. These frameworks are scalable and pay more attention to theoretical research of service selection. However, resource services are dynamic, non-real-time and autonomy for RSP in Cloud, which are not mentioned in the models and frameworks above. Thus, the practicability and universality of them are insufficient.

In summary, with the characteristics of Cloud, the resource characteristics in Table 1 can be got. The resources in Cloud may be provided by different RSP, so they are wide-area distributivities, heterogeneities and dynamics. Resource management on cloud (RMC) cannot control resources completely nor predict the resource states. Further, the heterogeneities of resource complicate the resource management. That's why it is difficult to manage the Cloud resources and tasks by previous network resource management mode. The above researches have proved that encapsulating resource in service can mask heterogeneity, distributivity, diversity and other problems caused by the different type of resources provided by RSP. Thus, the difficulty of selecting and scheduling resources algorithm is reduced efficiently. However, in practice, without considering private characteristics of resource, resource service ability cannot be fully displayed and resource view is uncompleted. Consequently, the resource optimization will not be realized completely and the ideal result cannot be reached. This paper does researches on the content in Table 1 and puts forward a scalable resource management architecture, in which RSP can choose resource attributes by their own demands, build resource models, and reduce the difficulty of management and schedule resources optimally by related algorithms.

Table 1 Analysis of resource characteristics in the cloud

Resource characteristics	Cause of the characteristic	Expansion of cloud resource management	Expansion functions
Distributional autonomy	In different organizations and geographic area, the resource owners can know, control and manage the resource well.	Providing resource integrating mode, scheduling resources by interacting with resource management systems in different area or the resource calendars provided by RSP.	To permit resource sharing and self-management, realize control of resource without knowledge uncompletely.
Heterogeneity	A great varieties of resources with different characteristics	To define global resource management model with individual extend support[16]	To contain heterogenic resources
Technology diversity	Different organizations apply different managing, scheduling and maintaining strategies to resources	To make the standard mechanism to describe resource and user demands [17] and build scalable resource management architecture	To permit to build different management strategies and mechanisms
Dynamics	The configuration, ability and running state of resource are changing continually during operation process.	Hold a certain self-adaption and fault-tolerant ability	To permit to configure, control and replace resources
Non real time	The production cycle covers a long time. The processes are independent.	To build resource calendar and monitor resource	To schedule resource and make resources cooperation
Cooperativity	Complex tasks need cooperation of resources from different providers	To know resource characteristics, trace the task and form many to many service model .	To realize the resource to allocation over different field
Awareness	Utilize Internet of things, RFID and other sensing technologies, sense the characteristic information in real time	Access Internet of things, RFID and other sensing technologies	To be aware and get the resource characteristics information in real time

2 Resource service model and resource management architecture

2.1 Resource attributes analysis

Knowing the resource attributes well in applications is the key of building a resource service model. As shown in Table 2, this paper lists attributes of several types of common resource in logistics transportation. The attribute set with abundant semantics can distinguish all characteristics of each resource, which benefits delicacy management and shows the heterogeneity of different resources. But in practice, if there are many types and a great number of resources, the complexity of management will rise up and the information extraction will become difficult. When matching resource's ability with a service demand, it will be difficult to get result due to too many input data. Thus, this management method is only adaptive when types of resource are insufficient. It is not suitable in Cloud with a lot of resources.

In order to control the complexity of management and computation, the resource attributes are divided into different types in the paper. As shown in Table 2, every type of resource has the identification, name, position and other attributes after getting rid of some special adjunct words. Getting them as common attributes of Cloud resource can help resource to register in Cloud easily. Adding the special attributes that make resources different from others into RSP self-management system can reduce the complexity of public management. Optimization choice of resource is based on the capability attributes. Simply by controlling these attributes, services and interests of resources can be gained. It is the base of resource service and the focus of both resource sharing in RMC (the resource information opened) and self-management (the resource information not opened). As a main function of RMC, RSP should open some administrator privileges to realize resource sharing. Here resources are classified as public attribute, private attribute and capability attribute. The attribute classification is helpful for solving distributive heterogeneity resource's involvement, sharing, resource self-management and other problems.

Table 2 Resource attribute analysis

Resource model		Attribute division		
Model Name	Attribute Set	Public Attribute Set	Private Attribute Set	Capability Attribute Set
Human Resource	Staff ID, Name, Type, Address, Salary, Organization, Status, Skill, Sex, ID-Card, Birthday, Education Background, Driving License(DL), DL-type, Nationality, …	Staff ID, Name, Address, Salary, Organization, Status, …	Staff ID, Sex, ID-Card, Birthday, Education, DL, DL-type, Nationality, …	Staff ID, Skill, …
Vehicle Resource	Vehicle ID, License No., Location, Transportation Cost, Provider, Status, Oil Consumption, Dead Weight (DW), Driving License, Engine Model…	Vehicle ID, Location, Transportation Cost, Provider, Status, …	Vehicle ID, License No., Location, Transportation Cost, Oil Consumption, Driving License, Engine Model…	Vehicle ID, Truck, DW, Hold Capacity, … Vehicle ID, Crane, DW, Lifting height, … Vehicle ID, Forklift, DW, Lifting height, … …
Container Resource	Container ID, Container Type, Location, Cost, Provider, Cost, Status, Size, Weight, Temperature Control, Capacity, …	Container ID, Type, Location, Cost, Provider, Status, …	Container ID, Size, Weight, …	Container ID, Temperature Control, Capacity, …
Ship	IMO-ID, Name, Nation, Custom No., Status, DWT, Voyage No, EAT, ETD, …	IMO-ID, Name, Nation, Custom No., Status, …	IMO-ID, Voyage No, …	IMO-ID, DWT (Dead Weight Tonnage), Voyage No., Estimated Arrival Time(EAT), Estimated Time of Departure(ETD), …
……	……	……	……	……
Resource General Model (abstract from Public Attribute Set)		Resource ID, Resource Name, Resource Type, Location, Resource price, Provider, Status		

By further processing the attributes, the Cloud resource management can be supported better, such as getting rid of the resource identification of public resource, emphasizing resource type and capability attribute, as shown in Table 3. The resource type is always described as a tree that is easy to search for. After aggregating capability attribute according to types, the rage of their value will become a continuous capability interval, such as the interval of van loading capability is $0 \leqslant w \leqslant 20$t, including van loading 4t, 8t, 16t and 20t. The coordination rule of resources is easier to formulate. For example, diver + van = transport service etc.. Besides, there are more advanced resource coordination composition, e. g. transport service + loading service = distribution service. Although these compositions are not final resource scheduling or task arrangement, these rules filter out a lot of mistakes to reduce the computation cost.

2.2 Resource service model (RSM)

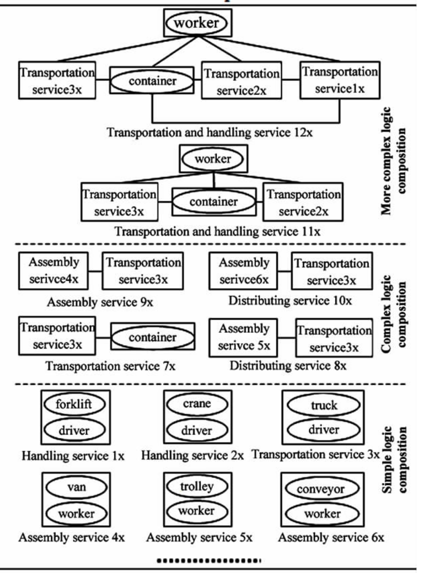
Although attribute classification can deduce the resource cooperative rules, it cannot resolve the conflicts between self-management and sharing. Here taking the management model of time-shared system for CPU as reference, introducing resource calendar to monitor the service time of resource for inside and outside enterprises can solve the problem above. Thus, the resource service model is defined as the following.

(1) Resource Meta Model. $RMM = \{G, P, C, RT, RE, F\}$, where G is resource general model, which is a common service attribute set owned by every resource; P is the resource private model, which is the set of private attribute belonging to every different resource; C is the resource calendar, which is composed of a serious of customizable time windows and can restrain RE; RE is a resource service capability attribute set. RT is a resource type, which is a resource classification tree oriented to different fields and can restrain G and P. F can restrain and customize P and RE to ensure the content shown to outside.

(2) Resource General Model. $G = \{(g, f_g, RID, R\_Name, R\_Location, R\_Status, t, RSP) \mid g = f_g(rid, R\_Name, R\_Location, R\_Status, t, RSP), t \in T, f_g \in F\}$。Resources are integrated into

the Cloud by G, which identifies resource individual by *rid* including name, position, RSP, state, type and other semantics. In order to ensure that a resource can be identified, it cannot be customized but inherited. RSP can expand G according to their own demands; f_g is the operating method of resource attributes.

Table 3 the regulation and collaborative model derived from attribute and classification

Name	Type	Attribute		Rule	Resource cooperation model
		Public Attribute	Capability Attribute		
Human Resource	driver	Resource type, Location range, Cost range, …	Driving ability range(e.g. A1, A2)…	driver + truck driver + crane driver + forklift container + truck worker + conveyor worker + van ……	
	worker		Skill Level (e.g. 1~8)…		
	……		……		
vehicle	truck		Load rang (e.g. 0~4t), Load capacity range (e.g.0~20*5*4m)…		
	crane		Load rang (e.g. 0~40t), Lifting height (e.g. 0~6m)…		
	forklift		Load rang (e.g. 0~10t), Lifting height (e.g. 0~2m), Gradeability range (e.g. 0~30°)…		
	……		……		
Container	container		Load capacity range (e.g. 0~20*5*4m)…		
	refrigerator		Load capacity range (e.g. 0~20*5*4m), refrigeration temperature (e.g. 0~ -5℃)…		
	……		……		
ship	roll-on/off ship		Load capacity range (e.g. 0~200TEU), Load range (0~100ton)…		
	container ship				
…	…		…		

(3) Resource Private Model. $P = \{(t, A, a, f_p) \mid a = f_p(A, t), a \in A, t \in T, f_p \in F\}$, where $A = (a_1, a_2, \cdots, a_n)$ means that t type resource has n different private attributes. RSP can customize resource private attributes by f_p. RSP should inherit and expand P.

(4) Resource Calendar. $C = \{(c_g, a_i, b_i, g, f_c) \mid c_{ird} = f_c(a_i, b_i, g)$, and $a_i < b_i$, $a_i, b_i \in Date$, $g \in G$, $i \in N\}$, where both a and b denote time , i is an ordered consequence 1, 2, 3, ……. C is a time period that can provide service outside. It can let RMC know the resource state at any time.

(5) Resource Type Tree. $RT = \{(T, R, root_k, t, f_t) \mid t = f_t(root_k, T, R)$ and $\forall i, j, i \neq j, T_i \cap T_j = \varnothing, R_i \cap R_j = \varnothing\}$, where T is a set of resource types, T_i and T_j are any two partitions of T, R is the relation among type elements, R_i and R_j are any two partitions (the adjacency matrix or adjacency list) of R and RT are the resource type of the domain. The resource classification is unified and non-duplicated in the same field. f_t is the operation to the resource tree which can improve the efficiency of resource search.

(6) Resource Service Capability. $RE = \{(g, E, c_g, t, e_c, f_e) \mid e_c = fe(g, E, c_g, t), e \in RE, g \in G, c \in C, t \in T, f_e \in F\}$, where $E = \{E_1, E_2, E_3, \cdots, E_n\}$, shows that it can describe resource capability from multi-dimension. e_c means the service ability provided by g and is obtained by f_e in the time window c_g, where the service ability varies with time period.

Here shows the above definitions and their relations by a UML class chart. As shown in Fig. 1, *RMM* is composed of G, P, C, *RT* and *RE*. The five parts restrain and depend on each other to form a complete resource management view, namely resource service model (RSM). Different resources have their own special attributes and service abilities. Thus *RT* can restrain *RE* and P in a certain extent and help RSD to find resources they need and help RSP to manage resources. The formation of *RT* is related to the field and generally a tree structure. In general, the type is defined in the xml file based on ontology analyzed by *RT* engine in order to provide more flexible definition way, and classified according to the system standard, for example, the ship classification in Fig. 2. There is no further discussion to this point. The service provided by resource at different time may be different. Sometimes the resource provides service for inside and sometimes for outside. Even though a resource is totally rented for outside, it still needs some maintained time. Thus, the resource calendar restrains it. The basic content of RSM is shown in the left part of Fig. 1.

In order to satisfy the diversity demands of tenants for resource management, RSM provides customizable and extensible functions. Tenants can expand and inherit

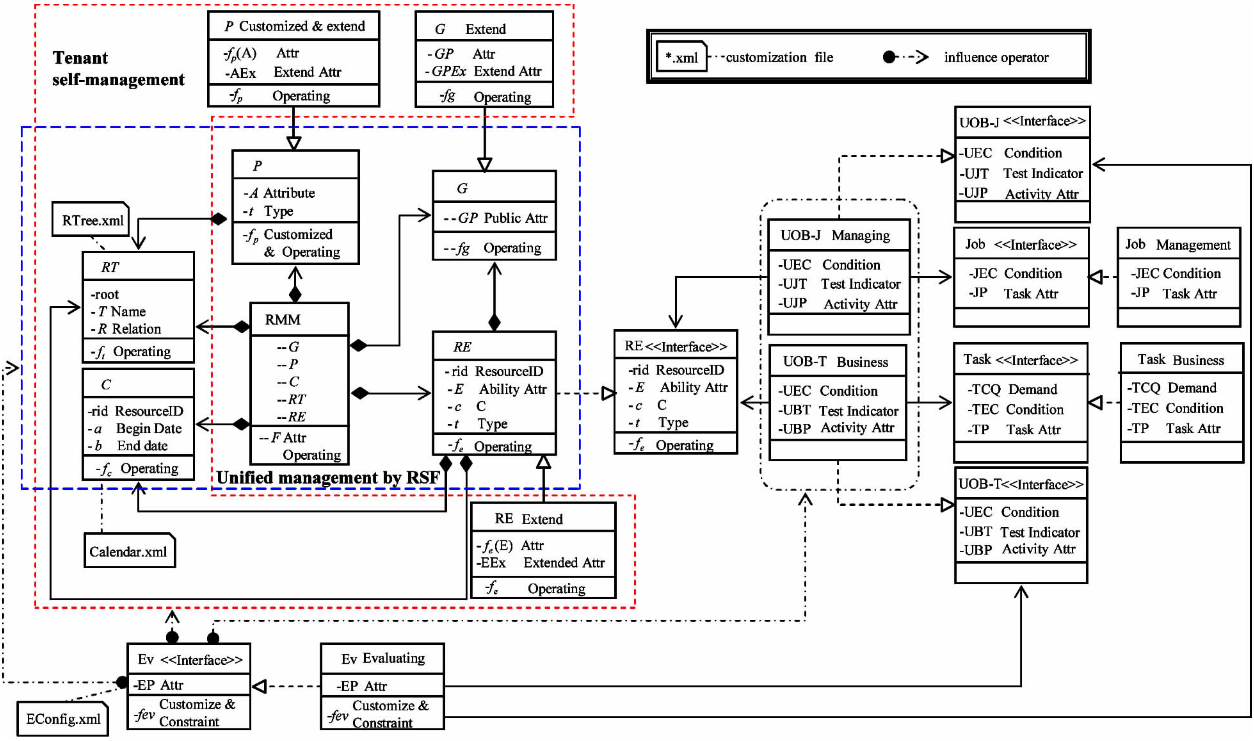
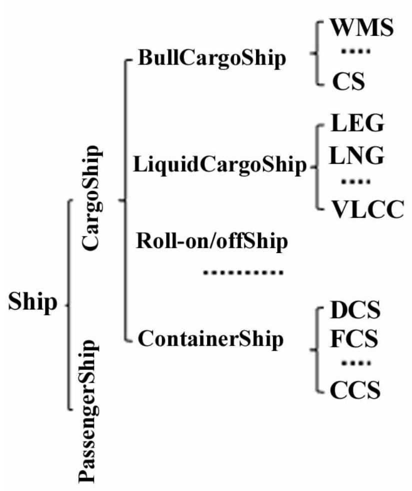


Fig. 1 Resource service model




```
<xs:complexType name= "Ship" >
    <xs:sequence>
        <xs:element name= "CargoShip"  type= "CargoShip" />
        <xs:element name= "PassengerShip"  type= "PassengerShip" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name= " CargoShip" >
    <xs:sequence>
        <xs:element name= "BulkCargoShip"  type= "BulkCargoShip" />
        <xs:element name= "LiquidCargoShip"  type= "LiquidCargoShip"
        <xs:element name= "Roll-on/offShip"  type= "xs:string" />
        ......
        <xs:element name= "ContainerShip"  type= "xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name= "BulkCargoShip" >
    <xs:sequence>
        <xs:element name= "WoodMaterialShip"  type= "xs:string" />
        ......
        <xs:element name= "CoalShip"  type= "xs:string" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name= "LiquidCargoShip" >
    <xs:sequence>
        <xs:element name= "LEG"  type= "xs:string" />
        ......
        <xs:element name= "LNG"  type= "xs:string" />
        <xs:element name= "VLCC"  type= "xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name= "ContainerShip" >
    <xs:sequence>
        <xs:element name="DiversityOfContainerShip" type="xs:string"/>
        <xs:element name="FullContainerShip" type="xs:string"/>
        ......
        <xs:element name="ConvertibleContainShip" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
```

Fig. 2 Ship classification

P and *RE* and customize the related attributes and operation by $f_p(A)$ and $f_e(E)$. *G* is the identification by which the system recognizes resource. It is the basic content, which can only be inherited and expanded but can be modified arbitrarily. In the same field, the resource management is better to be built in the united classify system. Thus, the expansion of *RT* mainly concentrates on the expanding and contracting of data entries and branches, but not on expansion and rebuilding of the structure and engine. *C* is the complementary time similar to *RT*. The resource does not provide service outside when the internal is busy. In other words, the resource provides service outside only when internal is free. As shown in Fig. 1, the components in the C dashed rectangle is the RSM expanding region. The diverse customized and expanding technologies enrich the autonomous function of the tenant to resource. At the same time, it can also be used to improve and perfect RSM itself.

The only external interface of resource management is resource capability, which is the main aim of resource management that resource can complete its designated tasks. Since the task management is not the main focus of this paper, the task is roughly divided into two parts: task and task management. The task is the business activity satisfying certain conditions, e. g. transportation commission. The task management is to monitor the task, e. g. the implementation of task. UOB-T business reflects the implementation of the task, e. g. the assignment bill, and UOB-J management is the result of executing the task such as back to single about task. Both of two activities need the participation of resources. Besides, evaluating components Ev can be introduced into the system which can gather the task executing information to evaluate the task execution and transmit the evaluate result to RSP to improve the task management state. What's more, RSP can also customize Ev to get the demanded data.

2.3 Resource service management architecture

RSM describes the most essential characteristics of resources. As shown in Fig. 1, it shows the most basic service relation between the resource and the task. Although it deals with heterogeneous resource introduced by attribute extraction, supports resource diversity management by customizability and expansibility, and completes resource sharing and self-management, resource distribution, state tracking, non-real time scheduling and other problems by the resource classification and the calendar, however, as mentioned in Section 2.1, when the type and number of resource are large, the complexity of management is still very high and the cooperation of different resources is still difficult. The main reason why the above problems cannot be completely solved is that the resource management is underlying. Thus, the resource is encapsulated as the service based on RSM and the resource service management architecture (RSMA) can be reached, as shown in Fig. 3.

(1) Resource providing layer. It is composed of the physical resources provided by RSP and it is the task execution layer. RSP can complete the five parts of information required by RSM to transform resources to virtual resource to be scheduled by RMC.

(2) Resource aggregation layer. Dividing resources in different type to form an elastic resource aggregation layer, as mentioned in Section 2.1, forms a continue service ability valued range, reduces the difficulty of selecting service, designing algorithm and coding. It can be seen from the definition of RSM that no constraint of region, RSP or RSD and permitting a lot of resources being introduced into the system, it ensures enough resource supply[18] during task assigned process. In order to cooperate with the distributed self-management, the system can regulate the resource to keep balance automatically by setting resource calendar and service ability provided outside.

(3) Resource service layer. One or more resources can be combined into an integeral with more capability by composing rules. For example, a driver and a forklift can be combined into a loading service with 1t lifting ability and 90cm lifting height and a driver and a truck can be combined into a transporting service with 10t load and 70cm loading height. Then the loading service and the transportation service can be connected to form a distribution service. The output service ability will be restrained mainly by some kind of resource. In this example, the restraint of distribution service is the load of the truck, 10t. The distribution time is equal to loading time + transporting time. However, when the transporting time is much longer than the loading time, the loading time can be ignored. The loading service is supplementary to the transportation service. The mode can reduce the complexity of computation (for details, see Section 3). Here, it is also allowed to package resource into service to enter into the system. In other words, it can access into the system crossing two different layers.

(4) Service flow layer. Resource services can be composed to form a fully functional service by the task demand. It may include non-resource service. Because of the uncertainty of tasks, it shall be formed dynamically.

(5) Service demand layer. In this layer, the user

can divide a task into a serious of subtasks. On one hand, it can choose a related flow model from the service flow layer. On the other hand, it is also possible to choose resource services and compose them to form the service flow dynamically according to task demand.

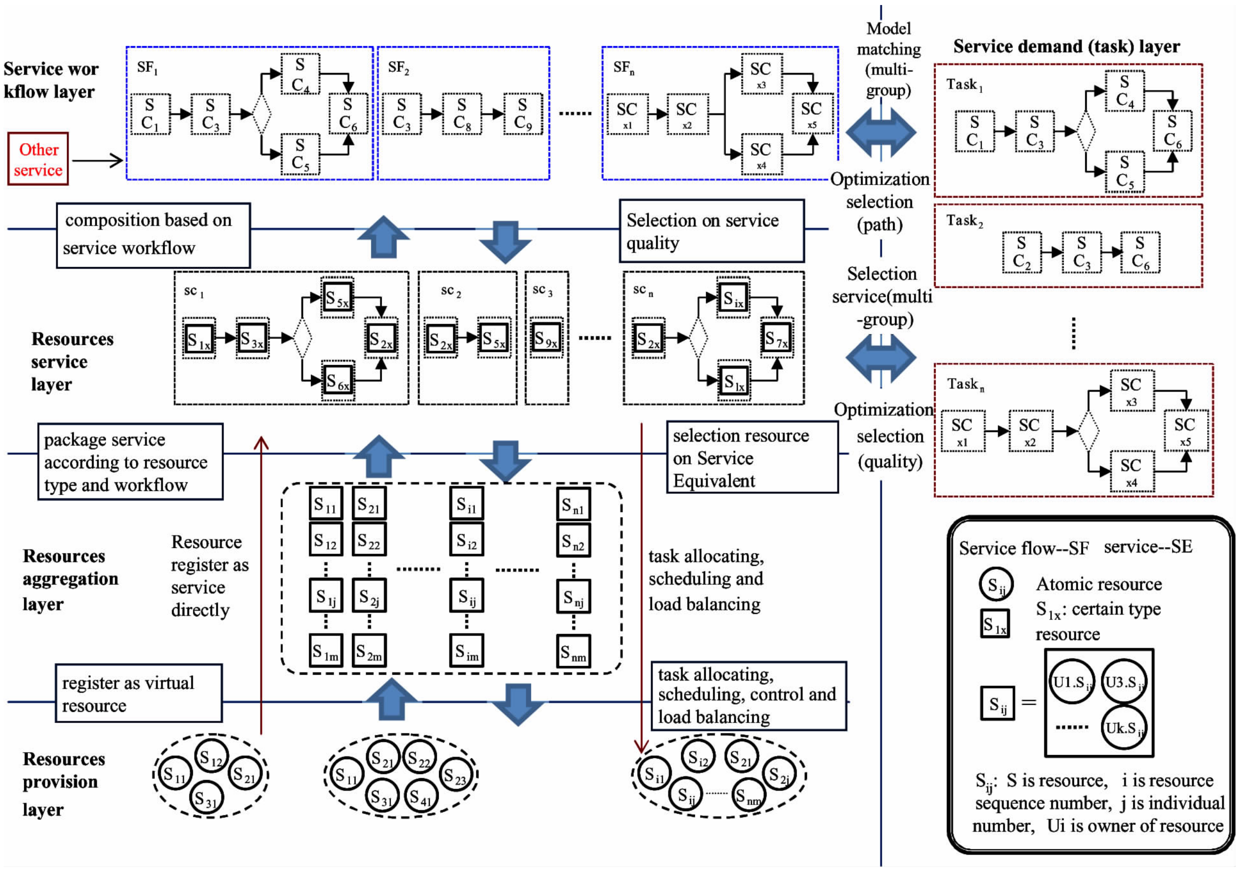


Fig. 3 Resource service architecture

The selection of service resource is a process across five layers and top-down: task decomposition→flow/service selection → service binding → resource binding→resource execution. The optimization of service flow layer is the optimization of service route. And the optimization of service layer is eventually the optimization of resource selection.

2.4 Resource service optimal scheduling model

After encapsulated in several levels, the scheduling resources will be simplified and the cooperation among resources will be easier. In RSMA, there are four ways to select resource services: model matching, optimal scheduling based on route, service selection, and optimal scheduling based on quality. The model matching is that the subtasks are composed to form an executive flow and then matched with the service flow in the service layer. So it can be abstracted as similarity judgment of graph. The optimal scheduling based on quality can get the information of QoS by the Ev components mentioned in RSM to optimize itself. There have been many related researches. Further discussion will not be provided due to the limitations of space. Service choices refer to finding related service from resource service layer directly when the tasks or the subtasks are relatively simple. This paper provides a general route optimization algorithm with time window (GROATW) about optimal scheduling based on route.

Suppose completing a task needs n steps and the state of every step is called a task node. The task execution forms a directed graph G which is composed of n task nodes, original node 0 and terminal node $n+1$. The set of task nodes is TN, $N=|TN|+2$. From node i to j, the standard time consumption is t_{ij} and the cost consumption is w_{ij}. Task node i has a time window $[a_i, b_i]$ and s is the beginning of task time, the task requires service capability q_i. According to the resource calendar, resource g has the time window like $[a_g, b_g]$. Let x_{ijg} represent the service provided by resource g from node i to node j. If the value is only 1 or 0, it is arrived or not. Otherwise, it represents the service capability value provided by the resource. According to the definition above, the mathematical model aiming to optimize the cost shall be:

$$\min\left\{\sum_{g\in R}\sum_{i\in N}\sum_{j\in TN} w_{ij}x_{ijg}\right\} \quad (1)$$

$$\text{s.t.}\begin{cases}\sum_{g\in R}\sum_{j\in N} x_{ijg} = 1 & \forall i \in TN \quad (2)\\ \sum_{i\in TN} q_i \sum_{j\in N} x_{ijg} \leqslant e_{ci} & \forall i \in TN \quad (3)\\ \sum_{j\in N} x_{0jg} = 1 & \forall g \in R \quad (4)\\ \sum_{i\in N} x_{ihg} - \sum_{j\in N} x_{hjg} = 0 & \forall h \in TN, \forall g \in R \quad (5)\\ \sum_{i\in N} x_{i,n+1,g} = 1 & \forall g \in R \quad (6)\\ s_{ig} + t_{ij} \leqslant s_{jg} & \forall i,j \in N, \forall g \in R \quad (7)\\ a_g \leqslant s_{ig} + t_{ij} \leqslant b_g & \forall g \in R \quad (8)\\ a_i \leqslant s_{ig} \leqslant b_i & \forall i \in N \quad (9)\end{cases}$$

This method is generic. When the variables are given more meanings in practice, the expression can be simplified. In the expression, Constraint Eq. (2) shows that every node can only be visited one time. The inequation (3) shows that the resources are called satisfy constraints. Eqs(4), (5) and (6) ensure that the task starts from original node, executes through every sub-node and finishes at the terminal node. The inequation (7) shows that the executing time constrains the same resource at different task nodes. In other words, the task that is incomplete at a node cannot arrive at another node. The inequation (8) shows the time that the resource executes the task should satisfy the constraint of resource time window. The inequation (9) is the time that the resource executes the task should satisfy the constraint of task time window. This model can provide for a variety of changes in practice. For example, let x denote the weight between task nodes, Eqs(2), (4), (5) can be omitted. If service demand is calculated dynamically formula (3) can be expanded. If it is used as resource scheduling algorithm for the enterprise internal, the resource time window can be removed. When all the resources are vehicles, it can be seen as vehicle routing problem with time window (VRPTW). It becomes vehicle routing problem (VRP) without the time window. If there is only one vehicle, the model will be the travelling salesman problem (TSP). Because the first two variations of this algorithm are more practical, it will be analyzed in the experimental part.

3 Experimental results and analysis

3.1 Background of experiment

In international trade, there is usual imbalance of trade between two countries because of their economic gap. There are different demands of empty containers from both sides. Therefore, empty container transportation between any two container yards (CYs) is demanded. A country is taken as a district. CYs maybe lie in the same district or in the different districts. Goods are transported between CYs in different districts by ship and in the same district by vehicle. Obviously, each transportation process is the above subtask node. To complete a task of empty container transportation, many resources are demanded to participate, such as ship, container, container truck, forklift, crane, driver, worker, sailor, etc. After they have been registered into RSMA, they can compose many services according to the rules and are placed in the resource service layer, as mentioned in Section 2. There are two ways to schedule them optimally, in which one is that field experts make the service flow in advance and then match these service flow with the task flow by the rule, the other is that the above service can be chosen to complete a subtask node directly by optimization algorithm. The latter is used as an example to discuss in the text.

Take a 20-foot container as a transmission extension unit (TEU). For the consideration of the transportation cost, the number of empty containers carried by ships is usually not the largest deadweight, but the residual load after loading the heavy containers, and it varies according to different voyages, which is expressed as $e_k = f_e(g.rid, E, c, t)$, where $g.rid$ is the voyage number, c is shipping schedule, $k \in c$, as shown in Table 4. Double-district is bidirectional traveling,

Table 4 Shipping schedule

Voyage No.	Vessel No.	Course	ETA	EDT
1029E	1029	eastern	2-24 08:00:00	2-24 20:00
1029W	1029	western	2-25 08:00:00	2-25 20:00
1030E	1030	eastern	2-26 08:00:00	2-26 20:00
1030W	1030	western	2-27 08:00:00	2-27 20:00
1031E	1031	eastern	2-28 08:00:00	2-28 20:00
1031W	1031	western	3-1 08:00:00	3-1 20:00
1032E	1032	eastern	3-2 08:00:00	3-2 20:00
1032W	1032	western	3-4 08:00:00	3-4 20:00

4 ships are divided into eight voyages. In the international marine logistics, shipping service time window is the hard time window, and can't be changed easily. So ships are the core of the service resources, while the loading service and land carriage service in the same district serve as the assistance resource service, scheduling is comparatively flexible, as a result this experiment uses shipping schedule as the prime resource calendar.

Empty container demand is decided by the available margin/shortage amount of each voyage and each CY, as shown in Table 5.

Table 5 Available margin/shortage amount of empty container all voyages and CYs (TEU)

District	CY	Voyage 1	Voyage 2	Voyage 3	Voyage 4
N	A	2	-25	-51	-67
	B	5	-1	-7	-12
	C	20	10	-1	-11
	Total	27	-16	-59	-90
M	D	6	31	56	89
	E	3	18	34	47
	F	-4	-10	-16	-22
	Total	5	39	74	114

3.2 Optimal scheduling model on international marine logistics

Because the eastward voyage is before the westward voyage for the same voyage No., containers can be scheduled from the eastward to the westward, but the contrary case does not exist. Furthermore, due to the voyage No. increasing along with time, there is no case that scheduling container is from big voyage No. to the small voyage No.. Therefore, the voyage direction's decision variable is a two-dimensional triangular matrix, if combined with the CY No., the decision-making variable is four-dimensional variable x_{ijkl}. x_{ijkl} can be interpreted as the number of scheduling containers from CY i in shipment date k to CY j in shipment date l. Because vehicle and human resources' calendar is software time window, it must meet the requirements of shipping schedule. Here x_{ijkl} can also refer to land resource scheduling. This kind of design will package the loading service with the shipping and land carriage service respectively in the resource service layer, so that the resource No. does not have to appear in the decision-making variable x, but can also reflect the characteristics of multiple-resource scheduling. w_{ij} is transportation cost here, using the Yuan/TEU as standard service equivalent, then objective function changed from (1) to (10), matched with the first variation in Section 2.4, then formulas (2), (4) (5) are removed. Formula (12) shows the upper bound of scheduled empty containers in CY k. Formula (7) changes formula (13) while the former omits resource No.. The task time window matches with the shipping schedule. The vehicle and human resource time window must follow the shipping schedule, both formulas (8) and (9) are merged to (14) which explains that the total time of many scheduling containers on land is no longer than the shipment date, where K is transportation time in the same district. Eventually, the double-district multi-ship-scheduling multi-container-yard empty containers model, which is evolution of resource allocation general model, is as follows:

$$\min\{\sum_{i}\sum_{j}(w_{ij}\sum_{l}\sum_{k=0}^{l}x_{ijkl})\} \tag{10}$$

$$\text{s.t.}\begin{cases} q_{ik} - \sum\limits_{j\neq i}\sum\limits_{l=k}^{n}x_{ijkl} + \sum\limits_{j\neq i}\sum\limits_{l=0}^{k}x_{jilk} \geqslant 0 \\ \qquad \forall i,j \in M\cup N,\ l,k\in c \qquad (11) \\ 0 \leqslant x_{ijkl} \leqslant e_k,\ x_{iikl}=0 \\ \qquad \forall i,j \in M\cup N,\ l,k\in c \qquad (12) \\ s_i + t_{ij} \leqslant s_j \qquad \forall i,j\in M \| N \qquad (13) \\ a_k \leqslant s_i + K\times t_{ij} \leqslant b_k \\ \qquad \forall i,j\in M\| N,\ \forall k\in c \qquad (14) \end{cases}$$

3.3 Solving

When solving the transportation problems with complex constraints, genetic algorithms (GA) is an ideal choice. Then let's find the optimal solution of the above equation.

(1) GA coding

This paper uses decimal encoding. Multi-ship-scheduling and multi-container-yard require using four-dimensional matrix's chromosome encoding, and Eq. (15) shows the allocation matrix of scheduling empty containers problem of six yards and eight shipment dates.

$$X_p = \begin{bmatrix} \begin{bmatrix} x_{0000} & x_{0100} & x_{0500} \\ x_{1000} & x_{1100} & x_{1500} \\ \vdots & \ddots & \vdots \\ x_{5000} & x_{5100} & x_{5500} \end{bmatrix} \begin{bmatrix} x_{0001} & x_{0101} & x_{0501} \\ x_{1001} & x_{1101} & x_{1501} \\ \vdots & \ddots & \vdots \\ x_{5001} & x_{5101} & x_{5501} \end{bmatrix} \cdots \begin{bmatrix} x_{0008} & x_{0108} & x_{0508} \\ x_{1008} & x_{1108} & x_{1508} \\ \vdots & \ddots & \vdots \\ x_{5008} & x_{5108} & x_{5508} \end{bmatrix} \\ \begin{bmatrix} x_{0010} & x_{0110} & x_{0510} \\ x_{1010} & x_{1110} & x_{1510} \\ \vdots & \ddots & \vdots \\ x_{5010} & x_{5110} & x_{5510} \end{bmatrix} \begin{bmatrix} x_{0011} & x_{0111} & x_{0511} \\ x_{1011} & x_{1111} & x_{1511} \\ \vdots & \ddots & \vdots \\ x_{5011} & x_{5111} & x_{5511} \end{bmatrix} \cdots \begin{bmatrix} x_{0018} & x_{0118} & x_{0508} \\ x_{1018} & x_{1118} & x_{1518} \\ \vdots & \ddots & \vdots \\ x_{5018} & x_{5118} & x_{5518} \end{bmatrix} \\ \vdots \qquad\qquad \vdots \qquad\qquad \ddots \qquad\qquad \vdots \\ \begin{bmatrix} x_{0080} & x_{0180} & x_{0580} \\ x_{1080} & x_{1180} & x_{1580} \\ \vdots & \ddots & \vdots \\ x_{5080} & x_{5180} & x_{5580} \end{bmatrix} \begin{bmatrix} x_{0081} & x_{0181} & x_{0581} \\ x_{1081} & x_{1100} & x_{1581} \\ \vdots & \ddots & \vdots \\ x_{5081} & x_{5181} & x_{5581} \end{bmatrix} \cdots \begin{bmatrix} x_{0088} & x_{0188} & x_{0588} \\ x_{1088} & x_{1188} & x_{1588} \\ \vdots & \ddots & \vdots \\ x_{5088} & x_{5188} & x_{5588} \end{bmatrix} \end{bmatrix}_{6\times6\times8\times8} \tag{15}$$

where X_p represents the pth chromosome, and x_{ijkl} is the corresponding decision variable.

(2) The calculation of fitness

An objective function is used to evaluate the value of Xp's fitness, Eq. (16) is obtained by Eq. (10).

$$eval(X_p) = \sum_i \sum_j (w_{ij} \sum_l \sum_{k=0}^{l} x_{ijkl}) \tag{16}$$

(3) The process of initialization

A group of decision variables x_{ijkl}s can be obtained by generating the subscript of each x_{ijkl} randomly. Referring to the direction d_{ij} limit (i.e., $d_{ij} = 0$ indicates the same district, $d_{ij} = 1$ denotes the direction from east to west and $d_{ij} = -1$ from west to east), the group of x_{ijkl}s can be filtered by formulas (11) to (14) in the model, and hereby the initial populations that satisfy the initial constraints can be produced.

(4) Crossover operator design

Suppose $X_1 = (x_{ijkl}^1)_{mmnn}$ and $X_2 = (x_{ijkl}^2)_{mmnn}$. The crossover needs the following three steps:

Step 1: Construct two temporary matrix $D = (d_{ijkl})_{mmnn}$ and $R = (r_{ijkl})_{mmnn}$:

$$d_{ijkl} = \lfloor (x_{ijkl}^1 + x_{ijkl}^2)/2 \rfloor \tag{17}$$

$$r_{ijkl} = (x_{ijkl}^1 + x_{ijkl}^2) \bmod 2 \tag{18}$$

Matrix D is the average integer value of the parents. Matrix R judges if the average value of the parents is integer. The following is the relation between D and R:

$$q_{ik} - \sum_{j=1}^{m} \sum_{l=k}^{n} d_{ijkl} = \frac{1}{2} \sum_{j=1}^{m} \sum_{l=k}^{n} r_{ijkl}$$
$$i, j = 1,2,\cdots,m;\ k, l = 1,2,\cdots,n \tag{19}$$

$$q_{jl} - \sum_{i=1}^{m} \sum_{k=1}^{n} d_{ijkl} = \frac{1}{2} \sum_{i=1}^{m} \sum_{k=1}^{n} r_{ijkl}$$
$$i, j = 1,2,\cdots,m;\ k, l = 1,2,\cdots,n \tag{20}$$

Step 2: R is divided into two matrix

$R^1 = (r_{ijmn}^1)$ and $R^2 = (r_{ijmn}^2)$

$$R = R^1 + R^2 \tag{21}$$

$$\sum_j^m \sum_k^n \sum_l^n r_{ijkl}^1 = \sum_j^m \sum_k^n \sum_l^n r_{ijkl}^2 = \frac{1}{2} \sum_j^m \sum_m^n \sum_n^n r_{ijkl};$$
$$i, j = 1,2,\cdots,m;\ k, l = 1,2,\cdots,n \tag{22}$$

$$\sum_i^m \sum_k^n \sum_l^n r_{ijkl}^1 = \sum_i^m \sum_k^n \sum_l^n r_{ijkl}^2 = \frac{1}{2} \sum_i^m \sum_k^n \sum_l^n r_{ijkl};$$
$$i, j = 1,2,\cdots,m;\ k, l = 1,2,\cdots,n \tag{23}$$

$$\sum_i^m \sum_j^m \sum_l^n r_{ijkl}^1 = \sum_i^m \sum_j^m \sum_l^n r_{ijkl}^2 = \frac{1}{2} \sum_i^m \sum_j^m \sum_l^n r_{ijkl};$$
$$i, j = 1,2,\cdots,m;\ k, l = 1,2,\cdots,n \tag{24}$$

$$\sum_i^m \sum_j^m \sum_k^n r_{ijkn}^1 = \sum_i^m \sum_j^m \sum_k^n r_{ijkn}^2 = \frac{1}{2} \sum_i^m \sum_j^m \sum_k^n r_{ijkn};$$
$$i, j = 1,2,\cdots,m;\ k, l = 1,2,\cdots,n \tag{25}$$

Step 3: generate two descendants X'_1 and X'_2

$$X'_1 = D + R^1 \tag{26}$$

$$X'_2 = D + R^2 \tag{27}$$

(5) Mutation operator design

The followings are three steps of variance:

Step 1: First, construct a sub-matrix matching with the previous generation chromosomes. Second, to choose $\{i_1, \cdots, i_r\}$, $\{j_1, \cdots, j_s\}$, $\{k_1, \cdots, k_t\}$, $\{l_1, \cdots, l_o\}$ randomly to build a sub-matrix $(r \times s \times t \times o)$ of $Z = \{z_{ijkl}\}$, where $\{i_1, \cdots i_r\}$, $\{j_1, \cdots, j_s\}$ $(\{1,2,\cdots,m\}$, and $2 \leqslant r,\ s \leqslant m$, $\{k_1, \cdots, k_t\}$, $\{l_1, \cdots, l_o\} \subset \{1,2,\cdots,n\}$, and $2 \leqslant t,\ o \leqslant n$; z_{ijkl} get value from element with subscript $ijkl$ in its parent matrix.

Step 2: Reassign a sub-matrix of transporting container amount. Scheduling of empty container amount and demanding empty container amount in the sub-matrix are calculated by the following:

$$q_{ik}^z = \sum_{j \in \{j_1, \cdots, j_s\}} \sum_{l \in \{l_1, \cdots, l_o\}} z_{ijkl};$$
$$i = i_1, \cdots, i_r;\ k = k_1, \cdots, k_t \tag{28}$$

$$q_{jl}^z = \sum_{i \in \{i_1, \cdots, i_r\}} \sum_{k \in \{k_1, \cdots, k_t\}} z_{ijkl};$$
$$j = j_1, \cdots, j_s;\ l = l_1, \cdots, l_o \tag{29}$$

Step 3: The elements of the parents-matrix are replaced by the new elements of the reassignment sub-matrix Z and then the descendants are got.

(6) The simulation of multi-ship-scheduling empty container transportation model in double-district

Assuming there are six container yards between the ship route of district M and N, where three CYs are in zone M, and the other three in zone N. Transporting empty container costs per TEU between any two CYs are shown in Table 6. Costs of the same district is mainly composed of land transportation and loading and unloading charges, the costs of transportation between different partitions are mainly composed of the inspection fee, disinfection fee, handling fee and agency fee, what's more, importers and exporters are all to be charged.

Table 6 Transporting empty container Cost w_{ij} between any two CYs Yuan/TEU

	CY A	CY B	CY C	CY D	CY E	CY F
CY A	#	1370	1910	648	2308	1858
CY B	1370	#	1096	2018	3416	2506
CY C	1910	1096	#	2568	4218	3508
CY D	648	2018	2568	#	1660	1210
CY E	2308	3416	4218	1660	#	1570
CY F	1858	2506	3508	1210	1570	#

The empty container transportation direction between any two CYs is as shown in Table 7 (the left of '/'), where 0 represents two yards in one same district, and the empty container needs not to be transported by ship. 1 indicates that the CY of line-number transporting to a CY of row-number need to go through the voyage eastward. And −1 indicates that the CY of line-number transporting to a CY of row-number need to go through the voyage westward. The time cost of empty container transportation between any two CYs in the same district is as shown in Table 7 (the right of '/'), where # represents two yards in different district, and the number represents the time cost of land transportation between two CYs.

Container margin/shortage amount of each yard in eight different shipping schedules is shown in Table 5.

Table 7 The direction matrix/ the time matrix hour

No.	CY 1	CY 2	CY 3	CY 4	CY 5	CY 6
CY 1	#/#	0/2	0/5	1/#	1/#	1/#
CY 2	0/2	#/#	0/5	1/#	1/#	1/#
CY 3	0/5	0/5	#/#	1/#	1/#	1/#
CY 4	−1/#	−1/#	−1/#	#/#	0/4	0/6
CY 5	−1/#	−1/#	−1/#	0/4	#/#	0/3
CY 6	−1/#	−1/#	−1/#	0/6	0/3	#/#

Under these input conditions above, GA parameters are set as: crossover probability is 0.8, mutation probability is 0.1, and the population size is 50. The generation of evolution is found better at around 300. During this simulation, optimal fitness appears in 283 generations, transportation cost is 156,300 Yuan. Populations' average transportation cost is 338,631.1 Yuan. The mean-square deviation is 59048.5, as shown in generations of fitness curves in Fig. 4.

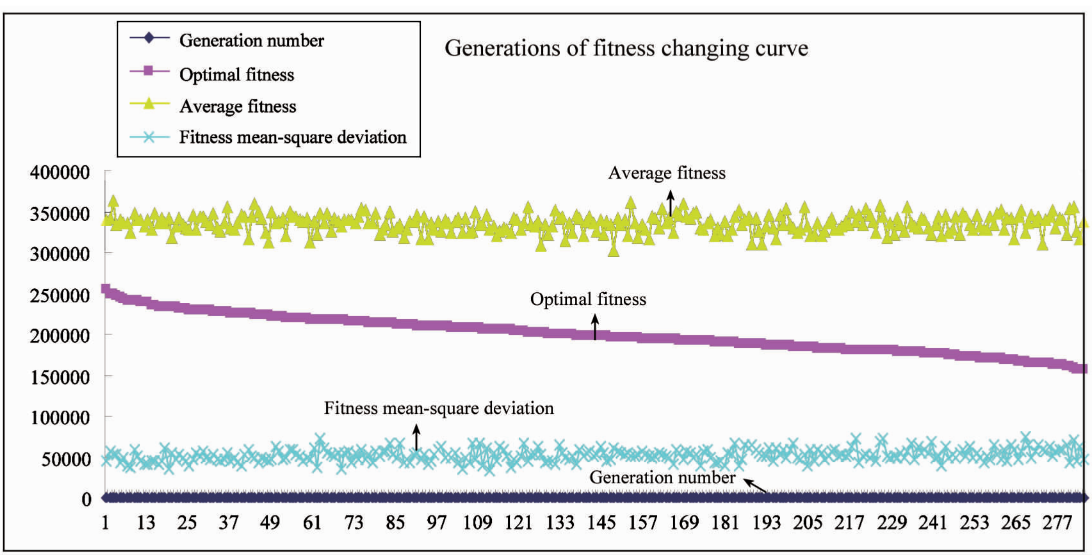


Fig. 4 Generations of fitness changing curve

4 Conclusion

According to the characteristics of the application resource management in Cloud, a resource model RSM is constructed. Based on it, a resource service management architecture RSMA is established, and a general resource scheduling model GROATW is introduced into the RSMA. Finally, they are applied successfully to double-district multi-ship-scheduling multi-container-yard empty containers transporting and solved by GA. The application proves the feasibility and validity of RSM. This application is related to several different CYs and shipping companies, corresponding with the characteristics of the RMC. The RSMA shows good openness and scalability and achieves a seamless integration of the distributed self-management and centralized scheduling. It can monitor and schedule application resources dynamically through the RSMA, achieve resource sharing and promote collaboration between enterprises to improve their overall competitiveness. For the next step, further study will be done to accumulate data for analyzing, and improve data analysis algorithms by means of the model's openness and scalability, promote the further evolution of analysis and optimization models.

References

[1] Mell P, Grance T. The NIST definition of cloud computing. *National Institute of Standards and Technology*, 2009, 53(6): 50

[2] Doelitzscher F, Sulistio A, Reich C, et al. Private cloud for collaboration and e-Learning services: from IaaS to SaaS. *Computing*, 2011, 91(1): 23-42

[3] Buyya R. Economic-based Distributed Resource Management and Scheduling for Grid Computing: [Ph. D dissertation]. Melbourne: Monash University, 2002. 30-35

[4] Lassila O, Swick R R. WD-rdf-syntax-19981008 Resource Description Framework (RDF) Model and Syntax Specification. US: W3C, 1998

[5] McBride B. Handbook on Ontologies. Berlin: Springer Berlin Heidelberg, 2004. 51-65

[6] McCuinness D L. Ontologies come of age. In: Spinning the semantic web: bringing the World Wide Web to its full potential. Cambridge: MIT Press, 2002. 171-194

[7] Peng J, Zhang X, Lei Z, et al. Comparison of several cloud computing platforms. In: Proceedings of the 2009 IEEE International Symposium on Information Science and Engineering, Shanghai, China, 2009. 23-27

[8] Wang Z J, Xu X F. Bilateral resource integration service mode for value innovation. *Computer Integrated Manufacturing Systems*, 2009, 15(011): 2216-2225 (In Chinese)

[9] Buyya R, Yeo C S, Venugopal S. Market-oriented cloud computing: vision, hype, and reality for delivering it services as computing utilities. In: Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, Dalian, China, 2008. 5-13

[10] Zhang J X. Survey of research progress on cloud computing. *Application Research of Computers*, 2010, 27(2): 429-433 (In Chinese)

[11] Keahey K, Figueiredo R, Fortes J, et al. Science clouds: Early experiences in cloud computing for scientific applications. *Cloud computing and applications*, 2008 (2008): 825-830

[12] Li Q, Zheng X. Research survey of cloud computing. *Computer Science*, 2011, 38(4): 32-37 (In Chinese)

[13] Li B H, Zhang L, Wang S L, et al. Cloud manufacturing: a new service-oriented networked manufacturing model. *Computer Integrated Manufacturing Systems*, 2010, 16(1): 1-7 (In Chinese)

[14] Li B H, Zhang L, Ren L, et al. Further discussion on cloud manufacturing. *Computer Integrated Manufacturing Systems*, 2011, 17(3): 449-457 (In Chinese)

[15] Casati F, Ilnicki S, Jin L J, et al. eFlow: A platform for developing and managing composition e-services. In: Proceedings of the IEEE Academia/Industry Working Conference on Research Challenges, New York, USA, 2000. 341-348

[16] Czajkowski K, Foster I, Karonis N, et al. A resource management architecture for metacomputing systems. In: Job Scheduling Strategies for Parallel Processing. Heidelberg, Germany, 1998. 62-82

[17] Raman R, Livny M, Solomon M. Matchmaking: Distributed resource management for high throughput computing. In: Proceedings of the IEEE International Symposium on High Performance Distributed Computing, Chicago, USA, 1998. 140-146

[18] Xu X. From cloud computing to cloud manufacturing. *Robotics and computer-integrated manufacturing*, 2012, 28 (1): 75-86

Zhang Xiaodong, born in 1971. He received his M. S. degree in School of Computer Science and Technology, Harbin Institute of Technology. His research focuses on software service engineering and service computing, cloud computing.