Computers & Geosciences 47 (2012) 46–59

Contents lists available at SciVerse ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Geo-information processing service composition for concurrent tasks: A QoS-aware game theory approach

Haifeng Li [a,b,*], Qing Zhu [c], Xiaoxia Yang [d], Linrong Xu [a]

[a] School of Civil Engineering, Central South University, China
[b] School of Geosciences and Info-Physics, Central South University, China
[c] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, China
[d] College of Earth Sciences, Chengdu University of Technology, China

ARTICLE INFO

Article history:
Received 29 April 2011
Received in revised form
12 October 2011
Accepted 13 October 2011
Available online 18 November 2011

Key words:
Geo-information processing service composition
Task concurrency
QoS
Non-cooperative game
Best response

ABSTRACT

Typical characteristics of remote sensing applications are concurrent tasks, such as those found in disaster rapid response. The existing composition approach to geographical information processing service chain, searches for an optimisation solution and is what can be deemed a "selfish" way. This way leads to problems of conflict amongst concurrent tasks and decreases the performance of all service chains. In this study, a non-cooperative game-based mathematical model to analyse the competitive relationships between tasks, is proposed. A best response function is used, to assure each task maintains utility optimisation by considering composition strategies of other tasks and quantifying conflicts between tasks. Based on this, an iterative algorithm that converges to Nash equilibrium is presented, the aim being to provide good convergence and maximise the utilisation of all tasks under concurrent task conditions. Theoretical analyses and experiments showed that the newly proposed method, when compared to existing service composition methods, has better practical utility in all tasks.

1. Introduction

The quality status of a remote sensing field can be described as "data-rich but analysis-poor" (Clery and Voss, 2005; Durbha and King, 2005; Baraniuk, 2011). Service-oriented science is a promising approach to changing modern geographical information (GI) processing methods within the scientific community (Foster, 2005), and it may allow manual data processing tasks to be automated via a geo-information processing service chain(GIPSC) (Alameh, 2003) by aggregating and combining small granularity GI services into complex processing service chains (PSC) to provide on-demand GI services flexibly. In distributed geospatial service environments, many services (referred to as concrete services (Canfora et al., 2008)) have the same level of functionality (referred to as an abstract service (Canfora et al., 2008)) but are fundamentally different in their non-functional attributes, such as performance, availability and reliability. These are also known as Quality of Service attributes (QoS). Each abstract service can be bound to one or more concrete services. With broad standardisation, an increasing number of functionally similar GI services are available on the Internet. When creating and executing a PSC, the number of component services involved in the PSC may be large; in addition, the number of GI services from which these component services are selected may be even larger. Hence, binding a concrete service to each abstract service, within the GIPSC, and according to a utility function under single user QoS constraints, imposed by the service level agreement (SLA) (called QoS-aware GI services composition), is a substantial challenge (Onchaga, 2006).

A large number of concurrent tasks, however, often exist within GI service-based applications, especially in crisis-orientated management. Therefore, if every task "selfishly" seeks an optimum solution without considering the performance of the entire service system, conflicts will emerge where each task competes for limited resources. This competition means that many tasks will be assigned to the same optimal service at the same time, resulting in the degradation of GI service capability, subsequently causing service quality to decline in all PSCs. This problem becomes even more serious for GI services, which are data dense and processing intensive in nature. In addition, concurrent tasks may form a queue for each GI service, and thus, the response time is not only influenced by the processing ability of the GI service itself but also by the number of tasks to be processed. The calculation of a PSC QoS aggregate value, particularly in terms of response time, is further complicated when considering the GIPSC control flow structure.

* Correspondence to: School of Civil and Architectural Engineering, Central South University, 22 Shaoshan South Road, Changsha 410004, PR China. Tel.: +86 15608418 565.
E-mail address: lihaifeng@csu.edu.cn (H. Li).

0098-3004/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2011.10.007

The goal of this paper, as indicated above, is to present a new approach to solve the conflict problem that results from competing optimal GI services caused by concurrent tasks in time-sensitive applications (Zhu et al., 2009), and to put forward a balancing mechanism for GIPSC with the aim of equally assigning the tasks to each GI service and hence maximising task utilisation. A non-cooperative game method is proposed to assign tasks to each GI service in a balanced manner, to decrease the conflict caused by competition for optimal services, and allow each task to achieve the highest performance. The tasks achieved in this study are as follows:

(i) Modelling the competition for optimal GI services as a non-cooperative game in which each task is to create a composite of concrete services to obtain the best utility according to the service composition strategies of the other tasks. To the best of the writer's knowledge, this is the first model that considers the GI service composition problem from the viewpoint of the entire system to address the issue of competition amongst the best GI services.
(ii) A queue theory-based GIPSC response time computing method, which considers the queue caused by concurrent tasks, is proposed. A mathematical model, i.e. a best response function, is then proposed to quantify task conflicts. An iterative algorithm is put forward that reaches the Nash equilibrium point based on the model.

Table 1
Sensors and the scope of cover in Wenchuan earthquake.

Sensor	Scope	Sensor	Scope
QB/World-View	8475 km^2	TerraSAR	75,160 km^2
SPOTS	98,253 km^2	ZY-2	6711 km^2
P5	59,185 km^2	Fore	57,574 km^2
EROS-B	447 km^2	COSMO	138,526 km^2
ALOS	132,249 km^2	RS-1	191,400 km^2
Total	7.7 million km^2		

The rest of this paper is organised as follows. The second section describes scenarios and applications related to the proposed method. The non-cooperative game-based model of QoS-aware GI services composition is discussed in the third section. Described in the fourth section is the best response iterative algorithm based on a best response function. The fifth section provides experimental results and analyses. Related work is described in the sixth section. A summary of all the work of this study, together with forecasts for further work are given in the seventh section.

2. Scenario

2.1. Application background

To illustrate the use of the proposed method, the Wenchuan earthquake which measured 8.0 on the Richter scale in China, at 14:28 on May 12, 2008, is used as an example. It resulted in a series of catastrophes (Li, 2008). The use of Air-borne and space-borne optical imagery and radar data is increasing, and the integration of such data with Geographic Information Systems (GISs), has opened a new dimension for scientists and planners/managers to identify, assess, and make decisions soon after deadly natural hazardous events and also in relation to rebuilding plans (Li, 2009; Singh, 2010). For example, the CBERS-02/02B satellite gathered 186 scene images during 13 May to 7 June, and 504 scene images before the earthquake. Beijing-1 satellite collected multispectral (32m) images covering the whole disaster area (1.39 million km^2) before the earthquake, and covered 3.49 million km^2 from 13 May to 4 June. Table 1 gives details on sensors and the scope of cover and Fig. 1 shows the region covered by different satellites after the Wenchuan earthquake

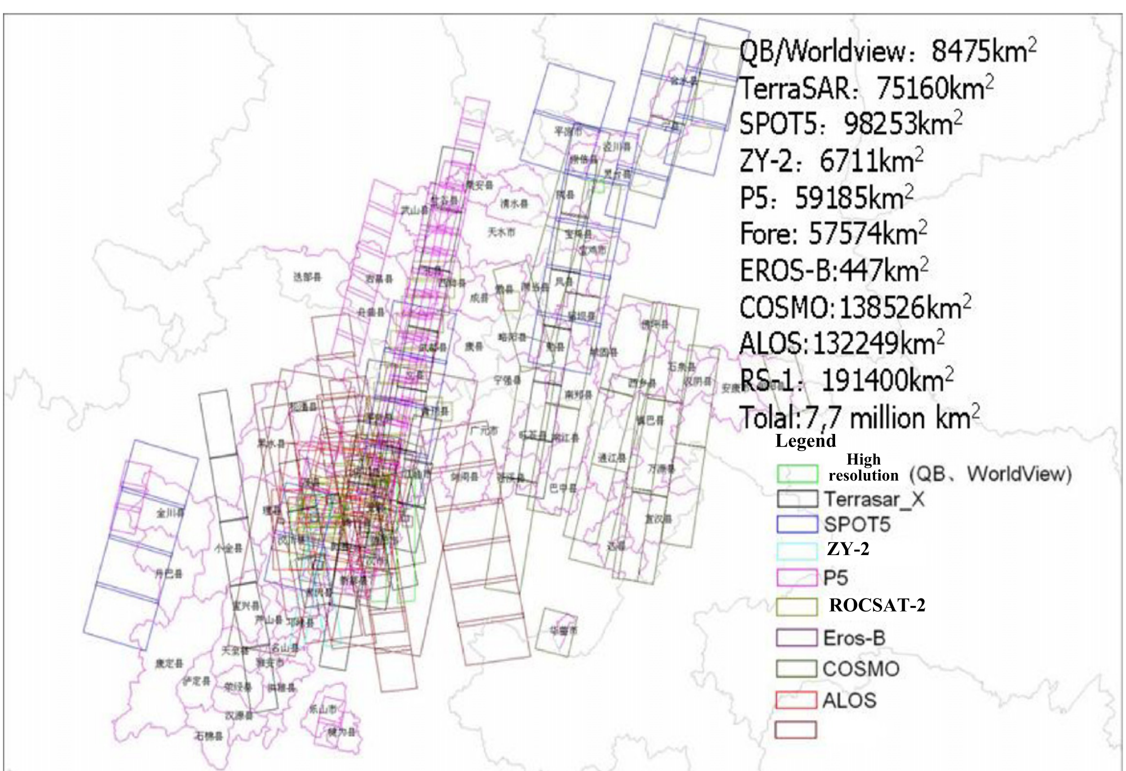

Fig. 1. Spatial distribution of remote sensing image (Li, 2009).

(Li, 2009). These both together show that much data can be obtained and the timely processing, and how the use of these data can be a challenge. In addition, different departments, involved in disaster mitigation have used spatial information techniques to provide decision making support and contributed much to earthquake relief (Li et al., 2008).

2.2. User requirements: a QoS view

In deriving user requirements, QoS characteristics of GI services are considered and some commonly defined quality criteria are also included.

Firstly, GI services, which are special remote sensing services, require large volumes of data and can therefore be bandwidth intensive. The process of GI is becoming more computing intensive. Hence, response time is an important factor. For example, in earthquake disaster, the response time should be controlled over a time scale measured in minutes (Li et al., 2010).

Secondly, user requirements can be specified including type of imagery, i.e., a choice between black and white and colour, image format such as jpeg, tiff, bitmap, spatial resolution, as well as cost due to budget constraints. Accordingly, interactivity, positional accuracy, completeness, consistency (format), and cost are relevant factors that can be specified for a GI service application (Onchaga, 2006).

Lastly, due to the urgency necessary during disaster response, availability is an important QoS criterion for a GI service. Additionally, other QoS criteria can be mapped in accordance with availability, e.g. latency (Onchaga, 2006).

2.3. The challenge

Change detection is a very important task in hazard assessment (Brunner et al., 2010). It is a technology for evaluating the degree of damage after an earthquake through the comparison of two different temporal images.[1] The change detection process flow can be modelled by the execution of a *PSC*. The main services included in the change detection PSC are a radiometric correction service, geometric correction service, image registration service and change detection service. The services function description is shown in Table 2. Fig. 2 shows change detection PSC using the GI services composition tool (Li et al., 2010) according to function property of services, and this PSC is used in the experiments.

As stated above, many departments are involved, such as assessment department, rescue department and decision-making department, in a disaster response. Further, there are very large number of remote sensing images with different degrees of temporal and spatial resolution gathered by continuous observation, e.g. the image coverage reached 7.7 million km^2 as shown in Table 1. The large number of applications, scene images will "squeeze" the performance of GISC if all the tasks are assigned to the same GI services, for instance, Digital Photogrammetric Grid (DPGrid), a parallel photogrammetric station, spends 20 h handling only 202 scene images (Zhang et al., 2008).

If every user (task) adopts a "selfish" strategy and maximises its own utility without regard for others, then every task would select the optimal resource (see dotted line in Fig. 3). Fig. 3 illustrates the concept of concurrent tasks. In the middle line is the change detection PSC composed of abstract services. Each abstract service can be bound to one or more concrete services. The dash-dot line denotes an equilibrium composition strategy in which each task is assigned to a different concrete service; the dotted line denotes a non-equilibrium composition strategy in which each task is assigned to the "best" concrete service. This strategy results in degradation in the performance of the services and thus diminishes the utility of all tasks

Hence, the allocation of tasks to particular services to assure all tasks achieve best performance or best utility is a challenge. Compared to the traditional method, the strategy employed in this research is to avoid sending all the tasks to the best services, hence "squeezing" the performance of each, but rather to achieve equilibrium. A better objective is to make sure that each task keeps optimal utility by considering others from the point of view of the system. A non-cooperative game based method provides a possible solution and is presented below.

Table 2
GI services included in the change detection PSC and corresponding function descriptions.

GI Service name	Service description
Radiometric correction (RC), $RC_1 \ldots RC_n$, concrete services achieve radiometric correction	RC is used to modify DN values of pre- and post-earthquake images to account for noise (i.e., contributions to the DN that are not a function of the feature being sensed).
Geometric correction (GC), $GC_1 \ldots GC_n$, concrete services achieve geometric correction	GC is used for the conversion of data to ground coordinates by the removal of distortions, due to sensor geometry, from pre- and post-earthquake images. For instance, software system was developed by Wuhan University that can perform rectification of optical and SAR imagery, such as SPOT 5, Quickbird, Terra SAR-X, and COSMOS, with limited ground control (Li, 2009).
Image registration (IR), $IR_1 \ldots IR_n$, concrete services achieve image registration	IR is used to transform pre- and post-earthquake images into one coordinate system. For example, the automatic selection of ortho image mosaic line by DPGrid (Zhang et al., 2008).
Changing detection (CD), $CD_1 \ldots CD_n$, concrete services achieve changing detection	CD is used for assessing the extent of damage after an earthquake according to pre- and post-earthquake images (Xu et al., 2010).

3. QoS-aware GI service composition method based on a non-cooperative game

3.1. The non-cooperative game model

A GI service composition comprised of concurrent tasks can be modelled as a non-cooperative game in which all tasks are dynamically adjusted according to the strategy of the other tasks. The target of this research study is to achieve Nash equilibrium (Fudenberg and Tirole, 1991), which is the state in which every task can attain the best utility by taking into account the strategy of all other tasks.

Definition 1. The non-cooperative game model of GI service composition is an assembly of the following players, strategy space, and utility function:

(1) *Players*: Supposing there are I types of continuous tasks, and every type of task[2] has the same QoS constraints. The arrival ratio of every task i follows the Poisson distribution with the

[1] We emphasise that abstract GIPSCs have existed previously and can be generated by GI service composition tools manually or automatically.

[2] Throughout this paper, "task" means a type of task flow with an arrival rate λ_i.

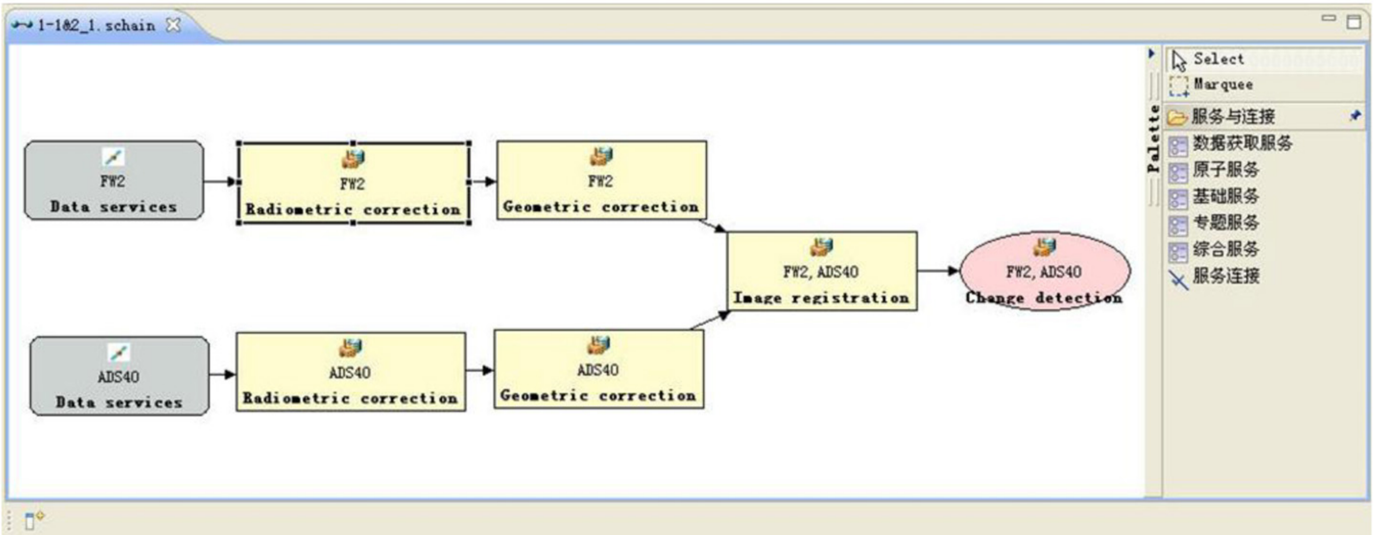

Fig. 2. *GIPSC* of change detection by GI services composition tool.

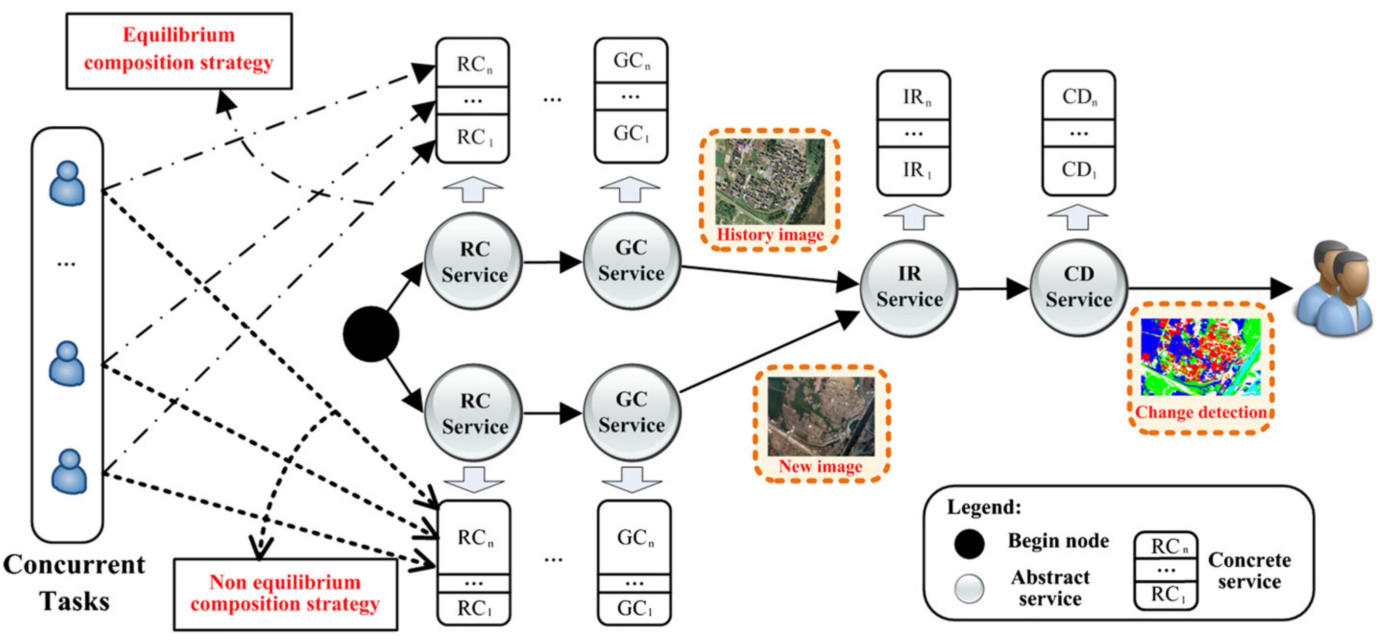

Fig. 3. Change detection PSC of concurrent tasks.

rate $\lambda_i, i=1,\ldots,I$. A GIPSC is composed of the abstract service set $\mathcal{A}=\{a_1,\ldots,a_L\}$, and every abstract service a_l comprises J concrete services written as $\mathcal{C}_l=\{c_1^l,\ldots,c_j^l\}$. Suppose the processing time of every concrete service follows an exponential distribution with the even speed μ_j.

(2) *Strategy*: Let $s_{i,j}^l$ represent the ratio of task i allocated to GI service j in step l, which is the serial number of service in PSC (see Fig. 4). Vector $s_i^l=(s_{i,1}^l,\ldots,s_{i,J}^l)$ represents the service combination strategy of task i in step l, and the service combination strategy of task i is $s_i=(s_i^1,\ldots,s_i^l)$; vector $s=(s_1,\ldots,s_I)$ is the combination strategy for all tasks, which is called the combination strategy of GI service composition. In Fig. 4, the dashed lines describe how the tasks are assigned to different concrete services denoted by $s_{i,j}^l\lambda_i$, for example, $s_{1,2}^1\lambda_1$ signifies that the $s_{1,2}^1$ ratio of the first task with rate λ_1 (such as 10 tasks in 1 min), was assigned to the second set of concrete services at the first abstract service, i.e. the first step. When the tasks were assigned to the same concrete service, the competition relationship between the tasks was formed (note the arc line in the figure).

(3) *Utility function*: $\mathcal{U}_i(s)$ is the expected utility of every task in the GIPSC execution. Task i selects s rather than s', iff $\mathcal{U}_i(s) \prec \mathcal{U}_i(s')$.

The essence of the model is that the combination strategy for every task is the best response compared to the other strategies. The best response of each task can be calculated through the notion of processing ability of the concrete service. As shown in Fig. 4, the processing ability of the concrete service can be described as two parts: the loaded part to which the processing resource has been allotted (see the shaded part of concrete service), the residual load part in which the processing resource is still free (see the blank part of the concrete service). The calculation method for the residual load is discussed in Section 4.1. Consequently, the best response can be used to define the competitive relationship between concurrent tasks: the best response of task i to combination strategy s_{-i} is $s_i^* \in s$, which

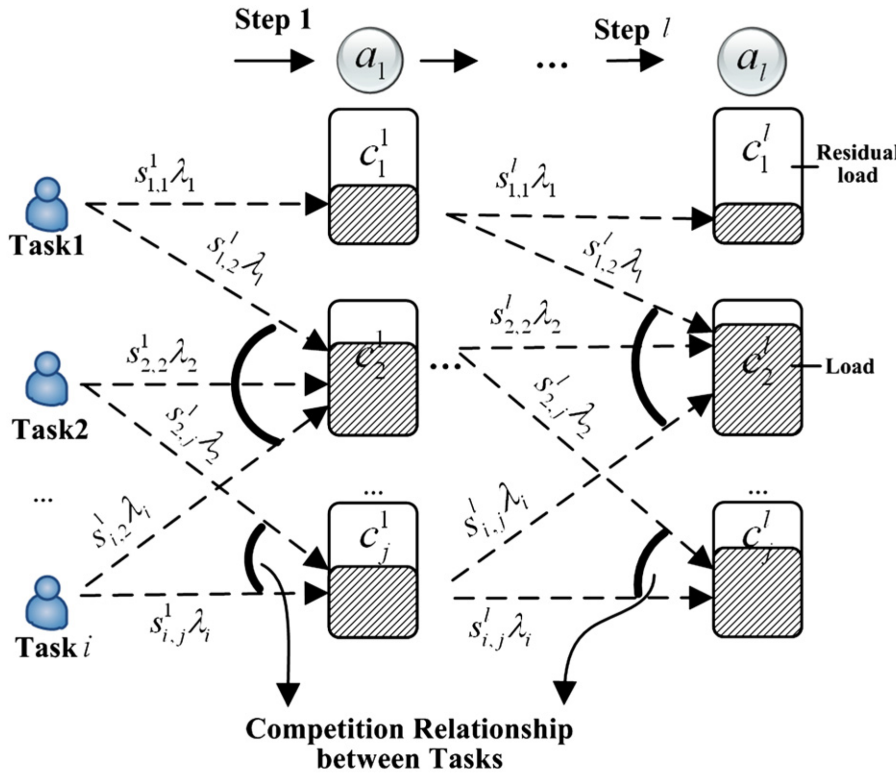

Fig. 4. Non-cooperative game model of GI service composition.

means that the other task strategies s_i, will not have a utility greater than s_i^* (Weibull, 1997)

$$\mathcal{U}_i(s_i^*,s_{-i}^*) \prec \mathcal{U}_i(s_i,s_{-i}^*) \tag{1}$$

The equation $s_{-i}=(s_1,\ldots,s_{i-1},s_{i+1},\ldots,s_n)$ represents the combination strategy of all players excluding player i.

The GIPSC QoS model under concurrent task conditions is next defined.

3.2. QoS model of GIPSC for concurrent tasks

To denote the accumulation, multiplication, and extreme value theorem, cost, availability, and response time are included in the QoS model. The cost and availability can be regarded as being uninfluenced by the concurrent tasks; the response time, however, will change because of the existing queue, which can be modelled as $M/M/1$.

In the $M/M/1$ queue model, the arrival time interval for every task i follows an exponential distribution with a speed of λ_i, and the processing ability of every GI service time follows an exponential distribution with the parameter μ. The processing time, therefore, for each service is $W=1/(\mu-\lambda_i)$ (Gross and Harris, 1985). The aggregation response time of the GIPSC is calculated according to four basic structures: sequential, parallel, branch, and loop (Martin et al., 2006).

(1) *Sequential structure*: According to Burke's theorem (Gross and Harris, 1985), for an $M/M/1$ queue with an arrival rate of λ, the output is also a Poisson process with a rate of λ. That is to say, for all GI services of a sequential structure, the arrival and departure processes follow the Poisson distribution. As a result, the method for computing the response time of the sequential structure is $W=\sum_{l=1}^{L} 1/(\mu^l-\lambda)$, where L represents the total length of the sequential structure and l indicates the step index in the sequential structure (Fig. 5). The total response time of the sequential structure is accumulated by all services including that of the PSC.

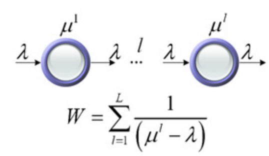

Fig. 5. Computational model of the response time of a sequential structure.

(2) *Parallel structure*: In a parallel structure, the total response time is determined by the longest parallel branch (i.e., the key path) (Zhu et al., 2009). Therefore, to solve the response time for a parallel structure, the key path must first be solved. The parallel structures can then be serialised (Fig. 6). Since the response time is decided by the key path in the parallel branch, the total response time of the parallel structure is the accumulated times for all services including those in the key path of PSC. The response time of the parallel structure is computed as $W=\max_M \sum_{l=1}^{L} 1/(\mu_m^l-\lambda)$, where k_p represents the number of parallel branches and M indicates the branch index of the parallel structure.

(3) *Branch structure*: The branch structure describes the possibility of an execution route being selected if there are $N=\{1,\ldots,n\}$ branches and the possibility that every branch n chosen is b_n, the sum of which satisfies $\sum_{n=1}^{N} b_n=1$. Accordingly, the arrival rate of every branch task is $b_n\lambda$ (Fig. 7). Tasks are allocated to different branches, with different possibilities within the branch structure. Thus, the serialisation method can be used to calculate the response time (Fig. 7). The response time of the branch structure is calculated as $W=\sum_{n=1}^{N}\sum_{l=1}^{L} 1/(\mu_n^l-b_n\lambda)$.

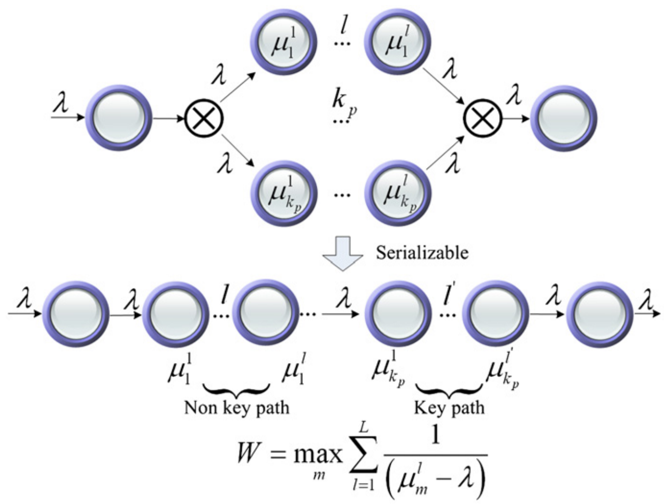

Fig. 6. Computational model of the response time of a parallel structure.

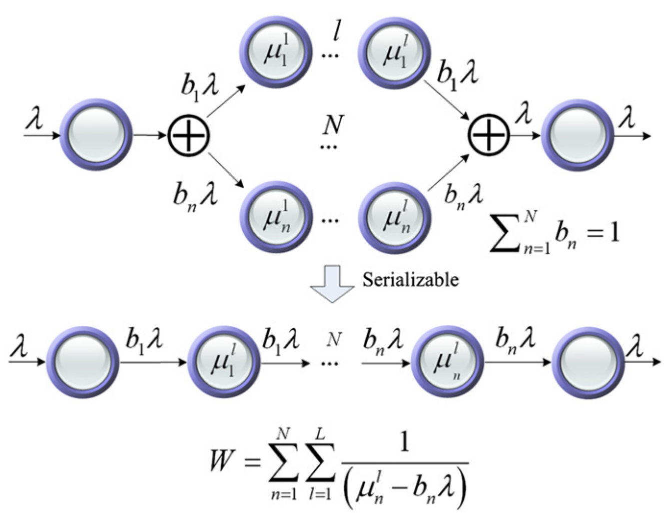

Fig. 7. Computational model of the response time of a branch structure.

(4) *Loop structure*: In contrast to loop peeling and unfolding methods (Ardagna and Pernici, 2007), the loop structure is considered as feedback to the execution within the queue model (Fig. 8). The loop structure is a part of the $M/M/1$ queue network. According to the Jackson theorem, supposing the internal arrival rate of the loop structure is r, the feedback possibility is $1-p$, and consequently:

$r = \lambda + (1-p)r$, or $r = \lambda/p$

The response time in the loop structure is $W = \sum_{l=1}^{L} 1/\mu_l - \lambda/p = \sum_{l=1}^{L} p/(p\mu_l - \lambda)$.

By recognising that the total tasks assigned to each GI service is $\sum_{m=1}^{I} s_{m,j}^{l}\lambda_m$, the response time of the GIPSC execution can be determined for each task i. All QoS aggregate functions are listed in Table 3. For cost calculating methods and availability see Zeng et al. (2004) for details not introduced here, due to space limitations.

Theorem 1. *Parallel, branch, and loop structures in the GIPSC can be serialised with equal values to attain a sequential structure without changing the values of the aggregate function.*

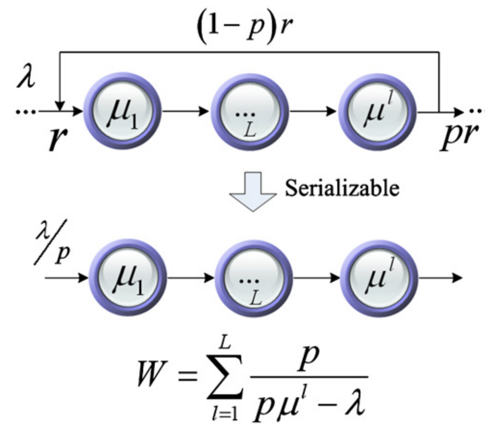

Fig. 8. Computational model of the response time of a loop structure.

Proof. A cost aggregate function is calculated by summation. Although the availability function is calculated by multiplication, these functions can be converted into summation via a logarithmic function. The response time aggregate function is calculated by maximum or minimum value, but after finding the key path, the calculation can still be converted into a summation. Therefore, these calculations are linear structures and can be serialised.

The significance of serialisation lies in the removal of the different expressions of QoS dimension aggregate functions due to the difference in control structure and in building a unified model. Here, the structure factor κ is used to unify the expression.

Definition 2. Structure factor κ depicts the compensator value of the serialisation for different control structures

$$\kappa = \begin{cases} 1, N=1 & \text{if } l \in S \text{ or } l \in P \\ \eta, N=1 & \text{if } l \in C \\ b_{k_n} & \text{if } l \in R \end{cases}$$

$$F_i(\mathscr{L}) = \sum_{l=1}^{L}\sum_{j=1}^{J} \kappa f_{n,j}^{l} s_{i,j}^{l} \lambda_i$$

$$\ln(AV_i(\mathscr{L})) = \sum_{l=1}^{L}\sum_{j=1}^{J} \kappa s_{i,j}^{l} \lambda_i \ln(av_j^l)$$

$$T_i(\mathscr{L}) = \sum_{l \in cp}^{L}\sum_{j=1}^{J} \frac{s_{i,j}^{l}}{(\mu_j^l - \kappa \sum_{m=1}^{I} s_{m,j}^{l}\lambda_m)}$$

where $\mathscr{L}$ is the PSC after serializing.

3.3. The computational model of the GIPSC utility function

QoS dimension values differ in their range and units, and normalisation is the first step. Formulae (2) and (3) represents the positive and negative rule, respectively

$$v_i^h = \left\{ \begin{array}{ll} \frac{q_i^h - \min q_i^h}{\max q_i^h - \min q_i^h} & \text{if } \max q_i^h \neq \min q_i^h \\ 1 & \text{if } \max q_i^h = \min q_i^h \end{array} \right\} \quad (2)$$

$$v_i^h = \left\{ \begin{array}{ll} \frac{\max q_i^h - q_i^h}{\max q_i^h - \min q_i^h} & \text{if } \max q_i^h \neq \min q_i^h \\ 1 & \text{if } \max q_i^h = \min q_i^h \end{array} \right\} \quad (3)$$

Cost, availability, and response time are represented by q_1, q_2 and q_3, respectively; $\max q_i^h$ ($\min q_i^h$) represents the maximum

Table 3
QoS aggregate function for GIPSC[a]

Control structure	Cost (F_i)	Availability ($\ln(AV_i)$)	Response time (T_i)
Sequential(S)	$\sum_{l\in S}\sum_{j=1}^{J} f_j^l s_{i,j}^l \lambda_i$	$\sum_{l\in S}\sum_{j=1}^{J} s_{i,j}^l \lambda_i \ln(av_j^l)$	$\sum_{l\in S}\sum_{j=1}^{J} \frac{s_{i,j}^l}{(\mu_j^l - \sum_{m=1}^{I} s_{m,j}^l \lambda_m)}$
Parallel (P)	$\sum_{l\in P}\sum_{j=1}^{J} f_j^l s_{i,j}^l \lambda_i$	$\sum_{l\in P}\sum_{j=1}^{J} s_{i,j}^l \lambda_i \ln(av_j^l)$	$\max_{cp\in P}\sum_{l\in cp}\sum_{j=1}^{J} \frac{s_{i,j}^l}{(\mu_j^l - \sum_{m=1}^{I} s_{m,j}^l \lambda_m)}$
Loop (C)	$\eta\sum_{l\in C}\sum_{j=1}^{J} f_j^l s_{i,j}^l \lambda_i$	$\eta\sum_{l\in C}\sum_{j=1}^{J} s_{i,j}^l \lambda_i \ln(av_j^l)$	$\sum_{l\in C}\sum_{j=1}^{J} \frac{s_{i,j}^l}{(\mu_j^l - \eta\sum_{m=1}^{I} s_{m,j}^l \lambda_m)}$
Branch (R)	$\sum_{k_n=1}^{k_r}\sum_{l\in C_{k_n}}\sum_{j=1}^{J} b_{k_n} f_{n,j}^l s_{i,j}^l \lambda_i$	$\sum_{k_n=1}^{k_r}\sum_{l\in C_{k_n}}\sum_{j=1}^{J} b_{k_n} s_{i,j}^l \lambda_i \ln(av_j^l)$	$\sum_{k_n=1}^{k_r}\sum_{l\in C_{k_n}}\sum_{j=1}^{J} \frac{s_{i,j}^l}{(\mu_j^l - b_{k_n}\sum_{m=1}^{I} s_{m,j}^l \lambda_m)}$

[a] *Note*: In the response time function, $\eta=1/p$ represents the feedback probability; in the cost and availability function, $\eta=\overline{p}$ represents the expected recycle times. The key path is represented by *cp*.

(minimum) aggregation value of the hth QoS dimension in the i task; $\max q_i^h - \min q_i^h$ is the normalised factor written as NF_i^h.

For each task i, the utility function can be expressed as $\mathcal{U}_i(s) = \sum_{h=1}^{3} w_h \mathcal{U}_i^h$, where w_h represents the QoS weights, and $\mathcal{U}_i^h$ is the utility of the hth dimension of the i task. The normalisation results are

$$v_i^1 = \frac{\mathcal{U}_i^1 - \min q_i^1}{NF_i^1}, \quad \mathcal{U}_i^1 = F_i(\mathscr{L}) \tag{4}$$

$$v_i^2 = \frac{\max q_i^2 - \mathcal{U}_i^2}{NF_i^2}, \quad \mathcal{U}_i^2 = AV_i(\mathscr{L}) \tag{5}$$

$$v_i^3 = \frac{\mathcal{U}_i^3 - \min q_i^3}{NF_i^3}, \quad \mathcal{U}_i^3 = R_i(\mathscr{L}) \tag{6}$$

The normalisation process is described in detail by Zeng et al.(2004).

4. Iteration algorithms with the best response

4.1. The best response of a single task

First, the quantitative model is established for the competitive relationship between concurrent tasks using the residual computing ability (RCA) of each GI service (Grosu and Chronopoulos, 2005).

Definition 3. RCA is the available processing ability of the j concrete service in the a^l abstract service of task i:

$$\mu_{i,-j}^l = \mu_j^l - \sum_{m=1, m\neq i}^{I} \kappa s_{m,j}^l \lambda_m \tag{7}$$

Obtained is $\mu_j^l - \sum_{m=1}^{I} s_{m,j}^l \lambda_m = \mu_{i,-j}^l - \kappa s_{i,j}^l \lambda_i$, and therefore $T_i(\mathscr{L}) = \sum_{l\in cp}^{L}\sum_{j=1}^{J}(s_{i,j}^l/(\mu_{i,-j}^l - \kappa s_{i,j}^l \lambda_i))$.

Obviously, tasks will impact each other by using the processing ability of GI services. The RCA illustrates how many abilities of the GI service remain when a task is assigned to this service (see Fig. 4). It is thus natural to question how the task chooses the optimal strategy under this condition.

Definition 4. Considering the RCA, the optimal problem $\mathcal{OP}_i$ of each task is

$$\min \mathcal{U}_i(s) = \sum_{h=1}^{3} w_i^h v_i^h \tag{8}$$

$$\text{s.t. } s_{i,j}^l \geq 0, \forall l \in L, \quad i = 1,\ldots,I, \quad j = 1,\ldots,J \tag{9}$$

$$\sum_{j=1}^{J} s_{i,j}^l = 1, \forall l \in L \tag{10}$$

$$\sum_{i=1}^{I} s_{i,j}^l \lambda_i \leq \mu_j, \forall l \in L \tag{11}$$

$$\frac{(-1)^{h+1}\mathcal{U}_i^h}{\lambda_i} \leq (-1)^{h+1} D_h \quad \forall h = 1,2,3 \tag{12}$$

where the feasible constraint conditions are as follows:

Formula (9) describes every GI server allocated with non-negative tasks;
Formula (10) describes conservation conditions, representing every task i allocated to GI services;
Formula (11) states stable conditions, showing that any arrival rate is smaller than the largest service rate to guarantee that the system does not "explode" because of queuing;
Formula (12) explains the constraint conditions of the task, where D_h, $h=1,2,3$ represent the average cost, availability, and response time constraint of task i.

Definition 5. The best response strategy (BR) of every task is the solution of $\mathcal{OP}_i$.

The proof of Theorem 2 demonstrates that the objective and constraint functions are second-order derivatives and are consistently greater than or equal to zero. Therefore, $\mathcal{OP}_i$ is a convex programming problem. Furthermore, a first-order Karush–Kuhn–Tucker condition is a necessary and sufficient condition for the $\mathcal{OP}_i$ solution to exist. The LaGrange function is

$$\begin{aligned} L(s_{i,1}^1,\ldots,s_{i,J}^1,\ldots,s_{i,J}^L,\alpha^1,\ldots\alpha^L,\beta_1^1,\ldots,\beta_J^1,\ldots,\beta_J^L,\gamma^1,\ldots,\gamma^h) \\ = \sum_{h=1}^{3} w_i^h v_i^h + \sum_{l=1}^{L}\alpha^l\left(\sum_{j=1}^{J} s_{i,j}^l - 1\right) + \sum_{l=1}^{L}\sum_{j=1}^{J}\beta_j^l s_{i,j}^l \\ + \sum_{h=1}^{3}\gamma^h\left(\frac{\mathcal{U}_i^h}{\lambda_i} - D_h\right) \end{aligned} \tag{13}$$

Based on the Karush–Kuhn–Tucker condition, $\alpha^l, \gamma^h \geq 0, \beta_j^l \geq 0$ for any $l = 1,\ldots,L$, $j = 1,\ldots,J$, $h=1,2,3$, the necessary and sufficient condition between $s_{i,j}^l$ and the solution of $\mathcal{OP}_i$ is

$$\frac{\partial L}{\partial s_{i,j}^l} = 0 \tag{14}$$

$$\frac{\partial L}{\partial \alpha^l} = 0 \tag{15}$$

$$\beta_j^l s_{i,j}^l = 0 \tag{16}$$

$$\gamma^h\left(\frac{(-1)^{h+1}\mathcal{U}_i^h}{\lambda_i}-(-1)^{h+1}D_h\right)=0 \tag{17}$$

Constraint condition (10) is the assumed condition of the entire system stability, and it can be considered as always true. Therefore,

$$\mathcal{O}_j^l+\mathcal{P}_j^l+\mathcal{Q}_j^l=0 \tag{18}$$

$$\sum_{j=1}^{J}s_{i,j}^l=1 \tag{19}$$

$$\beta_j^l s_{i,j}^l=0 \tag{20}$$

$$\gamma^h\left(\frac{(-1)^{h+1}\mathcal{U}_i^h}{\lambda_i}-(-1)^{h+1}D_h\right)=0 \tag{21}$$

$$l=1,\ldots,L, \quad j=1,\ldots,J, \quad h=1,2,3 \tag{22}$$

where

$$\mathcal{O}_j^l=\frac{w_i^1\delta 1_{i,j}^l}{NF_i^1}-\frac{w_i^2\delta 2_{i,j}^l}{NF_i^2}+\frac{w_i^3}{NF_i^3}\frac{\mu_{i,-j}^l}{(\mu_{i,-j}^l-\delta 3_{i,j}^l s_{i,j}^l)^2} \tag{23}$$

$$\mathcal{P}_j^l=\alpha^l+\beta_{i,j}^l \tag{24}$$

$$\mathcal{Q}_j^l=\gamma^1\delta 1_{i,j}^l-\gamma^2\delta 2_{i,j}^l+\frac{\gamma^3\mu_{i,-j}^l}{(\mu_{i,-j}^l-\delta 3_{i,j}^l s_{i,j}^l)^2} \tag{25}$$

$$\delta 1_{i,j}^l=\kappa f_j^l\lambda_i, \quad \delta 2_{i,j}^l=\kappa\lambda_i\ln(av_j^l), \quad \delta 3_{i,j}^l=\kappa\lambda_i \tag{26}$$

These nonlinear equations can be solved using Matlab toolbox, and the best response combination strategy $\overline{s}_i^{(x)}$ can be obtained in any iterations of task i by solving these equations.

4.2. The existence of Nash equilibrium

The existence of Nash equilibrium is a basic question for the non-cooperative game of GI service composition. Here, the sufficient condition of the existence of Nash equilibrium is discussed.

Theorem 2. *The non-cooperative game of GI service composition has the unique Nash equilibrium solution if* $s_{i,j}^l$,$\mathcal{K}_i^1+\mathcal{K}_i^3>\mathcal{K}_i^2$ holds where $\mathcal{K}_i^h=\partial v_i^h/\partial s_{i,j}^l$ (proof: Appendix A).

4.3. Best response iteration algorithm of concurrent tasks

The core concept behind using a best response-based algorithm to solve the Nash equilibrium is that every task resets its own combination strategy, based on other combination strategies. This process is iterated until it converges to Nash equilibrium.

For instance, the first task uses the initialisation value to obtain the combination strategy $\overline{s}_1^{(1)}$, where the superscript 1 is the iteration number. The second task then, obtains the combination strategy $\overline{s}_1^{(1)}$ in accordance with the previous task; that is, the best response of task 2 is compared with that of other tasks and so on until the last task gains its corresponding composition strategy $\overline{s}_I^{(1)}$ according to the combination strategy of the previous task $I-1$. Be noticed that each $\overline{s}_I^{(1)}$ can be set to zero for the initialisation value of it is not influence on the algorithms which is discussed in Section 5.2.1 in details. At the end of each iteration, $L_1=\sum_{i=1}^{I}|\mathcal{U}_i^{(x-1)}-\mathcal{U}_i^{(x)}|$, assigned by *sum* (Fig. 9), is calculated to denotes the sum of the differences of the utilities for all tasks, and marks the system utility error of each iteration, where $|\mathcal{U}_i^{(x-1)}-\mathcal{U}_i^{(x)}|$ is the utility difference between the adjacent iteration of each task i. Subsequently, a second iteration round is implemented; but this time, the first task will update its own combination strategy based on the combination strategies of the other tasks.

The ε is a small number used to denote the acceptance tolerance which is variable according to user's demand. A smaller ε means that the iteration solution is more close to the Nash equilibrium, however, with more compute complexity in time. The iteration algorithms terminates, when L_1 is less than ε. The steps can be called a basic progressive system (ESS) (Boulogne et al., 2002) as shown in Fig. 9. The iteration process terminates at Nash equilibrium.

4.4. Convergence of the iteration algorithm

An important problem associated with the best response iteration algorithm is whether it can ultimately converge to Nash equilibrium.

Theorem 3. *Beginning from any initial point, the iteration algorithm converges to the Nash equilibrium point, if it can satisfy* $\sum_{m=1,m\neq i}^{I}\sum_{l\in cp}^{L}\sum_{j=1}^{J}(\kappa s_{i,j}^l\lambda_i/(\mu_{m,-j}^l-\kappa s_{m,j}^l\lambda_i)^2)<1$ (proof: see Appendix A).

5. Experimental and analytical evaluation

5.1. Simulation environment

To test the efficiency of the proposed method, a simulation experiment was conducted. First, the abstract PSC (Figs. 3 and 2) was simulated. In this simulation every abstract service included 10 concrete services. Concrete services were modelled as an M/M/1 processing queue system. Afterwards, every QoS dimension value of a service was randomly created, while forcing all values to conform to a normal distribution (Cardoso, 2002).

5.2. Performance evaluation

The evaluation was conducted in terms of four aspects: algorithm convergence, task utility, fairness, and time complexity.

5.2.1. Algorithm convergence

One basic issue is whether the best response iteration algorithm can converge to the Nash equilibrium point. The algorithm initially set the combination strategy for every task as zero, and every task redesigns its own combination strategy according to the orderly combination strategy of other tasks. The experiment demonstrated that the best response iteration algorithm had good convergence and can subsequently converge to Nash equilibrium from any point (for the initial point is zero).

The algorithm convergence was tested by increasing the number of tasks and changing the norm, i.e., tolerable error, where the number of tasks ranged from 10 to 35, the average arrival rate of every task was λ_i=10, and the norm was reduced from 0.1 to 10^{-5} (Fig. 10). Fig. 10 shows the relationship between the norm (x-axis) and number of iterations (y-axis). It is found that the number of iterations increases approximately linearly with decrease of the norm, also, the number of iterations increases with the number of tasks to small extent. In all experiments, all the initial points were zero and L_1 was used to determine the termination condition. As the norm increased, the number of algorithm iterations increased, when the quantity of tasks was constant. Similarly, the number of the algorithm iterations increased as the number of tasks increased, when the tolerable error norm changed.

Fig. 11 denotes the relationship between norm (x-axis), number of task (y-axis), and number of iterations (z-axis $\log_2$ scaled), and also shows that the number of iterations increases

Variables:

$\bar{s}_i^{(x)}$ // composition strategy of task at the x^{th} iteration
$\mathcal{U}_i^{(x)}$ // utility value of task at the x^{th} iteration
x // the number of iteration.
$norm$// L^1 norm at the x^{th} iteration
$function\ Iterative\,(task\ i)$
 $x \leftarrow 0,\ norm, sum \leftarrow 1$ //sum is bigger than ε
 $\bar{s}_i^{(0)} \leftarrow 0, \mathcal{U}_i^{(0)} \leftarrow 0$
 $while\,(true)$
  // get the global value of x and sum
  $getGlobeInformation\,(x, sum)$
  $norm \leftarrow sum$
  $if\,(norm < \varepsilon)$
   $exit\ while$
  $\mathcal{BR}Iterative\,()$
  $x \leftarrow x + 1$
$end\ function$
$function\ \mathcal{BR}Iterative\,()$
 $sum \leftarrow 0$
 $for\,(each\ task\ i\ in\ task\ set\ \mathcal{T})\,(i = 1, \cdots, I)$
  $for\,(each\ step\ l\ in\ services\ chain)$
   $for\,(each\ concrete\ service\ j\ in\ step)$
    // calculate residual services capability $\mu_{i,-j}^l$ for each service
     $\mu_{i,-j}^l = \mu_j^l - \sum_{m=1,m\neq i}^{I} \kappa s_{m,j}^l \lambda_m$
  // calculate the best response composition strategy for each task at the x^{th} iteration
  $\bar{s}_i^{(x)} \leftarrow \mathcal{BR}$
  $sum \leftarrow sum + \left|\mathcal{U}_i^{(x-1)} - \mathcal{U}_i^{(x)}\right|$
 $updateGlobeInformation\,(x, sum)$
$end\ function$

Fig. 9. Iteration algorithm of the best response.

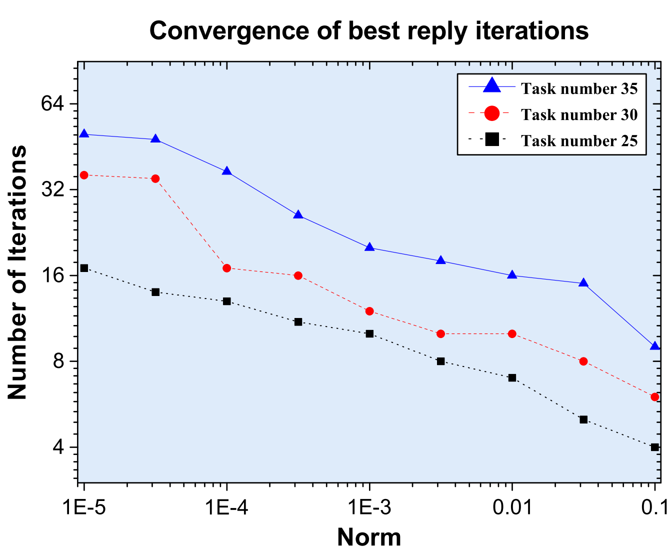

Fig. 10. Convergence comparison of the best response iteration algorithm.

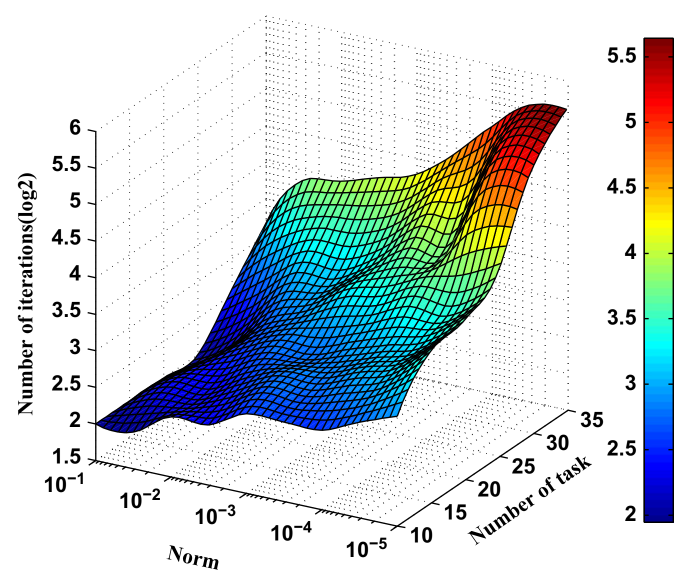

Fig. 11. Convergence surface of the best response iteration algorithm.

approximate linearly with the norm and number of tasks from a three dimensional viewpoint. For instances in the figure, when the number of tasks is 35 with the average arrival rate $\lambda_i = 10$, and the norm is 10^{-5}, the iteration number is 50 (the z-axis is $\log_2$ scaled).

During the process of algorithm convergence, every task constantly adjusted its own combination strategy according to the strategies of the other tasks; the load condition of every server also fluctuated regularly with changes in the combination strategies of the tasks and finally converged to a relatively fixed value. Fig. 12 shows the load change of a concrete service in the iterative convergence process, and illustrates how the single concrete service load responds to iterations from the concrete service point of view. The oscillation waveform in Fig. 12 represents process of competing services by tasks. For instance, a service

Fig. 12. Service task distribution convergence trajectory.

may be presented only a few tasks to choose from because of its relatively poor initial performance (wave trough in Fig. 12). In the second iteration round, however, its relative performance improves because no task was allocated; the service may cause load concentration (wave crest in Fig. 12). The fluctuation of the service load became very much smaller with each iteration process and ultimately converged at a fixed value.

5.2.2. *Task utility*

To evaluate the proposed method, two classic methods were compared. One was a classic distribution load balance method, and the other a classic global plan method developed for single tasks involving QoS-aware PSC composition:

A proportional scheme (PS) (Chow and Kohler, 1979) is a classic distribution method that employs a task combination strategy in which servers with higher processing ability acquire more tasks according to the particular processing ability of servers and the ratio distribution task equilibrium algorithm. Services with a smaller comprehensive utility value are allocated more tasks.

The mixed integer linear programming-based (MILP) method (Zeng et al., 2004) is a typical single-task service composition algorithm that does not consider the combination strategies of other tasks. This method attempts to find a global optimal solution under QoS constraints.

The goal of this study, as indicated in Section 4, is to minimise conflicts under concurrent task conditions to achieve the overall optimisation of all tasks. The expected task utility value (EXP) is calculated prior to GIPSC execution, without any consideration of the combination strategies of other tasks, denoting the expectation utility of tasks. EXP denotes the utility that each task can achieve in single task situation where no competition is considered. The actual task utility value (ACT) is the utility gained by the execution of the PSC. The value ERR is the difference between EXP and ACT, and denotes the performance difference caused by services competing.

As shown in Fig. 13, the histogram part gives the EXP value, the line part is the ERR value, and the ACT value is the sum of EXP and ERR, i.e. the total height of the histogram. Since utility is directly proportional to response time, a small utility is better. In

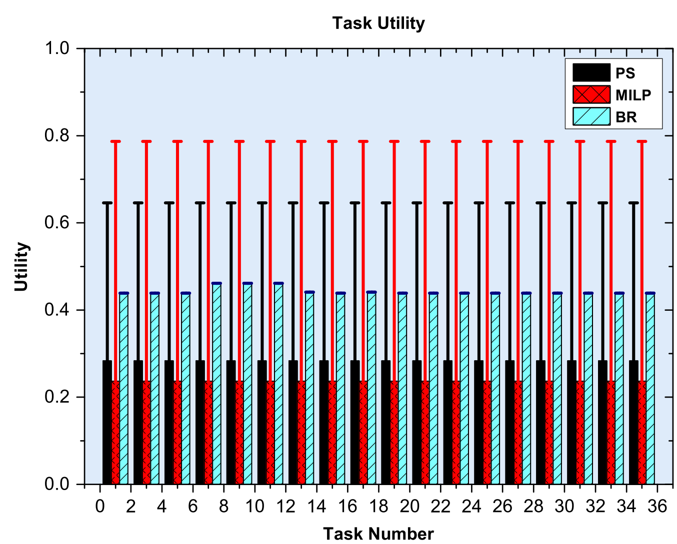

Fig. 13. Comparison between expected and actual task utilities.

contrast with PS and MILP methods (Fig. 13), although the BR's EXP is maximal (histogram part) among the three methods before GIPSC execution, the BR's ACT and ERR value are the minimum among the three methods after execution for considering the conflicts between tasks. Hence, the advantage of the BR method is that it ensures that each task can achieve a better utility for concurrent tasks, and an improvement can be achieved for the whole performance of concurrent tasks.

The BR method stipulates that every task constantly changes and finally converges to the Nash equilibrium point in accordance with other task composition strategies (Figs. 14 and 15).

In Fig. 14, the utility of each task (*y*-axis with 10 scaled) adjusts according to the iteration progress when considering the strategy of other tasks. At the beginning of the iteration, the utility of the current task is good (small) without other competitors. When other competitors are introduced, the utility of the task becomes worse than the initialisation value for the "best" services competing between tasks see the wave crest. The strategy of the task must be adjusted to adapt to other strategies.

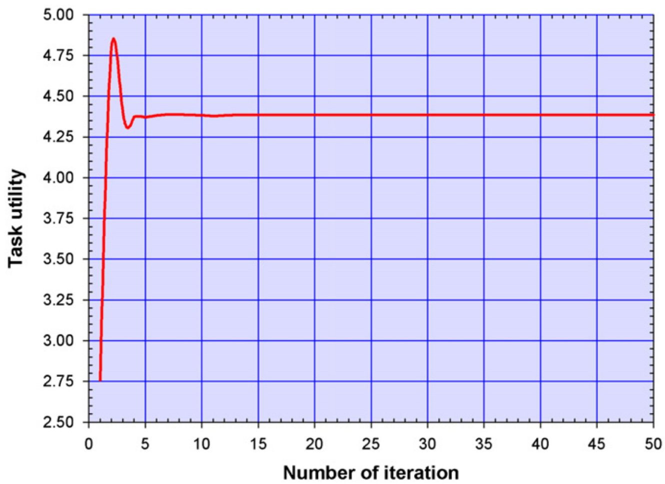

Fig. 14. Task utility convergence trajectory for one task.

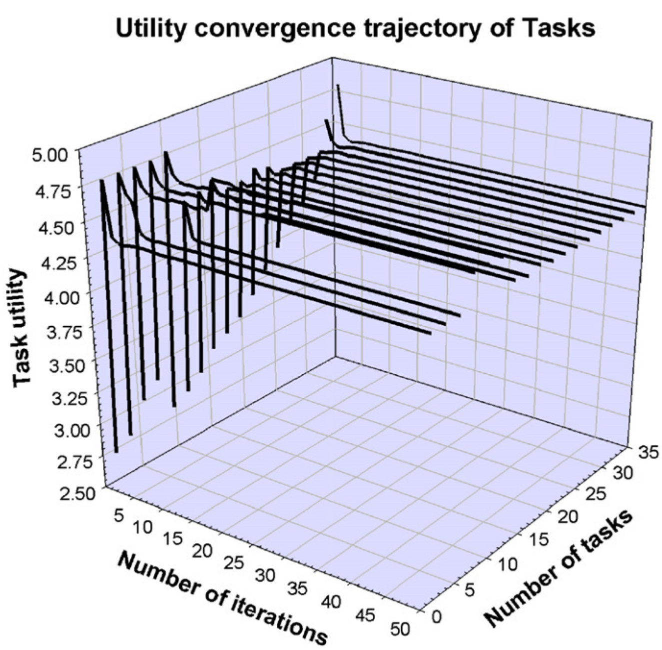

Fig. 15. Comparison convergence trajectory among actual task utilities.

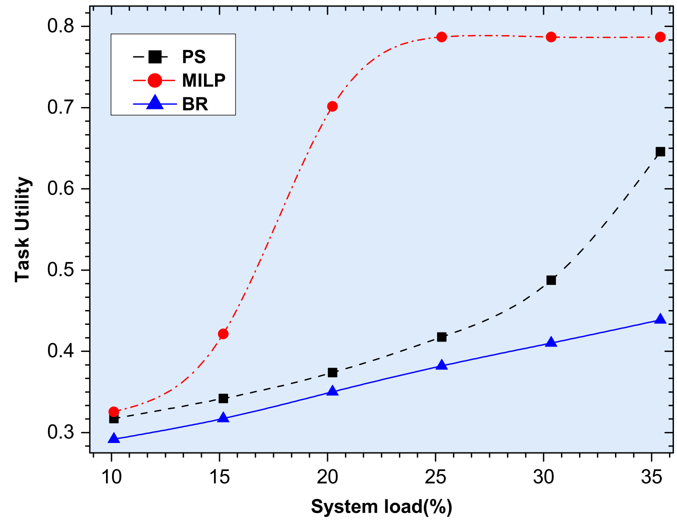

Fig. 16. Relationship between system load and ACT compared for three methods.

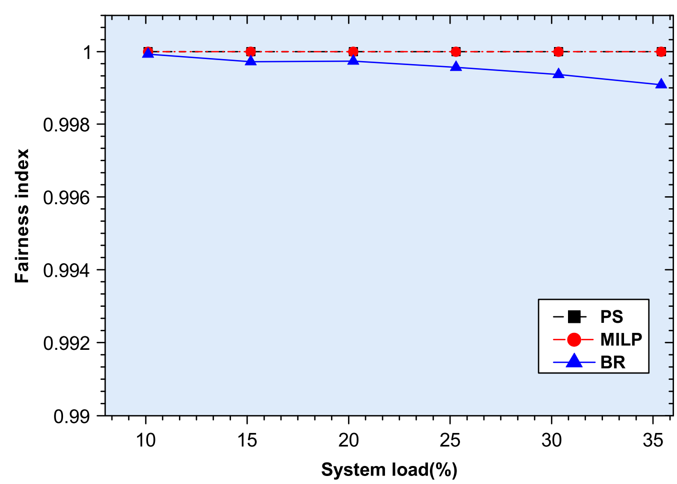

Fig. 17. Fairness indicator under system load changes.

After the iteration process, the task utility stabilizes and converges to a fixed value, i.e. Nash equilibrium point.

This Fig. 15 is extended from one task in Fig. 14 to all tasks. This figure shows the progress of all tasks converging to the Nash equilibrium point.

With an increased system load, the number of allocated concrete service tasks increases. The response time of the service also increased, with the ACT value increasing accordingly. Consequently, the task utility value was reduced (Fig. 16). The ACT values for the PS and MILP methods increased as the system load increased because the methods did not take account of the combination strategies of other users. Hence it was found that when the system load increased to a certain quantity, the task utility of the methods increased markedly.

The BR method states that every task must consider the combination strategies of the other tasks involved; thus, the utility of every task showed a nearly linear growth relative to the other two methods, and the method did not become unstable due to growth of the system load.

5.2.3. Fairness

Fairness describes the differences among all actual task utilities and whether all tasks have attained an optimal utility. A Fairness indicator can be calculated by

$$I(\mathcal{U}) = \frac{(\sum_i^I \mathcal{U}_i)^2}{I \sum_i^I \mathcal{U}_i^2} \tag{27}$$

To eliminate experimental influence on task preference, each task was assigned the same weight and QoS constraint value in the experiment. As a consequence, the fairness indicators of both the PS and MILP methods was one. However, the BR method caused every task to change constantly according to its own combination strategy; thus, the actual utility of every task differs.

Fig. 17 shows that as the system load increases, the BR fairness indicator diminishes; however, the decline in the fairness indicator is small.

As a result, the BR method can guarantee that all tasks achieved simultaneous utility optimisation.

5.2.4. Time complexity

Time complexity describes the time for the computer calculations to reach the Nash equilibrium point. In this method, the arithmetic execution time can be calculated as

$$Time = Iter * Tasks * T_{\mathcal{OP}} \tag{28}$$

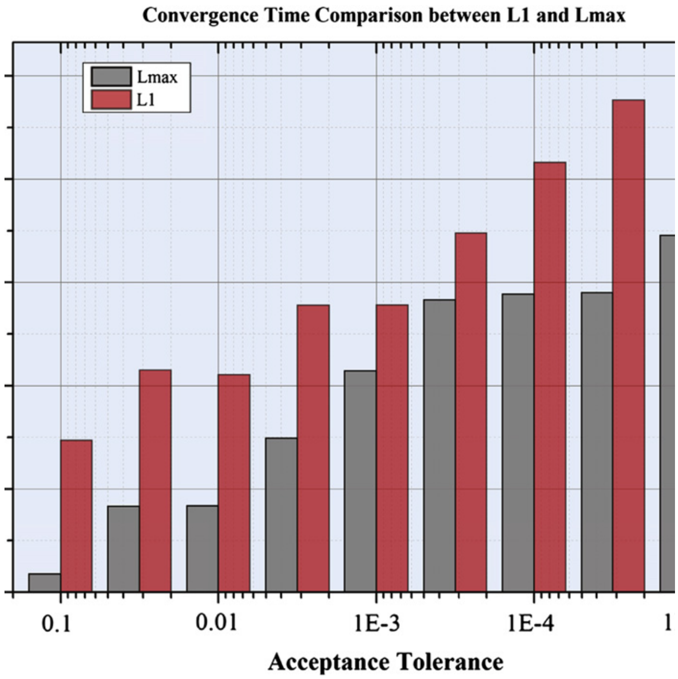

Fig. 18. Convergence time comparison between L_1 and $L_{\max}$.

where *Iter* is the number of iterations before reaching the Nash equilibrium; *Tasks* are types of continuous tasks; $T_{\mathcal{OP}}$ is the execution time of single task best responses. Fortunately, the $\mathcal{OP}$ problem mentioned in Section 4.1 is a convex optimisation, which means it can be solved in a reasonable time, even if it is a large-scale problem (Boyd and Vandenberghe, 2004). The key factor is to control iteration number when *Tasks* and $T_{\mathcal{OP}}$ are fixed. One way to reduce the iteration number is to replace the L_1 mentioned in Fig. 9 with L_{max}. Where $L_1 = \sum_{i=1}^{I} |\mathcal{U}_i^{(x-1)} - \mathcal{U}_i^{(x)}|$ is the sum of all the task utility differences in one iteration, and $L_{max} = Max_{i=1}^{I} |\mathcal{U}_i^{(x-1)} - \mathcal{U}_i^{(x)}|$ is the maximal task utility difference in one iteration.

From Fig. 18, in which *y*-axis is the time calculation, and *x*-axis is acceptance tolerance, it can be seen that L_{max} always represents less time than L_1 as acceptance tolerance changes from 0.1 to 10^{-5}. It is also found that time spent on the arithmetic calculations are at the level of 'minutes', whereas the remote sensing processing of mass data, is at the level of 'hours' (Zhang et al., 2008).

The proposed method is inferior to PS and MILP for the additional iteration, but it is felt that this additional time bonus is worthy of consideration. Equilibrium strategy can effectively reduce the "best services" competition and conflicts and hence avoid waiting in queues and network congestion. However, in this study, focus is on the effectiveness of the employment of game theories to model and resolve competition and conflict problems for concurrent tasks. Hence L_1 rather than L_{max} is used to test for convergence of the proposed method.

6. Related works

In recent years, a number of research groups have addressed the problem of GI service composition. In this section, the main approaches most relevant to the work of this study are presented, and differences between these approaches and the methodology presented in this paper are pointed out. The existing GI composition can be classified into function-oriented and QoS-aware methods.

6.1. Function-oriented GI service composition

The Open Geospatial Consortium (OGC), organisation is working towards this service-oriented approach and has developed geospatial GI service standard interface specifications including a Web Feature Service (WFS) (Vretanos, 2005), Web Map Service (WMS) (de la Beaujardiere, 2004), Web Coverage Service (WCS) (Evans, 2003), Catalogue Service for Web (CSW) (Chen et al., 2009), and Web Processing Service (WPS) (Schut, 2007).

These specifications have been developed according to the functional attributes of GI services. Likewise, GI service composition methods, such as semantic-based, rule-based, processing-based methods, are derived functions of GI services (Hoffmann et al., 2009; Lutz, 2007; Lutz et al., 2007; Yue et al., 2009, 2010).

However, as indicated above, the focus of this study is mainly on the non-function property of GI services, i.e. QoS. The proposed method depends on the abstract PSC generated by the function-oriented method.

6.2. QoS-aware GI service composition

QoS-aware GI services composition, is a process which binds a concrete service to each abstract service within the GIPSC according to a utility function under QoS constraints, imposed by the service level agreement (SLA) (Onchaga, 2006). This topic has been devoted, in the field of GI services and SOC, to finding an optimisation solution fitting user constraints. The existing methods can be grouped into two classes according to whether the running time can be adjusted:

(1) Static environment: These methods do not consider environment changes and model the QoS-aware PSC combination issues as a global constraint optimisation problem (Zeng et al., 2004). Based on a five-dimensional QoS model (cost, response time, reputation, success rate, availability), the method analyzes the working flow control structure and establishes a QoS-aware integer linear programming optimisation model (Zeng et al., 2004). Much work has been based on this model. For instance, intelligent algorithms, such as the genetic algorithm, the simulated annealing algorithm, were introduced to provide approximate solutions for large-scale programming problems, since global programming is a NP problem in essence (Ko et al., 2008).
(2) Dynamic environment: The QoS-aware PSC composition method under a dynamic environment includes a monitoring mechanism which sets a threshold. When a QoS property of any service changes more than the threshold or abnormal service conditions appear, a re-programming mechanism can be triggered quickly to guarantee that the PSC can be smoothly executed with optimal performance (Canfora et al., 2005). The representational approaches are local reconstruction based (Zeng et al., 2004), two-stage re-programming based (Berbner et al., 2007), and negotiation based methods (Ardagna and Pernici, 2007).

However, these QoS-aware service composition methods pursue performance optimisation (e.g., time, price, stability) under user QoS constraints (Ardagna and Pernici, 2007). They essentially centre on single task optimisation in both the GI services and information science areas, while disregarding the competition relationship between concurrent tasks. Therefore, the method presented in this paper, is different in this way from previous ones.

7. Summary and outlook

This paper presented a new approach, based on the Game Theory, to solve the problem of GI services competing for concurrent tasks in a services-oriented computing environment. The conflicting behaviour of these concurrent tasks was deeply analysed and modelled as a non-cooperation game model. In addition, a novel queue based method was proposed as a way to guarantee the calculated accuracy of the response times presented. This approach is entirely different from traditional ones because of the waiting in queues and network queues and network congestion created by concurrent tasks. The Nash equilibrium, as a balanced assignment strategy, is achieved by an iterative algorithm, which presents good convergence in both theory and experiment. Via analysis, and comparison with traditional methods in task utility, the results of the experiments show that this method can increase the performance of all tasks, and has strong anti-jamming and anti-waiting capability in the actual scenarios presented. As a result, the game theory-based, QoS-aware GI service composition methods used in this study proved beneficial in the analysis and solution of the optimal resource competition problem, from the perspective of the entire system (i.e., all tasks).

Future work includes issues related to the response time calculation method and composition of the best services caused by concurrent task conflicts. This requires analysis of the behaviour of tasks during the iterative process in order to design heuristic rules to speed up the performance of converging to the Nash equilibrium point. In addition, the evolutionary game model with environmental dynamics will also be used, and the relationship between system optimisation, Pareto optimisation and Nash equilibrium will be analysed.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC, 41001220 and 51178193), and the Postdoctoral Science Foundation of Central South University.

Appendix A

Theorem 2. proof:

(1) From (4), (6), (5), and (8), $\mathcal{U}_i(s)$ is clearly a continuous derivable function.
(2) According to $\mathcal{K}_i^1+\mathcal{K}_i^3>\mathcal{K}_i^2$, $(\partial\mathcal{U}_i(s)/\partial s_{i,j}^l)=\mathcal{K}_i^1+\mathcal{K}_i^3-\mathcal{K}_i^2>0$; thus, the function $\mathcal{U}_i(s)$ increases to $s_{i,j}^l$.
(3) $\mathcal{U}_i(s)$ is a second-order derivative: $(\partial^2\mathcal{U}_i(s)/\partial(s_{i,j}^l)^2)=(\partial\mathcal{K}_i^1/\partial s_{i,j}^l)-(\partial\mathcal{K}_i^2/\partial s_{i,j}^l)+(\partial\mathcal{K}_i^3/\partial s_{i,j}^l)$ because υ_i^1 and υ_i^2 are both first-order functions, as $(\partial^2\mathcal{K}_i^1/\partial(s_{i,j}^l)^2)$ and $(\partial^2\mathcal{K}_i^2/\partial(s_{i,j}^l)^2)$ are both equal to 0; however, $(\partial^2\mathcal{K}_i^3/\partial(s_{i,j}^l)^2)\geq 0$; thus, $(\partial^2\mathcal{U}_i(s)/\partial(s_{i,j}^l)^2)\geq 0$ holds for any $j=1,\ldots,J$ and $l=1,\ldots,L$. This indicates that Hessian matrix of $\mathcal{U}_i(s)$ is positive and definite. Therefore, $\mathcal{U}_i(s)$ is a convex function (Borwein and Lewis, 2000) of $s_{i,j}^l$, and the strict diagonal convexity is satisfied (Rosen, 1965).

The three points mentioned above illustrate that the object function of best response $\mathcal{OP}_i$ meets the conditions of continuity, increasing, and convex functions; the non-cooperative game of optimal service composition is a sole Nash equilibrium solution (Orda et al., 1993); thus, the theorem is proved.

Theorem 3. proof: $\mathcal{M}$ is defined as the subset of Euclidean space, and $\mathcal{M}$ is the feasible strategy set of task i. Let $\mathcal{M}=\cup_{i=1}^{I}\mathcal{M}_i, S\subset\mathcal{S}$ be the subsets of $\mathcal{M}$.

Let $\mathcal{BR}_i(s_{-i})$ be the best response function of the combination strategy s_{-i} of tasks $I-1$ other than task i, and $s_i=\mathcal{BR}_i(s_{-i})$.

Let $s^{(1)}=(s_1^{(1)},\ldots,s_I^{(1)}),s^{(2)}=(s_1^{(2)},\ldots,s_I^{(2)})$

$\forall s=(s_1,\ldots,s_I)\in S$, define:$g$:$S\to S$, $g(s)=(\mathcal{BR}_1,\ldots,\mathcal{BR}_I)$, Let $d(\cdot)$ be the L^1 norm distance; thus〈

$$d(g(s^{(1)}),g(s^{(2)}))=\sum_{i=1}^{I}\left|\mathcal{BR}_i^{(1)}-\mathcal{BR}_i^{(2)}\right| \tag{29}$$

Based on the mean value theorem:

$$\mathcal{BR}_i^{(1)}-\mathcal{BR}_i^{(2)}=\sum_{m=1,m\neq i}^{I}\left.\frac{\partial\mathcal{BR}_i(\phi_i)}{\partial s_m}\right|_{s_{-i}=\phi_i}(s_m^{(1)}-s_m^{(2)}) \tag{30}$$

Therefore

$$\begin{aligned} d(g(s^{(1)}),g(s^{(2)}))&=\sum_{i=1}^{I}\left|\mathcal{BR}_i^{(1)}-\mathcal{BR}_i^{(2)}\right| \\ &\leq\sum_{i=1}^{I}\sum_{m=1,m\neq i}^{I}\left|\frac{\partial\mathcal{BR}_i}{\partial s_m}\right|\left|s_i^{(1)}-s_i^{(2)}\right| \\ &\leq\sum_{m=1,m\neq i}^{I}\left|\frac{\partial\mathcal{BR}_i}{\partial s_m}\right|\sum_{i=1}^{I}\left|s_i^{(1)}-s_i^{(2)}\right| \\ &=\sum_{m=1,m\neq i}^{I}\left|\frac{\partial\mathcal{BR}_i}{\partial s_m}\right|d(s^{(1)},s^{(2)}) \end{aligned} \tag{31}$$

where

$$\frac{\partial\mathcal{BR}_i}{\partial s_m}=\left(\frac{\partial\mathcal{BR}_i}{\partial s_{m,1}^1},\ldots,\frac{\partial\mathcal{BR}_i}{\partial s_{m,J}^1},\ldots,\frac{\partial\mathcal{BR}_i}{\partial s_{m,J}^L}\right),$$

and $$\frac{\partial\mathcal{BR}_i}{\partial s_{m,J}^l}=\frac{\kappa s_{i,j}^l\lambda_i}{(\mu_{m,-j}^l-\kappa s_{m,j}^l\lambda_i)^2}.$$

According to the Banach fixed point theorem (Istratescu, 2001), when $\sum_{m=1,m\neq i}^{I}\left|\partial\mathcal{BR}_i/\partial s_m\right|<1$, there must be a fixed point $s^*=\{s_1^*,\ldots,s_I^*\}$ in which S satisfies $g(s^*)=s^*$; i.e., $s_i^*=\mathcal{BR}_i(s_{-i}^*)$. This point is the Nash equilibrium point:

$$\sum_{m=1,m\neq i}^{I}\left|\frac{\partial\mathcal{BR}_i}{\partial s_m}\right|=\sum_{m=1,m\neq i}^{I}\sum_{l\in cp}^{L}\sum_{j=1}^{J}\frac{\kappa s_{i,j}^l\lambda_i}{(\mu_{m,-j}^l-\kappa s_{m,j}^l\lambda_i)^2}$$

Thus, when

$$\sum_{m=1,m\neq i}^{I}\sum_{l\in cp}^{L}\sum_{j=1}^{J}\frac{\kappa s_{i,j}^l\lambda_i}{(\mu_{m,-j}^l-\kappa s_{m,j}^l\lambda_i)^2}<1$$

the aforementioned algorithm can approach the Nash equilibrium starting from any point.

References

Alameh, N., 2003. Chaining geographic information Web services. IEEE Internet Computing 7, 22–29.
Ardagna, D., Pernici, B., 2007. Adaptive service composition in flexible processes. IEEE Transactions on Software Engineering 33, 369–384.
Baraniuk, R.G., 2011. More is less: signal processing and the data deluge. Science 331, 717–719.
Berbner, R., Spahn, M., Repp, N., Heckmann, O., Steinmetz, R., 2007. Dynamic replanning of web service workflows. 2007 Inaugural IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2007), pp. 211–216.
Borwein, J., Lewis, A.S., 2000. Convex Analysis and Nonlinear Optimization: Theory and Examples. Springer, Berlin 273 pp.
Boulogne, T., Altman, E., Pourtallier, O., 2002. On the convergence to Nash equilibrium in problems of distributed computing. Annals of Operations Research 109, 279–291.
Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press 730 pp.

Brunner, D., Lemoine, G., Bruzzone, L., 2010. Earthquake damage assessment of buildings using VHR optical and SAR imagery. IEEE Transactions on Geoscience and Remote Sensing 48, 2403–2420.
Canfora, G., Penta, M.D., Esposito, R., Villani, M.L., 2005. QoS-Aware Replanning of Composite Web Services. Proceedings of the IEEE International Conference on Web Services (ICWS'05). IEEE Computer Society, pp. 121–129.
Canfora, G., Penta, M.D., Esposito, R., Villani, M.L., 2008. A framework for QoS-aware binding and re-binding of composite web services. Journal of Systems and Software 81, 1754–1769.
Cardoso, J., 2002. Quality of Service and Semantic Composition of Workflows, Department of Computer Science. University of Georgia, Athens, GA, p. 215.
Chen, N.C., Di, L.P., Yu, G.N., Gong, J.Y., Wei, Y.X., 2009. Use of ebRIM-based CSW with sensor observation services for registry and discovery of remote-sensing observations. Computers & Geosciences 35, 360–372.
Chow, Y.-C., Kohler, W.H., 1979. Models for dynamic load balancing in a heterogeneous multiple processor system. IEEE Transactions on Computers 28, 354–361.
Clery, D., Voss, D., 2005. All for one and one for all. Science 308, 757.
de la Beaujardiere, J., 2004. Web map service (WMS) implementation specification. Open Geospatial Consortium, Inc., p. 85.
Durbha, S.S., King, R.L., 2005. Semantics-enabled framework for knowledge discovery from Earth observation data archives. IEEE Transactions on Geoscience and Remote Sensing 43, 2563–2572.
Evans, J., 2003. Web Coverage Service (WCS) Implementation Specification. Open Geospatial Consortium, Inc, p. 67.
Foster, I., 2005. Service-oriented science. Science 308, 814–817.
Fudenberg, D., Tirole, J., 1991. Game Theory. MIT Press, Cambridge 616 pp.
Gross, D., Harris, C.M., 1985. Fundaments of Queueing Theory, 2nd ed John Wiley & Sons, New York 579 pp.
Grosu, D., Chronopoulos, A.T., 2005. Noncooperative load balancing in distributed systems. Journal of Parallel and Distributed Computing 65, 1022–1034.
Hoffmann, J., Bertoli, P., Helmert, M., Pistore, M., 2009. Message-based web service composition, integrity constraints, and planning under uncertainty: a new connection. Journal of Artificial Intelligence Research 35, 49–117.
Istratescu, V.I., 2001. Fixed Point Theory an Introduction. Springer, the Netherlands 484 pp.
Ko, J.M., Kim, C.O., Kwon, I.-H., 2008. Quality-of-service oriented web service composition algorithm and planning architecture. The Journal of Systems and Software 81, 2079–2090.
Li, D., 2008. Spatial Information technique in rapid response to Wenchuan earthquake. International Society for Photogrammetry and Remote Sensing.
Li, D., 2009. Remote sensing in the Wenchuan earthquake. Photogrammetric Engineering and Remote Sensing 75, 506–509.
Li, D.-r., Chen, X.-l., Cai, X.-b., 2008. Spatial information techniques in rapid response to Wenchuan earthquake. Journal of Remote Sensing 12, 841–851 (in Chinese).
Li, D., Zhu, Q., Zhu, X., Sui, H., 2010. Task-Oriented Remote Sensing Information Focused Services, Earth Observing and Navigating. Science Publish, Beijing, p. 197 (in Chinese).
Lutz, M., 2007. Ontology-based descriptions for semantic discovery and composition of geoprocessing services. Geoinformatica 11, 1–36.
Lutz, M., Lucchi, R., Friis-Christensen, A., Ostlander, N., 2007. A rule-based description framework for the composition of geographic information services. Geospatial Semantics, Proceedings 4853, 114–127.
Martin, D., Burstein, M., McDermott, D., McIlraith, S., Paolucci, M., Sycara, K., McGuinness, D.L., Sirin, E., Srinivasan, N., 2006. Bringing semantics to Web services with OWL-S. World Wide Web Journal 10, 243–277.
Onchaga, R., 2006. Quality of service management framework for dynamic chaining of geographic information services. International Journal of Applied Earth Observation and Geoinformation 8, 137–148.
Orda, A., Rom, R., Shimkin, N., 1993. Competitive routing in multiuser communication networks. IEEE/ACM Transactions on Networking 1, 510–521.
Rosen, J.B., 1965. Existence and uniqueness of equilibrium points for concave N-person games. Econometrica 33, 520–534.
Schut, P., 2007. Web Processing Service (WPS) Implementation Specification. Open Geospatial Consortium, Inc., p. 45.
Singh, R.P., 2010. Satellite observations of the Wenchuan earthquake, 12 May 2008. International Journal of Remote Sensing 31, 3335–3339.
Vretanos, P., 2005. Web Feature Service (WFS) Implementation Specification. Open Geospatial Consortium, Inc., p. 131.
Weibull, J.W., 1997. Evolutionary Game Theory. MIT Press, Cambridge, MA, USA 292 pp.
Xu, M., Cao, C., Zhang, H., Guo, J., Nakane, K., He, Q., Guo, J., Chang, C., Bao, Y., Gao, M., Li, X., 2010. Change detection of an earthquake-induced barrier lake based on remote sensing image classification. International Journal of Remote Sensing 31, 3521–3534.
Yue, P., Di, L., Yang, W., Yu, G., Zhao, P., Gong, J., 2009. Semantic Web Services-based process planning for earth science applications. International Journal of Geographical Information Science 23, 1139–1163.
Yue, P., Gong, J., Di, L., 2010. Augmenting geospatial data provenance through metadata tracking in geospatial service chaining. Computers & Geosciences 36, 270–281.
Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H., 2004. QoS-aware middleware for Web services composition. IEEE Transactions on Software Engineering 30, 311–327.
Zhang, Z.-x., Guo, D.-h., Ke, T., Wang, J.-c., 2008. The use of aerial photogrammetry in the fast response for China earthquake rescue. Journal of Remote Sensing 12, 852–857 (in Chinese).
Zhu, Q., Yang, X., Li, H., 2009. Optimal Composition Algorithm with a CPM Response Time Guarantee for Remotely Sensed Image Processing Services, Geospatial Technologies for Earth Observation. Springer, pp. 381–396.