Advances on QoS-aware web service selection and composition with nature-inspired computing

Xinchao Zhao\textsuperscript{1} \eprint{xcbupt@126.com}, Rui Li\textsuperscript{1}, Xingquan Zuo\textsuperscript{2}

\textsuperscript{1}School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, People’s Republic of China
\textsuperscript{2}School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, People’s Republic of China

Abstract: Service-oriented architecture is becoming a major software framework for complex application and it can be dynamically and flexibly composed by integrating existing component web services provided by different providers with standard protocols. The rapid introduction of new web services into a dynamic business environment can adversely affect the service quality and user satisfaction. Therefore, how to leverage, aggregate and make use of individual component’s quality of service (QoS) information to derive the optimal QoS of the composite service which meets the needs of users is still an ongoing hot research problem. This study aims at reviewing the advance of the current state-of-the-art in technologies and inspiring the possible new ideas for web service selection and composition, especially with nature-inspired computing approaches. Firstly, the background knowledge of web services is presented. Secondly, various nature-inspired web selection and composition approaches are systematically reviewed and analysed for QoS-aware web services. Finally, challenges, remarks and discussions about QoS-aware web service composition are presented.

1 Introduction

Web services are autonomous software systems identified by URIs which can be advertised, located, and accessed through messages encoded according to XML-based standards and transmitted using Internet protocols [1]. The emerging service-oriented computing (SOC) paradigm [2] promises to enable businesses and organisations to collaborate in an unprecedented way by means of standard web services. To support rapid and dynamic composition of services in this paradigm [3, 4], web services that meet requesters’ functional requirements must be able to be located and bounded dynamically from a large and constantly changing number of service providers based on their quality of service (QoS). With the increase in scale and complexity of the web services, QoS has become a critical challenge for the performance of SOC [5, 6].

Service-oriented architecture (SOA) has been recognised as the next generation framework for building agile distributed applications over Internet. Applications are provided as web services which can be discovered and composed into more coarse-grained services, i.e. composite services. The aim is to search for the optimal set of services that are composed to create a new service and result in the best QoS under the constraints of users or service designers [7]. In the real world, a single web service usually cannot fulfil the requirement for a given enterprise task; however, a composition of different web services might meet the complex requirements. Hence, there is the potential silver bullet that enterprises have been looking for and it has attracted wide interests of many researchers [8–14]. Web services provide the basic fundamental unit for constructing such systems, representing a specific business activity or functionality. Once services are exposed and become available for use by other services and resources on the web, they can establish more complicated structures and interactions, providing new added value aggregate services. In order to effectively support the highly dynamic nature of such systems, services need to be handled as flexible, composite and adaptive units, so they can be substituted, converted or even composed easily with other services.

The web service composition can be viewed as a three-step process: (i) composite web service specification, (ii) selection of the component web services and (iii) execution of the composite web services. At the first step, the user submits the goal he/she wants the composite service achieve, along with some constraints and preferences that need to be satisfied [15]. During the second step, component web services fulfilling the user’s goal are selected among a set of available services. This web service selection could be done by hand or could be automatically decided by the system. When component services are selected at design time, the third step of the composition process consists of executing the selected component web service. At runtime, selection and execution of component web service are integrated and the selection is described as dynamic.

Dynamic composition of web services, which has been recognised as a flexible way for resource sharing and application integration, involves two important consecutive questions: composition schema planning and optimised service selection. The former process, deciding which abstract web services are appropriate for the user’s needs and how to construct them, is mostly based on functional requirements; non-functional properties are not formally considered. Artificial intelligence or graph search problem-based planning technology can be leveraged to automatically produce composition schema, including the involved component tasks and their order. Many researchers [16–18] were in favour of regarding it as an artificial intelligence (AI) planning problem and applied classic AI planning algorithms to the problem. Liang and Su [19] proposed a semi-automated method for service composition; the main idea of their work is to construct an AND/OR graph from the service dependency graph and to find a sub-graph for the solution by applying a bottom-up search algorithm. The latter process, dynamic web service selection [20–22], involves choosing specific web services to assign to each involving task and fulfil the functionality of the abstract web services schema under the guidance of non-functional properties, such as QoS.

Automatic service composition problem [23, 24] is to automatically select services from a registry by matching their input and output data. The automatic composition archetype can...
solve heterogeneity of protocol description and paradigm by designing composite services that comply with them. Automatic service composition is generally dynamic. Resulting from this selection and a set of input data, the composite service produces a set of output data, which is needed by users. When QoS values for each service are considered, this selection problem becomes an optimisation problem called the QoS-aware automatic service composition problem.

Our work focuses on the recent advance of the state of the art in technologies for web service composition and selection. The background of QoS-aware web services is discussed in Section 2. Various nature-computing approaches for web service selection and composition are presented, in Section 3. Section 4 presents some discussions, remarks and possible challenges for web service selection and composition.

2 Background

This section clarifies the foundational issues of the QoS-aware web service composition (QWSC) problem. Herein the related works are described with web services composition problem, QoS criteria and aggregation functions.

2.1 Related works and research motivation

The critical point is web service discovery before web service selection and composition. Ngn and Kanagasabai [25] presented an extensive review of semantic web service discovery, highlighting the state-of-the-art approaches, the key semantic formalisms employed, as well as benchmarks and testbeds for performance evaluation. Web service composition and selection result in an optimisation problem that has been studied both in the research areas of SOC for business processes and in grid environments. There are two major problems in using QoS for service discovery; and they are the specification and storage of the QoS information, and the specification of the customer’s requirements [26].

Zeng et al. [2, 27] presented a QoS-based middleware for web service composition. They did not derive aggregate QoS performance from workflow patterns, but split a service composition into execution paths represented by directed acyclic graphs. They applied integer programming to solve the objective function of each execution path and merge the solutions of all execution paths to help service requesters select an optimal service. Inevitably, the computational cost of this approach increases exponentially with the number of exclusive choices in the composition, and multiple execution paths may produce conflicting service selections. As a step forward, Ardagna and Pernici [28] extend the integer linear programming (ILP) model by introducing both loop peeling to deal with the composition structures with cycles and negotiation to handle the situation when a feasible solution cannot be identified. However, ILP method is only suitable for small problems, as the complexity of the algorithm increases exponentially with an increase in problem size.

Huang et al. [29] applied multiple criteria decision making with a weighted sum model and integer programming approaches to select the optimal service to help service requesters evaluate services numerically. Wu et al. [30] strengthened the current service composition mechanism by generation of QoS requirement and its selection algorithm based on the QoS reference vectors. Two algorithms (exhaustive search and ILP) are introduced to execute experiment and compared as two kinds of experiment examples. One aims at the most accurate service selection and the other chases for trade-off between selection cost and result. Wang et al. [31] proposed a fuzzy linear programming technology, in order to identify their dissimilarity on service alternatives, assist service consumers in selecting most suitable services with consideration of their expectations and preferences.

Xiao and Bouatta [32] present an autonomic service provisioning framework for establishing QoS-assured end-to-end communication paths across administratively independent domains. The authors model the domain composition and adaptation problem as the classic k-constrained optimal path (MCOP) problems and provide algorithms to solve it. Yu et al. [21] modelled QWSC problem in two ways: the combinatorial model and the graph model. The combinatorial model is based on a multi-dimensional multi-choice 0-1 knapsack problem. The graph model defines the problem as a MCOP problem. The solution ensures that all selected services meet the QoS requirements, and no critical path is identified before the service selection. Tsesmetzis et al. [33] focused on the service provider perspective that receives multiple concurrent requests for services demonstrating different QoS properties. It introduces the ‘selective multiple choice knapsack problem’ to identify the services, which should be delivered in order to maximise the provider’s profit, subject to maximum bandwidth constraints.

Ma et al. [34] proposed a dynamic directed and weighted acyclic graph model of service-correlation aware with improved Dijkstra algorithm and self-adaptive algorithm to carry out a better combination of the candidate web services. The feature of their work is to take the dependence among services into account and establish the service correlations. Jiang et al. [35] explored the essence of QoS-aware automatic service composition and mapped it to a new graph problem: how to find the single-source optimal directed (SSOD) acyclic graphs from a directed graph. To deal with this new SSOD problem, they presented its model formally and then its computing rules of overall weight or QoS.

Service composition problem is formalised into a dynamical searching process for the optimal plan for the stochastic discrete event system in [36]. Aiming at reducing the high computation complexity, authors designed a reliable service selection algorithm using an improved Markov decision process. Differently, local martingale difference (LMD) approach [37] requires neither predetermined threshold nor accurate parameters of the providers’ distribution. LMD becomes the first self-adaptable methodology in the domain since it makes the decision based on the dynamic information all gathered during the process of waiting, without any manual intervention. Xu et al. [6] use dynamic programming method to address the efficiency issue by developing a polynomial time algorithm for shortest sequence composition.

Many other methods are also proposed for this problem, such as analytic hierarchy process [38], branch and bound [39], clouding computing [40], clustering [41], grey relational analysis [42], multi-state bounded sets technique [43], principal component analysis [44], skyline analysis [45], statistical regression [46] and systematic search algorithm like Dijkstra [47].

2.2 Composite web service

In general, a composite service can be developed by different methods and deployed on different platforms, which is described as follows. Given a business workflow that includes a set of abstract services and a set of component services implementing for each abstract service, the goal is to find an optimal combination of component services. An abstract service describes the functionalities, but not the execution details of a web service (WS) [48]. The functionalities offered by a web service are usually performed by its provided operations. The functional characteristics of an abstract service contain the inputs, outputs and the classification type representing an ontology concept.

Definition 1 (Abstract service): An abstract WS is a tuple of \((N, OT, I, O)\), where

- \(N\) is the name of the abstract service;
- \(OT\) is the type of required operation in terms of ontology concepts;
- \(I = \{I_1, \ldots, I_m\}\) is a set of inputs of required operations, where \(m\) is the number of inputs. \(I_i\) service \((IN, IT)\), where \(i = 1, 2, \ldots, m, IN\) is the name of input and \(IT\) is the type of input;
- \(O = \{O_1, \ldots, O_n\}\) is a set of outputs of required operations, where \(n\) is the number of outputs. \(O_i = (ON, OT)\), where \(ON\) is
the name of output and $OT$ is the output type in terms of ontology concepts.

**Abstract service**, corresponding to a specific functionality, is not necessarily a single service implementation. At execution time this description is used to retrieve some services using a match algorithm (which will be referred as concrete or component services), and then select one from them that meets the constraints and maximises the QoS aggregation function. Many matching methods have been proposed [25, 49–51].

**Definition 2 (Selection plan):** A selection plan $SP$ is a set of $(ws_i, op_i)$ pairs, where $SP = \{(ws_1, op_1), \ldots, (ws_n, op_n)\}$. For each pair $(ws_i, op_i)$ in $SP$, the service operation $op_i$ provides the required function of an abstract service $ws_i$, where $i = 1, \ldots, n$.

The selection plan describes a combination of operations offered by concrete services for an abstract process. A composite service is obtained after each abstract service is assigned to a concrete/ component service, whose interaction is described by some workflow description language (e.g. BPEL4WS [7]). There are two different questions to be solved:

- determine the QoS of a composite service as a function of the QoS of its components;
- determine the set of concrete services for each abstract service of the workflow, which will maximise the QoS of the composite service.

Given $m$ abstract services and $n$ concrete services for each abstract service, there are $n^m$ possible combinations. The search space is discrete and a combinatorial optimisation problem is presented here. An exhaustive search algorithm is inadequate due to the huge search space, and an NP-hard difficulty.

### 2.3 Instance illustration of a composite service

A simple example is presented to illustrate the service composition process with end-to-end QoS constraints [21] in Fig. 1, in which there are six abstract services and four possible abstract plans, namely, $\{t_1, t_2, t_3, t_4\}$, $\{t_1, t_3, t_5\}$, $\{t_2, t_3, t_4\}$ or $\{t_2, t_5, t_6\}$. Each function $F_i$ can be implemented by any component service $(s_i)$ associated with abstract service $t_i$. QoS values of all component services are shown in Fig. 1b. End-to-end QoS requirements include:

- Time: response time $\leq 600$;
- Price: price $\leq 250$;
- Availability: availability $\geq 85\%$.

There are totally 40 feasible candidate combinations, i.e. selection plans, each with different utility and QoS values. The optimal selection plan of services, which maximises the sum of all service utilities, is $\{(s_{t1}, t_1), (s_{t2}, t_2), (s_{t4}, t_4), (s_{t5}, t_5)\}$ with a total utility of 823, a response time of 590, at a price of 240 and an availability of 86.64%.

### 2.4 QoS and related metrics

O’Sullivan et al. [52] claimed that complete service descriptions are helpful to service discovery, management, negotiation, composition and substitution, and that a service is completely represented if both functional and non-functional aspects are described well. Third-party specifications [53, 54] are likely to become de facto standards in the near future, and they can be used in conjunction with the proposed service selection scheme to provide viable solutions for QoS-based SOA.

Previous studies provide various definitions of QoS: ‘The degree to which a system, component or process meets customer or user needs or expectations’ [55] and ‘The standard of something when compared to other things like it; how good or bad something is’ [56]. Garvin [57] identified the term from five perspectives including transcendent, product-based, user-based, manufacturing-based and value-based. A general QoS definition is given by Campanella et al. [58] as: QoS is a generic term which takes into account several techniques and strategies that could assure application and users a predictable service from the network and other components involved, such as operating systems.

The main purpose of any web service is to satisfy an end-user and there still is no confidence that a user is satisfied even when the QoS is rather high. The most popular metrics for estimating such quality and user perception of web services are QoS and quality of experience (QoE) [59], which represent objective and subjective assessments correspondingly. Besides, the quality of business [60] and quality of design [61] are also the examples of such metrics. Fig. 2 illustrates how the sets of attributes for the mentioned metrics are related. The main purpose of Fig. 2 is to illustrate the automatic calculation of the user satisfaction (QoE) based on the known values of attributes.

### 2.5 QoS criteria

QoS criteria are used to differentiate the web services providing the same functionality during the service selection process of QWSC. According to ISO 8402 and ITUE.800 [62], each operation of a web service will be associated with a set of QoS criteria (non-functional properties), which can properly reflect the users’

---

**Fig. 1** Composite web service illustration

**Fig. 2** Relationship between different sets of metrics when evaluating the quality of web services
requirements and affect the users’ satisfaction from some aspects, such as execution time, reliability, availability, reputation and price, which are generally used in many references [2, 28, 48, 52]. These properties apply both to component web services and to composite web services.

**Execution time** \( (op) \): The expected delay between the time instant when a service operation \( op \) begins and the time when the result is obtained. Execution time is usually measured in seconds.

**Reliability** \( Rel(op) \): Reliability is a measure of the service invocation trustworthiness, i.e. the probability of a service operation \( op \) being correctly responded within the expected time.

**Availability** \( A(op) \): The probability that the service operation \( op \) is accessible. Availability is a number in the range \([0, 1]\).

**Reputation** \( Rep(op) \): It is a measure of the trustworthiness of service operation \( op \). It is defined as the ratio between the number of service invocations which comply the negotiated QoS over the total number of service invocations. Reputation is a number in the range \([0, 1]\).

**Price** \( P(op) \): The fee that a service requestor has to pay to the service provider for the service operation \( op \). Price is measured in dollars (S).

According to the importance and measurement of the service quality parameters, a QoS vector is used to assess the QoS metrics of a service.

**Definition 3:** QoS vector of a service operation \( op \) is \( QoS(op) = [T(op), Rel(op), A(op), Rep(op), R(op)] \), which describes some information of service quality in the execution process.

### 2.6 QoS aggregation functions

Web service has four basic structures: sequential, cycle, parallel and branch structure (Figs. 3a–d) [63]. In the sequential structure, tasks are executed in a sequential order: in the cycle structure, a task will be executed for multiple times; in the parallel structure, all the parallel tasks can be executed simultaneously, but it cannot go to the next task until all the parallel tasks are finished; in the branch structure, each task in the branch can finish the same component function and system can go to the next task once any task in the branch has been finished. For all basic patterns and their combinations the question of overall QoS evaluation has been elaborated properly as aggregation functions. The result of each aggregation function is the value of a corresponding attribute calculated for a proper pattern.

In order to establish a healthy SOA environment, it is very important to establish a standard to represent QoS parameters and measure them. One of the main problems in web services that integrator faces is to verify all the possible combinations, which is caused by the dynamic nature of SOA on which services will be selected and invoked. However, as Deora et al. [64] referred, QoS attribute values of a WS are not realistic to be provided/measured by service providers or third-party software modules. These attributes are usually measured by the values that users submit from their experience of using the particular service.

So, the calculation of the values of the described QoS attribute will be a function of what values it should have taken and what values were eventually offered. The overall QoS can be given by the (weighted) sum of every QoS value of each attribute, where the weights denote the trustfulness we have for these users. A service composition is composed with candidate services that come from different tasks. A service composition can finish a specific composite function defined by user, whose QoS attribute is calculated by the component candidate services.

The QoS criteria are difficult to compare because different criteria describe different aspects of the non-functional properties (QoS) of a web service. Based on the above basic structures, aggregation functions are defined as Table 1 for each QoS criterion in order to calculate various QoS of a web service. \( p_i \) is the probability that an abstract service \( ws_i \) will be executed for a conditional structure. The probability can be evaluated from statistical execution logs.

If an abstract process has never been executed before, the probability will be assigned a uniform probability \( (p_i = 1/n) \). For QoS aggregation functions, there may be a little difference for different researchers [2, 29, 48, 65]. The service selection would be the key for the construction of an abstract process from composite services.

In fact, the set of functions in Table 1 is incomplete since more attributes can be considered, such as encryption, confidentiality and popularity [59]. When involving some logic models, such as fuzzy logic, \( k \)-value logic and so on, or modular arithmetic models, aggregation functions which use only sum and multiplication operators do not seem to be sufficient and thus, novel corresponding aggregation functions have to be elaborated.

### 3 Nature-inspired computing (NIC) approaches for web service composition

The proliferation of web services is fostering a very active research area in recent years. However, there are two challenges facing the practicality of web services: (i) efficient location of the web service registries that contain the requested web services, and (ii) efficient retrieval of the requested services from these registries with high QoS. Numerous works have been proposed for composing services with multiple QoS constraints. Therefore, understanding its current development and the state-of-the-art solving methods are important to researchers for further research. This paper will mainly review various NIC approaches for QWSC besides the above-mentioned classic methods.
3.1 Why NIC approaches are considered

Service composition creates new functionalities by aggregating different component services based on a specific workflow [29]. When two or more functional-qualified services are available, they can form different combinations. These combinations deliver the same functionality, but differ from each other in the non-functional property (QoS). Obviously, service requesters should be able to select the optimal service set without trying all possible combinations, which is impractical even if there are a few services and services in the workflow. Therefore, heuristic and metaheuristic algorithms are required to solve the QWSC problem. This is the inherent motivation to review the recent situation of QWSC with NIC.

NP difficulty: Web service selection and composition problem is NP [2, 21, 66], and traditional methods, such as graph theory and dynamic programming, cannot bear the complexity burden. Correspondingly, NIC approaches can provide satisfactory solutions in a reasonable time.

Multiple choices: Users generally have different preferences and different persons may have different opinions. Correspondingly, population-based bio-inspired computing approaches can present multiple solutions. In terms of this consideration, population-based approaches, especially for evolutionary or swarm multi-objective optimisation methods, tally with the practical scenarios very well.

Dynamic: Web services are executed on internet and the composite services have great dynamic properties, for example, the selected component service are suddenly unavailable and QoS values changed. Simultaneously, users are possible to modify their choices and/or constraint conditions. At these scenarios, traditional methods may need to replan the composite service. Correspondingly, NIC methods are very possible to present another schema immediately at any time and any situation due to they maintain a solution population.

3.2 Evolutionary algorithm (EA) based computing approaches

EA [67] is a population-based parallel iterative search technique to achieve the satisfactory solutions (not always optimal) for the complex optimisation problems. Evolutionary search of crossover, mutation and selection operations will be directed according to the ‘survival-of-the-fittest’ principle.

The strengths of EA for QWSC problem are the natural coordination between its encoding method and genetic operations and the problem itself. The weakness of EA is its slow convergence for its blind or semi-blind generation methods for new solutions.

3.2.1 Evolutionary algorithm: Canfora et al. [7, 66] described an approach for a quick, coarse-grained QoS-aware service composition based on genetic algorithms (GAs), which is more scalable, however, slower than integer programming. The aim of the approach is to provide a fast way, even if rough, to find the (sub)-optimal service composition and estimate its overall QoS. In [68], the reduction formulas are adopted and the reoptimisation is considered, but only sub-optimal solutions are identified since tasks specified in cycles are always assigned to the same web service. Gerardo et al. [66] advocated GA for its advantage of permitting to deal with QoS attributes having non-linear aggregation functions and for its good scalability that can solve the problem when the number of concrete services increases.

Zhang et al. [69] presented a GA structure characterised by the special relation matrix coding scheme of chromosomes and the population diversity handling with simulated annealing (SA). The relation matrix has the ability to represent simultaneously the composite service re-planning, cyclic paths and many web service scenarios, such as probabilistic invocation, parallel invocation, sequential activation and so on. A quick convergent GA [70] is proposed with an enhanced initial population policy and an evolution policy based on the relation matrix coding scheme. To improve the convergence of GA, chaos theory [71] is introduced into GA with relation matrix coding scheme for web service selection with global QoS constraints. A dynamic GA [72] is presented with a dynamic evolution evaluation policy on the basis of relation matrix coding scheme of genome, which get more excellent composite service plans. An adaptive GA [73] is proposed with a population diversity measurement and an adaptive crossover strategy to further improve the efficiency and convergence of GA.

Vanrompay et al. [74] proposed a GA for the best service variant, where the chosen service composition is deployed on a set of available nodes in an optimal way. Gong et al. [75] proposed a web service composition algorithm which supports global optimal and dynamic re-planning. Algorithm adopts a position matrix coding scheme to express all of composite paths, simultaneously, and re-planning information. A query optimisation model and the corresponding GA [76] are proposed based on multi-attribute QoS aggregation of different web services for the personality and efficiency in evaluations. The model adjusts QoS with global constraints and users’ preferences, dynamic rating scheme and multilevel matching.

Ai and Tang [77] presented a minimal-conflict hill-climbing repair GA to address the web service composition problem, in which individuals can be quickly repaired with a value-ordering heuristics when they violate the constraints on inter-service dependencies and conflicts. A penalty-based GA [78] is proposed to tackle QWSC problem with interservice dependency and conflict constraints. In order to solve the optimal web service selection better, a hybrid GA is proposed in [79] with a local optimiser and a knowledge-based crossover operator. It provides better quality of solutions than the penalty-based GA [78] and the repairing-based GA [77], however, a little longer computation time. A new penalty-based GA [80] is proposed to solve the composite web service partitioning problem due to judicious partitioning of composite web services across parallel servers can increase the throughput. A fast local optimiser and a rank-based selection with elitism are incorporated into GA to obtain even excellent solutions.

Combining web service models with GA, Xiao [81] put forward a multi-target optimisation selection model based on similarity, QoS, matching degree, rating and feasibility of web services. Cao et al. [82] proposed a refinement GA for quality optimisation of a service process model, which is first transformed into a process structure tree for quality calculation and then GA is used to refine the service process. Yu et al. [83] proposed an integrated service query framework which incorporates a service query model and a twophase optimisation strategy. The query model defines service communities to organise the large and heterogeneous service space. The two-phase optimisation strategy with greedy algorithm and EA is able to ‘co-evolve’ multiple feasible execution plans simultaneously for the best plan.

Considering the difference between QoS of the services themselves and QoS of the network, Klein et al. [84] described a network-aware approach to service composition in a cloud, in which QoS of the services and QoS of the network are handled independently. Ukor and Carpenter [85] presented a service selection optimisation problem (SSOP) with the notion of request tokens and three approaches (i.e. integer programming, tree-search heuristic and GA) for solving SSOP. Considering both functional and non-functional requirements together, FanJiang and Syu [86] proposed a semantic and GA-based automatic service composition approach.

Due to the huge search space of the service composition, a genetic programming approach [87] is proposed for the desired outputs based on available inputs, as well as ensure that the composite service has the optimal QoS. Rodriguez-Mier et al. [88] proposed a context-free grammar-based genetic programming algorithm to generate the valid structures of the composite services. Moreover, the proposal tries to minimise the number of services, and looks for compositions with the minimum execution path. A co-evolutionary GA [89] is proposed for web service composition based on QoS, which fully considers the individual relationships among populations.
3.2.2 Hybrid EA: Different algorithms have different search mechanism and each computing paradigm has its own pros and cons. Therefore, a felicitous design of hybrid algorithm is a possible direction for complex optimisation problem, whatever classical optimisation methods, heuristic and meta-heuristic algorithms.

Ma and Dong [90] first discussed the limitations of simple additive weight method due to these weights have no physical meaning and hard to set. Then a new effective computing approach, linear physical programming, cooperated with GA, is proposed for quality-driven web service selection. Liang and Huang [91] proposed a novel hybrid approach for web services composition application, which incorporates GA and rough set theory. The hybrid GA and rough set theory can: (i) solve problems that can be decomposed into functional requirements, and (ii) improve the performance of GA, detect the infeasible solutions and exceptions by reducing the domain range of the initial population and constrained crossover using rough set theory. Based on global QoS constraints decomposition, Liu et al. [92] proposed a dynamic composition method, which consists of three steps: (i) global QoS constraints are decomposed into local constraints by a culture GA; (ii) QoS determination rules determine QoS values of candidate services; (iii) best web services satisfying the local constraints are selected for each task during the running time. Based on QoS decomposition idea, Mardukhi et al. [93] also designed a GA method.

Combining the advantages of GA and SA, a QoE/QoS driven SA-based GA [94] is presented for better efficiency and effect of web service selection. A web service selection method based on quantum GA [95] is proposed, which uses quantum bit encoding, dynamic step-length quantum gate angle adjustment, neighbourhood service search and dynamic punishment strategy to expand search scope and speed up convergence.

3.2.3 Multi-objective EA: Claro et al. [96] implemented a multi-objective evolutionary approach based on the famous NSGA-II (non-dominated sorting genetic algorithm [97]), which identifies a set of Pareto optimal solutions without introducing a ranking among different quality dimensions. In [96], some execution time is wasted when infeasible solutions are generated and sometimes, no solution can be identified even when the problem is feasible, in cases in which the global constraints are stringent. Yao and Chen [98] also utilised the NSGA-II framework to QoS-aware service composition problem and multiple different feasible solutions selection. A web services selection method based on quantum GA [99] is proposed, which uses quantum bit encoding, dynamic step-length quantum gate angle adjustment, neighbourhood service search and dynamic punishment strategy to expand search scope and speed up convergence.

Liu et al. [102] first modelled the dynamic web service selection problem as a multi-objective services composition optimisation with QoS constraints. Then, multi-objective GA provides multiple Pareto optimal services composition schema with global QoS constraints. Aimed at the SSOP, Wang and Hou [103] presented another multi-objective GA, in which they designed a chromosome encoding method to represent a feasible service selection solution, and then developed genetic operators and strategies for maintaining population diversity and avoiding being trapped by local optima. For the same aim, Hu et al. [104] proposed a multi-objective GA, whose efficiency is not affected when building the evolving solutions set at each iteration. A Pareto optimal solution set is constructed with the pseudo binary tree’s rule. Then the Pareto optimal solutions are sequenced and the fitness of each Pareto optimal solution is decided by individuals’ similarity.

Wagner et al. [105] described an advanced repair operation and provided a multi-objective optimisation algorithm that leverages background knowledge to discover robust QoS-optimised service selections in an open service environment. The algorithm takes the costs of potential service failures into account and ranks solutions on the basis of the risk preferences of the decision maker. Ramírez et al. [106] believe that QWSC requires multiple simultaneous QoS attributes. They explored the suitability of many-objective EAs for addressing the binding problem of web services on the basis of a real-world benchmark with nine QoS properties. A complete comparative study demonstrates that these techniques can achieve a better trade-off between all the QoS properties.

3.3 Particle swarm optimisation (PSO) based computing approaches

PSO, introduced by Kennedy and Eberhart in 1995 [107], uses a simple mechanism that mimics swarm behaviour in birds flocking to guide the particles to search for globally optimal solutions. The most important feature of PSO is its elitism learning to leading the all solutions to the found-so-far promising neighbourhood.

The strengths of PSO are easy to implement, less parameters and rapid convergence. The weakness of PSO is its rapid loss of population diversity and being trapped.

3.3.1 Particle swarm optimisation: Wang et al. [63] proposed an improved PSO with adaptive weight adjustment, local best and non-uniform mutation strategies to solve web service selection problem and obtained satisfactory results. Wang et al. [108] also presented a service pool model to help users save more composite plans in one selection process and an efficient service pool construction algorithm based on PSO. Based on the notion of skyline, Wang et al. [109] proposed a fast PSO in cloud computing environment for cloud-based web services (CWSs) composition. Skyline operator is to prune redundant CWS candidates and then employs PSO to select CWS from amount of candidates for composing composite services.

Fan et al. [110] designed a PSO to facilitate the dynamic web service selection, in which three kinds of velocity operator and one position evolution equation are proposed. A no-hope/re-hope criterion guarantees the diversity of PSO and improves the global search ability. The service selection method is independent of the candidate services number, whose service selection efficiency will not decrease with the increase of available services. Wang and He [111] designed a new and automatic composition algorithm based on global QoS optimisation and chaos PSO. In order to realise the effective integration and unified management of a logistics centre’s resources, Li et al. [112] proposed a logistics resource expression and service execution solution method to find the best concrete web service binding to each corresponding abstract service.

Li et al. [113] proposed a grouping PSO based on user preference for web service selection. The algorithm not only achieves a balance between convergence speed and swarm diversification, but also makes users’ satisfaction reach maximum by the strategies of particles grouping, different clusters evolving alone, timely updating each cluster and using fuzzy constraints to express user preferences. Ludwig [114] hybridised PSO with the Munkres approach for web service composition, which is different from related works in two aspects: (i) two meta-heuristic methods based on PSO are introduced to address the optimisation problem; and (ii) several workflow requests are processed simultaneously.

3.3.2 Hybrid PSO: Sun et al. [115] proposed a quick service selection approach, which takes full use of PSO with adaptive jump and fuzzy logical components to global QoS constraints to local ones with adaptive quality level. A hybrid PSO with clonal selection principle and several beneficial strategies is proposed [116] for QWSC problem. An improved local best strategy is presented in order to achieve the same impacts on the local fitness of a component task and the fitness function of a composite web service when a different candidate service is replaced.

A new cooperative evolution algorithm [117] is proposed on the basis of PSO and SA, which inherits the rapid convergence of PSO and global convergence of SA, and overcomes the
disadvantages of being prone to be trapped by local optimum of PSO and slow convergence of SA. A multi-objective PSO is further designed for the multi-objective service composition with QoS global optimisation. A PSO-based web services selection model [118] is presented to select the optimal web services combination by evaluating QoS attributes for the dynamic e-business application, and the proposed model is realised by a multi-agent system (MAS). In MAS, agents can represent autonomous service requester, service providers and other functions in the e-business applications.

Fethallah et al. [119] also proposed a reactive MAS based on local version PSO in which each group has a full mesh topology and is more immune to local optima than global version. Xu et al. [120] proposed a predatory search-based chaos Turbo particle swarm optimisation (PS-CTPSO) algorithm based on the predatory search strategy. PS-CTPSO algorithm identifies a feasible service via a global search, and subsequently, it obtains suitable candidate services within the corresponding chain. Different web services are grouped into the same class, depending on whether they have the same input and output sets, thus reducing the number of combinations and improving the searching efficiency.

3.3.3 Multi-objective PSO: The main contribution of [121] is that the dynamic web service selection problem with QoS optimal is transformed into a constrained multi-objective optimisation. A set of Pareto optimal services composition process with constraint principle are arrived by accelerating global and detail searching speed. An efficient multi-objective services selection algorithm [122] is proposed based on PSO. It also models the service selection problem as a constrained multi-objective optimisation problem. The services in each sub-service set are first sorted according to the pareto domination concept. The new sub-service set, whose size is far less than the original one, is then constructed and finally output the pareto optimal set. Similar works are from [123, 124]. Yu et al. [125] transformed the top-k ranking task into a multi-objective programming problem when user preferences are considered along with user queries. Then an improved discrete PSO is proposed to make the aggregate rankings satisfy user queries and user preferences both. It makes sure the candidate results fit the user’s preferences and can be located quickly and accurately in a large-scale discrete solution space.

Service selection on the grid consists of finding an efficient algorithm which can match multiple client requests and service providers efficiently, while optimising multiple QoS parameters to ensure unbiased allocations of requesters to the service providers. Guha and Ludwig [126] proposed a multiple objective PSO algorithm using crowding distance technique and the constraint satisfaction based matchmaking algorithm. In order to design effective decision-making method for the personalised composite service, Fan [127] constructed an efficient multi-objective multi-constraint PSO algorithm with Niche technology. Considering the correlations among resource services, a new multi-objective PSO [65] is proposed for solving the multiple grid resource service selection and composition problem. The main features are: (i) it combines the non-dominated sorting technique to select the global best and private best position; (ii) particle updating formulation are dynamically updated for a compromise between global exploration and local exploitation; and (iii) permutation-based and objective-based population trimming operators are applied to maintain population diversity.

3.4 Neural network-based computing approaches

Artificial neural networks (ANNs) [128] are computational models naturally performing a parallel processing of information. Essentially, an ANN can be defined as a pool of simple processing units (neurons) which communicate among themselves by means of sending analogue signals. These signals travel through weighted connections between neurons. Each of these neurons accumulates the inputs it receives, producing an output according to an internal activation function.

The strengths of NN are its powerful non-linear learning ability, fault tolerance and robust capacity, and predicting and generalising abilities. The weakness of NN is its difficulty of learning algorithm design, determining the number of layers and number of neurons.

3.4.1 Neural network: Utilising other users’ experience on services to fulfil the prediction task based on an neural net, Gao and Wu [129] designed a new NN-based QoS service selection framework and extended existing QoS ontology by adding new dynamic QoS attributes. Client may be more capable of picking out the best service given a personalised service request plus the service performance prediction. Based on a novel method to utilise the earlier information of the cooperation between the devices and the context information, Cai et al. [130, 131] proposed a novel ANN-based service selection (ANNSS) model and a service evaluation standard for service provider. User can acquire an effective guidance and choose the most appropriate service. ANNSS algorithm is implemented in an actual power supply system for communication devices and an actual ubiquitous web services system.

Al-Masri and Mahmoud [132] proposed a back-propagation NN framework aiming at efficient discovering the web services of interest based on service classification. Its main contribution is to apply the NN model to determine suitable web services to provide an overall assessment of web services. Through QoS evaluation of web services, NN is capable of identifying those services that belong to a variety of class objects.

Rajput and Haider [133] proposed a semantic annotation framework, BNOSA (Bayesian network and ontology based semantic annotation), which aims at annotating unstructured and ungrammatical web data sources. The framework is highly extensible as it has the capability to extract data from any problem domain once provided with a domain-specific ontology and a corresponding Bayesian network. The semantic annotation process performed by BNOSA comprises the following three steps: (i) ontology development for knowledge representation and Bayesian network learning; (ii) information extraction with the help of ontology and Bayesian network; (iii) semantic tagging of the extracted information using semantic web technologies. Handwritten Chinese character recognition is an important research field of pattern recognition, which has attracted extensive studies. Zhuang et al. [134] constructed a handwritten Chinese character recognition service based on convolutional neural network, which tries to make effective use of handwritten based printed fonts and existing handwritten database. It achieved the state-of-the-art performance for handwritten Chinese character recognition.

3.4.2 Hybrid neural network: Yang et al. [135] proposed a new QoS evaluation model for dynamic service selection, which takes account of the interface parameters matching degree in order to find services with the best operability. They took the service selection process as a multistage decision-making problem and proposed a structured transiently chaotic neural network algorithm with high parallel performance and powerful computation ability.

Fuzzy neural network with GA [136] is proposed with a four-layered fuzzy neural network framework for semantic web services. Input nodes represent input linguistic variables in layer one. Membership nodes are in layer two, each of which is responsible for mapping an input linguistic variable into a possibility distribution for that variable. The rule nodes reside in layer three and the last layer contains the output variable nodes. Tiwari and Kaushik [137] constructed a tool WSS-NFP for web service selection based on non-functional properties using soft computing techniques, which is basically a neuro-fuzzy system for ranking and discovering web services. This tool is comparable with the existing mathematical model [4] in terms of performance and is much easier and flexible to use.
3.5 Fuzzy set-based computing approaches

Soft computing [138] differs from conventional (hard) computing, it is tolerant of imprecision, uncertainty, partial truth and approximations. Fuzzy representation and fuzzy reasoning may shed a promising light on the web services problem due to the decision maker’s imprecise perceptions under incomplete information and vague preferences when they choose component services.

The greatest strength of fuzzy set theory is its capacity to represent important uncertainties of the human knowledge with linguistic variables. The weaknesses of fuzzy set theory are (i) incapable to generalise, (ii) not robust in relation the topological changes of the system, (iii) dependent on the expert to determine the inference logical rules.

Conventional quality evaluation models just focus on the attributes which are expressed by exact, numerical values. However, some quality attributes are related to the subjective feeling of human beings, which is preferable to be expressed by fuzzy values. Therefore, fuzzy set or rough set is very suitable to express the QoS of services.

3.5.1 Fuzzy set and rough set: As we know, the non-functional properties are strongly influenced by customer’s feeling and experience, which are imprecise and uncertain. A scheme [139] is proposed for right selection of QoS-aware of web services which exploits fuzzy logic to locate and select the right service based on customer’s preference or satisfaction degree. This scheme revises a traditional entropy weighting approach to objectively analyse the weights of QoS criteria from the consumers’ fuzzy preferences and confidence interval. Wang [140] proposed another new fuzzy decision model under vague information, which extends the Max–Min–Max composition of intuitionistic fuzzy sets for QoS-aware web services provisioning. The proposed model has the following features:

- It not only deals with the decision maker’s imprecise perceptions under incomplete information, but also objectively determines the importance weights of QoS criteria.
- It enables decision makers to select QoS-aware services from the marketplace.

In order to measure QoS of the subjective feeling of human beings, Liu et al. [141] presented a new web service quality evaluation model, which uses real numbers, interval numbers and linguistic expression to describe the quality of geo-services more accurately.

Ladner et al. [142] extended the rough set approach for web services discovery to enhance and improve the interoperability of meteorological web services. This entails the use of rough set techniques in the feature selection component of the classifier.

Lin et al. [143] aim to resolve such conflicts that the distinct views of service providers and customers on the services QoS and to ensure consensus on the QoS characteristics in the web services selection. It proposes a QoS consensus moderation approach in order to perform QoS consensus and to alleviate the differences on QoS characteristics in the selection of web services.

Since service requesters usually have vague preferences, Wu and Yuan [144] proposed a user-preference-based selection engine to compose services, which allows users to define non-quantifiable factors and policies to represent their preferences. The workflow is divided into two processes: (i) the user process allows users to define the subjective factors and also to express these opinions in formal policies; (ii) the engine process performs all the calculation works that include acquiring values and using the users’ policies with fuzzy logic to give an evaluation for every possible web service combination.

Sora et al. [145] proposed a fuzzy-QoS approach, which is a complete architecture for incorporating the user preferences and QoS characteristics into the selection process of web services. It uses fuzzy inference for ranking the candidates, in which fuzzy inference is performed on sets of automatically generated fuzzy rules for each set of individual preferences. To help the non-expert end users define their QoS queries and understanding on the services registry, Molaeipour and Ding [146] proposed an enhanced selection model to handle both exact and fuzzy requirements, and return super-exact and partial matches and then rank them using a two-level ranking algorithm.

Wang et al. [147] proposed a web service selection approach based on QoS estimation (WSSAQE). Features of WSSAQE are: (i) the fuzzy synthetic evaluation method makes QoS estimation quite reasonable; (ii) the proposed slight non-uniform mutation operator makes the estimation more accurate; (iii) it is insensitive to the time cost of service selection; (iv) it considers not only the context of customers, but also the uncertainty of QoS, which greatly alleviates the deviation between requiring and receiving QoS.

Wang et al. [148] also introduced a novel QoS measure approach to efficiently measure the QoS of services when selecting. The essence is to exploit three factors, namely, service providers, the context of customers and historical statistics into QoS measure. A fuzzy synthetic evaluation method is adopted to obtain the initial weights for service providers and the context of customers. The initial weight of historical statistics dynamically depends on other two factors. Finally, a weighted average method is used to measure the QoS attribute values of web services on the basis of the received weights. In order to measure the QoS attributes, Tao et al. [149] took the user’s feeling and transaction experiences into concern on the resource service optimal-selection in a manufacturing grid system.

On the basis of semantic web services and fuzzy-set theory, Bai and Liu [49] presented a fuzzy set-based matching approach for semantic annotation attributes of web services intended to improve the precision rate and the recall rate for material selection services. Based on the semantic web service framework, semantic matchmaker, specification matching and probabilistic matching, Liu et al. [150] proposed another fuzzy-set approach for semantic annotation attributes of web services. They first described the fuzzy definition for semantic attributes of web services to expand the service matching scope, and then advised a fuzzy matchmaking framework for collaborative material selection services.

Sharifi et al. [151] proposed a fuzzy logic-based framework that makes service description, discovery and selection more flexible. The purpose of this research is to expand the scope of the existing descriptions of web services, enabling them to describe uncertain knowledge, and to provide an integrated solution for fuzzy description-based service organisation and matching.

3.5.2 Hybrid fuzzy set: A decision model combining subjective and objective QoS factors [152] is given for personalised web service selection, and a fuzzy query interface is designed to accept users’ service demands to solve the problem that the web service lacks service quality information. In order to research the advanced feedback control strategies for better web servers QoS, Loudini et al. [153] utilised fuzzy logic controller (FLC) to enhance the system capabilities. Then, the SA is used to optimise the FLC parameters with efficient tuning procedures.

Pernici and Siadat [154] used a fuzzy inference system for capturing overall QoS and selecting service adaptation strategies with fuzzy rules. First, the degree of QoS can be inferred by changes of QoS parameters using a fuzzy inference engine. Second, a decision making for adaptation can be inferred by trading-off between the overall QoS, its importance and substitution cost. Searching and composing services is always a computationally demanding work with the growth of data at an era of cloud computing. Zhu et al. [155] designed a system which combines fuzzy logic with GraphPlan algorithm for this problem. Fuzzy rules are enhanced according to users’ preferences. The solution with the optimal QoS value is selected and used in GraphPlan construction.
3.6 Ant colony optimisation (ACO) based computing approaches

ACO was introduced in [156] as a novel nature-inspired method for hard combinatorial optimisation, which is inspired by the foraging behaviour of real ants. When searching for foods, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates it and carries some food back to the nest. During the return trip, the ant deposits a pheromone trail on the ground. The deposited pheromone and its amount may depend on the quantity and quality of the food, guides other ants to the food source. It has been shown that indirect communication among ants via pheromone trails enables them to find shortest paths [157] between their nest and food sources.

The strengths of ACO are positive feedback accounts for rapid discovery of good solutions, inherent distributed parallelism and adaptability to problems and other methods. The weakness of ACO is its weak local search and universality, slow convergence and lack of theoretical explanation on its success.

3.6.1 Ant colony optimisation: QoS-aware service selection, inherently, is a complex combinatorial optimisation problem because there are so many similar component services waiting to be selected and composed harmoniously. According to the characteristic of grid services and the heuristic information of the user synthesis satisfaction function, Zheng [158] presented a QoS-aware service selection algorithm, in which a user satisfactory degree model for grid services and an extended ACO with a novel ant colony rule are incorporated. A cost minimising service composition model for data-intensive applications is proposed in [159] and an ACO algorithm is constructed to solve it.

Ambient media services in a smart home monitoring environment should be ubiquitous, adaptive and robust with respect to access and delivery. Based on dynamic user preferences and satisfaction, Hossain et al. [160] addressed this issue by proposing an ant-inspired service selection framework. The framework uses the potential of the ant-based service selection approach by considering user satisfaction and preferences, which is the key factor for media service selection to find the best way of selecting appropriate media services. It also enables different categories of residents to access various media services in such a way that their experiences are optimised with regard to their surrounding environment.

Pop et al. [161] proposed an ant-inspired semantic web service selection method, which used the previously described composition graph and a multi-criteria function to identify the optimal composition solution according to QoS user preferences and semantic quality. To improve the efficiency and accuracy of services discovery process, they organised the set of available services with similar functionality into service clusters.

Combined with the positive feedback mechanism of ACO, Wang et al. [162, 163] first analysed the web service selection problem with basic principle of ACO and then transformed the QoS-aware web service selection problem into a shortest path problem. Finally, they gave the solving steps of web service selection based on ACO and compared the algorithmic performance under different parameters.

3.6.2 Hybrid ACO: As we know, it is very important to select a best service composition plan quickly from numerous schemas with the constraints maximising the satisfaction of users and meeting the QoS requirements. A dynamic web services composition hybrid method [164] is presented based on the optimal path method of the weighted directed acyclic graph, which combines ACO and GA for the QWSC problem. A new dynamic ant-colony genetic hybrid algorithm [165] is proposed to determine the invoking timing of genetic and ant-colony algorithms for the best fusion evaluation strategy. So, the optimisation ability is maximised and the overall convergence speed is accelerated.

As more and more web services become available, QoS is becoming a decisive factor to distinguishing composite web services. A new optimisation algorithm, C-MMAS [166], is proposed by integrating Max–Min ant system into culture algorithm framework for QWSC problem. C-MMAS algorithm consists of population of ants, belief space and communication protocols between them. Population space contains a series of solutions. Belief space keeps the excellent solutions obtained in the evolutionary process of population space; these excellent solutions are regarded as experience and knowledge and used to guide the evolution of C-MMAS in population space. Wang et al. [167] extended the work of [166] to achieve web services selection efficiently with excellent comprehensive quality. First, a composite selection model based on generic QoS and domain QoS is presented. Second, a novel culture Max–Min ant optimisation algorithm is constructed to solve the service selection problem based on genetic and domain QoS.

3.6.3 Multi-objective ant colony optimisation: Yang et al. [168] applied multi-objective ant colony optimisation algorithm to solve the dynamic web service selection problem. The service selection model with QoS global constraints is converted into a multi-objective optimisation problem with user constraints. Zhang et al. [169] proposed a novel multi-objective optimal-path selection modelling for QoS-based dynamic services composition. A strategy is presented to decompose a composite service with a general flow structure into parallel execution paths, which is scalable to support composition of very complex web services.

3.7 Artificial immune system (AIS) based computing approaches

In biological immune system (BIS), an antigen is any molecule that can be recognised by provoking its response. This immune response is specific to each antigen. Inspired by the BIS, principles and models, AISs [170, 171] have been developed since 1990s as a new branch in computational intelligence. A number of AIS models have been proposed for problem solving, which include (i) clone selection algorithm; (ii) artificial immune network; (iii) negative selection algorithm; and (iv) danger theory. The candidate solutions in AIS are called antibodies, which are associated with an affinity measure that provides an indication of its performance.

The strengths of AIS are the existence of several well-established theories that reasonably explain immune functions, patterns recognition and memory mechanisms of the immune system. The weakness of AIS is its lack of information communications between antibodies and premature convergence.

3.7.1 Artificial immune optimisation: An immune optimisation algorithm (IOA) is first adopted to dynamic QWSC [172], which is based on the longest path method of weighted multistage graph. An abstract composition plan is mapped to concrete services and obtains a graph structure having services in the nodes and QoS values on the arcs.

A novel heuristic web service selection method [173] is proposed using IOA together with an efficient encoding and mutation method to reduce the search space. Algorithm involves two steps: (i) immune selection operation maintains the antibody population diversity and clonal selection; (ii) vaccine operation accelerates the convergence during the evolution provides heuristic information. During the clonal selection operation, the mutation activity is not randomly applied to all cells as usual, but only to a part of cells.

Pop et al. [174] proposed a framework, hybrid service composition method combining AI planning graph technique with an IOA. Main contributions of [174] are: (i) an enhanced planning graph model; (ii) a matrix of semantic links which stores semantic similarity links between the services involved in composition; (iii) an immune-inspired solution for extracting the optimal composition directly from the enhanced planning graph.

Ha and Chen [175] proposed an adaptive evaluation model of service selection based on artificial immune network and evaluation tree. The evaluation tree framework is able to collect the consumer feedback to service registry for trustworthiness QoS.
Immune network algorithm with new mechanism of antibody promotion and suppression, AiNet, is used to cluster the QoS attributes vectors.

3.7.2 Hybrid artificial immune optimisation: Neuroendocrine immune (NEI) system is inspired to design a decentralised, evolutionary, scalable and adaptive system for web service composition and management. Based on the novel intelligent network model inspired from NEI, Ding et al. [176] proposed a method for web service emergence by designing a bio-entity as an autonomous agent to represent web service. The service emergence system is constructed to obtain the desirable characteristics from self-organising, cooperating, and composing of web services. When one of the bio-entities cannot satisfy user’s request, they will mutually communicate and cooperate to finish the service emergence for the request.

It is well known that biological systems offer many clues to design optimisation techniques since they have exhibited excellent self-optimisation and self-organisation capabilities in a decentralised way without the existence of a central coordinator. Salomie et al. [177] analysed how principles inspired from immune and evolutionary systems can be combined with reinforcement learning to design a hybrid method for selecting the optimal or near-optimal composition solution. The hybrid algorithm adapts an enhanced CLONALG [178] with evolutionary computing and reinforcement learning to ensure that the optimal composition solution is obtained in a short time and without processing the entire search space.

Based on the analysis of clonal selection algorithm having no information communications and the neighbouring information of PSO being not fully exploited, a hybrid algorithm is proposed for [179] web service selection with several heuristic strategies. Based on the negative selection principle, Zhao et al. [180] also proposed a negative selection immune algorithm for web service selection. Que et al. [181] recently proposed an improved adaptive immune GA for optimal QoS-aware service composition selection in cloud manufacturing, in which a comprehensive mathematical evaluation model with four critical QoS-aware indexes (time, reliability, cost and ability) is constructed.

3.8 Social network (SN) based computing approaches

A SN is a graph representation of all the interactions that occur between people and services in a composition environment. In the increasingly popular Web 2.0 sites, like the SNs LiveJournal, Tribe.net, Renren and Facebook, people can create, use and share services in communities and SNs. These services can be simple web services or more sophisticated services as mashups. Many researchers have found SN analysis useful in analysing such networks; others provide models to leverage the social interactions in a way to enable and facilitate composite services creation.

The strengths of SN are: (i) allows users to discuss ideas, ask questions and share links; (ii) provides an opportunity to enhance information discovery and widen business contacts; (ii) provides an opportunity to enhance communications and directs interested people to specific services. These services can be simple web services or more sophisticated services as mashups. Many researchers have found SN analysis useful in analysing such networks; others provide models to leverage the social interactions in a way to enable and facilitate composite services creation.

Mobile social networking platform (MSNP) is a new form of SN in which users are capable of interacting with their surroundings via their mobile devices in public mobile peer-to-peer environments. Chang et al. [187] proposed an adaptive mobile-hosted service-oriented mediation framework for MSNP. The framework supports proactive service discovery by using context-aware user preference prediction scheme to reduce the latency caused by the service discovery process. Wang et al. [188] presented a service selection mechanism for mobile social networking service based on user context, trustworthy network and collaborative filtering algorithm. Thus, a ‘User-Service-Context’ three-dimensional collaborative filtering model is proposed. A novel service-oriented mobile social networking platform (MS2A) is proposed [189] for disaster situations. MS2A consists of two layers: service layer and application layer. With dynamic and automatic service collaboration support, it enables people to easily collaborate and help each other through their mobile devices in disaster situations.

Considering how to leverage social interactions to enable and facilitate composite services creation, Maaradji et al. [190] proposed a general framework, named Social Composer (SoCo), which relies on the interactions between users and web services to advise on the next actions to take. SoCo provides dynamic recommendations for services discovery and selection based on the users’ interactions and a SN is implicitly built from the interactions between users and services. When considering the incommensurability and confliction among non-functional properties of services, a mixed intelligent optimisation algorithm [191] is designed for services selection problem using social cognitive optimisation and maximum entropy method.
Elnaffar and Maamar [192] proposed an algorithm for searching a SN of web services in order to select an effective set of web services, which collaborate each other for the main goal of the composite web service at hand. The functional similarity constraint adds reliability to the operation of the composite web service in case one of the constituent services fails (i.e. better fault-tolerance). The communication cost is computed by measuring the SN diameter, which is the longest and shortest path between any two nodes.

To overcome the shortcomings (e.g. neglecting the rich information) of web services and inspired by the conventional human SNs on the net, like Facebook and Twitter, Maamar et al. [193] developed LinkedWS, a SNs discovery model to capture the different interactions that occur between web services. Based on these interactions, specialised relationships and substitution are spawned and discerned. Maamar et al. [193] also tells us that web services should no longer be treated as isolated components that respond to users’ queries. Contrarily, web services compete and collaborate with each other, different web services may replace each other. Maamar et al. [194] also described how service engineers can capitalise on web services’ interactions – namely, collaboration, substitution and competition – to build SNs for service discovery. Through these networks, web services identify those peers with which they would like to work, those that can replace them in case of failure, and those that compete against them for selection.

Most of trustworthy web service selection simply focus on individual reputation and ignore the collaboration reputation between services. Wang et al. [195] provided a trust model which not only considers individual’s reputation but also the collaboration reputation. Main contributions include, (i) construct a web service collaboration network (WSCN) to support trust web service selection; (ii) propose a collaboration reputation model, which is evaluated by invoking and invoked reputation. The invoking reputation depends on the community structure hiding in WSCN, thus, a web service with more recommenders indirectly proves its credibility. Individual service recommendation is a hot research spot nowadays. Gong et al. [196] presented a service recommendation algorithm URPC-Rec, which first clusters users based on their history behaviours, and then makes personalised recommendations for users considering both the clustering results, user basic information and relationships.

3.9 Other bio-inspired computing approaches for web service composition

NIC aims to develop new computing techniques after getting ideas by observing how nature behaves in various situations to solve complex problems. Research on NICs has opened many new branches, such as evolutionary computation, neural networks, fuzzy logic, AIs, swarm intelligence and so on. Besides, there are also many NIC approaches, which are also used to solve the web service selection and composition.

As the current applications are not yet too much, the detailed review is omitted. The NIC approaches, artificial bee colony algorithm [197], cross-entropy [198], cuckoo optimisation algorithm [199], differential evolution [200], finite state machines [201], gravitational search algorithm [202], harmony search [203], SA [204], support vector machine [205], taboo search [206], as well as the references of applying these methods, are listed in Table 2. Please refer to the initial references if you have further interests on any of them.

4 Discussions, remarks and future research

With the increasing availability of web services as a solution to various enterprise and daily application integration, web services selection and composition are becoming the chief priority for service providers and consumers.

Table 2 NIC approaches and corresponding references

| NIC approach                          | References                  |
|--------------------------------------|-----------------------------|
| artificial bee colony                 | [207-210]                   |
| cross-entropy                         |                             |
| cuckoo optimisation algorithm         | [212-214]                   |
| differential evolution                | [215]                       |
| deep learning                         | [210]                       |
| finite state machines                 | [5]                         |
| gravitational search algorithm        | [216]                       |
| harmony search                        | [217, 218]                  |
| simulated annealing                   | [219, 221]                  |
| support vector machine                | [222]                       |
| taboo search                          | [223]                       |
| fruit fly optimisation algorithm      | [224]                       |
| invasive weed optimisation            | [225]                       |
| whale optimisation algorithm          | [226]                       |
| glowworm swarm optimisation           | [227]                       |

4.1 Discussions

4.1.1 Nature-inspired approaches: EA, fuzzy set, social and cloud computing and petri net are widely developed for the QoS-aware web services problem solving. A little surprising is that some widely researched NIC approaches, such as neural network and differential evolution, have not applied to QWSS too much. Some well-known NIC variants for numerical optimisation may also deserve some considerations on their applications.

Semantic web services may pave a promising way for future research, which augments web service descriptions and empowers web services using semantic web technology. In order to enable the automatic or semi-automatic exploitation of semantic annotations with respect to discovery, composition, execution and inter-operability of web services, two major initiatives aim to realise it by providing appropriate description, namely: OWL-S [228] and WSMO [229].

4.1.2 Social and cloud computing: SN and cloud computing are the latest trends and they capitalise on users’ willingness to interact, share, collaborate and make recommendations. The growing number of initiatives reflecting the blend of social computing with SOC is certainly a positive sign of this area’s growing importance [230]. A social-based or cloud-based connection offers a wider view by stressing the interactions that occur between users, between web services, and between users and web services. However, why social web services? Social computing [231] may give some guidelines to us for QoS-aware web services compositions.

Social networking-inspired approaches attempt to involve humans in the loop to create a pool of services [25]. Social computing represents relationships that people experience daily, such as trustworthiness and fairness, via automated structures known as SNs. SOC abstracts application design using the principles of ’I offer services that somebody else might require’ and ’I require services that somebody else might offer.’ For those who offer or require services, issues arise regarding how to advertise services, discover them with respect to needs, compose them, trust them when you find them and replace them when they fail.

Secondly, current discovery techniques are registry-based and rely on syntactic and semantic descriptions of web services’ interfaces. Registries have several drawbacks – for instance, syntactic discovery returns results with low precision, web services are treated as independent elements in these registries, and the present registries do not record services’ past interactions. To address these issues, we embrace SNs to develop what we refer to here as social/cloud web services [193].

Recently, a new trend of cloud computing has been observed and studied in both academy and industry domains [322]. Essentially, cloud computing paradigm is a natural evolution of service computing towards a more flexible, dynamic, scalable and business-oriented environment. Compared with the various and approximately infinite application requirements from end users, the web services held by a cloud platform are usually limited.
Heterogeneity

User personalisation and privacy supported

Integration by service providers. These services cannot respond to

Web of Things (W2T) [4] Liu, Y.T., Ngu, A.H.H., Zeng, L.Z.: [5] Kondratyeva, O., Kushik, N., Cavalli, A., et al.: Evaluating web service quality using finite state models’ 13th Int. Conf. on Quality Software, Nanjing, China, 2013, pp. 95–102.

Web Services Architecture Requirements Working Group, 2004. Available at http://www.w3.org/TR/ws-reqa

‡ It is a big challenge to meet users’ various

Requirements, not only some specific users;

User personalisation and privacy supported: Ideally, the users’ preferences on different objectives will be captured automatically and precisely, especially for mobile users.

Heterogeneity: field study and application oriented service type heterogeneity.

From the view of these challenges, most literatures cannot be well suited for the SOA systems. It is either because only limited number of QoS criteria are supported, or user personalisation and privacy are not considered, or the algorithm is not well scalable with the increase of service providers.

4.2.2 Ubiquitous integration in the hyper world: Wisdom Web of Things (W2T) [236] is the next generation of networks, which provides ubiquitous wisdom services in a ubiquitous network in the hyper world. The hyper world consists of the physical world, cyber world and social world. The core of W2T is to adapt to the physical world and provide intelligent services, which can adapt to the changing requirements of human or organisations, as well as the changing data, computing resources, and security requirements. Adaptiveness is the key issue of realising the harmonious unity of human-information-thing. Therefore, how to provide ubiquitous and wisdom services in the hyper world based on the ubiquitous networks is a promising direction in future.

5 Acknowledgment

This research is supported by National Natural Science Foundation of China (71772060, 61873040, 61375066).

6 References

[1] Web Services Architecture Requirements Working Group, 2004. Available at http://www.w3.org/TR/ws-aereq

‡ Zeng, L.Z., Boualem, B., Anne, H.H., et al.: ‘QoS-aware middle ware for services composition’, IEEE Trans Softw. Eng., 2004, 30, (5), pp. 311–327

‡ Xu, C.H., Chen, X.H., Liang, X.M.: ‘Dynamic services selection algorithm in web services composition supporting cross-enterprises collaboration’, J. Central South Univ. Technol., 2009, 2, pp. 269–274

‡ Liu, Y.T., Nga, A.H.H., Zeng, L.Z.: ‘Qos computation and policing in dynamic web service selection’, WW 2004, New York, NY, USA, 2004, pp. 66–73.
This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/).
[226] Gavvala, S.K., Jatoth, C., Gangadharan, G.R., et al.: ‘Qos-aware cloud service composition using eagle strategy’, Future Gener. Comput. Syst., 2019, 90, pp. 273–290

[227] Liu, Z.C., Guo, S.S., Wang, L., et al.: ‘A multi-objective service composition recommendation method for individualized customer hybrid MPA-GSO-DNN model’, Comput. Ind. Eng., 2019, 128, pp. 122–134

[228] Martin, D., Burstein, M., Hobbs, J., et al.: ‘OWL-S: semantic markup for web services’. Available at http://www.w3.org/Submission/OWL-S

[229] Romana, D., Kellera, U., Lausena, H., et al.: ‘Web service modeling ontology’, Appl. Ontol., 2005, 1 (1), pp. 77–106

[230] Ko, M.N., Cheek, G.P., Shehab, M., et al.: ‘Social-networks connect services’, Computer, 2010, 43 (8), pp. 37–43

[231] Wang, F.Y., Carley, K.M., Zeng, D., et al.: ‘Social computing: from social informatics to social intelligence’, IEEE Intell. Syst., 2007, 22 (2), pp. 79–83

[232] Tan, B.X., Ma, H., Mei, Y., et al.: ‘Evolutionary multi-objective optimization for web service location allocation problem’, IEEE Trans. Serv. Comput., In press, 2018

[233] Qi, L.Y., Dou, W.C., Zhang, X.Y., et al.: ‘A QoS-aware composition method supporting cross-platform service invocation in cloud environment’, J. Comput. Syst. Sci., 2012, 78, pp. 1316–1329

[234] Jula, A., Sundararajan, E., Othman, Z.: ‘Cloud computing service composition: a systematic literature review’, Expert Syst. Appl., 2014, 41, (8), pp. 3809–3824

[235] Zhao, L.P., Ren, Y.Z., Li, M.C., et al.: ‘Flexible service selection with user-specific QoS support in service-oriented architecture’, J. Netw. Comput. Appl., 2012, 35, pp. 962–973

[236] Gao, Y., Lin, M., Wang, R.: ‘Adaptive support framework for wisdom web of things’, World Wide Web, 2013, 16, pp. 379–398