open science

QoS-Aware Web Services Selection Based on Fuzzy Dominance

Amal Halfaoui, Fethallah Hadjila, Fedoua Didi

▶ To cite this version:

Amal Halfaoui, Fethallah Hadjila, Fedoua Didi. QoS-Aware Web Services Selection Based on Fuzzy Dominance. 5th International Conference on Computer Science and Its Applications (CIIA), May 2015, Saida, Algeria. pp.291-300, 10.1007/978-3-319-19578-0_24 . hal-01789939

HAL Id: hal-01789939
https://inria.hal.science/hal-01789939

Submitted on 11 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

QoS-aware Web services Selection based on Fuzzy Dominance

Amal Halfaoui, Fethallah Hadjila, and Fedoua Didi

Computer Science Departement,
Tlemcen University, Algeria
{a_halfaoui,f_hadjila,f_didi}@mail.univ-tlemcen.dz
http://www.univ-tlemcen.dz/

Abstract. The selection of an appropriate web service for a particular task has become a difficult challenge due to the increasing number of web services offering similar functionalities. Quality of web services (QoS) becomes crucial for selecting web services among functionally similar components. However, it remains difficult to select an interesting Web services from a large number of candidates with a good compromise between multiples QoS aspect. In this paper, we propose a novel concept based on dominance degree to rank functionally similar services. We rank Web services by using a fuzzification of Pareto dominance called Average-Fuzzy-Dominated-Score($AFDetS()$). We demonstrate the effectiveness of the $AFDetS$ through a set of simulations by using a real Dataset.

Keywords: web service selection, dominance, Skyline, Ranking,QoS

1 Introduction

Nowadays, an increasing number of Web services is published and accessible over the web, they are designed to perform a specific task, which essentially consists of either altering the word state (e.g.,an on line shopping service) or returning some information to the user (e.g.,news Web service).

As the Web is populated with a considerable number of Web services, there exists a large number of service providers competing to offer the same functionality, but with different Quality Of Service(QoS) such as response time, price, etc. Consequently, QoS is thus a crucial criterion to select among functionally similar Web services.

Example. Consider a Web service for sending SMS, there are many Web services providing this functionality (e.g., Click Send, Inteltech, Etc.), but with different QoS. Table1 provides such functionality along with real QoS parameters taken from the publicly available Quality of Web services data.[1] Web services were obtained by using the keyword SMS which represents the tag associated to the functionality of the desired Web services. Each Web service has four QoS parameters q_1, q_2, q_3 and q_4, says respectively Response Time, Throughput

[1] http://www.uoguelph.ca/ qmahmoud/qws

(i.e.,Total Number of invocations/period of time) Reliability(Ratio: number of error messages/total messages) and Best Practices (the respect of the specifications). To select an adequate Web service, users need to examine all of them

Table 1. A set of Sending SMS Web Services

Service	provider	operation	$q1(ms)$	$q2$(hits/sec)	$q3$(%)	$q4$(%)
S_1	acrosscommunications.com	SMS	113.8	5.2	81	84
S_2	sjmillerconsultants.com	SMS	179.2	0.7	65	69
S_3	webservicex.net	SendSMS	1308	6.3	67	84
S_4	webservicex.net	SendSMSWorld	3103	5.3	79.3	91
S_5	smsinter.sina.com.cn	SMSWS	751	6.8	64.3	87
S_6	sms.mio.it	SendMessages	291.07	5.2	53.6	84
S_7	www.barnaland.is	SMS	436.5	4.5	43.2	84
S_8	emsoap.net	emSoapService	424.54	4.3	11.9	80

manually. The user may also face difficulties in balancing between different quality metrics. The skyline presents a good solution for reducing the number of candidate Web services [1],[2] and simplifying the process of selection as it overcomes the major limitation of the current approaches that require users to assign weights over different QoS attributes. The skyline is a subset of Web services that are not(Pareto) dominated by any other Web service. A Web service S_i is said to Pareto domine another Web service S_k if and only if S_i is better than or equal to S_k in all QoS parameters and better than S_k in at least on one QoS parameter.

According to our example (Table1), the service S_1 dominates S_6,S_7,S_8. Services S_1,S_3,S_4,S_5 belong to the skyline and they are no comparable between them. We can remark that computing skyline reduce the candidates services, in our example we eliminates 50% of the candidate services. However, it remains a challenge to compute skylines in high dimensional data [3],[4]. In addition to that, on the report of [5] the authors show that the skyline may lose some interesting Web services like S_6 which is dominated by S_4 while S_4 is the worst service in term of response time, however S_6 has a good response time and is closer to S_4 on the other QoS parameters.

Motivated by this, we propose an extension of Pareto dominance relationship called Averaged-Fuzzy-Dominated-score $AFDetS()$ to associate a score to each service and rank them, We also propose a comparison between the dominated-score $AFDetS$ and Dominating score used in [5] and confirm that the use of the Dominated score is more interesting than the Dominating score in Ranking service, this fact is also confirmed in [19]. The rest of the paper is organized as follows. In the next section, we discuss related work. In Section 3, we provide the formal definition of $AFDetS$ and show it application on our example Table1. Section 4 presents the results of our experimentation. Finally, section 5 gives conclusions and an outlook on possible continuations of our work.

2 Related work

A lot of efforts have been devoted to the problem of QoS-aware Web service selection. Some of them use the linear programming technique [7], [8]. Linear programming techniques are used in [7] to find the optimal selection of component services and gives an extensible model to evaluate the QoS parameters, Linear programming techniques are extended in [8] to include local constraints. Others work use combinatorial model and graph model [9] where the authors use heuristic algorithm to solve the problem of service selection with multiple QoS constraints. In [10] the authors present a selection algorithm to evaluates multiples QoS based on an ontology. Nevertheless, the majority of these approaches are more suitable for limited number of Services(the selection process has an exponential space complexity) and limited number of QoS, especially when the users has to assign weights on QoS attributes.

In recent research, the skyline paradigm is introduced as a good and efficient mechanism to reduce the number of service candidates and simplify the process of selection. The idea of skyline comes from the old research like contour problem, maximum vector and convex hull and was introduced into databases by Borzsonyi [11] who develops three algorithms: BNL, DC and B-tree, this leads to develop and ameliorate several other algorithms like SFS [12], SaLSa [13], Zorder,[14]and NN[3]. Some of these algorithms exploit index structures like [14],[3] to enhance the skyline computation process. However, the size of skyline increases under a high number of QoS and sometimes privileges Web services with bad compromise between QoS.

To handle the problem of large skyline, some works combine the advantage of the skyline and ranking and define variants of skyline like [1],[15],[16] and [17]. In [15] the authors present skyline frequency concept which is the number of subspaces where a point p is skyine, however this lead to calculate skyline of all subspaces and results in a hight computational time, further more authors introduce an approximate algorithm to reduce the computation space. In [1] Chan et al. present the notion of k-dominance which relax the pareto dominance to a subset of k parameters, however There exists cyclic dominance relationship (CDR) which leads to the loss of skylines in addition k-dominance often returns an empty set. In [17] lin et al. propose top-*krepresentative* skyline but this method is more suitable for anti-correlated data [18] in addition to that, k-representative skyline is considered as NP-hard for more than three dimensional dataset. In [16] the authors present the skyline graph which maps the dominance of different skyline subspaces into a weighted directed graph and use link-based techniques to rank skyline, however, the problem of dominance on a large space is still solved. These approaches rely on Pareto dominance relationship thus, they don't consider or privilege services with a good compromise between parameters, this drawback can be solved by the fuzzification of Pareto dominance in order to rank incomparable services.

The Fuzzy dominance was used in databases community like [20] the authors show the goal of fuzzification of the concept of Pareto dominance and it application in Evolutionary Multiobjectif Optimization. Other works use this principle and applied it in Genetic or particle Swarm Algorithm. In service computing community, [5] use the fuzzy-dominance and propose the α-dominance to rank Web service based on QoS parameters and associates the fuzzy-dominating score to Web services.

Like mentioned in [20] the measures between two vectors a, b "a dominates b by degree α" and "a is dominated by b to degree α" is not symmetric, In addition to that, in [19] the authors demonstrates that the use of the dominated measure is more efficient in selecting the top-k services than the dominating measure. Our work is close to [5]. However, [5] use fuzzy-dominating relationship to compare the services instead of use fuzzy-dominated measure in ranking services. According to these observations, we define the Fuzzy Dominated relationship $Fdet$ and the Average Fuzzy dominated Score $AFDetS()$. The next section presents the definition of this concept and it utilization in our context.

3 Problem formalization

In this section, we are going to study the fuzzification of the Pareto dominance relation , and show it application on our example (Table1). To allow for a uniform measurement of Web Services, we first normalize the different QoS value in the range [0,1].

3.1 Normalization of QoS parameters

let be S a set of similar functionally services $S = S_1, .., S_n$. Suppose that we have R quantitative Q_oS values for a service S_i. we use the vector $Q(S_i) = \{Nq_1(S_i), .., Nq_r(Si)\}$ to represent the Q_oS attributes of a service S_i where the function $Nq_k(S_{ij})$ represent the k-th Normalized quality attribute of S_i. We convert the negative attributes(time, cost) into positive attributes by multiplying their values by -1 so that the higher value is the higher quality. We normalize the different QoS values in the range [0, 1], as follow :

$$Nq_k(S_i) = \frac{q_k(S_i) - Qmin(q_k)}{Qmax(q_k) - Qmin(q_k)} \tag{1}$$

Where $Nq_k(S_{ij})$ is the normalized QoS value of the Web service S_{ij} on the QoS parameter q_k and $Qmin(q_k)$ (resp.$Qmax(q_k)$ is the minimum (resp. maximum) value of the QoS parameter q_k. Table2 shows the QoS values of Web services example of Table 1 after normalization.

3.2 Fuzzification of Pareto dominance relation

Services of the same functionality differ only in term of Qos. Like mentioned above, the skyline consists of the set of points which are not pareto dominated by any other.

Table 2. Web Services with Normalized QoS

Web service	$Nq1$	$Nq2$	$Nq3$	$Nq4$
s1	1	0.74	1	0.68
s2	0.98	0	0.77	0
s3	0.60	0.92	0.80	0.68
s4	0	0.75	0.98	1
s5	0.79	1	0.76	0.82
s6	0.94	0.74	0.60	0.68
s7	0.89	0.62	0.45	0.68
s8	0.90	0.59	0	0.50

Definition 1. *(Pareto Dominance)*
Let S_i and S_j be two Web services,Given a set of d QOS parameters $Q = \{q_1, ..., q_d\}$, We say that S_i dominates S_j denoted by $S_i \succ S_j$, iff $\forall q_k \in Q$, $q_k(S_i) \geq q_k(S_j)$ and $\exists q_t \in Q,\ q_t(S_i) > q_t(S_j)$.

Pareto dominance does not differentiate between Web services with good compromise and those with bad compromise, to clarify this, let us return to our example (Table2) and consider S_4 and S_5, in fact neither S_4 dominates S_5 nor S_5 dominates S_4, the two services are incomparable and belong to the skyline because S_4 is better than S_5 in q_3 and q_4, and S_5 is better than S_4 in q_1 and q_2. However we can consider that S_5 is better than S_4 since $q_1(S_5) = 0.79$ is much higher than $q_1(S_0) = 0$. In addition to that, $q_3(S_5) = 0.76$ and $q_4(S_5) = 0.82$ are almost close to (respectively) $q_3(S_4) = 0.98$ and $q_4(S_4) = 1$. For this reason, it is interesting to fuzzify the Pareto dominance. The goal of the fuzzification of Pareto dominance is to allow a practically usable numerical comparison between two service and express the extent to which a Web service (more or less) is dominated by another one.

To compute the Fuzzy dominance degrees it's important to distinguish between the measure of two concepts : the dominating score and the dominated Score between two service S_i and S_j. The first one express the degree to which S_i dominates S_j and the second express the degree to which S_i is dominated by Sj and the measure of dominance is not symmetric. We will use in our work the concept of dominated relation. We define bellow the fuzzification of the dominated relation.

Definition 2. *(Fuzzy-Dominated Score)*
let be S a set of functionally similar services, S_i and $S_j \in S$. Let $Q = \{q_1, ..., q_d\}$ be a vector of d QoS parameters.
First we define the monotone comparison function $\mu_{\varepsilon,\lambda}$ to express the degree to which u is dominated by v, where u represent $q_k(s_i)$ and v represent $q_k(s_j)$ as follow:

$$\mu_{\varepsilon,\lambda}(u, v) = \begin{cases} 0 & \text{if } (u - v) \geq \varepsilon \\ |u - v - \varepsilon| \,/\, |\lambda + \varepsilon| & \text{if } \lambda + \varepsilon \leq (u - v) < \varepsilon \\ 1 & \text{if } (u - v) < \lambda + \varepsilon \end{cases} \quad (2)$$

Where $\varepsilon, \lambda \in [-1, 0], \varepsilon + \lambda \geq -1$
Then, we define the Fuzzy-Dominated score $FDet(S_i, S_j)$ *to express the degree to witch* S_i *is dominated by* S_j *as follow:*

$$FDet(s_i, s_j) = \frac{1}{d} \sum_{k=1}^{d} \mu_{\lambda,\varepsilon}(q_k(s_i), q_k(s_j))) \quad (3)$$

Let us reconsider our example and compare Web services S_4 and S_5 by using $FDet()$, with $\varepsilon = -0.1$ and $\lambda = -0.2$ we have $FDet(S_4, S_5) = 0.5$ and $FDet(S_5, S_4) = 0$ this mean that S_5 is not fuzzy dominated by S_4 and is little more better than S_4. This concept gives a good compromise between QoS. In fact, this is more expressing than S_4 and S_5 not comparable by Pareto dominance. In what follows, we use the $FDet()$ to rank Web services

Definition 3. *(Averaged-Fuzzy-Dominated-Score)*
In order to rank a Web service S_i *in it class* S*, we first, make pairwise comparison with the other services and associate it a score by:*

$$AFDetS(S_i) = \frac{1}{|S| - 1} \sum_{j=1, i \neq j}^{n} FDet(S_i, S_j) \quad (4)$$

Then, we retain service with lower $AFDetS()$ *on a higher ranking position*

The Table3 show the services of our example (Table 1) after computing $AFDetS$ score and ranking with $\varepsilon = 0$ and $\lambda = -0.2$

Table 3. Services'Rank according to $AFDetS()$

Rank Web service	AFDedS()	$Nq1$	$Nq2$	$Nq3$	$Nq4$
s1	0,071	1	0,74	1	0,68
s5	0,107	0,79	1	0,76	0.82
s6	0,143	0,94	0,74	0,60	0,68
s3	0,25	0,60	0,92	0,80	0,68
s7	0,286	0,89	0,62	0,45	0,68
s4	0,312	0	0,75	0,98	1
s8	0,393	0,90	0,59	0	0,50
s2	0,571	0,98	0	0,77	0

We can observe that the top service is S_1 which is better than the others in q_1, q_2 and has a good value in the other QoS parameters. We remark that services that have some $QoS = 0$ are at the bottom of the ranking. Let us consider S_6 and S_4, according to the result provided by Pareto dominance S_4 belong to the skyline, but S_6 does not, however S_4 have the worst response time(q_1)) and S_6 has a good compromise between QoS parameters. According to (Table3: Fuzzy-Dominated Score)S_4 was downgraded to the Rank 7, On the other hand, the

Service S_6 which has a good compromise between QoS parameters was set up to the 3rd rank.

From this result, we confirm that the use of $Fed()$ can give more interesting results in term of balanced of QoS than the other approaches.

4 Experimental Evaluation

In order to evaluate and prove the effectiveness of our approach, we compare the result of using Fuzzy-Dominated with the Fuzzy-Dominating score. For this purpose, we implement the function fuzzy-dominating proposed in [6] and termed it $AFDingS$ and compare it to our Approach $AFDetS$. All the experiments are conducted on the same software and hardware, which were Intel i3-2365M CPU @ 1.40GHz 4 processors, 4.0GB of RAM, Ubuntu 13.10, Netbeans 7.4. Several simulations have been made by varying the parameters:

- ε, λ,
- d:number of QoS parameter,
- n:number of services of the same class S.

For each simulation we take the Top-5 services generated by the algorithms $AFDetS$ and $AFDingS$ and compare them. Different Services' subsets were taken from the real QoS dataset provided by [23]. The dataset includes informations about 2507 real-world web services. Each service comprise measurement of nine QoS parameters. The service name and its WSDL address are also included in the dataset. We group functionally similar Services into clusters, for example the cluster "sms" (sending sms) contains 30 real services. The cluster "search"(ie. Search Engine Web services such as Google Search,Amazone, etc.) contain 92 services.

a-Varying ε and λ: We present below two scenarios (Table4) and (Table5) by varying ε and λ on a set of 30 services belonging to the class SMS. Each service has 4 QoS parameters.

Table 4. Top-5 Services Rank according to $AFDingS$,$AFDetS()$ with $\varepsilon = 0, \lambda = -0.2$

Top-5 AFDingS			Top-5 AFDetS		
Si	$AFDingS$	$Qos(q_1, q_2, q_3 q_4)$	Si	$AFDetS$	$Qos(q_1, q_2, q_3 q_4)$
S5	0.566	[0.787, 1.0, 0.758, 0.818]	S12	0.071	[1.0, 0.738, 1.0, 0.682]
S4	0.551	[0.0, 0.754, 0.975, 1.0]	S5	0.107	[0.787, 1.0, 0.758, 0.818]
S12	0.529	[1.0, 0.738, 1.0, 0.682]	S6	0.143	[0.941, 0.738, 0.603, 0.682]
S30	0.423	[0.6, 0.918, 0.797, 0.682]	S30	0.25	[0.6, 0.918, 0.797, 0.682]
S6	0.329	[0.941, 0.738, 0.603, 0.682]	S7	0.286	[0.0, 0.754, 0.975, 1.0]

Table 5. Top-5 Services according to $AFDingS$, $AFDetS()$ with $\varepsilon = -0.1, \lambda = -0.2$

Top-5 AFDingS			Top-5 AFDetS		
Si	$AFDingS$	$Qos(q_1, q_2, q_3, q_4)$	Si	$AFDetS$	$Qos(q_1, q_2, q_3 q_4)$
S4	0.443	[0.0, 0.754, 0.975, 1.0]	S5	0.0	[0.787, 1.0, 0.758, 0.818]
S5	0.421	[0.787, 1.0, 0.758, 0.818]	S12	0.036	[1.0, 0.738, 1.0, 0.682]
S12	0.036	[1.0, 0.738, 1.0, 0.682]	S6	0.107	[0.941, 0.738, 0.603, 0.682]
S30	0.321	[0.6, 0.918, 0.797, 0.682]	S30	0.143	[0.6, 0.918, 0.797, 0.682]
S6	0.223	[0.941, 0.738, 0.603, 0.682]	S7	0.25	[0.892, 0.623, 0.453, 0.682]

We can observe from the results on (Table 4) and (Table 5) that the ranking given by $AFDetS$ is more interesting than the one given by $AFDingS$ even if we vary ε and λ the top-1 is always better according to $AFDetS$. The service S4 (Table 5)is the top-1 according to $AFDing$ while it does not belong to the top-5 according to $AFDetS$ because of its bad first criterion value. We can say that $AFDetS$ favors services with good value in all parameters and discards services with worst values in some QoS parameters even if the others are good.

b-Varying *d* and *n*: We present below two scenarios by varying d from 7 to 9 on a set of 92 services belonging to the class search. We fixed $\varepsilon = -0.1$ and $\lambda = -0.2$. The result of the top-5 services provided by $AFDtingS$ and $AFDetS$ approach are shown in (Table6) and (Table7).

Table 6. Top-5 Services($AFDingS()$ Vs. $AFDetS()$) with $d = 7$

	Si	$Score$	$Qos(q_1, q_2, q_3 q_4, q_5, q_6, q_7)$
AFDingS	S70	0.409	[0.183, 0.904, 0.618, 0.964, 0.767, 1 , 0.815]
	S30	0.388	[0.164, 0.904, 1 , 0.964, 0.767, 1 , 0.815]
	S24	0.385	[0.005, 1 , 0.829, 1 , 0.767, 1 , 0.667]
	S72	0.381	[0.474, 0.795, 0.260, 0.807, 0.767, 0.667, 0.815]
	S16	0.365	[0.003, 1 , 0.419, 1 , 1 , 0.667, 0.111]
AFDetS	S30	0.005	[0.164, 0.904, 1 , 0.964, 0.767, 1 , 0.815]
	S52	0.006	[0.016, 0.819, 0.955, 0.94, 0.767, 1 , 0.667]
	S24	0.006	[0.005, 1 , 0.829, 1 , 0.767, 1 , 0.667]
	S70	0.008	[0.183, 0.904, 0.618, 0.964, 0.767, 1 , 0.815]
	S45	0.022	[0.042, 0.831, 0.382, 0.940, 0.767, 1 , 0.667]

From (Table6), we can observe that the ranking given by $AFDetS$ is more interesting than the one given by $AFDingS$. The top-1($AFDetS$) is the service S30. This latter has better value than the top-1($AFDingS$) on q_3. Moreover, service S30 is close to service S7 on q_1 parameter. We can remark that the service S16 is included into top-5($AFDingS$) while it does not belong to the

Table 7. Top-5 Services($AFDingS()$ Vs. $AFDetS()$) with $d = 9$

	Si	$Score$	$Qos(q_1, q_2, q_3 q_4, q_5, q_6, q_7, q_8, q_9)$
AFDingS	S24	0.397	[0.050, 1 , 0.829, 1 , 0.767, 1 , 0.667, 0.004, 0.958]
	S16	0.366	[0.003, 1 , 0.419, 1 , 1 , 0.667, 0.111, 0.030, 0.358]
	S60	0.344	[0.016, 0.988, 0.955, 1 , 0.333, 1 , 0.259, 0.008, 0.337]
	S55	0.328	[0.179, 0.916, 0.244, 0.976, 0.767, 1 , 0.815, 0.066, 0.800]
	S70	0.318	[0.183, 0.904, 0.618, 0.964, 0.767, 1 , 0.815, 0 , 0.021]
AFDetS	S24	0.006	[0.050, 1 , 0.829, 1 , 0.767, 1 , 0.667, 0.004, 0.958]
	S45	0.018	[0.042, 0.831, 0.382, 0.940, 0.767, 1 , 0.667, 0.030, 0.937]
	S55	0.018	[0.179, 0.916, 0.244, 0.976, 0.767, 1 , 0.815, 0.066, 0.800]
	S52	0.024	[0.016, 0.819, 0.955, 0.940, 0.767, 1 , 0.667, 0.017, 0.105]
	S30	0.027	[0.064, 0.904, 1 , 0.964, 0.767, 1 , 0.815, 0.092, 0.053]

top-5($AFDetS$) because of its bad values on q_3 and q_7. In fact,it is replaced by service S45 witch has a good compromise between its QoS parameters.

Let us consider now the ranking with $d = 9$ (Table7). The two ranking methods have the same top-1 (service S24). However, the other services given by $AFDetS$ are different from those provided by $AFDingS$. The service S16 and the service S70 witch belong to (top-5($AFDingS$)) are discarded by $AFDetS$ from the top-5 because they contains some bad values (close /or equal to 0) on some Qos criteria. This two services are replaced by respectively the service S45 and the service S30 by the $AFDetS$ approach, we can remark that these two services present a good compromise between their QoS parameters.

5 conclusion

In this paper, we have presented an approach for ranking QoS-based-Web services. We have presented a fuzzification of the Pareto-dominance and introduced the concept $AFDetS$ which associates a score to a service according to the Fuzzy dominated relation. We demonstrate that the fuzzy dominated concept can offer an alternative to compare services when they are non comparable with pareto dominance. Experimental results show that the proposed approach is effective in comparison with the Fuzzy Dominating ranking. For future work, we can use this concept for the web service composition.

References

1. Chan, C.Y., et al., On high dimensional skylines. Advances in Database Technology-EDBT 2006, 2006: p. 478-495
2. Lee, J., G. You, and S. Hwang, Personalized top-k skyline queries in high-dimensional space. Information Systems, 2009. 34(1): p. 45-61.
3. Kossmann, D., F. Ramsak, and S. Rost, Shooting stars in the sky: an online algorithm for skyline queries, in Proceedings of the 28th international conference on Very Large Data Bases2002, VLDB Endowment: Hong Kong, China. p. 275-286.

4. Papadias, D., et al., Progressive skyline computation in database systems. ACM Transactions on Database Systems (TODS), 2005. 30(1): p. 41-82.
5. K. Benouaret, D. Benslimane, and A. HadjAli, On the use of fuzzy dominance for computing service skyline based on qos, in ICWS, 2011, pp. 540547
6. K. Benouaret, D. Benslimane, and A. Hadjali. A fuzzy framework for selecting top-k Web services compositions. Applied Computing Review, 2011.
7. L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas,J. Kalagnanam, and H. Chang, Qos-aware middleware for web services composition, IEEE Trans. Software Eng.,vol. 30, no. 5, pp. 311327, 2004..
8. D. Ardagna and B. Pernici, Adaptive service composition in flexible processes, IEEE Trans. Software Eng., vol. 33, no. 6,pp. 369384, 2007.
9. T. Yu, Y. Zhang, and K.-J. Lin, Efficient algorithms for web services selection with end-to-end qos constraints, TWEB, vol. 1, no. 1, 2007.
10. X. Wang, T. Vitvar, M. Kerrigan, and I. Toma, A qos-aware selection model for semantic web services, in ICSOC, 2006, pp. 390401.
11. Borzsonyi, S., D. Kossmann, and K. Stocker, The Skyline Operator, in Proceedings of the 17th International Conference on Data Engineering2001, IEEE Computer Society. p. 421-430.
12. J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, Skyline with Presorting, Proc. 19th IEEE Intl Conf. Data Eng. (ICDE), pp. 717- 816, 2003.
13. I. Bartolini, P. Ciaccia, and M. Patella, Efficient Sort-Based Skyline Evaluation, ACM Trans. Database Systems, vol. 33, no. 4, pp. 1-45, 2008.
14. K. C. K. Lee, B. Zheng, H. Li, and W.-C. Lee, Approaching the skyline in z order, in VLDB, 2007, pp. 279290.
15. Chan, C. Y., et al., Finding k-dominant skylines in high dimensional space. 2006: p. 503-514.
16. Vlachou, A. and M. Vazirgiannis, Ranking the sky: Discovering the importance of skyline points through subspace dominance relationships. Data and Knowledge Engineering, 2010. 69(9): p. 943- 964.
17. Lin, X., et al., Selecting stars: The k most representative skyline operator. 2007: p. 86-95.
18. M. Alrifai, D. Skoutas, and T. Risse, Selecting skyline services for qos-based web service composition, in WWW, 2010, pp. 1120.
19. D. Skoutas, D. Sacharidis, A. Simitsis, and T. Sellis. Ranking and clustering web services using multi-criteria dominance relationships. In IEEE Trans. on Services Computing , 2010.
20. M. Koppen and R. Vicente Garcia. A fuzzy scheme for the ranking of multivariate data and its application. In Proceedings of the 2004 Annual Meeting of the NAFIPS (CD-ROM), pages 140145, Banff, Alberta, Canada, 2004. NAFIPS
21. M. Koppen, Raul Vicente Garcia, and Bertram Nickolay. Fuzzy-pareto-dominance and its application in evolutionary multi-objective optimization.In Evolutionary Multi-Criterion Optimization, Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings, LNCS 3410, pages 399412. Springer Berlin / Heidelberg, 2005.
22. M. K oppen and C. Veenhuis, Multi-objective particle swarm optimization by fuzzy-Pareto-dominance meta-heuristic, Int. J. Hybrid Intell. Syst. , vol. 3, no. 4, pp. 179186, 20067
23. E. Al-Masri and Q. H. Mahmoud, Investigating web services on the world wide web, in WWW, 2008, pp. 795804.