A Multi-Objective Evolutionary Algorithm to optimize the Dynamic Composition of Semantic Web Services

Brahim Batouche[a] Yannick Naudet[a] Isabelle Jars[a] Thibaud Latour[a] Frédéric Guinand[b]

[a] Public Research Center Henri Tudor, Luxembourg

[b] University of Le Havre, France

Abstract

The optimization of the dynamic composition of web service is helpful to answer the user-query in the dynamic environment. The answers are the best service composites, which are robust, i.e., stay best even if the events intervene during the invocation of the composite service. To keep the best compositions of services during the events, we consider them in the resolution process, i.e., we answer the questions: *How find automatically the set of service composite? How optimizer the dynamic composition of web services?* How find the alternatives which solve the event and keep the best solution? This paper answers these questions and defines the problem of dynamic composition optimization, then proposes a resolution method. The Multi-Objective Evolutionary Algorithm (MOEA) are considered as optimization methods.

1 Introduction

The composition of web services is an efficient way to achieve a user goal by creating an ad hoc service with a selection and sequencing of available elementary services. The composite service obtained as a result of the query bears some dependencies with the nature, characteristics and availability of elementary (atomic or composite services) services. This requires a highly adaptive service composition system enable to dynamically create, invoke, and maintain the resulting composite service.

In most cases, many candidate composite services are possible as an answer to a given query. The set of candidates depends on the availability of a large number of existing elementary services or composite services resulting from previous compositions. The existing services are characterized by different functional and non-functional parameters, such as, price, response time, availability, reliability … etc. These parameters are used to match user constraints and objectives during the composition process. Considering non-functional aspects of the elementary services extends the scope of the target constraints leding the service composition to consider not only the problem of finding one composition that fulfill the global user's functional requirement, but

also of finding the best selection of the most suitable ones in non-functional terms. Eventually, finding the most suitable dynamic composition of web services is equivalent to the optimization of the dynamic composition.

The problem of optimizing the dynamic composition of web services addresses multiple problems like, e.g., heterogeneity of service descriptions, request interpretation and decomposition, automatic service discovery, composition modeling, and composition optimization. Adding semantics to web service descriptions is one possibility to enable or to facilitate (at least theoretically) automatic service discovery and composition, by adding semantic interoperability potential capacity into these descriptions. From this perspective, semantic web service languages like, e.g., OWL-S D. Martin, M. Burstein, J. Hobbs, et al. OWL-S: Semantic Markup for Web Services. W3C Member Submission, 22, 2004. provides elements to tackle service composition problems.

The purpose of this paper is to propose a method to optimally compose web services in a dynamic environment, in response to a user query expressed in both functional and non-functional objectives. The method ensures that the result remains optimum even if the system undergoes dynamic changes in terms of data and service availability over execution time. In the next section, we shall describe how to find automatically the set of composite services, and how to use of a Multi Objective Evolutionary algorithm (MOEA) enables the optimization of the dynamic service composition. The main feature of the MOEA is to provide automatically an optimum set of global solutions of the multi-objective and multi-constraint requests, by providing the set of compositions fulfilling individual objectives in various ways, leaving the final choice to the user.

The remained of this paper is structured as follows. In section 2 we present related works, the section 3 presents the problem formalization, the section 4 presents the resolution approach, the section 5 presents the events processing, and then we conclude in section 6.

2 Related Works

In a previous paper Batouche, B., Naudet, Y., & Guinand F., Semantic Web Services Composition Optimized by Multi-Objective Evolutionary Algorithms, ICIW: Internet and Web Applications and Services, 2010., we have presented how to use the MOEA to optimize static compositions of web services satisfying the static service composition with respect to the user non-functional objective, together with composite services that also satisfy the user functional objectives. In this contribution, we focus on the dynamic aspect of the composition, *i.e.*, when atomic service conditions and status (availability) change over time at invocation time. There is quite little literature describing the use of Genetic Algorithms in the context of web service composition. The existing related works focus in either one of the follows resolution points: the automatic composition of web services and the optimization of the web service composition. These resolution points can combine, i.e., made together, in the case where the optimization problem is mono-objectives Mohammad Alrifai and Thomas Risse. Combining global optimization with local selection for efficient qos-aware service composition. In 18th International World Wide

Web Conference (WWW2009), April 2009.Sorin M. Iacob, ao Paulo A. Almeida, Jo and Maria E. Iacob. Optimized dynamic semantic composition of services. In SAC '08 : Proceedings of the 2008 ACM symposium on Applied computing, pages 2286_2292, New York, NY, USA, 2008. ACM..

Most of papers dealing with the automatic composition of web services Matthias K and Andreas G, Semantic web service composition planning with OWLS-XPlan, 2005. Silva, E., & Sinderen, M. V. An Algorithm for Automatic Service Composition, ICSOFT: International Workshop on Architectures. 2007. Sorin M. Iacob, ao Paulo A. Almeida, Jo and Maria E. Iacob. Optimized dynamic semantic composition of services. In SAC '08 : Proceedings of the 2008 ACM symposium on Applied computing, pages 2286_2292, New York, NY, USA, 2008. ACM.Oh, S., On, B., Larson, E. J., & Lee, D. BF*: Web Services Discovery and Composition as Graph Search Problem, e-Technology, e-Commerce, and e-Services, IEEE International Conference on, 6-8, 2005., Matthias K and Andreas G, Semantic web service composition planning with OWLS-XPlan, 2005. presents an architecture of automatic web service composition, the architecture allows the fast and flexible composition of OWL-S service, Matthias K and Andreas G, Semantic web service composition planning with OWLS-XPlan, 2005. gives the interesting ideas to plan the composite service and to automatic the service invocation, but the result provided cannot be the search space of the optimization, because the search space contains the information about the input/output instance of the service, such as, the depart city of the transport, price, …etc. therefore, we must considered these information in the automatic composition.

The automatic composition algorithm Silva, E., & Sinderen, M. V. An Algorithm for Automatic Service Composition, ICSOFT: International Workshop on Architectures. 2007. provides the composition graph which answers a query. The query presented by the input/output (I/O) of the requested service. The algorithm compares this I/O with the I/O of the existing services. The algorithm selects the service which matched with query I/O, otherwise, cheeks the service which matched with query output. If the service matches then selects it and creates the node which presents its functionality. Then looks for the service where its input matches with a query input and its output matches with the selected service input, and so on. The algorithm ends if found the service which matches with the query I/O or all nodes of graph are covered, i.e., a set of sub-service are covered. The composite graph provided does not present the search space of the optimization. Also, the result does not give the link to invocate the service.

Oh, S., On, B., Larson, E. J., & Lee, D. BF*: Web Services Discovery and Composition as Graph Search Problem, e-Technology, e-Commerce, and e-

Services, IEEE International Conference on, 6-8, 2005. uses the flooding algorithm to automatic the web service composition, the algorithm provides the sequences of the functionality needed to answer a query, but the answer does not give the link to invocate the service. Also we cannot consider the answer as the search space of the optimization.

To deal the optimization of the dynamic composition, we define the optimization problem and simulate it to the classic optimization problem, and then we study this simulated problem. We detail this point in the section 4.2.

3 Problem Formalizations

We define those “complex queries” as requests for which no single service answer exist and that necessitate some composition. A classical example of such a query is: “I want to travel from Paris to London in air between 5 and August 7 and returning 10 days later, I want to stay in 3 * hotel and to rent a car, the whole at best price and service quality”. Additionally, we can add “total price should not exceed 3000 EUR and service quality should be at least satisfactory”. The whole request contains distinct parts that can be considered as sub-queries, as well as multiple constraints and objectives. This example query, which serves as our experimental case study, requires together services for travelling, hotel booking and car rental. Each requested service is characterized by the functional parameter: input, output, precondition and effect. It contains also different constraints: starting and arrival locations (Paris-London), transportation type (by air), date of departure (between 5 and August 7), length of stay (10 days), and the quality level of the hotel (3 *). It also contains price and quality of service objectives, the latter being reduced to reputation rating. The last requirements we have specified concerning the whole trip are also constraints, respectively on total price and quality of service. What is interesting is that there were already objectives defined for these parameters.

1.1 Query Formalization

The complex query requires many functionality, Q={Fct},where Fct is a functionality requested. For example, our query requires three functionalities: transport, booking hotel and rent a car. So, each functionality has the input and output.

We formalize the query by the quadruple: Q = <I, O, C, B>, where: the set of input is I=(i1, …,il)T, the set of output is O=(o1, …, om)T, the set of constraints is C=c1, …, cnTand the set of objectives is B=b1, …, bkT. Where I and O contain the URI of functional parameter, but C and B contains the URI functional or non functional parameter. The functional parameters of service are described in the domain ontology but the non-functional parameters of service are described in a specific ontology, such as, QoS ontology K.Kritikos and D.Plexousakis. Semantic QoS Metric Matching. ECOWS. 2006.Gustavo F. Tondello and Frank Siqueira. The qos-mo ontology for semantic qos modeling. In SAC '08 : Proceedings of the 2008 ACM symposium on Applied computing, pages 2336_2340, New York, NY, USA, 2008. ACM..

Generally, the constraints are imposed to some properties of the composition or to parts of it. Differently, objectives correspond to the maximization or minimization of such a property's value, e.g., minimizing a price; maximizing reliability, reputation or availability. Some properties like, e.g., a price can be the subject of both a constraint and an objective. For others like, e.g., reputation, their use in an objective only is more evident. The service answered a query is made according to these constraints and objectives, and optimization is performed accordingly. This mean, the constraints and the objectives influences the size of search space.

Finally, both constraints and objectives can be of three kinds: (a) related to a service (e.g. targeting the service's price); (b) related to a data manipulated by a service (e.g. the price of the hotel should not exceed X or/and should be minimized); and (c) related to the composition (e.g. the total trip cost should not exceed X or/and should be minimized). The latter is taken as a penalty during the optimization phase.

3.1 Answers Set Formalization (or Compositions answered Formalization)

1.2 ***The set of answers are illustrated by the multi layer composition graph, which represent a search space for the optimization method. We formalize the composition graph as a unidirectional multi-layered acyclic graph,G= <N,V>, where N is the multilayered set of service/data nodes, we add the begin/end node of the graph, V is the set of arcs. The arc weight corresponds to an objective value of destination node. Always the objectives correspond to a non-functional parameter of the service.***

We distinguish three types of nodes, informative service, active service and data. The informative service (IS) provides the information (data) and after its invocations does not change the data base of the service. The actives service (AS) provides the action and after its invocations changes the data base of the service. In our resolution the IS will be invocated during the search for the composition answering the query but the AS will be invocated after the optimization also after the user choose between the best composition.

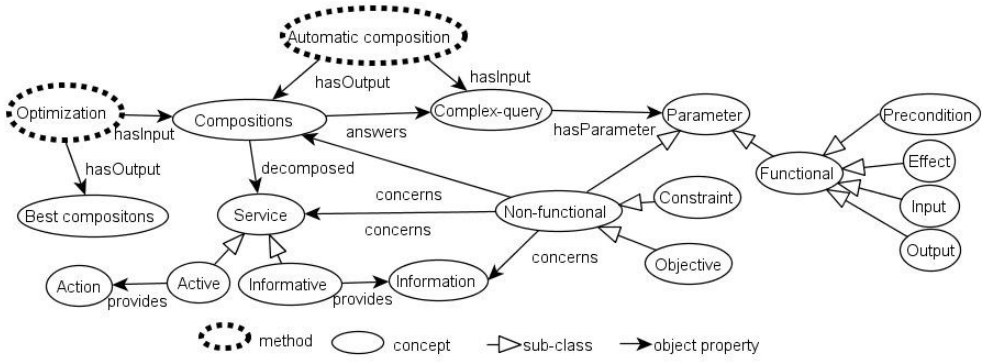

Figure 1. Problem description

Finally, the Figure 1 describes our problem and its concepts.

4 Resolution Approach

Our resolution approach are decomposed into two steps: (1) find a set of answers, which represents the search space of the composition (2) select the best composition by the MOEA optimization methods.

The query elements are used in a specific resolution steps, the Input/output of the query are used to select the services, then the constraints related to a service and a data are used to filter the selected services and data. The composition-related constraints and all type of objectives are used during the optimization.

4.1 Algorithm Finding the Search Space of Composition

The Figure 2 illustrates our algorithm of finding the answers set, the algorithm extends the flooding algorithm and deals progressively the query and provides the search space of the composition. We begin by defining the terminology used:

-The current layer "lk" corresponds to a set of current nodes, initially lk contains the begin node. The step of the algorithm corresponds to full covers of lk.

-The temporary layer corresponds to a set of nodes which follow a current node and not precede the end-node. The nodes which precede the end node will be regrouped in the end layer.

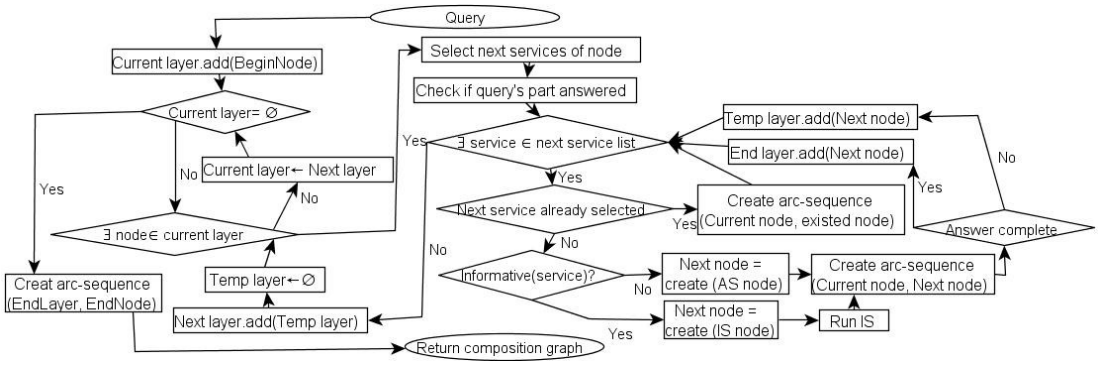

Figure 2. Find a set of answer service composite diagram

The Figure 2 resumes our algorithm, where IS is informative service, AS is active service.

The algorithm begins by covers the current layer and for each current node selects the next services according to the functionality realized. Initially, the current node "ni" is the begin node of the graph, which its output presents the input-query.

Then select the services which its input matches with a current node output. If the current node outputs match with one of the query output, i.e., a part of functionality requested is terminated. Therefore, select the service which its input matches with a next input of the query.

The selected service is represented in the graph by the next node "ni+1".

If Mni.output, Q.outputj≤ε then

if MQ.inputj+1, ni+1.input≤ε then select(ni+1).

else if Mni.output, ni+1.input≤ε then select(ni+1).

Where, M the matchmaker functions, ε the matching threshold.

If the selected service is already selected then the node which corresponds is already created and added in the set of nodes N. Consequently, the next node will be selected from N, otherwise, the next node will be created according to the service informative or active, and then create arc-sequence between the current node and the next node. The service informative will be running, if provide the data then create arc sequence between the next node and the data nodes, so replace the next node by the data nodes.

The next node is affected to end layer if all requested functionalities are realized, otherwise, affected to the temporary layer, then to the next layer.

If all nodes of current layer are covered then the next layer replaces the current layer and so on. If the next layer is empty then the algorithm is terminated. Finally, the nodes set are decomposed into three types: informative service, data and active service.

Each path in the composite graph corresponds to a composition of services, which have an answer level to a query. The evaluation of this level depends to the matching level between a query and the selected service, formally: fperfx=1ni=0 nMLi , where x is the composition of services (path), n is the number of the service in the composition and MLi is the matching level between output-input of the sub-service or the input-input of query and the first services. The function fperfx gives the performance level of the service composite, i.e., the evaluation of the functional parameter.

4.2 Selection of the Best Compositions

The selection of the best composition means the optimization of the composition. To choice the optimization method we bases into the definition of the optimization problem.

The problem of web services composition optimization has already been defined in terms of the search for a shortest path in a graph X.X. Wang and H. Wang, "Web Services Selection and Composition based on the Routing Algorithm," 10th IEEE International Enterprise Distributed Object Computing Conference Workshops, pages 57_66, 2006., which is a polynomial time mono-objective problem. However, in a more general context, a composition of services has to fulfill multiple objectives such as cost minimization, time response minimization, or consumer satisfaction maximization. These objectives are normally not related. When there are some relationships between them, the objectives can be combined into a single one (see, e.g., approach proposed in L. Jian-hua, C. Song-qiao, L.yong-jun, and L.gui-lin, "Application of genetic algorithm to QoS-aware Web Services composition," Industrial Electronics Applications, 2008. ICIEA, vol. 4582569, 2008.).

We address to the multi-objective optimization problem occurring when there is no relation between target objectives. In that case, this leads to search a trade-off between non related objectives, since a set of solutions (distributed in the multi-dimensional solution space) is obtained. We define the optimization problem of the composition as a Multi-objective Shortest Path Problem (MSPP), which it is solved by the MOEA Qiong Fan Fangguo He, Huan Qi. An evolutionary algorithm for the multi-objective shortest path problem. International Conference on Intelligent Systems and Knowledge Engineering , Chengdu, China, Oct. 15-16 2007.Christian Horoba. Analysis of a simple evolutionary algorithm for the multi-objective shortest path problem. In FOGA '09 : Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms, pages 113_120, New York, NY, USA, 2009. ACM. .

The tradeoff function for evaluating a service composition candidate (sc) is defined by the formula: fsci=fobj1, fobj2, …, fobjk, where k is the number of objectives and fobji is the evaluation function for objective i .

When the composition is dynamic (i.e., available services can change during composition search), the problem may be defined as a problem of multicast multi-objectives shortest path L.S. Randaccio and L. Atzori, "Group multicast routing problem : A genetic algorithms based approach," Comput. Netw, vol. 51,2007., we solve the problem by the approached optimization method such as MOEA.

There are different approaches solving multi-objective optimization problems: combine the objectives, optimizing every objective in parallel or look for a compromise between the objectives. The problem addressed does not allow the objectives combination, so the parallel resolution provides solutions may be best for one objective while being very poor for others.

Our way, look for a compromise between the objectives, considering all objectives expressing solutions in terms of all these objectives. The solutions illustrate in the multidimensional solution space and present a curve called a Pareto front. In our case the front represents all services compositions that are not dominated. Given a solution (a service composition sci) belonging to the front, this means that any other service composition can have better values for the objectives for at most *k-1* objectives and is always worse for the last one. We define this Pareto front as SC*={sci*}, with sci*=[obj1*, obj2*,…, objk*] a non-dominated solution. Non-domination of solutions sci* in the front verified the condition: ∀ sc*∈SC*, ∀ i ∈1,2,…,k, ∀ sc ∈CSS, fobjisc*≻fobjisc, where ≻ is the domination operator, i.e. (≤) in case of an objective to be minimized and ≥ in case of a maximization. The best MOEA are SPEA2 E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Computer Engineering, 2001, pp. 1-21. and NSGA2 K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II.", Evolutionary Computation, IEEE Transactions on, 6(2):18-197, 2002. algorithms. While both use an elitist strategy for keeping best solutions, selection of non-dominated solutions and diversification is handled differently, thus bringing different results.

Genetic algorithm (GA) are among well-know evolutionary approach, the crucial point in problem solving with GA is the coding and the evaluation of candidate solutions (called individuals) as chromosomes.

1.2.1 Multi-Caste Individual Coding and Initial Population

The individual coding is based into the optimization problem definition, but there are the works Daniela Barreiro Claro, Patrick Albers, and Jin-Kao Hao. Web services composition.Yves Vanrompay, Peter Rigole, and Yolande Berbers. Genetic algorithm-based optimization of service composition and deployment. In SIPE '08 : Proceedings of the 3rd international workshop on Services integration.Jia Yu, Michael Kirley, and Rajkumar Buyya. Multi-objective planning for workflow execution on grids. In GRID '07 : Proceedings

of the 8th IEEE/ACM International Conference on Grid Computing, pages 10-17, Washington, DC, USA, 2007. IEEE Computer Society.Wei-Chun Chang and Ching-Seh Wu and Chun Chang. Optimizing Dynamic Web Service Component Composition by Using Evolutionary Algorithms. Web Intelligence. 2005.which coding the individual of GA without the definition. According our definition, the multicast individual coding considers all possible fail of service during the invocation. For each fail proposes an alternative sub-composition to complete the initial composition. For this, we group all possible paths in the matrix, where each line present a path, e.g., the first line presents the path S1, S3, S5, S9 .

The figure 3 illustrates the multi caste coding, which presents all nodes in the composite graph, which have not precede the end node, in our example, the nodes formed the individual are, Begin, S1, …, S8, the service which precede the end node are not presented in the individual because are the last in the composition, i.e., the multi-caste member, in our example are S9, S10, S11.

The individual element value is the index of the matrix line, e.g., the element value of Begin have the index [1, 6] and corresponds to the initial composition CS*, S1 can have the index value [1, 9], S2 can have the index value [10, 12]… and S8 can have the index value [24, 24]. The multi-cast individual presentation can translate to another presentation of approached optimization method, such as the ant colony... etc.

The initial population is presented by the paths of the composition graph, which are chosen randomly. Then the evolutionary operation, crosser and mutation, will be chosen according to the coding, i.e., respect the node position in the individual.

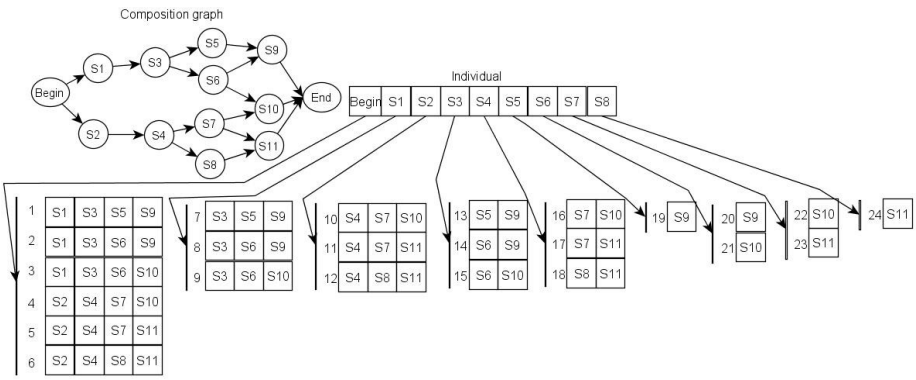

Figure 3. Multicaste individual coding

4.2.2 Multicast Evaluation (or Normalization of the Evaluation)

The multicast evaluation considers the fail probability γi of the service “Si” or the data “Di”, the service fail probability depends to the reliability or the availability of the service, which corresponds to a non functional parameter of the service. The data fail probability is constant and depends to the situation, for example if there are the perturbations in the transport then the information reliability are weak otherwise is important.

The importance level of the service θSi depends on its position in the composition, suppose that a service composite with the sequence: S1, S3, S5, S9, which there γi are respectively: 0.6, 0.5, 0.1, 0.9. According the sequence, the service S3 is more important than the service S5, because the invocation of S5 depends to the invocation of S3. If S3 fail then S5 have not the importance in the composition, for this, the importance costs of the service are respectively: θS1=0.6+ 0.5+ 0.1+ 0.9β , θS3= 0.5+ 0.1+ 0.9β , θS5= 0.1+ 0.9β and θS9= 0.9β. Generally : θSi=θDi=irγiβ, where r is the number of the services in the composition and β=γi= 2.1.

We can have tow value of θSi for the service Si, e.g., the sequences S1, S3, S6, S10 and S2, S4, S7, S10 are different but the both contain the service S10. In this case there are two θS10. Therefore, the value of θS10are calculated according to a path of the composition. The important cost will be multiplicities to the objective value, therefore the weight of the arc will be θSiObj1,…,ObjkSiT.

4.2.1 The Fitness

During the evolution phase the individuals (*ind*) are selected according to the value obtained by the evaluation function. The evaluation function formula depends to the composition structure and the evaluated objectives. We adopt the following functions for the sequence structure, where the objectives are, price and reputation rating (*rr*): frr(x)=1ni=1nθSi*rri and fPrice(x)=i=1, j=1n, m(θSi*pri+ θDj*prj), where: n and m are respectively the number of services and data nodes in the composition and x is the composition of services.

To consider the dynamic composition, i.e., assure that the best alternatives composition for the events. We optimize the objectives value of each possible alternative, therefore, the evaluation function will be: Frr(ind)=1ni=1nθSi*frr(Si) and FPrice(ind)=i=1, j=1n, m (θSi*fPrice(Si) +θDi*fPrice(Dj)), where Si,Dj∈ind. We use the functions Frr and FPrice during the dynamic composition optimization and the functions frrand fpriceto give the user the real value if there is not events.

Constraints related to the composition are taken into account by introducing penalties in the objectives evaluation functions. Hence, an individual who does not respects a constraint gets its fitting values (regarding objectives) lowered, so that it will get a bad evaluation with respect to other individuals during optimization. In the particular case where both a constraint and an objective concern a same parameter, a

penalty is put on the concerned evaluation function. In a situation where a constraint and an objective concern different parameters, the evaluation functions for all objectives are also penalized, in a lower manner. For our example, The penalized evaluation functions for price and customer satisfaction are: fPrice(x)= Δ+i=1, j=1n, m(θSi*pri+θSj*prj), and frr(x)=1ni=1nθSi*rri-Δ, where Δ>0 is a penalty rate and x is the path which corresponds to the index of Begin element. These penalized evaluation function do not consider the dynamic composition.

2 Events Processing

The Pareto front provided by the method, such as NSGA2 and SPEA2, presents a set of the best service compositions, where the user will be choose one among these.

The service composite chosen cs* represents the individual, e.g., if the individual sequence is “3-9-11-13-16-21-23-24”, then the initial sequence of the composition is S1, S3, S6, S10 . If the service S6 fail then the sub-sequence alternative of the composition correspond to the index of predecessor, i.e., S3, the index equal 13, which correspond to the best sub-sequence S5, S9, therefore the final sequence of the composition is S1, S3, S5, S9 .

If the service S3 fail then consider the index of S1, which equal 9 and begin by the fail service S3. To resolve this problem we consider yet the index of S1 predecessor, which equal 3 and begin by S1, S3. In this case, we propose to user the neighbors solution in the Pareto front.

```
If ((Sifail ){

    If(Si=Begin)

Propose neighbors;

    Else {Select sequence Si-1;

            While (Si ∈sequenceSi-1∧ Si≠Begin){

              i←i-1;

              Select sequence Si-1 ; }

          If(Si=Begin)

Propose neighbors; }

}
```

To propose the solutions neighbor of the chosen solution cs*, we define the neighbor according to the tolerance rate λ, where λ∈0, 1. The neighbors,

service composites cs, are defined if satisfy the condition: $cs \in \psi$, $csob1 \in \alpha 1 \wedge csob2 \in \alpha 2 \wedge \ldots \wedge csobk \in \alpha k$. Where ψ is the Pareto front, $\alpha obi = [csobji^* - \varepsilon i, csobji^* + \varepsilon i]$ and $\varepsilon i = maxobji - minobji^* \lambda$, if the set of neighbors is empty we increase λ (see figure 3).

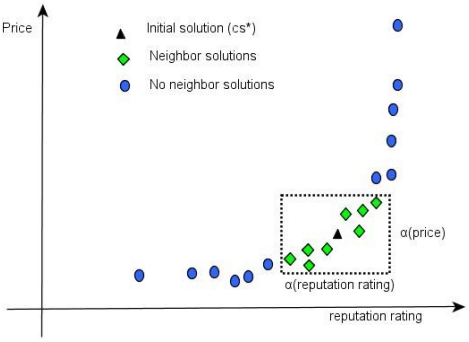

Figure 3. Repairing composition with Pareto front

3 Conclusion

We presented how optimize the dynamic composition of web services by the MOEA, which the MOEA method provides a set of robust and best compositions of services. The compositions of services meet the request criteria, which are distributed in the objectives space. This leaves the final choice to the end-user, who can choose a solution according to the priorities he implicitly gives to objectives, knowing the available solutions.

The future work will be generalized the method to consider different types of composition structure, such as, parallel, conditional structure, etc. Also considers the precondition and effect in the selection of the service, then apply the method in the project WiSafeCar[1].

5 References

D. Martin, M. Burstein, J. Hobbs, et al. OWL-S: Semantic Markup for Web Services. *W3C Member Submission*, 22, 2004.

Batouche, B., Naudet, Y., & Guinand F., Semantic Web Services Composition Optimized by Multi-Objective Evolutionary Algorithms, *ICIW: Internet and Web Applications and Services*, 2010.

Mohammad Alrifai and Thomas Risse. Combining global optimization with local selection for efficient qos-aware service composition. *In 18th International World Wide Web Conference (WWW2009)*, April 2009.

1 http://wisafecar.gforge.uni.lu/

Sorin M. Iacob, ao Paulo A. Almeida, Jo and Maria E. Iacob. Optimized dynamic semantic composition of services. *In SAC '08 : Proceedings of the 2008 ACM symposium on Applied computing*, pages 2286_2292, New York, NY, USA, 2008. ACM.

[1] Matthias K and Andreas G, Semantic web service composition planning with OWLS-XPlan, 2005.

[2] Silva, E., & Sinderen, M. V. An Algorithm for Automatic Service Composition, *ICSOFT: International Workshop on Architectures.* 2007.

[3] Oh, S., On, B., Larson, E. J., & Lee, D. BF*: Web Services Discovery and Composition as Graph Search Problem, *e-Technology, e-Commerce, and e-Services, IEEE International Conference on*, 6-8, 2005.

K.Kritikos and D.Plexousakis. Semantic QoS Metric Matching. *ECOWS*. 2006.K. Kritikos and D. Plexousakis. Semantic qos metric matching. *In ECOWS: Proceedings of the European Conference on Web Services*, 2006.

Gustavo F. Tondello and Frank Siqueira. The qos-mo ontology for semantic qos modeling. *In SAC '08 : Proceedings of the 2008 ACM symposium on Applied computing*, pages 2336_2340, New York, NY, USA, 2008. ACM.

[4] X.X. Wang and H. Wang, "Web Services Selection and Composition based on the Routing Algorithm," *10th IEEE International Enterprise Distributed Object Computing Conference Workshops, pages 57_66*, 2006.

[5] L. Jian-hua, C. Song-qiao, L.yong-jun, and L.gui-lin, "Application of genetic algorithm to QoS-aware Web Services composition," *Industrial Electronics Applications, 2008. ICIEA*, vol. 4582569, 2008.

Qiong Fan Fangguo He, Huan Qi. An evolutionary algorithm for the multi-objective shortest path problem. *International Conference on Intelligent Systems and Knowledge Engineering* , Chengdu, China, Oct. 15-16 2007.

Christian Horoba. Analysis of a simple evolutionary algorithm for the multi-objective shortest path problem. *In FOGA '09 : Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 113_120, New York, NY, USA, 2009. ACM.

L.S. Randaccio and L. Atzori, "Group multicast routing problem : A genetic algorithms based approach," *Comput. Netw*, vol. 51,2007.

E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," *Computer Engineering*, 2001, pp. 1-21.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II.", *Evolutionary Computation*, *IEEE Transactions on*, 6(2):18-197, 2002.

Daniela Barreiro Claro, Patrick Albers, and Jin-Kao Hao. Web services composition. *In Semantic Web Services, Processes and Applications*, p 195-225. 2006.

Yves Vanrompay, Peter Rigole, and Yolande Berbers. Genetic algorithm-based optimization of service composition and deployment. *In SIPE '08 : Proceedings of the 3rd international workshop on Services integration*. 2008.

Jia Yu, Michael Kirley, and Rajkumar Buyya. Multi-objective planning for workflow execution on grids. *In GRID '07 : Proceedings of the 8th IEEE/ACM International Conference on Grid Computing*, pages 10-17, Washington, DC, USA, 2007. IEEE Computer Society.

Wei-Chun Chang and Ching-Seh Wu and Chun Chang. Optimizing Dynamic Web Service Component Composition by Using Evolutionary Algorithms. *Web Intelligence*. 2005.