A Hybrid Evolutionary Multi-objective Optimization Algorithm for QoS-driven Service Selection Problem

Dalaijargal Purevsuren*,[1], Gang Cui[1], and Saif ur Rehman[1]

[1] School of Computer Science and Technology
Harbin Institute of Technology
**** corresponding author***

Abstract

The QoS-driven Service Selection (QSS) problem is a well-known NP-hard problem in the combinatorial optimization field. Although the QSS problem is naturally multi-objective optimization problem, most of the existing approaches solve the problem in single-objective optimization context. In the recent years, there have been some efforts to tackle the problem in multi-objective optimization context. In this paper, we propose a hybrid interactive Evolutionary Multi-objective Optimization (EMO) algorithm for solving the QSS problem in multi-objective context. The proposed algorithm hybridizes an interactive EMO algorithm with Path Relinking. The performance of the proposed algorithm is assessed for two and three objectives QSS problem sets. We show the performance advantage over other existing standard approaches such as NSGA-II. The comparative evaluations of results indicate that the proposed approach can converge to the preferred solution faster, and improve upon previously existing techniques up to 13.3% in terms of the requirement of user feedback.

Keywords: *QoS-driven Service Selection Problem, Evolutionary Computation, Multi-objective Optimization, Path Relinking.*

1. Introduction

The number of functionally similar services over the internet is growing continuously. Hence the role of QoS is getting more important in selection of appropriate concrete services to maximize the quality value for a composite service. This is a decision making problem, called the QoS-driven Service Selection (QSS) problem. It can be modeled as a combinatorial optimization problem that searches the optimal bindings between each abstract service and a set of the corresponding concrete services. It is well known that the problem is NP-hard. The problem should be considered in multi-objective context because QoS requirements have several properties such as cost, execution time, and reliability. Moreover, these objectives normally conflict with each other. It means that an improvement of one objective may demand some sacrifices in another. For example, minimizing cost and maximizing reliability are clearly conflicting. Therefore, there is no single optimal solution which can simultaneously optimize all these objectives unless decision makers' (DMs) preference information is available. Although the QSS problem involves multiple objectives by its nature, most approaches in the literature transform multiple objectives into single objective using Simple Additive Weighting (SAW) [1] in order to employ the methodologies of single objective optimization. SAW method is very sensitive to selecting weights, and needs much cognitive effort from DMs [2], [3]. A way to avoid the usage of SAW is the usage of Evolutionary Multi-objective Optimization (EMO) algorithms.

The EMO algorithms have demonstrated their efficiency and effectiveness in finding a well-distributed and well-converged approximation of the Pareto optimal front [10] for solving Multi-objective Optimization Problems. However, finding the whole Pareto Front eventually is not the final objective because most of decision makers want to find the most preferred point (MPP) instead of the whole Pareto Front. There are several recent works e.g., [4], [5] for finding the Most Preferred Solution by incorporating the decision maker preference information in an interactive manner, called interactive EMO. These interactive procedures require periodical interaction from the decision maker (DM) to get their preference information. The process getting the preference is called "DM calls". We propose that the interactive EMO algorithms can be usefully extended by hybridizing it with the Path Relinking technique.

Path Relinking (PR) is a metaheuristic optimization technique that generates new solutions by exploring trajectories connecting high-quality solutions. The basic hypothesis is that by exploring the trajectories between high-quality solutions, more high-quality solutions will be found. PR is actively studied by hybridizing with other heuristic methods as illustrated in the work of [6], [7], [8].

In this paper, we propose a hybrid algorithm by combining an interactive EMO algorithm with Path Relinking technique. The Path Relinking promotes the convergence of the interactive algorithm. As a result, the hybrid algorithm can converge to the preferred solution faster and require user feedback less frequently compared to previous proposals. The remainder of the paper is structured as follows. The second section gives preliminary. Section three details the proposed algorithm. The result of the experimentation is presented in section four. Finally, we conclude the paper and discuss some future work.

2. Preliminary

In this section, some background theories are briefly presented, and then the problem statement is described in detail.

2.1 Background

Definition (Utility function): A function $V: R^k \to R$ representing the preferences of the decision maker among the objective vectors is called a value (utility) function [9]. Assume that z and z* $\in$ Z be two different objective vectors. z* is preferred to z if and only if V(z*) >V(z).

Assume that there are m ($m \geq 2$) distinct points $z_1, z_2, \ldots, z_m$ and a quasi-concave, non-decreasing utility function V(x). Let the z_k is the least preferred one, i.e., $V(z_k) = \min_{i \in \{1,\ldots,m\}}\{V(z_i)\}$. Once the least preferred point is known, a cone can be constructed. The cone $C(z_1,\ldots,z_m; z_k)$ with vertex z_k is defined as follows:

$$C(z_1,\ldots,z_m; z_k) = \{z \mid z = z_k + \sum_{i=1; i\neq k}^{m} \mu_i (z_k - z_i), \mu_i \geq 0\}.$$

As shown in [11], for any point $y \in C(z_1,\ldots,z_m; z_k)$, $V(z_i) > V(z_k) \geq V(y)$ for i=1,…,m and $i \neq k$.

Definition (cone dominated): Every point $z \in C(z_1,\ldots,z_m; z_k)$ $z \neq z_k$, and any point z' dominated by z is called cone dominated.
The exact explanation of the above idea is presented in Theorem 1 of [11]. It states, "Assume a quasi-concave and nondecreasing function f(x) defined in a p-dimensional Euclidean space R^p. Consider distinct points $x_i \in R^p$, *i*=1,…,*m*, and any point $x^* \in R^p$ and assume that $f(x_k) < f(x_i), \mathrm{i} \neq \mathrm{k}$. Then, if $\varepsilon \geq 0$ in the following linear programming problem:

$$\text{Max} \quad \varepsilon,$$
$$\text{Subject to: } \sum_{i=1, i\neq k}^{m} \mu_i (x_k - x_i) - \varepsilon \geq x^* - x_k, \; \mu_i \geq 0,$$

it follows that $V(x_k) \geq V(x^*)$ ".

After applying cone dominance-based sorting to a set of solutions, there will be a trend which raise the most preferred solutions to the top of the set. Therefore, the cone dominance-based interactive procedure can be viewed as a utility function which defines the most preferred solution from a set of solutions.

Definition (Pareto and Cone-based utility function): Given a set of solutions, Pareto dominance-based sorting is applied on the set first. Then, Cone dominance-based sorting is applied to the non-dominated front of the set. The first solution in the ordered non-dominated set is returned as the most preferred solution. The procedure defining the most preferred solution from a set of solutions is called Pareto and Cone-based utility function.

Definition (Pareto and Cone-based sorting): Given a population of solutions, Pareto dominance-based sorting is applied on the population, and the non-dominated set of the population is obtained. Then, Cone dominance-based sorting is applied to the non-dominated set. The result of applying these two sorting is an order of the non-dominated set of a population. This procedure is called Pareto and Cone-based sorting.

2.2 Problem Statement

In SOA environment, a complex task can be decomposed into several abstract services, i.e., $T = \{S^1, S^2, \ldots, S^N\}$ where N is the total number of the decomposed abstract services and S_i denotes the *i*th abstract service of *T*, *i* = 1, 2, …, *N*. A set of concrete services (CS) are available for each abstract service, i.e., $T' = \{CS^1, CS^2, \ldots, CS^N\}$, where CS^i has the set of concrete services for i th abstract service and there are M_i concrete services available in CS^i, i.e., $CS^i = \{cs_1^i, cs_2^i, \ldots, cs_{M_i}^i\}$, where cs_j^i is the j th concrete service for i th abstract service. Each concrete service has three QoS parameters, i.e., execution time, invoking cost, and reliability. QSS problem is to select one concrete service from each corresponding set of concrete services (e.g., $CS = \{cs_1, cs_2, \ldots, cs_N\}$, where cs_i is the selected concrete service for abstract service S^i) under multi-objective (e.g., time minimization, cost minimization, and reliability maximization). Hence, there are $\prod_{i=1}^{N} M_i$ possible executing paths for task T in theory. After selection, the overall QoS parameters of the execution path are computed by using the aggregation function in Table 1. In this study, we assume that all objectives are to be minimized and all are equally important. Suppose there are k objectives (i.e., k service QoS dimensions). Formally, the optimization problem being addressing can be stated as follows:

$$\text{Min:} \quad \{f_1(x), f_2(x), \ldots, f_k(x)\}$$
$$\text{subject to:} \quad x \in X$$

where X is decision space, $x = \{x_1, x_1, ..x_N\}$ is a vector of X. In this case, k is three, N is the total number of the decomposed abstract services, and $f_i(x)$ is defined by using Table 1.

Table 1: Aggregation Function

	QoS parameter		
	Execution time	***Cost***	***Availability***
Aggregation method	$\sum_{i=1}^{N} T(a_i)$	$\sum_{i=1}^{N} C(a_i)$	$\prod_{i=1}^{N} A(a_i)$

3. Interactive EMO algorithm with Path Relinking

In this section, we propose a hybrid algorithm for solving QSS problem. The proposed hybrid algorithm is an extended version of our previous proposal Cone dominance-based interactive EMO algorithm (CDEMO) proposed in [12]. The CDEMO algorithm progressively interact with decision makers during optimization to learn the DMs' preference information. With DMs preference information, the CDEMO algorithm drives the search process toward a preferred region of the search space. For a more complete explanation of this interactive procedure, the interested reader is referred to [12], [13].

Decreasing user feedback requirement is one of the challenging issues of interactive algorithms. For decreasing user feedback requirement of the CDEMO algorithm, we propose a hybridization scheme on the CDEMO algorithm with Path Relinking (PR) technique. PR promotes the convergence of the interactive algorithm. It is accomplished by exploring trajectories connecting high-quality solutions set (elite set). Then, the population is updated with results of PR. The general overview of the proposed hybrid algorithm is given in Fig. 1, called EMO-Q.

For implementing it, EMO-Q is a modified version of the standard NSGA-II [14]. NSGA-II sorts solutions in a population according to two criteria. The primary criterion is the Pareto dominance-based sorting. It is described in detail in [14]. The result of this sorting is a partial order. The second criterion is the crowding distance-based sorting which sorts among solutions in the same dominance ranking in terms of Pareto dominance-based sorting. In the CDEMO algorithm, the crowding distance-based sorting is replaced with the cone dominance-based sorting. In this paper, we extend the CDEMO algorithm by joining Path Relinking in line 5 in Fig. 1.

```
Algorithm. EMO-Q
1. initialize population randomly
2. repeat
3.    get DM preference information;
4.    rank population using Pareto-Cone sorting;
5.    intensify population using Path Relinking;
6.    create next generation using standard
      evolutionary computation operations;
7. until Stopping criterion met
```

Fig. 1 General scheme of EMO-Q

3.1 Path Relinking (PR)

PR manages a set of promising solutions named the ''elite set''. In this study, the best N_{ES} solutions are selected for "elite set". In each iteration, PR randomly chooses two solutions from the elite set, named the initiating and guiding solution. Then, PR generates a sequence of successive solutions from the initiating to the guiding solution. This process is repeated until the termination criterion met. Each step is generated by replacing elements of the initial solution with the corresponding elements of the guiding solution.

In context of QSS problem, each step of any relinking path incorporates one service candidate from the guiding solution. It is worth noting that the order in which service candidates are incorporated defines different paths. It introduces the service candidates from the guiding solution in a greedy order that selects the best move in terms of Pareto and Cone-based utility function. Thus, there is only one path between the two solutions linked by PR. A local search technique is applied on the best solution of the path. The solution obtained by the local search will be the result of the PR. The population is updated with this resulting solution. An illustration of PR is shown in Fig. 2.

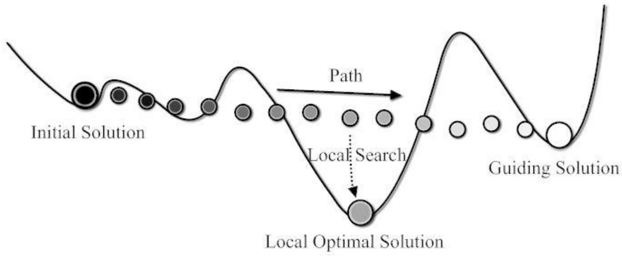


Fig. 2 An illustration of Path Relinking.

4. Experimentation

The experimentation consists of two sub-experiments. The aim of the first experiment is to compare the proposed algorithm with other algorithms, i.e., NSGA-II [14], CDEMO [12] in terms of speed of convergence over iterations. The aim of the second experiment is to compare the performance of the proposed algorithm with a previous proposal i.e., CDEMO in terms of the requirement of user feedback.

4.1. Experimental Input

The test problems are generated at random. The task numbers are 30 for composite web service instances. The number of candidate services for each task were randomly obtained from 15 to 50. The QoS attribute values of candidate services are randomly generated in [0, 1]. The same initial population is used for impartial comparisons among three algorithms. The proposed EMO-Q is implemented using MOEA Framework, version 2.0 (available at http://www.moeaframework.org/). Results for all approaches have been averaged over 30 independent runs. For selecting the DM's utility function, the functions given in [13] are used (Table 2). Where X is the set of feasible solutions.

Table 2: The utility function of the DM

# of obj	Function Form		
	Linear	Chebyshev	S.t:
Two	(0.6f1(x)+0.4f2(x))	max {0.6f1(x), 0.4f2(x)}	$x \in X$
Three	(0.4f1(x)+0.3f2(x) +0.3f3(x))	max {0.4f1(x), 0.3f2(x), 0.3f3(x)}	$x \in X$

We executed several preliminary experiments to define the right parameters of the proposed algorithm. Then, the parameters used in the experiments are selected and shown in Table 3. Crossover probability, mutation probability, and population size are the usual parameters associated with an EMO algorithm. Hence, we used the standard values and operators for these parameters. As it is mentioned in [15], the size of the elite set N_{ES} should have a small size (around 10). In addition, according to our preliminary experiments, the increase in the size of the elite set is not efficient. Therefore, we select 10 as the NES value. After executing preliminary experiments, we set Kmax = 0.01|V| and MaxIters = 0.1|V| where |V| is the number of vertex of the graph used as the input problem.

Table 3: Parameters of the proposed algorithm used in the experiment

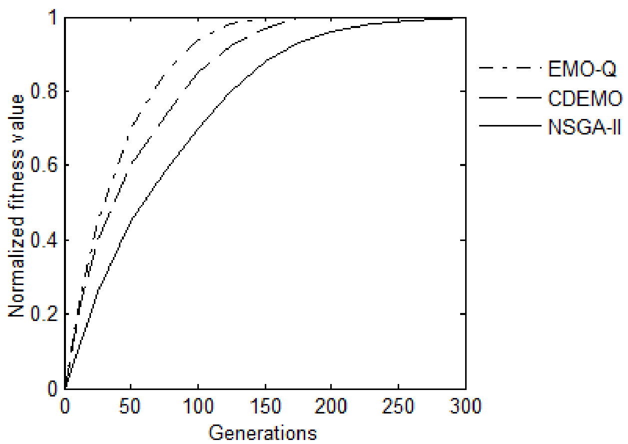
Technique	Parameter	Value
EMO-Q	NES	10
	Kmax	0.01|V|
	MaxIters	0.1|V|
	Population size	100
	Crossover probability	0.9 (Uniform crossover)
	Mutation probability	0.01 (Bit-flip mutation)

4.2 Experiment #1

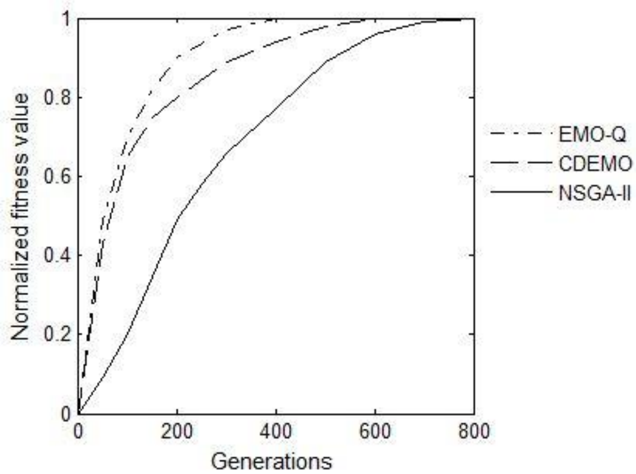
For evaluating the convergence speed, EMO-Q, CDEMO [12], and NSGA-II [14] are compared with respect to the number of generations required to reach the most preferred point. CDEMO is a recently proposed interactive EMO algorithm and explained in detail in [12]. NSGA-II is used for performance evaluation of the proposed algorithm because it is the most well established and known approach. Linear form was considered as the utility function, and the number of DM calls was set to 20 for all runs. The results have been normalized, along the whole run, in interval of [0, 1].

4.2.1 Results and discussions

The convergence plot can be seen in Fig. 3. It shows the utility function's value for the most preferred solution in the population over generations. EMO-Q finds the most preferred solution more quickly than the standard NSGA-II and the CDEMO algorithm (Fig. 3). The difference among these three algorithm was increasing by the increase in the number of objectives. For example, in case of two objectives, at 100th generation, the results of EMO-Q are better than the results of CDEMO at 140th generation, and that of NSGA-II at 180th generation (Fig. 3 (a)). In case of three objectives, at 200th generation, the results of EMO-Q are better than the results of CDEMO at 310th generation, and that of NSGA-II at 520th generation (Fig 3 (b)). We can conclude that the Path Relinking technique has promoted the convergence speed of the proposed algorithm.



(a). two objective case



(b). three objective case

Fig. 3 EMO-Q, CDEMO, and NSGA-II's fitness value versus the number of generation.

4.3 Experiment #2

In second experiment, we compared the proposed EMO-Q algorithm to CDEMO in terms of the number of DM calls required to reach the most preferred point. This measure enable us to evaluate the impact of Path Relinking technique with respect to the number of DM calls. To estimate this measure, we use the method explained in [13]. The authors of [13] calculates the deviation between the reported solution and the optimal solution obtained with the true utility function. The deviation is found by using the following formula:

$$\Delta = \frac{X^{opt} - X^{out}}{X^{opt} - X^{poor}}$$

Where X_{opt} is the solution obtained by the algorithm's a variant when the DM's utility function is known, X_{out} is the solution obtained by the algorithm's a variant when the DM's utility function is unknown, and X_{poor} is the poorest solution in the population. The deviation provides a measure for evaluating an interactive procedure. If deviation Δ is small, it implies that the interactive procedure works well and vice versa. If deviation Δ equals to zero, it implies that X_{opt}, and X_{out} are equal. This means that the interactive procedure has developed and learned the DM's preference over DM calls. In other words, it has reached with the level that can be obtained by using the underlying utility function.

To reach zero deviation, EMO-Q requires 13 DM calls for both utility function form while at least 15 DM calls for both utility function form is needed for CDEMO. The improvement is 13.3%. Note that reducing the number of DM calls is one of the major challenging issues for interactive algorithms [4].

5. Conclusion

The ever growing number of services and interoperability between various platforms is giving end users a large number of similar options to choose from. Selection of best combination of services based on multiple criteria can be a difficult problem for end users. The EMO-Q algorithm can help the users identify the right services more easily. In the proposed technique, using the interactive approach helps in guiding the optimization towards a more suitable solution from the user's perspective. The use of cone dominance-based EMO keeps the direction of optimization flexible enough so that it quickly converges towards the user preferred part of the Pareto front while the path relinking improves this process further by promoting the convergence speed. The combined effect of these techniques enables the proposed technique to perform much better than the previously developed techniques such as NSGA II and CDEMO.

Reference

[1] Strunk, A. (2010). "QoS-aware service composition: a survey," in proc. IEEE 8th European Conference on Web Services 2010 (ECOWS 2010), pp. 67–74.
[2] I. Das, and J.E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems," Structural Optimization, vol. 14, no. 1, pp. 63-69, 1997.
[3] J. Wallenius, J. S. Dyer, P. C. Fishburn, R. E. Steuer, S. Zionts, and K. Deb, "Multiple Criteria Decision Making, Multiattribute Utility Theory: Recent Accomplishments and What Lies Ahead," Management Science, vol. 54, no. 7, pp. 1336-1349, 2008.

[4] Sinha, A., Korhonen, P., Wallenius, J. and Deb, K. (2014) 'An interactive evolutionary multi-objective optimization algorithm with a limited number of decision maker calls', European Journal of Operational Research, Vol. 233 No. 3, pp. 674–688.

[5] Wang, R., Purshouse, R.C., Giagkiozis, I. and Fleming, P.J. (2015) 'The iPICEA-g: a new hybrid evolutionary multi-criteria decision making approach using the brushing technique', European Journal of Operational Research, Vol. 243 No. 2, pp. 46–60.

[6] Vallada, E. and Ruiz, R. (2010) 'Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem', Omega, Vol. 38 No. 1-2, pp. 57 - 67.

[7] Ribeiro. C.C. and Resende, M.G.C. (2012) 'Path-relinking intensification methods for stochastic local search algorithms', Journal of Heuristics, Vol. 18 No. 2, pp. 193 - 214.

[8] Duarte, A., Sánchez-Oro, J., Resende, M.G.C., Glover, F. and Martí, R. (2015) ' Greedy randomized adaptive search procedure with exterior path relinking for differential dispersion minimization', Information Sciences, Vol. 296, pp. 46–60.

[9] Miettinen, K.M. (1999) 'Nonlinear Multiobjective Optimization', Boston: Kluwer.

[10] Deb, K. (2014) ' Multi-objective Optimization', in: Burke, E.K. and Kendall, G. (Eds.), Search Methodologies, Springer, US, pp. 403-450.

[11] Korhonen, P., Wallenius, J., and Zionts, S. (1984) 'Solving the Discrete Multiple Criteria Problem Using Convex Cones', Management Science, Vol. 30, No. 11, pp. 1336-1345.

[12] Dalaijargal, P., Rehman, S., Cui, G., Nwe, N.H.W. and Jian-min, B. (2014) 'Cone Dominance-based Interactive Evolutionary Multiobjective Algorithm for QoS-driven Service Selection Problem', in CSE 2014: Proceedings of IEEE 17th International Conference on Computational Science and Engineering, Chengdu, China, pp. 940-945.

[13] Fowler, J.W., Gel, E.S., Köksalan, M., Korhonen, P.J., Marquis, J.L, and Wallenius, J. (2010) 'Interactive evolutionary multi-objective optimization for quasi-concave preference functions', European Journal of Operational Research, Vol. 206 No. 2, pp. 417-425.

[14] Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T. (2002) 'A fast and elitist multiobjective genetic algorithm: NSGA-II', IEEE Transactions on Evolutionary Computation, Vol. 6 No. 2, pp. 182-197.

[15] Laguna, M., and Marti, R. (2003) 'Scatter Search: Methodology and Implementations in C', Springer, 2003