Int J Theor Phys (2013) 52:2181–2186
DOI 10.1007/s10773-013-1544-0

Study of a Quantum Framework for Search Based Software Engineering

Nan Wu · Fangmin Song · Xiangdong Li

Received: 29 December 2012 / Accepted: 16 February 2013 / Published online: 9 March 2013

Abstract The Search Based Software Engineering (SBSE) is widely used in the software engineering to identify optimal solutions. The traditional methods and algorithms used in SBSE are criticized due to their high costs. In this paper, we propose a rapid modified-Grover quantum searching method for SBSE, and theoretically this method can be applied to any search-space structure and any type of searching problems.

Keywords Quantum algorithm · Software engineering · Searching · Complex structure · Grover algorithm

1 Introduction

With the rapid growth of software engineering and the expanse of computer science, various problems have been raised and no polynomial time complexity solution algorithm has been found so far [1]. These problems cover a broad class of important applications, such as the protein structure prediction, many-body problem simulation, multi-epitope gene sequence searching, coordination chemistry engineering and civil engineering, etc. These problems can only be solved by using the large-scale search with the state-of-the-art technologies.

The search based software engineering (SBSE) was firstly claimed in 2001 by computer scientists Harman and Jones [2]. The basic idea of SBSE is to use unstructured metaheuristic searching algorithms, such as the genetic algorithm (GA) [3], simulated annealing [4] and Tabu search [5], etc. The SBSE also exploits a set of software engineering framework for

N. Wu (✉) · F. Song
National Key Laboratory for Novel Software Technology, Nanjing University, Jiangsu 210023, China
e-mail: nwu@nju.edu.cn

N. Wu · F. Song
Department of Computer Science and Technology, Nanjing University, Jiangsu 210023, China

X. Li
Department of Computer System Technology, NYCCT, City University of New York, New York, NY 11201, USA

the proposed variety of searching algorithms to provide solutions for the "difficult problems of balancing competing constrains and may suggest ways to finding acceptable solutions in situations where perfect solutions are either theoretically impossible or practically infeasible" [2].

After nearly 10-year research, the traditional SBSE has been developed substantially in both theoretical and experimental field. However, two serious problems are still plaguing people when using SBSE. First, the SBSE aims to do the search within the completely unstructured data, but it is limited by the data structure and abstract data type of the computer can expressing, one cannot reuse a completely traditional data-structure model and a traditional (random) algorithm to meet the wide variety of requirements from the different solutions for the space structure of problems. Second, to obtain a "good" solution of a problem, the traditional searching algorithms always show that their efficiency is low—even using the randomized searching or stochastic algorithms; it is hard to get a satisfying solution with a polynomial time.

2 Quantum Searching Methods

People have studied to develop the quantum computing device (called as "quantum computer") to expedite the traditional searching speed since the property of quantum parallelism was discovered. In 1996, Grover first proved that by using the quantum parallelism and quantum superposition effects we could speed up the searching over a classical database was possible [6]. Compared with the Shor's factoring algorithm that is actually to search (or find) an order (or period) in a hidden subgroup, Grover's algorithm can be used for a wide-area search in unstructured data or objects. From this point of view, Grover's algorithm is treated as a more important quantum-searching algorithm than Shor's algorithm. Afterwards, various universal quantum searching methods have been proposed and studied, such as the quantum adiabatic evolution searching algorithm [7], fixed-point quantum search (FPQS) [8, 9], quantum space search in hierarchical structures [10] and quantum walks [11, 12]. Among these quantum searching methods, people think Grover's algorithm is the best choice for large-scaled unstructured data searching.

3 Unstructured Problem Spaces and Grover's Algorithm

Now consider a finite unstructured set $P = \{1, 2, \dots, N_G\} \in \{0, 1\}^{\otimes n}$, where $n \in \mathbb{N}^+$ and $N_G = 2^n$. We want to find a specified target element x_t, where $x_t \in X$. Let $f : \{0,\ 1\}^{\otimes n} \to \{0,\ 1\}$ be an well-defined externally supplied function (or say a *Oracle*), such that $f(x_t) = 1$ and $f(x \neq x_t) = 0$. Grover's original algorithm has the following steps [6]:

Algorithm (Grover's algorithm)
Step 1. Initialize the quantum system as initial (ground) states of size N_G, with amplitude uniformed as $|\psi_{init}\rangle = \frac{1}{\sqrt{N_G}} \sum_x |\psi_x\rangle$ (where computational basis states is $\{|0\rangle, |1\rangle\}^{\otimes n}$).
Step 2. Repeat the following quantum operations (Grover Iterate) for $O(\sqrt{N_G})$ times:

(i) Let $|x\rangle$ be any computational basis state, **if** $f(x_t) = 1$ **then** rotate state by π, **else if** $f(x) = 0$ **then** leave the state unaltered;
(ii) Apply diffusion transform D such that:
$D_{ij} = \frac{2}{N_G}$ when **if**
$i \neq j \wedge D_{ii} = -1 + \frac{2}{N_G}$.

Step 3. Implement projective measurement to the state and sample the searching result.

With simple analysis we can show that Grover's algorithm cannot be always successful in a deterministic situation (with an error $O(1/N_G)$), so Grover's original algorithm is also a stochastic algorithm. Long et al. [13] analyzed Grover's algorithm in group $SO(3)$ structure and proposed a modified version with a new database-scale-related phase match equation (i.e. change rotate phase π in **Step 2(i)** as $2\arcsin[\sin(\frac{\pi}{4J+6})\sqrt{N_G}]$, where $J = \frac{\pi}{4\beta} - \frac{1}{2}$, $\beta = \arcsin\frac{1}{\sqrt{N_G}}$) [13, 14]. The Long's modified version of Grover's algorithm is a deterministic algorithm that can reach the targeted answer with a constant probability of 1 [13]. Grover's algorithm has also been proven to be optimal and faster than any other possible classical algorithm in 1997 [15], and it has been implemented in quantum devices [16]. Grover's algorithm is also applicable to multiple-target unstructured data search [6].

4 Analysis of SBSE-Based Unstructured Searching Problems

The important goal of search-based software engineering is to reformulate the problems for software engineering as searching problems. Three criteria are proposed below [2]:

(1) A representation of the problem which is amenable to symbolic manipulation;
(2) A fitness function (defined in terms of this representation);
(3) A set of manipulation operators.

Here we adopt the above criteria into the quantum computational space. First, a representation of a candidate solution must show the characteristic of searching problems in Hilbert space, a suitable method should be used to recode the original problem into quantum grey codes which are in the form of pure ground state—the qubits. The reason to use the pure state, not the mixture or superposition form, is that we can easily code the answer of targeted problem into the final ground computational basis by using the grey transformation matrices. In computer science, the solution space of an unstructured searching problem can be rebuilt with the discrete form, each solution can also be represented as a series of basis vectors—which can be easily presented as the state vectors in the geometric interpretation of Grover's algorithm. And more important, the accurate unitary transformation can be implemented in these state vectors. See Fig. 1.

Second, in computer science we use fitness function to characterize what is considered to be a "good solution". In the theory of software measurement, "the fitness function need merely impose an ordinal scale of measurement upon the individual solutions it is applied to" [2]. So before the implementation of quantum projective measurement, it is critical to know which of the candidate solutions is a better solution based on their properties. In the

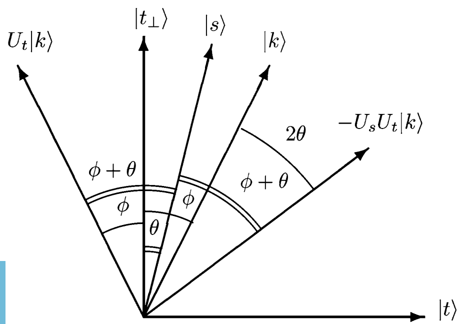

Fig. 1 The geometry interpretation of Grover's algorithm. The steps of the unstructured search can be represented as the rotations in the plane formed by the states $|s\rangle$ and $|t\rangle$. The angle $\theta = \arcsin t|s\rangle$

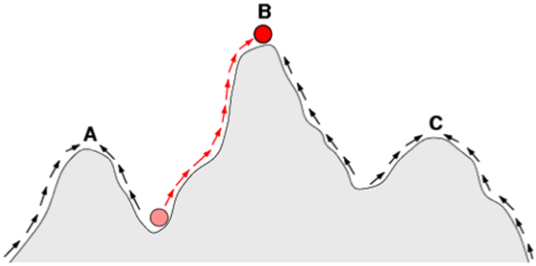

Fig. 2 Fitness function can evaluate the metric distance between the "good solution" (point B) and other local optimal solutions (points A and C). The QFT can transform the quantum evolution process into a global optimal solution

quantum version of fitness function, a well-designed function is used to implement probability evaluation of the trace distance (i.e. Kolmogorov distance) and we can use quantum Fourier transform (QFT) to adjust the distance to the "good solution", see Fig. 2. In some cases, it is hard to define a fitness function due to the difficulty on determining a given problem's ordinal scale metric. However, other metaheuristic searching approach can be used as a classic post-processing.

Third, in the proposed framework of quantum searching used for SBSE, the different operators are used to implement different searching techniques. We need to build a universal-architecture framework for various types of searching problems, and then define the variety of corresponding quantum operators. In the proposed framework, all the operators are developed based on Grover's algorithm, which only considers the problem-dependent information (see Sect. 5).

5 Modified Grover's Search—A Universal Quantum Searching Framework

Actually, for many problems in software engineering, even in the situations where there is no known general solving algorithm, there may exist known examples which can be constructed by hands, where the individual solutions are known for particular elements of the problem domain [2]. It means that the "correctness" of a solution can be easily characterized by a pre-designed characteristic function and the "goodness" of a solution can be validated and/or measured by a fitness function.

With the quantum searching method, it's easy to find a single target in a complex and unstructured dataset, however, to find all the best or optimal solutions is very hard. The fitness function used to evaluate a searching result always needs additional information—called the *priori*, this affects the Grover searching's iteration level for completing a successful search. The basic iterative deepening Grover's search (IDGS) algorithm [17] is shown as the following:

Algorithm (iterative deepening Grover's search)

Step 1: Initialization

Initialize the quantum system as initial (ground) states of size N_G, with the normalized amplitude as $|\psi_{init}\rangle = \frac{1}{\sqrt{N_G}} \sum_x |\psi_x\rangle$ (where the computational basis states are $\{|0\rangle, |1\rangle\}^{\otimes n}$). Let a characteristic function $f : \{0, 1\}^{\otimes n} \rightarrow \{0, 1\}$ be an externally supplied function (i.e. *Oracle*) defined as $f(x_t) = 1$ and $f(x \neq x_t) = 0$.

Step 2: Iteration

The function *Grover*() is the standard Grover's searching procedure:

for i := 1 to **sizeof** (f) **skipping** 2 **do**

  $r := Grover(f, \lfloor \frac{\pi}{4} 2^{\frac{i}{2}} \rfloor)$;

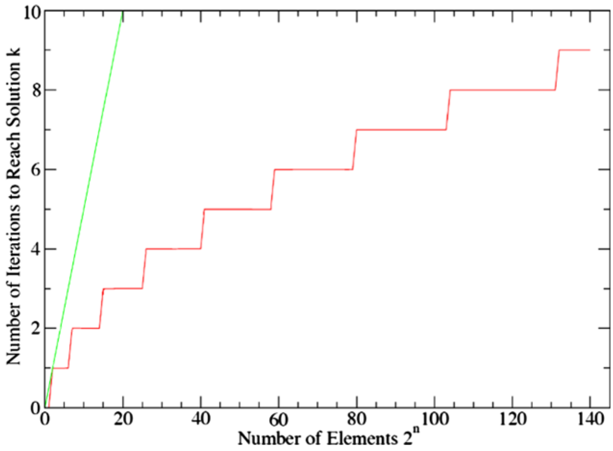

Fig. 3 Iteration count vs. number of searched elements in IDGS algorithm, *green line*: classical search in $2^n/2$ scale; *red line*: IDGS estimate of $\frac{\pi}{4}\sqrt{2^n}$

  if $(f(r) == 1)$ **then return** r;
endfor;
return (-1).
Step 3: Sample r for the searching results.

Iterative deepening Grover's search is a basic framework in this paper and we now consider the costs. Figure 3 shows the iteration count vs. the number of searched elements. It is easily find above algorithm is more efficient in iteration level [18].

The basic steps to find solution from an unstructured dataset are:

(1) Modeling the problem;
(2) Design the characteristic function f to proposed problem;
(3) Use IDGS for deepening Grover searching;
(4) Evaluate the estimated solution by a classic fitness function.

6 Conclusion and Further Work

To summary, we propose a quantum searching framework based on Grover's quantum searching algorithm for solving variety classes of problems in software engineering. We firstly show that many software engineering problems can be translated into a multi-target searching problem, then secondly, analysis the characteristics of unstructured problem space in search based software engineering. Thirdly, we mention three criteria in SBSE and propose the quantum version of these criteria, and finally, we propose a basic frame work by using iterative deepening Grover search for finding multi-target in large-scale and unstructured dataset in SBSE scenario.

The related tools of our proposed framework is under developing and we also want to design the algorithm for dynamic adjusting when the problem size changes.

Acknowledgements This work is supported by the Chinese National Natural Science Foundation of Innovation Team (Grant No. 61021062); the Natural Science Foundation of Jiangsu Province, China (Grant No. 2011560).

References

1. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge University Press, New York (2000)
2. Harman, M., Jones, B.: Inf. Softw. Technol. **43**, 833 (2001)
3. Goldberg, D.E.: Algorithms in Search, Optimization & Machine Learning. Addison-Wesley, Reading (1989)
4. Laarhoven, P., Aarts, E.: Simulated Annealing: Theory and Practice. Kluwer, Dordrecht (1987)
5. Glover, F.: Interfaces **20**, 74 (1990)
6. Grover, L.: In: Proc. 28th Annual ACM Sym. on Theory of Comp. (STOC), p. 212 (1996)
7. Childs, A., Deotto, E., Farhi, E., et al.: Phys. Rev. A **66**, 032314 (2002)
8. Grover, L.: Phys. Rev. Lett. **95**, 150501 (2005)
9. Tulsi, T., Grover, L., Patel, A.: Quantum Inf. Comput. **6**, 483 (2006)
10. Benioff, P.: AMS Contem. Math. Ser. **305** (2002)
11. Tulsi, A.: Phys. Rev. A **78**, 012310 (2008)
12. Tulsi, A.: Ph.D. thesis, Indian Institute of Science (2009)
13. Long, G., Li, Y., Xiao, L., et al.: Nucl. Phys. Rev. **21**, 114 (2004)
14. Long, G., Li, Y., Zhang, W., et al.: Phys. Lett. A **262**, 27 (1999)
15. Bernstein, E., Bennett, C., Brassard, G., et al.: SIAM J. Comput. **26**, 1510 (1997)
16. Bhattacharya, N., van den Heuvell, H., Spreeuw, R.: Phys. Rev. Lett. **88**, 137901 (2002)
17. Hall, R.: In: Proc. IEEE/ACM Internal. Conf. on Auto. Soft. Engineer, p. 40 (2009)
18. Smith, A., Alsing, P., McDonald, J., et al.: (2012). arXiv:1211.3407v2