IEEE Access
practical innovations : open solutions

Received May 14, 2014, accepted May 26, 2014, date of publication June 4, 2014, date of current version June 10, 2014.

Digital Object Identifier 10.1109/ACCESS.2014.2327596

Quantum-Assisted Routing Optimization for Self-Organizing Networks

DIMITRIOS ALANIS, (Student Member, IEEE), PANAGIOTIS BOTSINIS, (Student Member, IEEE), SOON XIN NG, (Senior Member, IEEE), AND LAJOS HANZO, (Fellow, IEEE)

School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

Corresponding author: L. Hanzo (lh@ecs.soton.ac.uk)

This work was supported in part by the EPSRC under the grant EP/L018659/1, in part by RC-UK under the India-UK Advanced Technology Centre (IU-ATC), in part by the EU under the CONCERTO Project and that of the European Research Council, Advanced Fellow Grant, and in part by the Royal Society's Wolfson Research Merit Award.

ABSTRACT Self-organizing networks act autonomously for the sake of achieving the best possible performance. The attainable routing depends on a delicate balance of diverse and often conflicting quality-of-service requirements. Finding the optimal solution typically becomes an nonolynomial-hard problem, as the network size increases in terms of the number of nodes. Moreover, the employment of user-defined utility functions for the aggregation of the different objective functions often leads to suboptimal solutions. On the other hand, Pareto optimality is capable of amalgamating the different design objectives by providing an element of elitism. Although there is a plethora of bioinspired algorithms that attempt to address this optimization problem, they often fail to generate all the points constituting the optimal Pareto front. As a remedy, we propose an optimal multiobjective quantum-assisted algorithm, namely the nondominated quantum optimization algorithm (NDQO), which evaluates the legitimate routes using the concept of Pareto optimality at a reduced complexity. We then compare the performance of the NDQO algorithm to the state-of-the-art evolutionary algorithms, demonstrating that the NDQO algorithm achieves a near-optimal performance. Furthermore, we analytically derive the upper and lower bounds of the NDQO algorithmic complexity, which is of the order of $O(N)$ and $O(N\sqrt{N})$ in the best and worst case scenario, respectively. This corresponds to a substantial complexity reduction of the NDQO from the order of $O(N^2)$ imposed by the brute-force method.

INDEX TERMS SONs, quantum computing, Pareto optimality, Grover's QSA, BBHT-QSA, NDQO, ACO, NSGA-II, complexity reduction.

NOMECLATURE

ACO	Ant Colony Optimization
BBHT	Boyer, Brassard, Høyer and Tapp
BER	Bit Error Ratio
BF	Brute Force
BSC	Binary Symmetric Channel
CD	Classical Domain
CF	Cost Function
CFE	Cost Function Evaluation
CLT	Central Limit Theorem
CNOT	Controlled-NOT quantum gate
DAF	Decode And Forward
DCCP	Dynamic Coverage and Connectivity Problem
DHA	Durr-Høyer-Algorithm
DN	Destination Node
DSS	Direct-Sequence Spreading
HGR	Hybrid Geographic Routing
HYMN	HYbrid Multihop Network
MMAS	Min-Max Ant System
MODE	Multi-Objective Differential Evolution
MUD	Multi-User Detection
NDS	Non-Dominated Sort
NDQO	Non-dominated Quantum Optimization
NP	Non-Polynomial
NSGA-II	Non-dominated Sort Genetic Algorithm II
OF	Objective Function
OPF	Optimal Pareto Front
OW	Oracle Workspace
PF	Pareto Front
PLR	Packet Loss Ratio
QACO	Quantum Ant Colony Optimization
QAE	Quantum Amplitude Estimation
QCR	Quantum Control Register
QD	Quantum Domain

QIR	Quantum Index Register
QMA	Quantum Mean Algorithm
QoS	Quality of Service
QP	Quantum Parallelism
QSA	Quantum Search Algorithm
QWSA	Quantum Weighted Sum Algorithm
RN	Relay Node
SN	Source Node
SO	Soft-Output
UF	Utility Function
VoIP	Voice over Internet Protocol
WSN	Wireless Sensor Network
XOR	Exclusive OR gate.

I. INTRODUCTION

SELF-ORGANIZING Networks (SONs) [1] are capable of reorganizing their structure in order to improve their performance. Their operations include *self-configuration*, which allows uninitialized *ad hoc* nodes to be seamlessly inserted in a SON without any service disruptions, while their *self-optimization* and *self-healing* capabilities improve the network's robustness. One of their salient *self-optimization* procedures is routing. The concept of SONs may be readily applied in both *ad hoc* networks [2] and in *Wireless Sensor Networks* (WSNs) [3], which are usually based on mobile nodes. Each node has a restricted amount of power and typically aims for minimizing the Bit-Error-Ratio (BER) or Packet-Loss-Ratio (PLR). Moreover, the transmission power of each link as well as the end-to-end delay should also be given cognizance, according to the Quality of Service (QoS) requirements. Consequently, routing optimization is essential for satisfying these requirements. However, as the network size increases, the total number of legitimate paths increases exponentially [4], potentially rendering routing a *Non-Polynomial Hard* (NP-hard) problem, which requires highly sophisticated optimization methods.

A. RELATED WORK

Our literature review is based on a pair of parallel orientations. On the one hand, we will present the classical computing approaches conceived for solving the multi-objective optimization problems. On the other hand, we survey the advances in quantum computing, which facilitate the employment of *Quantum Search Algorithms* (QSA) [5], [6].

1) CLASSICAL ROUTING APPROACHES

Zhu *et al.* [7] have introduced a protocol, in which the nodes of a WSN are organized into clusters using Hausdorff clustering [7] for minimizing the energy consumption. Chen *et al.* [8] proposed a Hybrid Geographic Routing (HGR) scheme for satisfying the end-to-end delay requirements considered, while optimizing the energy dissipation. Furthermore, Abdulla *et al.* [9] conceived a range of HYbrid Multihop Network (HYMN) performance metrics for minimizing the power consumption and hence for maximizing the lifetime of WSNs.

Numerous single-objective approaches have been advocated in the literature, because focusing on a single requirement may unduly degrade the remaining metrics. This problem may be avoided [10] by using a multi-objective approach. Likewise, all the requirements considered may be optimized jointly without the need for user-defined parameters in order to aggregate the different design objectives [11].

In fact, there is a plethora of comprehensive studies in the literature [4], [12]–[15], which investigate diverse aspects of WSNs using the multi-objective approach relying on evolutionary algorithms. For example, Yetgin *et al.* [4] used both the Non-dominated Sorting Genetic Algorithm II (NSGA-II) and the Multiobjective Differential Evolution Algorithm (MODE) for optimizing the transmission routes in terms of their end-to-end delay and power dissipation. They used the concept of Pareto Optimality [16] for evaluating the fitness of multi-objective problems. While considering a similar context, Camelo *et al.* [12] employed the NSGA-II in order to satisfy the same QoS requirements for both the ubiquitous Voice over Internet Protocol (VoIP) and for file transfer. Moreover, Perez *et al.* [13] used a multi-objective model for optimizing both the number of sensor nodes used in a WSN and the total energy dissipation of the network, which allowed the minimization of the WSN's deployment cost. Martins *et al.* [14] employed a hybrid multi-objective evolutionary algorithm for solving the Dynamic Coverage and Connectivity Problem (DCCP) of WSNs subjected to node failures.

Among the evolutionary algorithms, the so-called *Ant Colony Optimization* (ACO) conceived by Dorigo and Di Caro [17], [18] has been extensively used for optimizing routing problems. In the so-called *Ant Network* (AntNet) [18], each ant representing a legitimate route travels from a source node (SN) to the destination node (DN), while traversing a different number of nodes. Each ant moving from one node to another deposits an amount of pheromone across its route depending on the heuristic value of the Objective Function (OF) that is being optimized. To elaborate further, as the value of the OF increases throughout the route-search, the intensity of the pheromone would increase as well. Based on the deposited pheromone, the ant in nature are capable of sensing in binary fashion, whether a specific route is leading up to the source of food or not. Meritorious routes attract more ants and as additional ants choose to follow a specific link between two nodes, the pheromone they deposit is cumulatively superimposed so that the optimal routes would have the highest pheromone intensity. An additional factor, namely the so-called *intrinsic affinity*, was introduced for avoiding premature convergence to local optima, which particularly corresponds to the *a priori* probability of each solution being the globally optimum solution.

Further extensions of the *AntNet* model have also been proposed. Both Golshahi *et al.* [19] and Chandra *et al.* [20] have implemented a routing protocol using the *AntNet* model; two types of agents have been used: a forward-oriented one which would seek to explore the network and a backward-oriented

one, which would perform the ACO operations including the pheromone update and fitness-evaluation of the routes. Finally, Wang and Wu [21] developed an ACO routing algorithm for optimizing the performance of fault tolerant *Hypercube Networks* at reduced complexity by exploiting the regularities exhibited by these networks. Additionally, Pinto *et al.* introduced the concept of Pareto Optimality in [15] for conceiving a *Multiobjective Max-Min Ant System* (MMAS) for solving the multi-objective mutlicast routing problem.

Finally, a quantum-inspired version of the ACO (QACO) algorithm has been proposed by Wang *et al.* [22]. Jiang *et al.* [23] invoked an enhanced version of the QACO design for increasing the lifetime of a WSN by minimizing its energy consumption.

2) ADVANCES IN QUANTUM COMPUTING

The advances of the most recent decades in Quantum Computing [5] provide us with substantial new insights on the design of quantum search algorithms and on quantum communications [24]–[26]. In particular, in 1981 Feynman [26] proposed a framework in which a quantum computer simulated the evolution of quantum states. Then in 1985, Deutch [27] proposed a quantum algorithm for finding a global maximum or minimum of a function $f : \{0, 1\} \rightarrow \{0, 1\}$ whilst relying on a single call of the function f by exploiting the so-called *Quantum Parallelism* (QP) [5], [6]. Later in 1992, Deutch and Jozsa [28] extended this algorithm to functions having n-bit binary arguments $f : \{0, 1\}^n \rightarrow \{0, 1\}$ for determining, whether f is balanced or constant[1]. These innovations prepared the ground for the development of more sophisticated quantum algorithms.

In 1996 Grover [29] proposed a Quantum Search Algorithm (QSA), which would find a particular solution[2] in an unsorted database by relying on as few as $O(\sqrt{N})$ calls to the database. The theoretical proof of this algorithm has been provided by Boyer *et al.* [30], who also proposed an extension of this algorithm in the same treatise for the case, where the number of solutions is not known. Moreover, Zalka in [31] proved that Grover's QSA is optimal in terms of the number of database calls. A further improved extension to the original algorithm proposed by Boyer *et al.* [30] has been provided by Durr and Høyer [32] in form of the so-called *Durr-Høyer-Algorithm* (DHA), which is capable of finding the index of the minimum value of a database in *quadratic* time.

Apart from QSAs, a range of further important contributions have been made in the field of computation. In 1994, Shor [33] introduced an algorithm, which would perform the factorization of a polynomizal into its prime functions in polynomial time. This invention eventually led to the concept of *Quantum Amplitude Estimation* (QAE) and to the *Quantum Counting Algorithm* [34]. Based on these principles, Brassard *et al.* proposed the *Quantum Mean Algorithm* (QMA) [35] for calculating the mean of the values in a database at a reduced complexity. Botsinis *et al.* [36] expanded the QMA by proposing the *Quantum Weighted Sum Algorithm* (QWSA), which calculates the weighted sum of a function $f : \{0, \ldots, 2^n - 1\} \rightarrow \{0, 1\}$. Moreover, the same authors [37] investigated the impact of reduced-complexity early termination of the DHA using deterministic inputs on the attainable performance of *Multi-User Detection* (MUD) conceived for multiple access wireless systems. A further improved version of the algorithm was used by the same authors in [38] for implementing an optimal *Soft-Output* (SO) MUD scheme for *Direct-Sequence Spreading* (DSS) and *Slow Subcarrier-Hopping* (SSH) aided *Space-Division Multiple Access - Orthogonal Frequency Division Multiplexing* (SDMA-OFDM) systems.

In a nutshell, quantum computing offers a well-equipped parallel-processing toolbox for solving NP-hard problems, such as the multi-objective routing problem.

B. CONTRIBUTIONS AND PAPER STRUCTURE

The classical computing methods mentioned in the previous subsection are suboptimal. To elaborate further, not only they fail to spot all the paths that belong to the *Optimal Pareto Front* (OPF) [39], but they also often mistakenly identify others which are not optimal, since they may converge to local minima. In this paper we propose a QSA, which addresses these problems and at the same time exploit the property of QP, resulting in a beneficial complexity reduction.

In particular, our contributions may be summarized as follows:

1) *We have proposed a novel quantum-assisted algorithm, namely the Non-dominated Quantum Optimization (NDQO) algorithm, which optimizes the multi-objective routing problem using the exponential search algorithm of [30]. We have also improved the latter algorithm and derived the NDQO algorithm's complexity upper- and lower-bounds by taking into consideration the classical cost function evaluations (CFE) invoked by this algorithm.*
2) *We have solved of the OPF partial generations, by ensuring that all the optimal legitimate routes will be identified despite imposing a reduced complexity.*
3) *We have characterized the performance versus complexity of the NDQO algorithm and have demonstrated that it achieves the optimal performance at a complexity, which is on the order of $O(N^{3/2})$ in the worst-case scenario.*

The rest of this paper is organized as follows. In Section II, we will detail the assumptions concerning the network investigated. In Section III we will introduce the quantum computing framework of the NDQO algorithm, which is then analysed in Section IV. Finally, the evaluation of the NDQO in terms of its performance versus complexity is presented in Section V.

[1] A function is called constant iff for all the possible input values the same binary value is output.

[2] Given a known value y, an argument x of a function f is termed as a solution, iff $f(x) = y$.

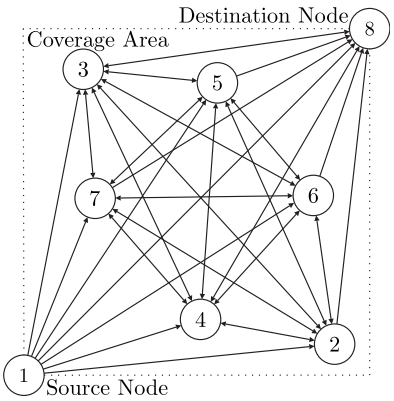

FIGURE 1. Exemplified network topology for an 8-node SON.

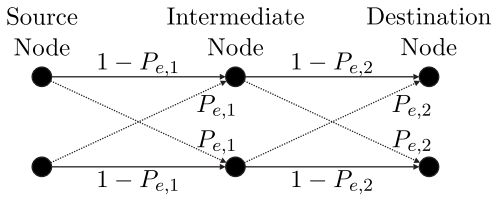

FIGURE 2. Two stage Binary Symmetric Channel for the case of an intermediate node.

II. SYSTEM OVERVIEW

A. NETWORK SPECIFICATIONS

As far as the network architecture is concerned, a fully interconnected network has been assumed. The coverage area was assumed to be a (100×100) m^2 square block. The relay node (RN) locations were generated using a uniform random distribution within this area, whereas the SN and the DN were located at the two opposite corners of this square block. Moreover, each path is assumed to traverse through each node at most once. In this way, unnecessary loops, which would result in potentially excessive PLR, may be avoided. An illustrative example of the topology considered is shown in Fig. 1, where the formation of all the legitimate paths for an 8-node SON is shown. We note that although the SN is capable of transmitting its message towards all nodes, it is not possible for the other nodes to send their message back to the SN, due to the previous assumption. Therefore, the reverse principle would apply for the DN.

According to the physical layer model considered, each message transmission uses QPSK modulation over an uncorrelated Rayleigh fading environment [40], where the *Bit Error Ratio* (BER), P_e, versus the *Bit-to-Noise power Ratio*, E_b/N_0, relationship is given by [41]:

$$P_e = \frac{1}{2}\left(1 - \sqrt{\frac{E_b/N_0}{E_b/N_0 + 1}}\right). \tag{1}$$

In the multiple access layer, each RN is capable of retransmitting the received messages using the classic *decode-and-forward* (DAF) scheme [42]. More specifically, each node decodes the received messages and then performs encoding and modulation in order to forward it to the next node. As the received message may be corrupted by errors due to erroneous detection at the previous nodes and also since QPSK is used, the channel can be modelled by a two-stage *Binary Symmetric Channel* (BSC) [40] as presented in Fig. 2. The RN corresponds to the intermediate node of Fig. 2 and the route has two BER values, one for each of the two links established. It is possible to transform this channel into a single-stage one, which would be described by a single overall BER $P_{e,12}$, given by:

$$P_{e,12} = P_{e,1} + P_{e,2} - 2P_{e,1}P_{e,2}. \tag{2}$$

The last term in (2) corresponds to the propagated errors, i.e. to the errors that were introduced by the first link and have been erroneously "corrected" by the introduction of another error within the second link.

Additionally, (2) may be used for recursively calculating the overall BER of a particular route. Owing to the *Central Limit Therorem* (CLT) [41] the interference caused by multiple users accessing the same channel may be treated as *Additive White Gaussian Noise* (AWGN) at each node. To elaborate further, during the network's initialization a second random process obeying a normal distribution has been invoked for assigning a specific noise level to each node. The mean of this noise was set to -90 dBm with a standard deviation of 10 dB.

Furthermore, another factor to be considered is the network's delay. In the proposed system the delays are introduced by the DAF scheme, since a finite time-duration would be required for a RN to perform all the necessary operations before forwarding a message. For simplicity, the service queue is assumed to have zero length, hence the messages would be forwarded almost instantly with a short delay equal to the DAF signal processing operation duration. Hence, the total delay of the route would be proportional to its number of established hops.

Moreover, as far as the power consumption is concerned, only the free-space path-loss of each link has been considered. In particular, the *path loss exponent* was set to $\alpha = 3$. Hence, the path loss L of single link may be formulated as [40]:

$$L = P_{\mathrm{Tx,dB}} - P_{\mathrm{Rx,dB}} = 10\alpha \log_{10}\left(\frac{4\pi d}{\lambda_c}\right) [\mathrm{dB}], \tag{3}$$

where $P_{\mathrm{Tx,dB}}$, $P_{\mathrm{Rx,dB}}$, d and λ_c stand for the transmitted power, the received power, the distance between the nodes of a link and the carrier wavelength respectively. The transmission power was set to 20 dBm for each link, while the carrier frequency was set to 1.8 GHz, which would result in a wavelength of 0.1667 m.

Last but not least, bearing in mind all the above assumptions, an N-node network may be modelled as a graph $G(E, V)$ having E edges and V vertices, which is

formulated as:

$$v_{i,j} = \begin{cases} \left[\text{SNR}_{i,j}, L_{i,j}, D_{i,j}\right] & \forall i,j \in E : i \neq j, \\ & j \neq 1, i \neq N, \\ \varnothing & \text{otherwise}, \end{cases} \quad (4)$$

where $v_{i,j}$ stands for the transition weight, $\text{SNR}_{i,j}$ is the received SNR, $L_{i,j}$ denotes the power losses due to path loss and $D_{i,j}$ represents the delay of forwarding a message from node i to node j. Additionally, the symbol $\varnothing$ in Eq. (4) corresponds to a transition that is not legitimate. Moving on to the route fitness evaluation, assuming a legitimate path x, which belongs to the set of all the possible legitimate routes S, its *Utility Function* (UF) $\mathbf{f}(x)$ is described by a vector formulated as:

$$\mathbf{f}(x) = \left[P_{e,x},\ CL_x,\ CD_x\right], \quad (5)$$

where $P_{e,x}$, CL_x, CD_x stand for the overall BER, the cumulative linear-domain sum of the path losses and the cumulative sum of the delays for route x respectively. Finally, the system parameters considered are summarized in Table 1.

TABLE 1. SON Network Parameters.

Network Coverage Area	100×100 m^2 Square Block
Modulation	QPSK
Mean Node Interference, μ_I	-90 dBm
Node Interference Std, σ_I	10 dB
Power Consumption Model	Log-Distance Path Loss Model with $a = 3$
Transmission Power	20 dBm
Carrier Frequency	1.8 GHz
Delay Unit	Single Hop

B. PARETO OPTIMALITY

Since the proposed approach is a multi-objective one according (5), the optimality of the route-solution vectors should be defined. The principle of *Pareto Optimality* introduced by Pareto for solving multi-objective maximization problems [16] will be employed for examining the optimality of a route-solution vector. However, Deb *et al.* [10] extended its employment for minimization problems as well. The *Pareto Optimality* may be summarized using Definitions 1, 2 [4] and 3.

Definition 1 (Pareto Dominance): A particular solution vector $\mathbf{f}(x_1) = [f_1(x_1), f_2(x_1), \ldots, f_n(x_1)]$ is said to dominate another particular solution vector $\mathbf{f}(x_2) = [f_1(x_2), f_2(x_2), \ldots, f_n(x_2)]$ if and only if $\mathbf{f}(x_1) \succeq \mathbf{f}(x_2)$; explicitly for $\forall\, i \in 1, 2, \ldots, n$, we have $f_i(x_1) \leq f_i(x_2) \wedge \exists i \in 1, 2, \ldots, n : f_i(x_1) < f_i(x_2)$, where n is the number of optimization objectives considered in the optimization problem.

To elaborate further, let us assume having a pair of solution vectors $\mathbf{f}(x_1)$ and $\mathbf{f}(x_2)$, which are shown in Fig. 3(b) as well another particular solution $\mathbf{f}(x_3)$, which may be located in each of the patterned areas of the f_1f_2 plane. If $\mathbf{f}(x_3)$ resides in the plain white area, then no dominance relationship exists

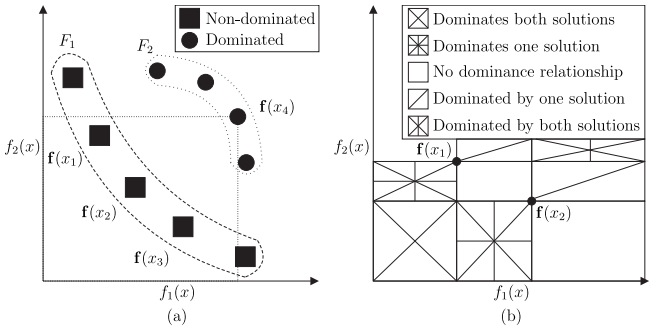

FIGURE 3. (a) Optimal Pareto Front for two objective functions and (b) dominance relationship between a front of two solutions.

since we have either $f_1(x_3) < f_1(x_i)$ and $f_2(x_3) > f_2(x_i)$ or $f_1(x_3) > f_1(x_i)$ and $f_2(x_3) < f_2(x_i)$ where $i \in \{1, 2\}$. In prose, no dominance relationship exists, when none of the solutions has coordinates, all of which are lower than their corresponding counterparts in the others. Moreover, if $\mathbf{f}(x_3)$ resides in the single-line-patterned area, it will then be dominated by either the $\mathbf{f}(x_1)$ or $\mathbf{f}(x_2)$ vectors, while it will not dominate the other one. By contrast, if the solution $\mathbf{f}(x_3)$ resides in the four-line-patterned area, it will then dominate only one of the $\mathbf{f}(x_1)$ and the $\mathbf{f}(x_2)$ vectors. Finally, if it resides in the two-line-patterned area, it will dominate both solutions, since we have $f_1(x_3) < f_1(x_i)$ and $f_2(x_3) < f_2(x_i)$, where $i \in \{1, 2\}$.

Definition 2 (Pareto Optimality): A particular route-solution vector $\mathbf{f}(x_1)$ is said to be *Pareto Optimal*, if and only if, there is no route-solution vector x, which dominates $\mathbf{f}(x_1)$.

Using Definition 1, the dominance relationship for all the route-solution vectors may be extracted. This facilitates the grouping of the route-solution vectors into so-called *Pareto Fronts* (PFs) [10]. The specific points, which belong to the same PF, share the characteristic of being dominated by the same number of points. Therefore, according to Definition 2, the specific PF which contains the particular route-solution vectors that are not dominated by any other route-solution vector is the optimal one and it is hence termed as the *Optimal Pareto Front* (OPF). An example of grouping the route-solution vectors into PFs is presented in Fig. 3(a), where the route-solution vectors in F_1 form the OPF, while the ones in F_2 share the property of being dominated by exactly 3 route-solution vectors. This property of the PFs facilitates the quantitative evaluation of a specific route-solution vector leading to the OPF.

*Definition 3 (Pareto Distance):*Given a set of route-solutions S and a particular route-solution x_i, belonging to the set $x_i \in S$, its distance from the OPF may be defined as the probability P_d of being dominated by the other solutions of S. This is formally formulated as:

$$P_d(x_i) = \frac{\#\{\mathbf{f}(x_j) \succeq \mathbf{f}(x_i) \quad \forall j, \quad i \in \{0, 1, \ldots, |S| - 1\}\}}{|S|}, \quad (6)$$

where the operator $\#\{\cdot\}$ quantifies the number of times that the condition in the curly brackets is satisfied, while the operator

$|\cdot|$ represents the total number of elements of a set and $\mathbf{f}(\cdot)$ is the UF vector defined in (5).

According to Definition 3, the Pareto Distance function $P_d(\ )$ is limited to the range $[0, 1]$. Naturally, the points belonging to the OPF will have the minimum possible distance, which is equal to 0, whereas the non-optimal points would have higher distances. Therefore, the optimization problem takes the form of:

$$\begin{aligned} &\text{find} && x \\ &\text{s. t.} && x \in S,\ P_d(x) = 0. \end{aligned} \tag{7}$$

Finally, let us define the complexity metric invoked for the quantification of the computational complexity. Since the calculation of the Pareto Distance in (6) requires the invocation of the dominance operator ($\succeq$), we will define the *Cost Function* (CF) of our optimization problem as a single application of this operator between two solution vectors. The calculation of each route-solution vector requires a single UF evaluation, based on (5).

Let us now apply the principles of *Pareto Optimality* to our SON model. Based on the definition of the route-solution vector in (5), we attempt to jointly optimize the performance of our SON in terms of the time delay, the BER and the energy dissipation. Since these performance metrics conflict with each other, the OPF consists of each performance metric's global minimum and the specific routes corresponding to route solution vectors, which lie in the space defined by the global minima and they are not dominated by any other route-solution vector. Our main interest lies in determining the latter solutions rather than the three global minima. For instance, the optimal route in terms of the time delay would be the direct route from the SN to the DN without traversing through any RNs. However, this link would potentially suffer from an excessive power dissipation, since the distance between the SN and the DN may be long, potentially leading to a low E_b/N_0 as well. Similar disadvantages may apply in the general case for all the global minima of each parameter.

Furthermore, as mentioned in the previous section, determining all the OPF routes provides us with useful information about the trade-offs of the diverse parameters considered [10], hence resulting in a more beneficial design in terms of the various QoS requirements. For this reason, all the legitimate routes have to be examined in terms of their *Pareto Distance* for the sake of identifying those that have a *Pareto Distance* of zero. Assuming that $|S| = N$, where N corresponds to the total number of legitimate routes, the examination of a single route would invoke the dominance operator $N/2$ times on average and N times in the worst-case scenario. The total number of legitimate routes increases exponentially with the number of nodes N_{nodes}, and it is equal to [4]:

$$N = \sum_{i=0}^{N_{\text{nodes}}-2} \frac{(N_{\text{nodes}}-2)!}{(N_{\text{nodes}}-2-i)!}. \tag{8}$$

Since this operation is carried out for every legitimate route, the resultant average and maximum brute force (BF) complexity, $L_{\text{BF}}^{\text{avg}}$ and $L_{\text{BF}}^{\text{max}}$ respectively, are equal to :

$$\begin{aligned} L_{\text{BF}}^{\text{avg}} &= N^2/2 = O(N^2), \\ L_{\text{BF}}^{\text{max}} &= N^2 = O(N^2). \end{aligned} \tag{9}$$

Hence, sophisticated search methods are required for determining the OPF in polynomial time. As already mentioned in Section I, this ambitious goal may be achieved with the aid of QSAs.

III. FUNDAMENTALS OF QUANTUM COMPUTING

Before proceeding with the portrayal of the proposed QSA, the four postulates of quantum mechanics should be mentioned, [5]:

1) State Space: A quantum system's state is given by:

$$|\phi\rangle = \textstyle\sum_{i=0}^{M-1} \varphi_i |\phi_i\rangle = (\varphi_0, \varphi_1, \ldots, \varphi_{M-1})^T, \tag{10}$$

where the complex valued φ_i represents the amplitude of the *basis state* $|\phi_i\rangle$ and there are $M = 2^K$ basis states in total. The squared modulus $|\varphi_i|^2$ of the amplitude corresponds to the probability of observing the quantum system being in the state $|\phi_i\rangle$, the probability of observing one of the states the quantum system is superimposed in should naturally be equal to 1, as encapsulated in:

$$\sum_{i=0}^{M-1} |\varphi_i|^2 = 1. \tag{11}$$

Moreover, for the complex conjugate transpose of $|\phi\rangle$ the notation used is $\langle\phi|$, which is equal to:

$$\langle\phi| = |\phi\rangle^\dagger = (\varphi_0^*, \varphi_1^*, \ldots, \varphi_{M-1}^*). \tag{12}$$

Since the amplitudes assume complex values, the system state's argument would lie within the Hilbert space. An interesting aspect observed from (10) is that given a quantum register (QR) of L_{qr} qubits, the QR may assume all states simultaneously, which is often termed as being in the superposition of basis states. It is exactly this fact, which the quantum algorithms take advantage of and hence they are capable of carrying out operations in parallel.

2) Time Evolution: The evolution of a physical system states versus time may be characterized by a set of unitary transformations, which is formulated as:

$$|\psi\rangle = U |\phi\rangle, \tag{13}$$

where U is a unitary matrix, In other words, we have $U^{-1} = U^\dagger$, where $U^\dagger$ is the complex conjugate transpose of U. Eq. (13) stems from the Schrödinger equation [5]. Unitary matrices are linear ones and this assumption of linearity assists in preventing the occurrence of some "strange" phenomena such as *time travel* [5] due to non-linearities. Moreover, this property enables us to break down the operations encapsulated in Eq. (13) into simpler ones using *quantum gates*, hence assisting in reducing the complexity of quantum algorithms. In fact, there is a suite of components having a unitary response. Some of the most common single-qubit quantum gates which are important components of

the proposed quantum-assisted algorithm are the *Hadamard* gate H and the *Rotation* gate R_θ. Their single-qubit transfer matrices are [5]:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \; R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \quad (14)$$

The *Hadamard* gate is mainly used for mapping the ground state $|0\rangle$ to the superposition of the states $|0\rangle$ and $|1\rangle$, while the *Rotation* gate rotates the qubit state by an angle of θ. These operations are formally expressed as:

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle\,, \quad (15)$$

$$|1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle\,, \quad (16)$$

$$|0\rangle \xrightarrow{R_\theta} \cos\theta\,|0\rangle - \sin\theta\,|1\rangle\,. \quad (17)$$

Apart from these simple operations, it is possible to carry out controlled operations. A commonly used gate belonging to this family of components is the *Controlled-NOT* (CNOT) gate [6]. It performs the *Exclusive OR* (XOR) operation of its two inputs storing the output on the second qubit or QR. Thus, it is equivalent to the classic XOR gate and its function is formulated as:

$$|c\rangle\,|t\rangle \xrightarrow{\text{CNOT}} |c\rangle\,|c \oplus t\rangle\,, \quad (18)$$

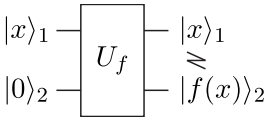

FIGURE 4. Quantum circuit implementing a specific function $f(x)$; the subscripts of the "kets" are used in order to distinguish the two input QRs and the hashed line denotes the *entanglement* between the two output QRs.

where the state $|c\rangle$ is often referred to as the *control* register, while $|t\rangle$ is the *target* register. In fact, the CNOT gate is a special case of a family of quantum gates, which are commonly known unitary operators U_f [5]. They are capable of implementing a binary function $f : \{0, 1, \ldots, N-1\} \rightarrow \{0, 1\}$ in the quantum domain. Their quantum circuit is shown in Fig. 4; due to the superposition of states of the QR $|x\rangle_1$ it is possible to carry out the function's calculations in parallel, which is the main advantage of quantum computing. Their operation may be formulated as:

$$|x\rangle_1\,|0\rangle_2 \xrightarrow{U_f} |x\rangle_1\,|0 \oplus f(x)\rangle_2 \equiv |x\rangle_1\,|f(x)\rangle_2\,. \quad (19)$$

We note that the subscripts of the "kets" are used for distinguishing the two inputs of the QRs. These unitary operator are the main component for the construction of *Quantum Oracles* [5]. Therefore, the QR $|x\rangle_1$ is often referred to as a *Quantum Index Register* (QIR), since it points to the indices of the input states, while the second input is commonly known as the *Oracle Workspace* (OW), since all the Oracle operations are carried out in this QR.

3) Measurement: Quantum measurements may be described by a vector of measurement operators $\{M_m\}$, which are applied to the state space, when the system is subjected to a "measurement" or observation. The index m indicates that the measurement's outcome will be equal to m and the probability $p(m)$ of this outcome is given upon assuming a general initial state of $|\psi\rangle$ as:

$$p(m) = |\psi\rangle\, M_m^\dagger M_m\, \langle\psi|\,, \quad (20)$$

where the state of the related qubit after the above-mentioned measurement procedure becomes:

$$|\psi'\rangle = \frac{M_m\,|\psi\rangle}{\sqrt{p(m)}}. \quad (21)$$

In the special case, where there are two basis states, namely $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$ and an arbitrary state of the qubit $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$, we may arrive at:

$$p(0) = |\psi\rangle\, M_0^\dagger M_0\, \langle\psi| = |a|^2\,, \quad (22)$$

$$p(1) = |\psi\rangle\, M_1^\dagger M_1\, \langle\psi| = |b|^2\,, \quad (23)$$

where the post-measurement state will be respectively:

$$|\psi'\rangle_{m=0} = \frac{M_0\,|\psi\rangle}{|a|} = \frac{a\,|0\rangle}{|a|}, \quad (24)$$

$$|\psi'\rangle_{m=1} = \frac{M_1\,|\psi\rangle}{|b|} = \frac{b\,|1\rangle}{|b|}. \quad (25)$$

It may be observed from (24) and (25) that the qubit collapses into a classical bit state after the measurement operation on the computational basis of $\{|0\rangle, |1\rangle\}$.

4) Composite Systems: The fourth postulate of quantum mechanics describes the state of a length K QR, which is composed by individual QRs of length 1. The resultant state can be expressed as the tensor product of the individual register states. For instance, in case of a length-2 QR, the resultant state will become:

$$|\psi\rangle = |\psi_1\rangle_1\,|\psi_2\rangle_2 \quad (26)$$

$$= (\alpha\,|0\rangle_1 + \beta\,|1\rangle_1)\,(\gamma\,|0\rangle_2 + \delta\,|1\rangle_2) \quad (27)$$

$$= \underbrace{\alpha\gamma}_{a_{00}}|00\rangle + \underbrace{\alpha\delta}_{a_{01}}|01\rangle + \underbrace{\beta\gamma}_{a_{10}}|10\rangle + \underbrace{\beta\delta}_{a_{11}}|11\rangle\,, \quad (28)$$

with

$$\sum_{\forall i,j} |a_{ij}|^2 = 1. \quad (29)$$

Some notable consequences constituted by the above postulates are the *no cloning theorem* [5] and *entaglement* [6]. The first imposes the constraint that only qubits having known and/or orthogonal states may be copied. This theorem provides security in quantum communications and it is the main idea of *Quantum Key Distribution* (QKD) [43]. The latter one expresses a linkage between single qubits of a multiple qubit state, when the multiple qubit state is an entangled one. This linkage occurs even when these single qubits are delivered to two arbitrary remote locations.

IV. NON-DOMINATED QUANTUM OPTIMIZATION ALGORITHM

A. ORACLE CONSTRUCTION

Calculating the Pareto distance for each point is a rather demanding task in terms of CF evaluations, since it would require N^2 CF evaluations, which is equal to the classic exhaustive search complexity. Although some complexity reduction is offered by the *Quantum Mean Algorithm* of [35], [36], the accuracy of the algorithm is affected. To elaborate further, assuming l qubits to be in the *Quantum Control Register* (QCR), which is used for the parallel calculation of the nominator in (6), an error $\epsilon \in O(1/l)$ [35] would be introduced and increasing l for the sake of minimizing this error would result in an excessive complexity.

This imperfection may be mitigated, if we inspect the concept of Pareto Optimality in a more meticulous manner, given in Definition 2. In fact, the calculation of the Pareto distance itself is unnecessary. In other words, a particular route would belong to the OPF, if and only if there is no other route which would dominate it. From this perspective, our problem is simplified to an existence problem and all we have to find is a route, which would dominate the examined one. If the search is unsuccessful and, consequently, there is no legitimate route which would dominate the examined one, the latter would belong to the OPF.

Our new approach would involve ''asking'' the oracle gate O whether a particular route-solution vector is dominated by the other route-solution vectors. Assuming that the particular route corresponds to the state $|i\rangle$, let us now define the oracle function $g(i, x)$, which the QSA would query as follows:

$$g(i,x) = \begin{cases} 1 & \mathbf{f}(x) \succeq \mathbf{f}(i) \\ 0 & \text{otherwise} \end{cases}. \tag{30}$$

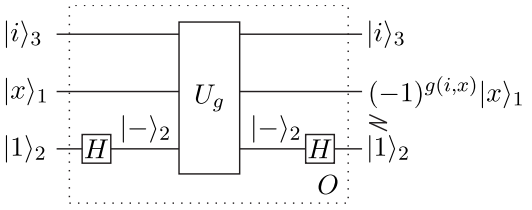

FIGURE 5. Quantum circuit of the quantum oracle gate *O*, which implements the utility binary function *g*(*i*, *x*); even though *O* would alter the phase of $|-\rangle_2$ due to the *entanglement* it would be valid to assume that $|x\rangle_1\,[(-1)^{g(i,x)}\,|-\rangle_2] = [(-1)^{g(i,x)}\,|x\rangle_1]\,|-\rangle_2$ [44]. These quantum oracle gates belong to the family of *Phase-Kickback Quantum Circuits* [6] as they alter the phase of some selected states.

It may be observed from (30) that the oracle function would require two inputs: a classic state $|i\rangle$, which would point to a specific route, and a quantum state $|x\rangle$ initialized to the equally weighted superposition of all the states. Hence, the oracle gate O implementing $g(i, x)$ would map the state $|x\rangle$ to $-|x\rangle$ if and only if the respective route dominates the examined route i. The quantum circuit of O is shown in Fig. 5 and it should be noted that the classic control input $|i\rangle_3$ corresponds to the examined route. Therefore, for each examined route it is possible to construct a database containing the dominance relationship of the rest of the routes with respect to the examined one.

Additionally, we note that a single activation of the oracle gate O would impose a single CFE, since the dominance operator would be employed only once. The function of this gate can be simulated in a classical computer. To elaborate further, simulating the oracle gate in a classical computer results in checking serially whether each of the legitimate routes dominates the reference route with index i and ''marking'' the specific routes that indeed dominate it. However, in this case the actual number of CFEs imposed would be equal to the number of the legitimate routes in the absence of QP.

B. QUANTUM SEARCH ALGORITHMS

Having constructed a database containing the specific routes which dominate each route by the application of the quantum oracle gate O, we may employ a QSA [29], [30], [44] for finding the routes that correspond to the minimum Pareto distance. A search algorithm succeeds in finding the specific index x in a database or the argument for which the function g satisfies $g(i, x) = \delta$, which is termed as the *solution*. In a search maze of size N, Grover's QSA [29] finds a solution with $\sim$ 100% probability after $O(\sqrt{N})$ database queries or function evaluations, provided that the number of solutions t is equal to $t = 1$. Hence, it achieves a quadratic reduction in the computational complexity, when compared to the optimal BF search in unsorted databases. The most recent Boyer, Brassard, Høyer and Tapp (BBHT) QSA [30] is based on a successive application of Grover's QSA and finds a solution to the search problem with $\sim$ 100% probability, even if multiple solutions exist, which is achieved without requiring *a priori* knowledge about the exact value of t. Explicitly, the BBHT-QSA succeeds in finding a solution after $4.5\sqrt{N}$ database queries in the worst-case scenario. Naturally, in both Grover's QSA and in BBHT-QSA, the value δ has to be known. However, in many communication applications where the index $x_{\min}$ minimizing the cost function f is required to be found, the exact value of $f(x_{\min})$ cannot be known until after all the possible CF values have been evaluated. As for the solution, the Dürr-Høyer Algorithm (DHA) [32] employs the BBHT-QSA multiple times and manages to find $x_{\min}$ representing the minimum entry of a database, even if the value of that particular entry is not known beforehand and also if we have $t \geq 1$. The DHA performs a minimum of $4.5\sqrt{N}$ and a maximum of $22.5\sqrt{N}$ database queries in the best-case and the worst-case scenario, respectively.

In our specific application we have to find a route that dominates the examined one in the previously constructed database. Since the particular nature of our problem, we know that the intended entry is equal to $\delta = 1$ and that multiple routes may dominate a single route. Hence, the BBHT-QSA is the most appropriate quantum algorithm for finding a solution x_s in our application. It should be noted that the DHA will also succeed in finding a solution, since we have $x_s = x_{\min}$ but it will introduce more unnecessary database queries. Let us now

proceed by introducing Grover's QSA, the BBHT-QSA and, finally, the NDQO algorithm.

1) GROVER'S QUANTUM SEARCH ALGORITHM

Grover's QSA [29] assumes having $t = 1$ solution and initially creates an equiprobable superposition of $n = \log_2 N$ qubits in the $|0\rangle$ state by passing them through a Hadamard gate H [5], which results in

$$|\psi\rangle = \sum_{x=0}^{N-1} \frac{1}{\sqrt{N}} |x\rangle . \tag{31}$$

The Grover operator $\mathcal{G} = HP_0H \cdot O$ is applied $L_{opt} = \lfloor \pi/4\sqrt{N/t} \rfloor$ successive times to the quantum state $|x\rangle$ in (31), where H is the Hadamard gate H [5], while P_0 [29] is a unitary operator that maps $|x\rangle$ to $-|x\rangle$ if and only if we have $|x\rangle \neq |0\rangle^{\otimes n}$ and leaves the state $|x\rangle = |0\rangle^{\otimes n}$ unaltered. Finally, O is the Oracle Gate [29], which is a quantum circuit that considers the database's entries in parallel and maps $|x\rangle$ to $-|x\rangle$ if and only if the entry with index x is equal to δ. When the final state of the system is measured after L_{opt} iterations $|\psi_f\rangle = \mathcal{G}^{L_{opt}} |\psi\rangle$, the probability P_s of finding a solution x_s is equal to [30]:

$$P_s = \sin^2 \left[(2L_{opt} + 1)\, \theta \right], \tag{32}$$

where we have $\theta = \arcsin \sqrt{t/N}$.

2) BBHT QUANTUM SEARCH ALGORITHM

The number of solutions t of a problem does affect the number L_{opt} of optimal Grover iterations. Since the BBHT-QSA [29], [36] assumes having no *a priori* knowledge about the number of solutions S in the database, it employs Grover's operator a pseudo-random number of consecutive times in a structured way. In our routing application we have $\delta = 1$, but we are unaware of the number of routes t that dominate the i-th route. The BBHT-QSA applied in our system is formally stated in Algorithm 1 [36]. The BBHT-QSA applies the Grover operator L consecutive times to the initial equiprobable superposition of states in (31) (Step 1.4[3]) and then measures the resultant QR $|x_f\rangle$ (Step 1.5). The extraction of the database's entry that corresponds to the observed index will verify whether the observed state $|j\rangle$ is a solution or not (Step 1.8). If the latter case is true, the process described is repeated after an update of the parameters (Steps 1.11–1.16) until a solution is found or a predetermined maximum number $L_{BBHT}^{QD,\ \max}$ of affordable Grover iterations has been reached. The algorithm keeps track of the total number L_{BBHT}^{QD} of CF evaluations in the *Quantum Domain* (QD) and the total number L_{BBHT}^{CD} of CF evaluations in the *Classical Domain* (CD) (Step 1.7). Naturally, the QD CFEs would be increased from the total number of Grover iterations L with each BBHT-QSA iteration, and the number of CD CFEs would are based on the check of Step 1.8. The function implemented by the Oracle's action is invoked in the CD.

[3]The notation refers to Step 4 of Algorithm 1.

Algorithm 1: BBHT-QSA in NDQO Algorithm

1: Import reference route index i.
2: Set $m \leftarrow 1$, $\lambda \leftarrow 6/5$ and $L_{BBHT}^{QD} \leftarrow 0$, $L_{BBHT}^{CD} \leftarrow 0$.
3: Choose L uniformly from the set $\{0, \ldots, \lfloor m \rfloor\}$.
4: Apply the $\mathcal{G}$ operator L times starting from the initial state $|\psi\rangle$ in (31), resulting in the final state $|x_f\rangle = \mathcal{G}^L |\psi\rangle$.
5: Observe $|x_f\rangle$ in the QD and obtain $|j\rangle$.
6: Compute $g(i, j)$ in the CD.
7: Update $L_{BBHT}^{CD} \leftarrow L_{BBHT}^{CD} + 1$ and $L_{BBHT}^{QD} \leftarrow L_{BBHT}^{QD} + L$.
8: **if** $g(i, j) = \delta = 1$ or $L_{BBHT}^{QD} \geq L_{BBHT}^{QD,\ \max}$ **then**
9: Set $x_s \leftarrow j$, output x_s, L_{BBHT}^{CD}, L_{BBHT}^{QD} and exit.
10: **else**
11: Set $m \leftarrow \min \left\{ \lambda m, \sqrt{N} \right\}$.
12: **if** $m = \sqrt{N}$ **then**
13: Choose L uniformly from the set $\{1, \ldots, \lfloor m \rfloor\}$ and go to step 4.
14: **else**
15: Go to step 3.
16: **end if**
17: **end if**

An improvement the original BBHT-QSA in [30] is proposed, as detailed in Section V. To elaborate briefly, as soon as the upper limit m of the range, from which the number of Grover iterations is selected, becomes higher than or equal to $\sqrt{N}$ (Step 1.11), i.e. higher than the value which corresponds to the worst case scenario of having only a single solution, a bias is imposed on it for excluding the value 0 from the range (Step 1.13). This stage of the BBHT-QSA is often referred as the *critical stage* [30], since there is at least 25% probability of finding a solution, as long as there exists one.

By applying the BBHT-QSA to the database constructed by the oracle gate O of Eq. (30), we will be able to identify a route, if there exists one, that would dominate the input one with $\sim$ 100% probability, while using a number of $O(\sqrt{N})$ database queries. If there is no route that dominates the input one, the BBHT-QSA will output a random route $x_{s,r}$ from the search database, which could be disregarded with a simple final check. This check, which is similar to the first check of Step 1.8, may be invoked after the completion of the BBHT-QSA process for identifying at the expense of a single CFE as to whether the index exported from the BBHT-QSA corresponds to a valid route-solution, i.e. whether the condition $g(i, x_{s,r}) = \delta = 1$ is satisfied. Therefore, it is possible to avoid the identification of false solutions stemming from a potential BBHT-QSA QD-CFE time-out.

3) NDQO ALGORITHM

Having defined the core procedure of finding a route, which dominates the examined one, we may now proceed to the presentation of our proposed approach. We will exploit the observation [29] that if there is no solution for which we have $\delta = 1$, then O will mark no solutions and a single application of the $\mathcal{G}$ operator will leave the probabilities of the routes unaltered. Consequently, as the BBHT-QSA will not find any

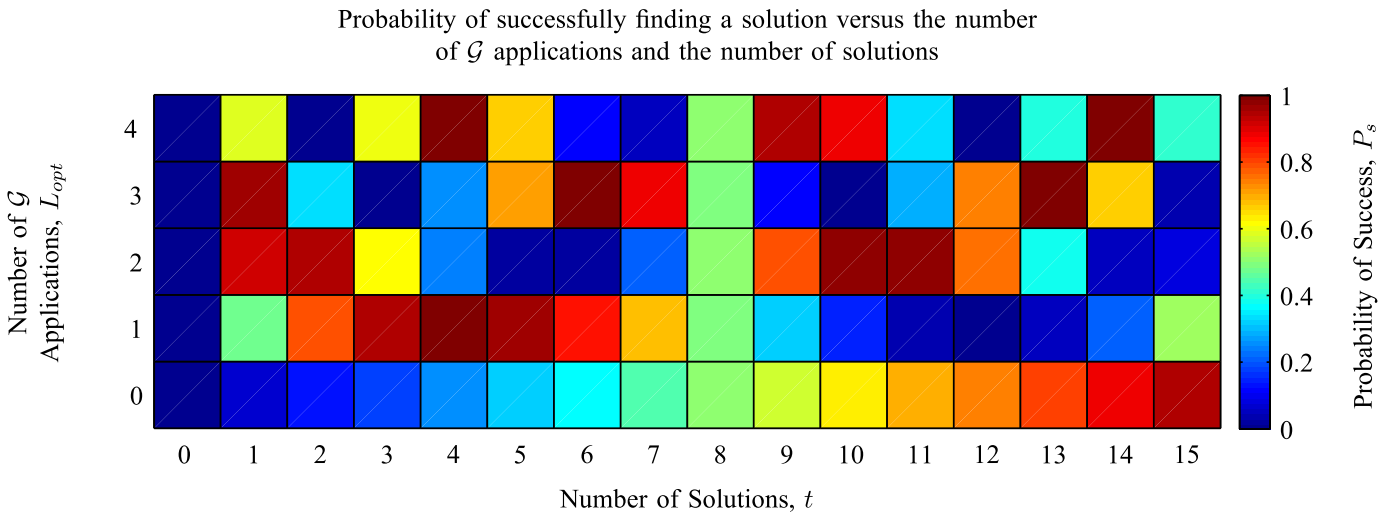

FIGURE 6. Probability P_s of successfully finding a solution versus the number of solutions t and the number of Grover iterations L_{opt} for the database of Table 2, based on Eq. (32). The number of Grover iterations L_{opt} is varied in the range $\{0, 1, .., \sqrt{N}\}$; the upper bound of the range corresponds to the optimal number of Grover iterations, where only a single solution is available. In our example, we have $\sqrt{N} = \sqrt{16} = 4$.

Algorithm 2 NDQO Algorithm

```
Initialize solution flag vector, F, to zero.
Initialize OPF = ∅.
for i = 0 to N − 1 do
    if F_i = 0 then
        if |OPF| > L_BBHT^max or ∄j ∈ OPF : f(j) ⪯ f(i) then
            Set l ← i.
            repeat
                Set k ← l.
                Define the oracle function g(k, x) from (30).
                Invoke the BBHT-QSA with input g(k, x) and
                output x_s.
                Set l ← x_s and F_k ← 1.
            until f(l) ⋡ f(k).
            Append x_k into the OPF.
        end if
    end if
end for
Output the OPF and exit.
```

solutions, it will reach its time-out after $L_{BBHT}^{\max} = \lfloor 4.5\sqrt{N} \rfloor$ Grover iterations and output a random route from the total set of routes chosen randomly. At this point, let us check whether the output of the BBHT-QSA dominates i; the i-th route will belong to the OPF if and only if the check outcome is false, i.e. we have $g(i, x_s) = 0$, which implies that there is no solution dominating it. Otherwise, if the BBHT-QSA outcome dominates the solution examined, we can proceed with checking whether the outcome is optimal. This action may be repeated until an OPF route-solution is extracted, which would then terminate this *chain of BBHT activations*. Having extracted either a single or multiple route-solutions from the OPF, it becomes possible to avoid an excessive number of CFEs by checking whether the solution examined is dominated by the OPF generated so far. Naturally, the solutions that are dominated by the OPF generated so far have a high probability of outputting an already generated OPF point, which would imply having unnecessary *BBHT chain* activations.

Having provided all the necessary discussions concerning the NDQO subroutines, we may now proceed to its detailed description relying on Algorithm 2. We will define a binary check flag vector $\mathcal{F}$ which would indicate whether a specific route has already been processed and it is initialized to a vector of zeros. Then, for each route we will check whether it has already been considered. In this case, if the specific route is not dominated by the already generated OPF, then a *BBHT-QSA chain* will be activated relying on the examined route as its initial input. The BBHT-QSA chain will be then terminated, when an OPF route-solution is examined and the flags of the routes processed by the chain are set equal to 1. This procedure is repeated until all the legitimate routes have been processed either by the OPF dominance check or by the BBHT-QSA chain and, thus, all the OPF route-solutions are exported. The goal of this procedure is to extract possible the entire OPF as promptly as possible.

C. A DETAILED 5-NODE EXAMPLE

Let us now provide an illustrative example portraying the main concepts of our proposed algorithm. We will consider a 5-node network obeying the architecture of in Fig. 7(a) along with the respective node interference imposed. The legitimate route-solutions produced by this setup along with their UFs are shown in Table 2. We note that only the BER and the *CL* are taken into account for simplifying the solutions' graphical representation, which is shown in Fig. 7(b). Moreover, the actual routes are assumed to be stored in a list in ascending lexicographical order. This routing operation may be implemented using the *Lehmer Encoding/Decoding* technique of [45]. Furthermore, we present in Fig. 6 the probability P_s of successfully finding a route-solution, when using Grover's QSA, versus the number of solutions t present and the number of $\mathcal{G}$ applications for our 5-node SON routing problem example of Fig. 7(a). The respective probabilities have been calculated using Eq. (32).

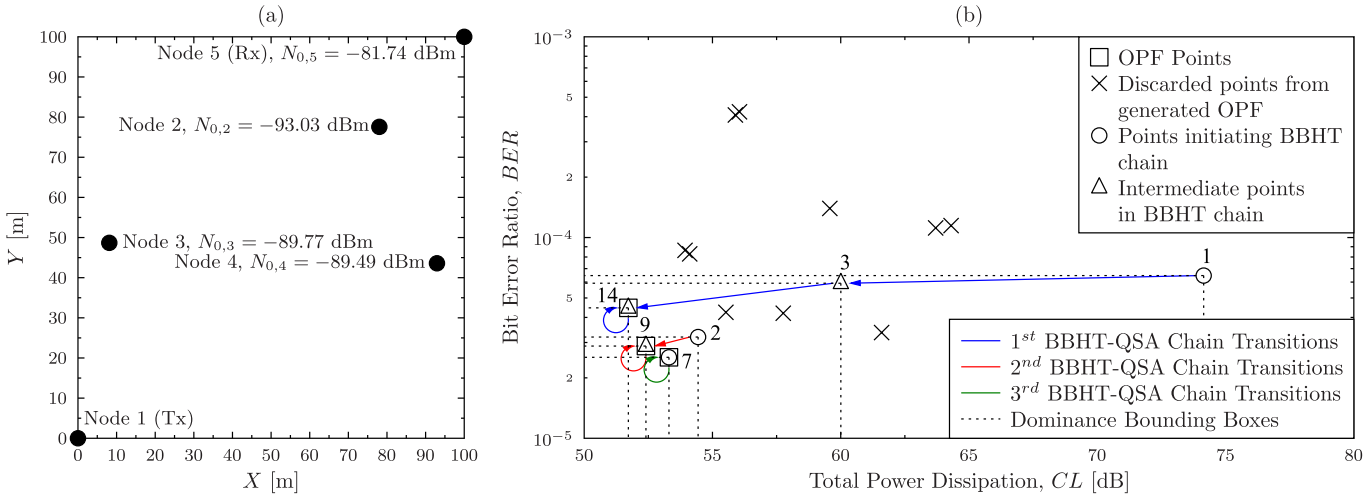

FIGURE 7. (a) Exemplified architecture for a 5-node SON, and (b) its optimization process using the NDQO algorithm. In this example only two UF are used per solution for the sake of simplicity. The routes that belong to the OPF are noted with a square marker (□), the routes that have already been processed as intermediate points in BBHT-QSA chains and will be skipped in the serial parsing step (Step 2.4) are marked with a triangle (△), whereas those that have not been processed and, at the same time, are not dominated by the generated OPF initiating a BBHT-QSA chain are marked with a circle (○). Moreover, the route-solutions, which have not already been processed but are dominated by the hitherto generated OPF and thus they will be skipped, are marked with a cross (×). Moreover, the indices of the routes as shown in Table 2 are marked in (b). Finally, the round arrows in (b) denote that a BBHT-QSA has been activated with input the respective point but in the absence of potential route-solutions a random route is output by the BBHT-QSA, classifying the input route-solution as Pareto Optimal (Step 2.13). The current problem solution is not the unique one; different solutions could be derived depending on the BBHT-QSA chain intermediate outcomes.

TABLE 2. Routes along with their UFs and indices for the exemplified 5-node SON of Fig. 7.

Index i	Route	$P_{e,i}$ ($\times 10^{-4}$)	CL_i [dB]	Index i	Route	$P_{e,i}$($\times 10^{-4}$)	CL_i [dB]
1	{1 5}	0.646	74.147	9	{1 4 2 5}	0.288	52.407
2	{1 2 5}	0.319	54.440	10	{1 4 3 5}	1.147	64.302
3	{1 3 5}	0.592	60.004	11	{1 2 3 4 5}	1.397	59.575
4	{1 4 5}	0.336	61.593	12	{1 2 4 3 5}	4.230	56.053
5	{1 2 3 5}	1.117	63.700	13	{1 3 2 4 5}	0.829	54.113
6	{1 2 4 5}	0.424	55.524	14	{1 3 4 2 5}	0.446	51.721
7	{1 3 2 5}	0.253	53.304	15	{1 4 2 3 5}	4.079	55.893
8	{1 3 4 5}	0.420	57.759	16	{1 4 3 2 5}	0.863	53.931

Moreover, all the steps carried out by the NDQO algorithm for exporting the OPF are shown in Fig. 7(b), where the solution[4] transitions that are facilitated by the BBHT-QSA are represented by the arrows. Moreover, the routes that belong to the OPF are indicated by a square marker (□). Furthermore, the points that have already been processed as intermediate points in the BBHT-QSA chains and hence will be skipped during the serial parsing step (Step 2.4) are marked by a triangle (△). Still referring to Fig. 7(b), those points that have not been processed and, at the same time, are not dominated by the generated OPF initiating a BBHT-QSA chain are marked with a circle (○). Additionally, the points that have not yet been processed but are dominated by the generated OPF and thus are skipped are marked by a cross (×). Moreover, the transitions that are carried out by the BBHT-QSA chains of Steps 2.6 – 2.11 are indicated by arrows. Additionally, arrows of different color has been used for each BBHT-QSA chain.

[4]We define a solution as an output route of the BBHT-QSA that dominates the input one.

Let us now proceed with a more detailed description of the NDQO algorithm.

The algorithm will initialize the binary check flag vector $\mathcal{F}$ to a vector of zeros and the OPF to an empty set ($\varnothing$), according to Step 2.2. Then, the 1st route of Table 2, {1 5}, is checked. Since its flag value is equal to zero and the OPF set is empty, the BBHT-QSA process of Alg. 1 is initiated with this specific route as its input. The legitimate successful outputs of the BBHT-QSA, i.e. the solutions that dominate {1 5}, are located within the rectangle of Fig. 7(b) which has the solution argument and the zero point of coordinates as its opposite corners, as indicated by the doted lines. Our database length is equal to $N = 16$, while the number of present solutions,, which are located within the aforementioned rectangle, is equal to $t = 7$. Each of these solutions will have the same probability of becoming the single output. According to Fig. 6, the optimal number of $\mathcal{G}$ applications would be for $L_{opt} = 3$, which gives a 88.45% probability of finding a route-solution that dominates the direct route. The BBHT-QSA process initializes the upper bound of the

L_{opt} selection range to $m = 1$ (Step 1.2) and the specific L_{opt} value is chosen from the set $\{0, 1\}$. Assuming that $L_{opt} = 0$ is chosen, based on Eq. (32), the probability of successfully finding a solution would be equal to $P_s = 43.75\%$. The quantum algorithm then observes its QIR (Step 1.5) and outputs $j = 13$, i.e the route {1 4 3 2 5}. A check is then performed whether the 13th route dominates the direct one (Step 1.8), which is unsuccessful, since $P_{e,1} < P_{e,13}$ and $CL_1 < CL_{13}$ and hence $g(1, 13) = 0$. Then parameter m is increased to $m = \lambda m = 6/5$ (Step 1.11) and the upper bound is modified to $\lfloor m \rfloor = \lfloor 6/5 \rfloor = 1$, while the range remains the same as in the previous iteration, i.e. the parameter L_{opt} will be selected from the set $\{0, 1\}$. At this point, assuming that $L_{opt} = 1$ is chosen, the probability of successfully finding a route-solution that dominates the input one would be equal to $P_s = 68.36\%$. Let us assume that the output of the BBHT-QSA is the 3rd route ($i = 3$), {1 3 5}, of Table 2. Once again, this route will be checked whether it dominates the input route and since it lies within the rectangle of Fig. 7(b), it will indeed dominate it. The transition between the direct route and the third one is noted using a blue coloured arrow in Fig. 7(b).

Then, the input solution flag value is set to $\mathcal{F}_1 = 1$ and hence a new BBHT-QSA process is invoked with the output of the previous BBHT-QSA as its input. Again, the successful BBHT-QSA outputs will be located within the rectangle defined with the new reference point and the center of coordinate axes in its opposite corners; after the completion of the BBHT-QSA of Alg. 1, the flag value of the input route is set to one, i.e. to $\mathcal{F}_3 = 1$. Following a similar procedure as in the first BBHT-QSA iteration, the algorithm outputs the 14th route, {1 3 4 2 5}, of Fig 7(b) which dominates the input and happens to be a Pareto Optimal solution. A BBHT-QSA will be invoked with this route as its input and since no solutions would exist which dominate the new reference, the BBHT-QSA will reach its time-out after at a minimum of $L_{BBHT}^{QD,\ \max}$ Grover operator $\mathcal{G}$ applications and it will hence output a random route-solution selected from the set of all the legitimate ones. Upon applying the dominance operator to the BBHT-QSA output, its response will become false, which would indicate that no point dominating the input route exists, which, in turn, would imply that the route belongs to the OPF. Hence, the route with index $i = 14$ in Fig. 7(b) is appended to the OPF. Moreover, its flag value will be set to $\mathcal{F}_{14} = 1$. This transition is noted in marked Fig. 7(b) with the blue round arrow, since the NDQO algorithm will return to the input route-solution and classify it as optimal.

Then, the second route with index $i = 2$ of Table 2 will be checked (Step 2.3). Since we have $\mathcal{F}_2 = 0$ and this route is not dominated by the route with index $i = 14$ (Steps 2.4 and 2.5), the BBHT-QSA process of Alg. 1 will be initiated with the second route as its input. The possible successful outcomes of the BBHT-QSA would be the 7th ($i = 7$) and the 9th ($i = 9$) routes with 50% probability, both of which happen to be Pareto Optimal. Following the same process as the very first BBHT-QSA activation, we may now assume that the 9th route is the output of the BBHT-QSA, and since it will indeed dominate the input route, the flag value of the input route is toggled to $\mathcal{F}_2 = 1$. The respective transition is noted in Fig. 7(b) using a red arrow. A new BBHT-QSA process will be invoked with the 9th route of Fig. 7(b) as its input, which would exhaust the maximum number of $\mathcal{G}$ applications, therefore outputting a random solution from the solution space. Hence, the 9th route is incorporated into the OPF and its flag value will be modified accordingly. This operation corresponds to the round red arrow seen in Fig. 7(b). Afterwards, the 3rd route will be skipped, because we have $\mathcal{F}_3 = 1$. Additionally, the rest of the routes until the 7th route of Fig. 7(b) will be discarded as they are dominated by the 9th route and, thus, are not Pareto Optimal (Step 2.5). As for the 7th route, since it has not been processed ($\mathcal{F}_7 = 0$) and will not be dominated by any solution due to the fact that it is Pareto Optimal, the BBHT-QSA process of Alg. 1 will be invoked with its input given by this route. This process will exhaust the maximum number of $\mathcal{G}$ applications and will hence output a random solution, thus leading to appending the 7th route-solution to the OPF. Moving back to Fig. 7(b), this step corresponds to the round green arrow. Finally, the rest of the solutions will be discarded, since albeit they have not been processed, they will be dominated by the OPF generated.

In the exemplified description of the NDQO algorithm we have not mentioned the CF evaluations. Thus, the complexity the BBHT-QSA will be discussed in the next section, where a variety performance metrics will be introduced.

D. BENCHMARKING ALGORITHMS

Apart from the classical BF method, we will use two additional benchmarking algorithms, namely the NSGA-II [4], [39] and the ACO algorithm [17]. As far as the first is concerned, we have used the exact the algorithm proposed in [4] by Yetgin *et al.*. The only deviation from approach of [4] was the fact that the number of generations involved was set to the population of the parents and the children at each generation. For the ACO algorithm, a similar approach to that of [15] was followed. To elaborate further, the *intrinsic affinity* matrix was extracted by using a Non-Dominated Sort (NDS) across all the legitimate node-transitions for each RN and the SN, exporting the number of node-transitions $N_{i,j}^{dom}$ in the network, which dominate a specific node-transition $v_{i,j}$, as it was defined in (4). As for the intrinsic affinity $\eta_{i,j}$ of each transition, we followed a similar approach to [46] and it would be equal to:

$$\eta_{i,j} = e^{-N_{i,j}^{dom}}. \tag{33}$$

As for the pheromone update also defined in [47], multi-level pheromones were used, since the concept of Pareto Optimality was relied upon. However, they we were combined into a single-level pheromone for exporting the probabilities of the node-transitions circumventing the need for aggregation of the objectives. Once again, an NDS was used across all the possible transitions exporting the number of transitions

$N_{i,j}^{\tau,dom}$, which dominate a specific multilevel pheromone $\tau_{i,j}$. Consequently, the combined pheromone will be equal to:

$$T_{i,j} = e^{-N_{i,j}^{\tau,dom}}. \tag{34}$$

Then, the transition probability $P_{i,j}$ will become [18]:

$$P_{i,j} = \sum_{k=0}^{N_{\text{nodes}}-1} \frac{\left(T_{i,j}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}}{\left(T_{i,k}\right)^{\alpha}\left(\eta_{i,k}\right)^{\beta}}, \tag{35}$$

where α corresponds to the combined pheromone weight and β is the intrinsic affinity weight. In both algorithms, a classical NDS is used at the end of each generation in order to extract the OPF. Assuming that the NSGA-II is confined for N_G generations and N_{pop} individuals in the initial population and the ACO is initialized to have Ξ generations and ζ ants, their complexities will be:

$$L_{NSGA-II} = N_G\,(2N_{pop})^2, \tag{36}$$

$$L_{ACO} = \Xi\,\zeta^2. \tag{37}$$

For the sake of simplicity, we will set the number of NSGA-II individuals evaluated by the end of each generation equal to the number of generations, i.e. we rely on $N_G = 2N_{pop}$ and $\Xi = \zeta$. Assuming now that the maximum complexity of the NDQO algorithm is $\max\left\{L_{CFE}^{tot}\right\}$, in order to match the complexities of the NDQO, the ACO and the NSGA-II will rely on:

$$N_G = \Xi = \sqrt[3]{\max\left\{L_{CFE}^{tot}\right\}}. \tag{38}$$

Finally, the input parameters of the NSGA-II and the ACO used in our comparative case study are shown in Table 3.

V. COMPUTATIONAL ACCURACY VERSUS COMPLEXITY

In this section, we will provide simulation results concerning the accuracy of the NDQO algorithm versus its complexity. Before delving into the presentation of the results, the complexity of the classical BF algorithm should be quantified. In the classical domain, each route is compared to all the other legitimate routes for determining whether it is dominated by any solutions. According to Eq. (9), this would impose a complexity on the order of $O(N^2)$. However, this operation, which we will refer to as *naive-BF*, would find the OPF route-solutions along with sorting all legitimate routes into PFs. The latter operation is not performed by our proposed algorithm, making their comparison rather unfair.

For the sake of fairness, we will also use another version of the BF which exports only the OPF, as stated formally in Alg. 3. This is the same as the NDQO algorithm stated in Alg. 2. The only difference would be that a BF serial search is used instead of the BBHT-QSA chains for identifying, whether a solution is optimal. The actual complexity of this BF method is random and would explicitly depend on the number of OPF route-solutions and on their order of appearance in the solution space. Therefore, due to the complex structure of the SONs examined we will derive its complexity

Algorithm 3 BF Method

1: Initialize $OPF = \varnothing$.
2: **for** $i = 0$ **to** $N-1$ **do**
3: Set $f \leftarrow 0$
4: **if** $\nexists j \in OPF : \mathbf{f}(j) \succeq \mathbf{f}(i)$ **then**
5: **for** $k = 0$ **to** $N-1$ **do**
6: **if** $\mathbf{f}(k) \succeq \mathbf{f}(i)$ **then**
7: Set $f \leftarrow 1$ and terminate inner loop.
8: **end if**
9: **end for**
10: **if** $f = 0$ **then**
11: Append i into the OPF.
12: **end if**
13: **end if**
14: **end for**
15: Output the OPF and exit.

using *Monte Carlo* simulations and compare it to the respective complexity of the NDQO algorithm for the same SON setups. Moreover, both the upper and the lower bounds of this BF method may be derived. The upper bound corresponds to the rather unrealistic case, where all the legitimate routes are optimal. In this case, the direct route would require N CFEs for its identification as an optimal route-solution, the second route in the database will require in turn $(N+1)$ CFEs, since it will be checked against the OPF generated, which consists of a single solution. Finally, the last one will be checked against all the solutions forming part of the OPF, which would consist of $(N-1)$ route-solutions and another N CFEs would be required for its identification as an optimal route-solution. Therefore, the resultant maximum complexity may be derived by exploiting the following property of the sum of arithmetic series as:

$$L_{BF}^{\max} = N^2 + \sum_{i=0}^{N-1} i = N^2 + \frac{N}{2}(N-1) = \frac{3}{2}N^2 - \frac{1}{2}N. \tag{39}$$

Hence, the upper bound of the BF complexity is still on the order of $O(N^2)$. On the other hand, assuming that there is only a single optimal path, which happens to be the first in our database, namely the direct route, the BF method would require N CFEs for classifying this route as an optimal one (Steps 3.5-3.12) and another $(N-1)$ CFEs for classifying the rest of the routes as suboptimal. As a result, the lower bound of the BF complexity is equal to:

$$L_{BF}^{\min} = 2N - 1 = O(N). \tag{40}$$

Therefore, the BF method would involve a complexity on the order of $O(N)$ and $O(N^2)$ for the best- and the worst-case scenario, respectively.

Additionally, we note that the simulation results presented in this section have been generated using the *Monte Carlo* simulation method and they have been averaged over 10^8 runs. Finally, since we had no quantum computer at our disposal, the simulations of the QSAs were carried out using a classical cluster. Explicitly, since the quantum Oracle O calculates in parallel the UF vectors of all the legitimate routes in the QD, they were pre-calculated. We note that

TABLE 3. Input Parameters of the Benchmarking Algorithms.

NSGA-II [4], [39]		ACO [15], [18]	
Number of Generations, N_G	$\sqrt[3]{\max\{L_{CFE}^{tot}\}}$	Number of Generations, Ξ	$\sqrt[3]{\max\{L_{CFE}^{tot}\}}$
Initial Population N_{pop}	$\sqrt[3]{\max\{L_{CFE}^{tot}\}}/2$	Number of Ants, ζ	$\sqrt[3]{\max\{L_{CFE}^{tot}\}}$
Mutation Probability, P_m	0.5	Pheromone weighting, α	1.2
Crossover Probability, P_c	0.9	Intrinsic Affinity weighting, β	0.6

this results in an actual complexity higher than that of the BF method. Therefore, the deployment of the NDQO in a quantum computer is essential for observing a complexity reduction stemming from the QP. Hence, in our simulations, we have made the assumption of employing a quantum computer for our algorithm and we count the total number of O-activations for quantifying the NDQO's complexity. This number would be the same for both classical and quantum implementations. Let us now proceed by characterizing the complexity of the NDQO algorithm.

A. NDQO COMPLEXITY PERFORMANCE

Within the BBHT-QSA iterations, each quantum oracle application would result in a single QD-CFE, whereas the validation check of Step 1.4 would require a single CD-CFE. We note that the complexity of a single CD-CFE and of a single QD-CFE are assumed to be identical for simplicity. The total complexity L_{CFE}^{tot} may be derived as the sum of the number of $\mathcal{G}$ applications L_{NDQO}^{QD}, that of the total classic comparison activations within the BBHT-QSA iterations L_{NDQO}^{CD} and that of the comparisons with the route-solutions of the already generated OPF $L_{\text{NDQO}}^{\text{OPF}}$, yielding:

$$L_{CFE}^{tot} = L_{\text{NDQO}}^{QD} + L_{\text{NDQO}}^{CD} + L_{\text{NDQO}}^{\text{OPF}}. \tag{41}$$

Therefore, in order to derive both the upper and lower bounds of complexity in terms of the number of CFEs in both the quantum and classic domains, we have to consider two extreme cases, which are identical to the ones considered for the BF method. For the lower bound, we will assume that the optimization problem has only a single solution, which happens to be the first route in the solution database, namely the direct route. The NDQO algorithm will exhaust the maximum affordable complexity of $L_{BBHT}^{QD,\ \max} = \left\lfloor 4.5\sqrt{N} \right\rfloor$ for the first route and the rest of the routes will be discarded, since they will be dominated by the first one. Since we are examining the lower bound, each of the terms in the sum of (41) needs to be minimized. In terms of L_{NDQO}^{QD}, only the first route will invoke the BBHT-QSA process which will reach the maximum number of $\mathcal{G}$ applications and the lowest possible number would be:

$$L_{\text{NDQO}}^{QD,\ \min} = \left\lfloor 4.5\sqrt{N} \right\rfloor + 1 > 4.5\sqrt{N}. \tag{42}$$

Hence, the minimum number of the $\mathcal{G}$ applications would be for $L_{\text{NDQO}}^{QD,\ \min} = 4.5\sqrt{N}$. As for the CD-CFEs, we could can a greedy approach in order to find the associated minimum value: we may assume that the maximum number of $\mathcal{G}$ iterations $\lceil m \rceil$ is selected in Step 1.2. In this way, the maximum number of $\mathcal{G}$ applications will be reached with as few CD-CFEs as possible. Under this perspective, we would get:

$$\sum_{i=0}^{L_{\text{NDQO}}^{QD,\min}-1} \lambda^i m \geq L_{BBHT}^{QD,\ \max} \equiv 4.5\sqrt{N}, \tag{43}$$

where λ and m are the BBHT-QSA initialization parameters. Therefore, in order to find the minimum value, all that has to be done is to solve Eq. (43) in terms of $N_{QD}^{\min}$, yielding:

$$L_{\text{NDQO}}^{CD,\min} = \log_\lambda \left(4.5\, \frac{\lambda - 1}{m}\, \sqrt{N} + 1 \right) + 1. \tag{44}$$

As for the comparisons with the hitherto generated OPF, all the routes except for the first one will be dominated by the direct route leading to $L_{\text{NDQO}}^{\text{OPF},\ \min} = N - 1$ and hence all but the first routes will be discarded. Finally, the lower bound of the NDQO algorithm's complexity may be expressed as:

$$\begin{aligned} L_{CFE}^{tot,\min} &= 4.5\sqrt{N} + \log_\lambda \left(4.5\, \frac{\lambda - 1}{m}\, \sqrt{N} + 1 \right) + N \\ &= O(N). \end{aligned} \tag{45}$$

Consequently, the lower bound complexity of the NDQO algorithm is on the order of $O(N)$ providing a quadratic speed-up down from $O(N^2)$.

As far as the complexity upper bound is concerned, we will consider the extreme case, where all the routes are optimal. Naturally, having that many route-solutions in the OPF would result in excessive an excessive number of CFEs in the sub-process of the NDQO algorithm, where a particular route-solution is examined to ascertain whether it is dominated by the OPF generated. Hence, a restriction should be imposed on the grounds that this process should not exceed the BBHT-QSA maximum number of activations. To elaborate further, since the BBHT-QSA involves in the worst case $4.5\sqrt{N}$ CFEs in the quantum domain, it is reasonable to impose an upper bound also on the number of the classic domain CFEs, which may be set to the number of comparisons with the generated OPF for maintaining the number of classic CF evaluations. In our approach this upper bound was set to the BBHT-QSA time-out of $L_{BBHT}^{QD,\ \max} = 4.5\sqrt{N}$. Therefore, if the length of the generated OPF is higher than or equal to the BBHT-QSA time-out, the examined solution will not be compared to the generated OPF and a BBHT-QSA chain will be invoked directly, since it would involve fewer CF evaluations. Under this perspective, the number of CD-CFEs

due to OPF comparisons would be upper bounded by:

$$L_{NDQO}^{\text{OPF, max}} = \sum_{i=1}^{4.5\sqrt{N}} i = 2.25\sqrt{N}\left(1 + 4.5\sqrt{N}\right), \quad (46)$$

involves a complexity on the order of $O(N)$. As for the BBHT-QSA, an additional restriction should be imposed. There exists an extreme case, when only zero $\mathcal{G}$ applications are "chosen" to be applied. In this case, since the time-out is quantified in terms of the number of oracle queries, there is an extremely low probability that the algorithm will fall into an infinite loop, where $L = 0$ is continuously chosen in Step 1.2. This event would yield an upper bound of infinity. In fact, the effect of this exceptional case could be mitigated. Upon reaching the *critical stage*[5] [30] of the algorithm, a valid solution would be output with a probability equal to 25%. Hence, it may seem reasonable to exclude from the range the specific event of "choosing" 0 $\mathcal{G}$ applications for avoiding the infinite loop trap. Additionally, the probability of success upon reaching the *critical stage* will remain unaltered. Under this perspective, the worst case scenario, as far as the CD-CFEs of the inner BBHT-QSA iterations are concerned, would be to "choose" $L = 0$ in Step 1.3, until the *critical stage* (condition in Step 1.12) is reached and where $L = 1$ is selected in Step 1.13 $4.5\sqrt{N} - 1$ times, or in other words until we are a single QD-CFE away from the time-out condition, and at $\sqrt{N}$ $\mathcal{G}$ applications during the last iteration, following a greedy approach. Consequently, this operation will be repeated for each solution yielding a complexity of:

$$L_{\text{NDQO}}^{CD,\ \text{max}} = N\left(\log_{\lambda}\sqrt{N} + 4.5\sqrt{N}\right), \quad (47)$$

$$L_{\text{NDQO}}^{QD,\ \text{max}} = N\left(5.5\sqrt{N} - 1\right). \quad (48)$$

Finally, the upper bound of the NDQO algorithm's complexity may be quantified by substituting Eqs. (46-48) into (41), yielding:

$$\begin{aligned} L_{CFE}^{tot,\text{max}} &= 9.5N\sqrt{N} + N\log_{\lambda}\sqrt{N} + 9.125N + 2.25\sqrt{N} \\ &= O(N\sqrt{N}) \\ &= O(N^{3/2}). \end{aligned} \quad (49)$$

Therefore, the upper bound would involve a complexity on the order of $O(N^{3/2})$, which is still lower than that of the classical BF, which is $O(N^2)$.

The average complexity of the NDQO algorithm $E\left[L_{CFE}^{tot}\right]$ is shown in Fig. 8 for SONs consisting of $N_{nodes} = 2$ to $N_{nodes} = 9$ nodes. These average complexities are also compared to both the upper and the lower bound of (49) and (45) respectively, as well as to the respective complexity of the classical BF algorithm, as they were derived in Eqs. (39) and (39). The average complexity of the NDQO algorithm is presented in Fig. 8 with the aid of boxes surrounding the bold dots and having different aspect ratios in order to portray its stochastic nature. Explicitly, the upper and lower edges of

[5]The *critical stage* will be reached at exactly $\left\lceil \log_{\lambda}\sqrt{N} \right\rceil$ BBHT-QSA iterations [30].

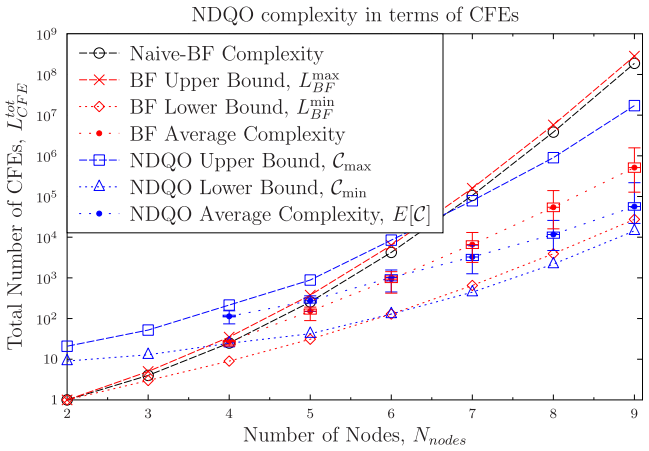

FIGURE 8. Evolution of complexity of the NDQO algorithm compared to the respective complexity imposed by the BF method of Alg. 3; the mean number of CF evaluations is shown along with the upper and the lower bounds, as they were derived in (45) and (49) and they are compared to the naive-BF complexity and upper and lower bounds of the BF method of Alg. 3, based on Eqs. (39) and (40) respectively. Both the NDQO algorithm and the BF method average complexities are presented using box plots; the upper and lower bounds of the boxes correspond to the 75% and 25% quartiles respectively. In addition, maximum and minimum observed complexity values are presented using horizontal lines. The mean complexity results have been averaged over 10^8 runs.

the box boundaries at each SON size represent the 25% and 75% quartiles, while the vertical bars correspond to the maximum and minimum value of L_{CFE}^{tot} found by our simulations. Observe in Fig. 8 that, naturally, the interquartile distance will increase as the number of SON nodes increases. In fact, the increase in the total number of routes would entail a higher variation in the number of routes belonging to the OPF yielding a higher variation in the number of BBHT-QSA time-outs needed in order to check the entire solution space. Additionally, the average NDQO complexity tends to be closer to its upper bound for SONs up to $N_{\text{nodes}} = 5$ nodes. This could be justified by the fact that, since three optimization objectives were used in our case based on Eq. (5), the OPF would be formed by at least three routes, namely one for the minimum of each objective. Furthermore, the 4-node and 5-node SONs involve $N = 5$ and $N = 16$, routes in total respectively based on Eq. (8), hence then approach the worst case scenario. However, as the network size is increased, the ratio of the number of optimal routes over the number of the total legitimate ones will decay, hence approaching the lower bound.

Furthermore, it may be observed from from Fig. 8 that the NDQO tends to require less CFEs than the classical BF method for SONs having more than $N_{\text{node}} = 6$ nodes. Explicitly, the NDQO algorithm becomes more efficient for realistic practical databases, where a complexity reduction would be achieved by the use of the BBHT-QSA quantified in terms of the QD-CFEs. This is achieved, if the BBHT-QSA's complexity is lower than that of its respective classical BF counterpart, which would serially check whether there exists a route-solution that dominates the examined one, implying that the minimum required size of a database should satisfy the

condition of $L_{BBHT}^{tot} < N$, where L_{BBHT}^{tot} corresponds to the complexity imposed by a single BBHT-QSA activation. Based on Eqs. (42) as well as (44) for the lower bound and on Eqs. (48) as well as (47) for the upper bound, the respective complexities of a single BBHT-QSA activation will be equal to:

$$L_{BBHT}^{tot,\min} = 4.5\sqrt{N} + \log_{\lambda}\left(4.5\,\frac{\lambda-1}{m}\,\sqrt{N}+1\right)+1, \quad (50)$$

$$L_{BBHT}^{tot,\max} = 10\sqrt{N} + \log_{\lambda}\sqrt{N} - 1. \quad (51)$$

Therefore, based on Eqs. (50) and (51), the minimum database required size for achieving a complexity reduction would be $N_{\min}^{best} > 39$ and $N_{\min}^{worst} > 123$ routes for the best- and the worst-case scenarios, respectively. In our routing application, this condition becomes valid for the best-case scenario in SONs having six or more nodes, where the total number of routes is $N = 65$[6], whereas in the 4-node and 5-node SONs it would be equal to $N = 5$ and $N = 16$ routes, respectively, according to Eq. (8). Hence, we observe in Fig. 8 that some complexity reduction is offered by the NDQO lower bound for SONs consisting of seven nodes.

Moreover, as far as the complexity upper bound is concerned, the condition for achieving a complexity reduction by the BBHT-QSA is satisfied for SONs having seven or more nodes. This may be verified by the change in trend of the upper bound complexities observed in Fig. 8 for the SONs consisting of $N_{nodes} = 7$ nodes. Explicitly, the upper bound of the NDQO algorithm is seen to impose a lower complexity quantified in terms of CFEs compared to the BF and the naive-BF methods. Furthermore, we observe in Fig. 8 that the BF method's complexity upper bound imposes a higher number of CFEs compared to the naive-BF one. As for the NDQO lower bound, where the only optimal route is the direct one, we will achieve a better performance with respect to the BF method for SONs having more than four nodes.

Additionally, as for the average complexity, observe in Fig. 8 that the same average complexity is imposed for both the average NDQO algorithm and the BF method for SONs consisting of six nodes. This is justified by considering the fact that even though a beneficial complexity reduction is offered by the BBHT-QSA chains, there is a computational overhead which is imposed by padding our database so that it has a size equal to a power of 2. Hence, the NDQO algorithm has a slightly higher average complexity than the BF method. On the other hand, a substantial complexity reduction of about 50.5% is offered on average for a 7-node SON, while for the 8-node and 9-node SONs we have a complexity reduction of about 78.6% and 89% respectively, and it increases as the SON becomes larger. This complexity reduction may be translated into a routing-latency reduction of 202%, 466% and 908% for SONs consisting of seven, eight and nine nodes, respectively.

[6]A database of $N = 128$ entries is used for the 6-node SON, since the database length has to be explicitly a power of 2 due to the binary nature of qubits.

Last but not least, a substantial complexity reduction is offered by the NDQO algorithm compared to the naive-BF method for SONs consisting of five or more nodes. According to Fig. 8, a complexity reduction of about an order of magnitude is offered by the NDQO algorithm for 6-node SONs, which is increased to several orders of magnitude for SONs supporting more than six nodes.

B. NDQO COMPUTATIONAL ACCURACY

Before delving into the related NDQO accuracy discussions, the three metrics of computational accuracy that were used should be defined. To begin with, the optimization accuracy may be quantified by the distance from the OPF, which is equal to the average Pareto distance $E[P_d(x)]$ of the OPF exported from the true OPF. Assuming that the exported OPF routes form a set S_0 having a length of $|S_0|$, the *average Pareto Distance* $E[P_d(x)]$ from the OPF becomes:

$$E[P_d(x)] = \sum_{x\in S_0}\frac{P_d(x)}{|S_0|}. \quad (52)$$

Its physical interpretation is given by the average probability of a route belonging to the hitherto generated OPF being dominated by the rest of the legitimate routes. Inherently, if the generated OPF consists exclusively of routes of the true OPF, which is generated by the BF method, the value of this metric would be equal to zero. On the other hand, should the OPF consist of suboptimal points, its value will be bounded by the range (0, 1]. Consequently, it becomes plausible that as its value decays and tends to zero, the generated OPF approaches the true OPF.

Additionally, the accuracy may also be quantified in terms of the *average normalized euclidean distance* between the exported routes, that are erroneously included in the exported OPF, and the specific routes of the true OPF that are closer to the particular erroneous one and they dominate them at the same time. Assuming an exported route x_i from the OPF and another x_j, which corresponds to its counterpart from the true OPF, the related error function $e(x_i)$ can be formulated as:

$$e(x_i) = \frac{1}{\sqrt{N_{UF}}}\sqrt{\sum_{k=1}^{N_{UF}}\left(\frac{f_k(x_i)-f_k(x_j)}{f_k(x_i)}\right)^2}, \quad (53)$$

where N_{UF} is the total number of UFs. From this perspective, the average error $E[e(x)]$ would be equal to:

$$E[e(x)] = \sum_{x\in S_0}\frac{e(x)}{|S_0|}. \quad (54)$$

This metric has the advantage that its value becomes independent from the distribution of the routes within the higher-rank Pareto Fronts. This is desirable, because there may exist route-solutions, which would potentially belong to a suboptimum PF, which is pretty close to the OPF, while their solution vectors are rather distant from those of their OPF counterparts. However, its value in Eq. (54) would inherently depend on the actual values of the route solution vector.

This is in contrast to the average Pareto distance. Explicitly, the value of Eq. (54) would be bounded by the range [0, 1].

The third metric considered is what we refer as the *Optimal Pareto Front Completion* C, which is defined as follows. Assuming that the optimization process has generated an OPF of length equal to $|OPF|$ and that the number of erroneous points not belonging to the true OPF is equal to $|OPF|_e$, the Optimal Pareto Front Completion C may be defined as:

$$C = \frac{|OPF| - |OPF|_e}{|TOPF|}, \tag{55}$$

where $|TOPF|$ is the length[7] of the TOPF. Naturally, this metric is bounded to the range [0, 1] and should the entire true OPF be successfully generated it will be equal to unity.

Having defined these metrics, let us now carry out a comparative case study: we will evaluate the OPF generated by the NSGA-II, the ACO and the NDQO algorithms. For the NSGA-II and the ACO the evaluation of the hitherto generated OPF will be provided at the end of each generation process. For the NDQO algorithm, there is no notion of generations, hence the evaluation process will be invoked each time a route is appended to the OPF (right after Step 2.13). However, since the total number of CFEs required by the BBHT-QSA chains is a rather stochastic process upper bounded by $L_{BBHT}^{\max}$ CFEs as defined in Eq. (51), the evaluation process will be activated at different L_{CFE}^{tot} values. We will assume that between these evaluation processes the metrics remain constant, which results in a sum of step functions for each simulation. We can then extract a continuous distribution of these metrics versus the number of CFEs by performing an averaging operation.

The accuracy metrics are shown in Fig. 9 a 7-node SON. As far as the average Pareto distance $E[P_d(x)]$ is concerned, it becomes clear from Fig. 9(a) that the NDQO algorithm exhibits a far better performance than the NSGA-II and the ACO algorithm. Explicitly, observe in Fig. 9(a) that for the 7-node SON the NDQO performs optimally for 502 CFEs and then the average Pareto distance $E[P_d(x)]$ would be about 10^{-8}. The order of these values suggests that our NDQO algorithm attains a near-optimal performance compared to the BF method, while its complexity is a about an order of magnitude lower than the complexity of the BF method. Moreover, according to Fig. 9(a), the computational accuracy in terms of the average Pareto distance is several orders of magnitude lower than that of both the NSGA-II and of the ACO algorithm. It should be noted that the associated errors of the NDQO algorithm arise from the inclusion of suboptimal route-solutions into OPF. To elaborate further, the BBHT-QSA involves a small arbitrary error [30], which would imply that there is a slight possibility that a BBHT-QSA time-out will result in a suboptimal route owing to it inability to find another dominating route, which results in misinterpreting it as the optimal one. However,

[7]The length of a PF may be defined as the number of route-solution, which it consists of.

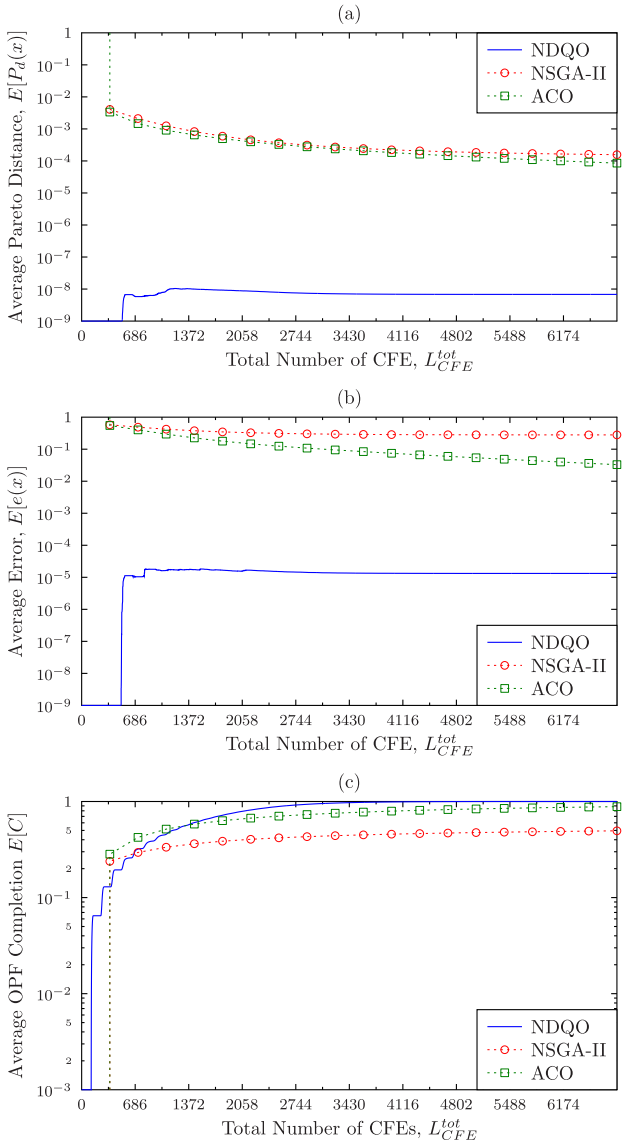

FIGURE 9. Perfomance comparison between the NDQO and the benchmarking State-of-the-Art algorithms NSGA-II and ACO for 7-node SONs in terms of the Average Pareto Distance $E[P_d(x)]$ (a), Average Error $E[e(x)]$ (b) and Optimal Pareto Front Completion $E[C]$ (c). For the sake of fairness, the comparison is made for the number of CFEs for all the algorithms examined. For both the NSGA-II and the ACO, the number of agents has been chosen equal to the number of the generations, which, in turn, is equal to the cubic root of the maximum NDQO complexity. Therefore, they will be set equal to 19. The maximum observed complexity of the BF method is equal to 13058 CFEs, as it may be seen in Fig 8. The results have been averaged over 10^8 runs.

the inclusion of the entire true OPF route set is guaranteed, because all the routes will be explicitly considered. This leads to a generated OPF which consists of all the true OPF routes along with with some low-probability suboptimal ones.

This trend is shown in Fig. 9(c) in terms of their Optimal Pareto Front Completion C. To elaborate further, it may be observed that although the NSGA-II and the ACO algorithm fail to converge to unity, the NDQO algorithm succeeds in exporting all the routes consisting the true

OPF after 5575 CFEs, yielding a 234.22% and 1906.30% improvement for our 7-node SON compared to the BF and the naive-BF methods, respectively. This gain is further increased the number of nodes increases. This property of the NDQO algorithm is of great importance, since it enables the reconstruction of the OPF by invoking a classical NDS for discarding the erroneous route-solutions. Nevertheless, this operation would require additional CFEs. This action cannot be invoked for our benchmarking algorithms, since their completeness does not converge to unity and, consequently, they will fail to export all the tue OPF route-solutions.

As far as the average error $E[e(x)]$ is concerned, identical trends to these seen for the average Pareto distance are observed in Fig. 9(b); however, the value of this metric is higher than that of the $E[P_d(x)]$. Quantitatively, they are on the order of 10^{-5} for our 7-node SON. This error is four orders of magnitude lower than the the respective error of both the NSGA-II and the ACO algorithm. Explicitly, our proposed algorithm has a near-optimal accuracy, since each generated OPF route would differ from the closest true OPF, which would potentially dominate it, by about 0.001%.

VI. CONCLUSIONS

We have proposed an optimal algorithm for multi-objective routing in SONs using Pareto Optimality. The theoretical upper and lower complexity bounds of the NDQO algorithm have been analytically derived, yielding a complexity between $O(N))$ and $O(N\sqrt{N})$. This implies a significant CFE reduction compared to the classical BF method, which exhibits a complexity on the order of $O(N^2)$ in the worst-case scenario. Naturally, this complexity reduction becomes more significant, as the number of nodes increases. As far as its accuracy is concerned, we have demonstrated that the NDQO algorithm exhibits a near-optimal performance whilst attaining several orders of magnitude better accuracy than the state-of-the-art classical evolutionary NSGA-II and ACO algorithms.

REFERENCES

[1] H. Hu, J. Zhang, X. Zheng, Y. Yang, and P. Wu, "Self-configuration and self-optimization for LTE networks," *IEEE Commun. Mag.*, vol. 48, no. 2, pp. 94–100, Feb. 2010.

[2] M. Gerla, *Ad Hoc Networks*. New York, NY, USA: Springer-Verlag, 2005, pp. 1–22.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[4] H. Yetgin, K. T. K. Cheung, and L. Hanzo, "Multi-objective routing optimization using evolutionary algorithms," in *Proc. IEEE WCNC*, Apr. 2012, pp. 3030–3034.

[5] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. press, 2010.

[6] S. Imre and F. Balazs, *Quantum Computing and Communications: An Engineering Approach*. New York, NY, USA: Wiley, 2005.

[7] X. Zhu, L. Shen, and T.-S. P. Yum, "Hausdorff clustering and minimum energy routing for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 990–997, Feb. 2009.

[8] M. Chen, V. C. M. Leung, S. Mao, Y. Xiao, and I. Chlamtac, "Hybrid geographic routing for flexible energy—Delay tradeoff," *IEEE Trans. Veh. Technol.*, vol. 58, no. 9, pp. 4976–4988, Nov. 2009.

[9] A. E. A. A. Abdulla, H. Nishiyama, J. Yang, N. Ansari, and N. Kato, "HYMN: A novel hybrid multi-hop routing algorithm to improve the longevity of WSNs," *IEEE Trans. Wireless Commun.*, vol. 11, no. 7, pp. 2531–2541, Jul. 2012.

[10] K. Deb, "Multi-objective optimization," in *Search Methodologies*, E. K. Burke and G. Kendall, Eds. New York, NY, USA: Springer-Verlag, 2005, pp. 273–316.

[11] E. Masazade, R. Rajagopalan, P. K. Varshney, C. K. Mohan, G. K. Sendur, and M. Keskinoz, "A multiobjective optimization approach to obtain decision thresholds for distributed detection in wireless sensor networks," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 40, no. 2, pp. 444–457, Apr. 2010.

[12] M. Camelo, C. Omaña, and H. Castro, "QoS routing algorithm based on multi-objective optimization for wireless mesh networks," in *Proc. IEEE Latin-Amer. Conf. Commun.*, Sep. 2010, pp. 1–6.

[13] A. J. Perez, M. A. Labrador, and P. M. Wightman, "A multiobjective approach to the relay placement problem in WSNs," in *Proc. IEEE WCNC*, Mar. 2011, pp. 475–480.

[14] F. V. C. Martins, E. G. Carrano, E. F. Wanner, R. H. C. Takahashi, and G. R. Mateus, "A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks," *IEEE Sensors J.*, vol. 11, no. 3, pp. 545–554, Mar. 2011.

[15] D. Pinto and B. Barán, "Solving multiobjective multicast routing problem with a new ant colony optimization approach," in *Proc. 3rd Int. IFIP/ACM Latin Amer. Conf. Netw.*, 2005, pp. 11–19.

[16] W. Stadler, "A survey of multicriteria optimization or the vector maximum problem, part I: 1776–1960," *J. Optim. Theory Appl.*, vol. 29, no. 1, pp. 1–52, Sep. 1979.

[17] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[18] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Proc. Congr. Evol. Comput.*, 1999, vol. 2.

[19] M. Golshahi, M. Mosleh, and M. Kheyrandish, "Implementing an ACO routing algorithm for AD-HOC networks," in *Proc. ICACTE*, 2008, pp. 143–147.

[20] S. Chandra, U. Shrivastava, R. Vaish, S. Dixit, and M. Rana, "Improved-AntNet: ACO routing algorithm in practice," in *Proc. 11th Int. Conf. Comput. Model. Simul.*, Mar. 2009, pp. 25–29.

[21] H. Wang, Z. Wu, X. Yang, and H. Liu, "A novel ACO-based multicast path algorithm in hypercube networks," *Intell. Autom. Soft Comput.*, vol. 17, no. 5, pp. 541–549, 2011.

[22] L. Wang, Q. Niu, and M. Fei, "A novel quantum ant colony optimization algorithm," in *Bio-Inspired Computational Intelligence and Applications*. Berlin, Germany: Springer-Verlag, 2007, pp. 277–286.

[23] H. Jiang, M.-R. Wang, M. Liu, and J.-W. Yan, "A quantum-inspired ant-based routing algorithm for WSNs," in *Proc. IEEE 16th Int. Conf. CSCWD*, May 2012, pp. 609–615.

[24] Z. Babar, S. X. Ng, and L. Hanzo, "Near-capacity code design for entanglement-assisted classical communication over quantum depolarizing channels," *IEEE Trans. Commun.*, vol. 61, no. 12, pp. 4801–4807, Dec. 2013.

[25] S. Imre and L. Gyongyosi, *Advanced Quantum Communications: An Engineering Approach*. New York, NY, USA: Wiley, 2013.

[26] R. P. Feynman, "Simulating physics with computers," *Int. J. Theoretical Phys.*, vol. 21, no. 6, pp. 467–488, 1982.

[27] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer," *Proc. Roy. Soc. London. A. Math. Phys. Sci.*, vol. 400, no. 1818, pp. 97–117, 1985.

[28] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proc. Roy. Soc. London. Ser. A, Math. Phys. Sci.*, vol. 439, no. 1907, pp. 553–558, 1992.

[29] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, Jul. 1996, pp. 212–219.

[30] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, "Tight bounds on quantum searching," *arXiv preprint quant-ph/9605034*, 1996.

[31] C. Zalka, "Grover's quantum searching algorithm is optimal," *Phys. Rev. A*, vol. 60, no. 4, p. 2746, 1999.

[32] C. Durr and P. Høyer, "A quantum algorithm for finding the minimum," *arXiv preprint quant-ph/9607014*, 1996.

[33] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, 1997.
[34] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," *arXiv preprint quant-ph/0005055*, 2000.
[35] G. Brassard, F. Dupuis, S. Gambs, and A. Tapp, "An optimal quantum algorithm to approximate the mean and its application for approximating the median of a set of points over an arbitrary distance," *arXiv preprint arXiv:1106.4267*, 2011.
[36] P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum search algorithms, quantum wireless, and a low-complexity maximum likelihood iterative quantum multi-user detector design," *IEEE Access*, vol. 1, pp. 94–122, 2013.
[37] P. Botsinis, S. X. Ng, and L. Hanzo, "Fixed-complexity quantum-assisted multi-user detection for CDMA and SDMA," *IEEE Trans. Commun.*, vol. 62, no. 3, pp. 990–1000, Mar. 2014.
[38] P. Botsinis, D. Alanis, S. Ng, and L. Hanzo, "Low-complexity soft-output quantum-assisted multiuser detection for direct-sequence spreading and slow subcarrier-hopping aided SDMA-OFDM systems," *IEEE Access*, vol. 2, pp. 451–472, 2014.
[39] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[40] R. Steele and L. Hanzo, *Mobile Radio Communications: Second and Third Generation Cellular and WATM Systems*. New York, NY, USA: Wiley, 1999.
[41] L. Hanzo, S. X. Ng, W. Webb, and T. Keller, *Quadrature Amplitude Modulation: From Basics to Adaptive Trellis-Coded, Turbo-Equalised and Space-Time Coded OFDM, CDMA and MC-CDMA Systems*. New York, NY, USA: Wiley, 2004.
[42] M. Salem *et al.*, "An overview of radio resource management in relay-enhanced OFDMA-based networks," *IEEE Commun. Surv. Tuts.*, vol. 12, no. 3, pp. 422–438, Jul. 2010.
[43] W. Buttler *et al.*, "Practical free-space quantum key distribution over 1 km," *arXiv preprint quant-ph/9805071*, 1998.
[44] E. Aïmeur, G. Brassard, and S. Gambs, "Quantum speed-up for unsupervised learning," *Mach. Learn.*, vol. 90, no. 2, pp. 261–287, 2013.
[45] D. H. Lehmer, "Teaching combinatorial tricks to a computer," in *Proc. Symp. Appl. Math. Combinat. Anal.*, vol. 10. 1960, pp. 179–193.
[46] C. Xu, B. Hu, L.-L. Yang, and L. Hanzo, "Ant-colony-based multiuser detection for multifunctional-antenna-array-assisted MC DS-CDMA Systems," *IEEE Trans. Veh. Technol.*, vol. 57, no. 1, pp. 658–663, Jan. 2008.
[47] I. Alaya, C. Solnon, and K. Ghedira, "Ant algorithm for the multi-dimensional knapsack problem," in *Proc. Int. Conf. Bioinspired Optim. Methods Appl.*, 2004.

DIMITRIOS ALANIS (S'13) received the M.Eng. degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2011, and the M.Sc. degree in wireless communications from the University of Southampton, Southampton, U.K., in 2012, where he is currently pursuing the Ph.D. degree with the Communications, Signal Processing and Control Group, School of Electronics and Computer Science.

His research interests include quantum computation and quantum information theory, quantum search algorithms, cooperative communications, resource allocation for self-organizing networks, bioinspired optimization algorithms, and classical and quantum game theory.

PANAGIOTIS BOTSINIS (S'12) received the Diploma (equivalent to M.Eng.) degree from the School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece, in 2010, and the M.Sc. (Hons.) degree in wireless communications from the University of Southampton, Southampton, U.K., in 2011, where he is currently pursuing the Ph.D. degree with the Communications, Signal Processing and Control Group, School of Electronics and Computer Science. Since 2010, he has been a member of the Technical Chamber of Greece.

His research interests include quantum-assisted communications, iterative detection, OFDM, MIMO, multiple access systems, coded modulation, channel coding, cooperative communications, and combinatorial optimization.

SOON XIN NG (S'99–M'03–SM'08) received the B.Eng. (Hons.) degree in electronics engineering and the Ph.D. degree in wireless communications from the University of Southampton, Southampton, U.K., in 1999 and 2002, respectively. From 2003 to 2006, he was a Post-Doctoral Research Fellow involved in collaborative European research projects known as SCOUT, NEWCOM, and PHOENIX. Since 2006, he has been a member of the academic staff with the School of Electronics and Computer Science, University of Southampton. He is involved in the OPTIMIX and CONCERTO European projects, and the IU-ATC and UC4G projects. He is currently a Senior Lecturer with the University of Southampton.

His research interests include adaptive coded modulation, coded modulation, channel coding, space-time coding, joint source and channel coding, iterative detection, OFDM, MIMO, cooperative communications, distributed coding, quantum error correction codes and joint wireless-and-optical-fiber communications. He has published over 170 papers and co-authored two John Wiley/IEEE Press books in this field. He is a Chartered Engineer and a Fellow of the Higher Education Academy in the U.K.

LAJOS HANZO (F'04) is a fellow of REng, IET, and EURASIP. He received the D.Sc. degree in electronics and the Ph.D. degree in 1976 and 1983, respectively. He was a recipient of the Honorary Doctorate Doctor Honoris Causa Award from the Technical University of Budapest in 2009. During his 35-year career in telecommunications, he has held various research and academic posts in Hungary, Germany, and the U.K. Since 1986, he has been with the School of Electronics and Computer Science, University of Southampton, Southampton, U.K., where he is the Chair of Telecommunications. He has successfully supervised 80 Ph.D. students, co-authored 20 John Wiley/IEEE Press books on mobile radio communications totaling in excess of 10 000 pages, published more than 1 300 research entries at IEEE Xplore, acted both as a TPC and the General Chair of IEEE conferences, presented keynote lectures, and has been awarded a number of distinctions. He is currently directing 100-strong academic research team, involved in a range of research projects in the field of wireless multimedia communications sponsored by the industry, the Engineering and Physical Sciences Research Council, U.K., the European IST Program, and the Mobile Virtual Centre of Excellence, U.K. He is an enthusiastic supporter of industrial and academic liaison, and he offers a range of industrial courses. He is also a Governor of the IEEE Vehicular Technology Society. From 2008 to 2012, he was the Editor-in-Chief of the IEEE Press and a Chaired Professor also at Tsinghua University, Beijing, China. His research is funded by the European Research Council's Senior Research Fellow Grant. He has more than 16 000 citations.