Article

Quantum Inspired General Variable Neighborhood Search (qGVNS) for dynamic garbage collection

Christos Papalitsas [1]*, Panayiotis Karakostas [2], Theodore Andronikos [1], Spyros Sioutas [1], and Konstantinos Giannakis [1]

[1] Ionian University, Department of Informatics, 7 Tsirigoti Square, Corfu, Greece
Emails: {c14papa, andronikos, sioutas, kgiann}@ionio.gr
[2] University of Macedonia, Department of Applied Informatics, Thessaloniki, Greece
Email; pkarakostas.tm@gmail.com
* Correspondence: c14papa@ionio.gr; Tel.: +30 2661087712

Abstract: General Variable Neighborhood Search (GVNS) is a well known and widely used metaheuristic for efficiently solving many NP-hard combinatorial optimization problems. Quantum General Variable Neighborhood Search (qGVNS) is a novel, quantum inspired extension of the conventional GVNS. Its quantum nature derives from the fact that it takes advantage and incorporates tools and techniques from the field of quantum computation. Travelling Salesman Problem (TSP) is a well known NP-Hard problem which has broadly been used for modelling many real life routing cases. As a consequence, TSP can be used as a basis for modelling and finding routes for Geographical Systems (GPS). In this paper, we examine the potential use of this method for the GPS system of garbage trucks. Specifically, we provide a thorough presentation of our method accompanied with extensive computational results. The experimental data accumulated on a plethora of symmetric TSP instances (symmetric in order to faithfully simulate GPS problems), which are shown in a series of figures and tables, allow us to conclude that the novel qGVNS algorithm can provide an efficient solution for this type of geographical problems.

Keywords: Metaheuristics; VNS; Quantum Inspired; qGVNS; Optimization; TSP; Routing Algorithms; GPS application; Garbage Collection

1. Introduction

Many complex real world problems can be formulated as Combinatorial Optimization (CO) problems. Technically, CO problems require a proper solution from a discrete finite set of feasible solutions in order to simultaneously achieve the minimization (or maximization) of a cost function and the satisfaction of the problem's given constraints. One such problem is finding the shortest path (or a path "close" to the shortest with respect to some appropriate metric) for Global Positioning System (GPS) in a short period of time. The above real-world problem can be perfectly modeled by the Travelling Salesman Problem (TSP).

The Traveling Salesman Problem (TSP) is one of the most widely studied combinatorial optimization problems. Solving the TSP means finding the minimum cost route so that the salesman (the person or entity who travels along a specific route containing many nodes) can start from a point of origin and return to the origin after passing from all given nodes once. The first use of the term "Traveling Salesman Problem" appeared around 1931-1932. Remarkably, a century earlier, in 1832 a book was printed in Germany [1], which, although dedicated to other issues, in the last chapter deals with the essence of the TSP problem: "With a suitable choice and route planning, you can often save

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 November 2017 doi:10.20944/preprints201711.0128.v1

so much time to make some suggestions. ... The most important aspect is to cover as many locations, without visiting a location for the second time." In that book, the TSP is expressed for the first time using some examples of routes through Germany and Switzerland. However, not an in-depth study of the problem is attempted in this book. The TSP was expressed mathematically for the first time in the 19th century by Hamilton and Kirkman [1]. A *cycle* in a graph is a closed path that starts and ends at the same node and visits each other node exactly once. A cycle containing all vertices of the graph is called *Hamiltonian*.

In short, TSP is the problem of finding the shortest Hamiltonian cycle. The Hamiltonian graph problem, i.e., deciding whether a graph has a Hamiltonian cycle, is reducible to the traveling salesman problem. One can see that by assigning zero length to the edges of the graph and constructing a new edge of length one for each missing edge. If the solution of the TSP for the resulting graph is zero, then the original graph contains a Hamiltonian cycle; if it is a positive number, then the original graph contains no Hamiltonian cycle [2]. TSP is NP-hard and is of great significance in various fields, such as operational research and theoretical computer science. In practice, TSP amounts to finding the best way one can visit all the cities, return to the starting point, and minimize the cost of the tour.

Typically, TSP is represented by a graph. Specifically, the problem is defined as a complete graph $G = (V, A)$, in which $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $A = \{(v_i, v_j) : v_i, v_j \in V \text{ and } v_i \neq v_j\}$ is the set of the directed edges or arcs. Each arc is associated with a weight c_{ij} representing the cost (or the distance) of moving from node i to node j. If c_{ij} is equal to c_{ji}, the TSP is symmetric (sTSP), otherwise it is called asymmetric (aTSP). Obviously GPS routing refers to symmetric TSP. The fact that TSP is NP-hard implies that there is no known polynomial-time algorithm for finding an optimal solution regardless of the size of the problem instance [3].

Real world problems, such as those related to GPS, can be formulated as instances of the TSP. This class of routing problems requires good solutions computed in a short amount of time. For the acceleration of computational time, sacrificing some of the solution's quality by adopting heuristic and metaheuristic approaches is commonly accepted [4–6]. Heuristics are fast approximation computational methods divided into construction and improvement heuristics. Construction heuristics are used to build feasible initial solutions and improvement heuristics are applied to achieve better solutions. It is customary to apply improvement heuristics iteratively. Metaheuristics are general optimization frameworks which can be appropriately modified in their individual characteristics in order to generate efficient methods for solving specific classes of optimization problems.

The main contribution of this paper is the application of the recently introduced quantum inspired General Variable Neighborhood Search (qGVNS for short) metaheuristic to a GPS problem. We achieve efficient solutions in a short period of time for a GPS application used for garbage trucks, which is modeled as an instance of the TSP. The proposed method guarantees optimal or near-optimal solutions for a real life routing application. GPS applications typically use the Nearest Neighborhood or some of its variations in order to achieve good results for the routing section. The routing section is a critical part of a typical GPS application because if the provided routes are optimal or near-optimal, then the end result will also be near-optimal. The routing solutions produced by the qGVNS were compared to the ones obtained by the well known Nearest Neighborhood algorithm and some of its most widely used variations. The computational results reveal that both qGVNS's first and best improvement solutions provide efficient routes that are either optimal or near-optimal and outperform classic, widely used methods, such as Nearest Neighborhood and its modifications, with a wide margin. qGVNS is a quantum inspired expansion of the conventional GVNS in which the shaking function is based on complex unit vectors.

This paper is organized as follows. In Section 2 we present related works, in Section 3 we explain metaheuristics, the Variable Neighborhood Search, and describe our algorithm in detail. A GPS application and the specific problem we tackle with our algorithm is presented in Section 4. In Section 5 the experimental results of our implementation are presented in a series of matrices and

 doi:10.20944/preprints201711.0128.v1

figures which clearly demonstrate that our method outperforms standard approaches like Nearest Neighborhood algorithm. Finally, conclusions and ideas for future work are described in Section 6.

2. Related Work

The research community has shown great interest in solving tangible, real world problems via methods applicable to combinatorial optimization problems. Recently, many authors have been actively trying to enhance the conventional optimization methods by exploiting advantages of unconventional methods of computation (e.g., quantum inspired computation) in order to take advantage of their conjectured superiority over traditional approaches. For example, Sandip et al. proposed several novel techniques which they called quantum inspired Ant Colony Optimization, quantum inspired Differential Evolution and quantum inspired Particle Swarm Optimization, respectively, for Multi-level Colour Image Thresholding. These techniques find optimal threshold values at different levels of thresholding for colour images [7].

A new quantum inspired Social Evolution algorithm was proposed by hybridizing a well-known Social Evolution algorithm with an emerging quantum-inspired evolutionary one. The proposed QSE algorithm was applied to the 0-1 knapsack problem and the performance of the algorithm was compared to various evolutionary, swarm and quantum inspired evolutionary algorithmic variants. Pavithr et al. claim that the performance of the QSE algorithm is better than or at least comparable to the different evolutionary algorithmic variants it was tested against [8].

Wei Fang et al. proposed a decentralized form of quantum-inspired particle swarm optimization with a cellular structured population for maintaining population diversity and balancing global and local search [9]. Zheng et al. conducted an interesting study by applying a novel Hybrid Quantum Inspired Evolutionary Algorithm to a permutation flow-shop scheduling problem. They proposed a simple representation method for the determination of job sequence in the permutation flow-shop scheduling problem based on the probability amplitude of qubits [10].

Lu et al. designed a quantum inspired space search algorithm in order to solve numerical optimization problems. In their algorithm the feasible solution is decomposed into regions in terms of quantum representation. The search progresses from one generation to the next, while the quantum bits evolve gradually to increase the probability of region selection [11]. Wu et al. in [12] proposed a novel approach using a quantum inspired algorithm based on game-theoretic principles. In particular, they reduced the problem they studied to choosing strategies in evolutionary games. Quantum games and their strategies seem very promising, offering enhanced capabilities over classic ones [13].

Variable neighborhood Search (VNS) based solutions have been applied to route planning problems. Sze et al. proposed a hybrid adaptive variable neighborhood search algorithm for solving the capacitated vehicle routing problem (capacitated VRP) [14]. A two level VNS heuristic has been developed in order to tackle the clustered VRP by Defryn and Sorensen [15]. In [16] a VNS approach for the solution of the recently introduced Swap-Body VRP is proposed. Curtin et al. made an extensive comparative study of well known methods and ready-to-use software and they concluded that no software or classic method can guarantee an optimal solution to the TSP problems that model GIS problem with more than 25 nodes [17]. Papalitsas et al. proposed a GVNS approach for the TSP with Time windows [4] and a quantum inspired GVNS (qGVNS) for solving the TSP with Time Windows [18].

In this work we demonstrate that for small problems, our method achieves the optimal value and for larger problems it guarantees a close to optimal solution with a deviation of 1-3%.

3. Quantum Variable Neighborhood Search (qVNS)

In this section we describe the quantum Variable Neighborhood Search method. For completeness we state the necessary background notions and formal definitions (such as on metaheuristics, VNS, etc.).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 November 2017 doi:10.20944/preprints201711.0128.v1

3.1. Metaheuristics

A metaheuristic is a high level heuristic, designed to find, create, or select a lower level heuristic (for example a local search algorithm), which can provide a pretty good solution to an optimization problem. It is particularly useful for instances with missing or incomplete information, or when the computing capacity is limited. According to the literature on metaheuristic optimization [19], the word "metaheuristics" was devised and proposed by Glover. Metaheuristic algorithms are able to make assumptions about the optimization problem to be solved and thus, they can be used for a wide variety of problems. Obviously, compared with exact methods, metaheuristic procedures do not guarantee a global optimal solution for each category of problems [20].

Many metaheuristic algorithms apply some form of stochastic optimization. This implies that the generated solution depends on a set of random variables. By searching in a large set of feasible solutions, the metaheuristic procedures can often find good solutions with less computational effort than exact algorithms, iterative methods, or simple heuristic procedures. Therefore, metaheuristic procedures are useful approaches for optimization problems in many practical situations.

3.2. Variable Neighborhood Search (VNS)

Variable Neighborhood Search (VNS) is a metaheuristic for solving combinatorial and global optimization problems, proposed by Mladenovic and Hansen [21,22]. The main idea of this framework is the systematic neighborhood change in order to achieve an optimal (or a close-to-optimal) solution [23]. VNS and its extensions have proven their efficiency in solving many combinatorial and global optimization problems [24].

Each VNS heuristic consists of three parts. The first one is a shaking procedure (diversification phase) used to escape local optimal solutions. The next one is the neighborhood change move, in which the following neighborhood structure that will be searched is determined; during this part, an approval or rejection criterion is also applied on the last solution found. The third part is the improvement phase (intensification) achieved through the exploration of neighborhood structures through the application of different local search moves. Variable Neighborhood Descent (VND) is a method in which the neighborhood change procedure is performed deterministically.

General Variable Neighborhood Search (GVNS) is a VNS variant where the VND method is used as the improvement procedure. GVNS has been successfully tested in many applications, as several recent works have demonstrated [25,26].

3.3. Description of the qGVNS

Similarly to the original GVNS, the quantum inspired GVNS (qGVNS) consists of a VND local search, a diversification procedure and a neighborhood change step. In our method, the pipe-VND (exploitation in the same neighborhood while improvements are also being made) is used during the improvement phase. During the improvement phase of pipe-VND, two classic local search strategies are applied: the relocate and the 2-opt. In relocate, the solutions are obtained by removing a node and inserting it in a different position of the current route. In 2-opt the solutions are obtained by breaking 2 edges and reconnecting them in a different order.

The biggest difference between qGVNS and the classic GVNS is in the diversification phase. The main use of a shaking function is to resolve local minima traps within a VNS procedure. In our approach, perturbation is achieved by adopting techniques from the field of quantum computation. In each shaking call, a simulated quantum n-qubit register generates a complex n-dimensional *unit* vector. The dimension n of the complex unit vector is greater than or equal to the dimension of the problem. Our algorithm takes as input the complex n-dimensional vector and produces a real n-dimensional vector. The i-th component, $1 \leq i \leq n$, of the real vector is equal to the modulus squared of the i-th component of the complex vector. Obviously, the components of the real vector are real numbers in the interval $[0, 1]$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 November 2017 doi:10.20944/preprints201711.0128.v1

Each node of the current solution is associated with precisely one of the components of the real n-dimensional vector. In effect the vector components are used as a flag for each node in the current solution. Under this correspondence between vector components and nodes, the sorting of the components of the real vector, will induce an indentical ordering among the nodes in the solution. The ordered route thus produced after this shaking move, will drive our exploration effort in another search space.

At this point, it should be mentioned that the Nearest Neighbor heuristic is used in order to produce an initial feasible solution (the first node is set as a depot). From an algorithmic perspective, qGVNS is summarized in the next pseudocode fragment [27].

Algorithm 1: Pseudocode of qGVNS

```
Data: an initial solution
Result: an optimized solution
Initialization of the feasibility distance matrix
begin
    X ← Nearest Neighbor heuristic;
    repeat
        X′ ← Quantum-Perturbation(X)
        X″ ← pipeVND(X′)
        if X″ is better than X′ then
            X ← X″
        end
    until optimal solution is found or time limit is met;
end
```

4. GPS application for garbage trucks modeled as a Travelling Salesman Problem

An operation with substantial importance for the handling of everyday's scheduling of a city's traffic is the routing of garbage trucks from their depots to every dustbin on their routes and back to their depots. The optimal routes correspond to minimum required transportation time and minimum distance. Finding optimal routes typically proves to be time-consuming, especially in the case of metropolitan cities with very dense road networks. However, by exploiting recent advances from the field of metaheuristics, it is possible to attain efficient, near-optimal solutions in a short amount of time for many practical cases. We take advantage of the performance improvement brought by a novel metaheuristic procedure based on Variable Neighborhood Search and called quantum inspired General Variable Neighborhood Search (qGVNS). qGVNS, in combination with the minimal required computational time, can provide a significantly enhanced the solution for these kind of problems. The incorporation of the enhanced VNS procedure within the GIS will lower the system's response time, and provide close to optimal solutions.

GIS technology integrates common database operations such as query and statistical analysis with the unique visualization and geographic analysis benefits offered by maps [28,29]. Among other things, a GIS facilitates the modeling of spatial networks (e.g., road networks) offering algorithms to query and analyze them. Spatial networks are modeled with graphs. In the case of road networks, the graph's arcs correspond to street segments whereas the nodes correspond to street segment intersections. Each arc has a weight associated with it, representing the cost of traversing it.

A GIS usually provides a number of tools for the analysis of spatial networks. It generally offers tools to find the shortest or minimum route through a network and heuristic procedures to find the most efficient route to a series of locations. Such a problem is typically modeled as an instance of the traveling salesman problem. Our implementation solves efficiently the TSP problem, finding near-optimal solutions for a range of small, medium or bigger benchmark problems. Distance matrix calculation can be used to calculate distances between pairs of nodes representing origins and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 November 2017 doi:10.20944/preprints201711.0128.v1

destinations whereas location-allocation functions determine site locations and assign demand to sites. These capabilities of GIS for analyzing spatial networks enable them to be used as decision support systems for the districting and routing of vehicles [30,31].

Routing a garbage truck from its depot to each dustbin and back is modelled by finding routes for the Travelling Salesman Problem. The GIS will be used to find the optimal routes corresponding to minimum required transportation time. The GIS can also present the driver with directions corresponding to the routes generated. These directions will be transmitted to the garbage truck. In a real-time system, the time performance of the routing function is of vital significance. Metaheuristics, like our implementation presented here, can guarantee that.

4.1. Our approach

In this paper, we describe a system offering a solution to the problem of garbage truck routing management. It is based on quantum inspired metaheuristics applied on TSP and integrated to GIS/GPS technologies. Our approach is an integrated waste management solution. Based on the functional requirements and some case studies, [32] the components of our application are designed and decomposed into subsystems and smaller functional units. Operations and relationships between subsystems are defined for each subsystem:

- Bin sensors: equipment that estimates the waste bin fill level and collects, stores, and transmits bins data. This will help our main system to take into account specific bins and avoid compute on final route empty bins.
- Data Gathering from bins: A unit that communicates with the Bin sensors and retrieves the collected information to the Central System. It can be installed on passing vehicles and consists of three components:

  1. Communicator, which implements the communication with the bin.
  2. Storage procedure which temporarily stores the data until transferred to the Central System.
  3. Transfer procedure which implements the data transfer to the Central System.

  All these operations will be applied via GSM network.
- GPS Navigation Application integrated on trucks: It is a classic navigation application through GPS, which will provide navigation guidance to the truck driver, and instructions about which bins should be collected.
- Central System: It is the back-end system of our application. Its main part is the Data storing, bins data, vehicles data and all needed data for computing the most efficient routes. Furthermore, Data storing will keep any information retrieved from the other subsystems, particularly from the Data Mining Subsystem and the Routing Optimization Subsystem. All generated spatial information for the current route will be stored locally in a Spatial Database.
- Map substructure: A restful API based on maps that will provide all the required functionality for creating rich-web applications based on geographic and descriptive data.
- Data Mining Subsystem: Mainly used to estimate fullness of bins when we do not have the available information updated.
- Routing Optimization structure: Our main contribution, is mainly based on implementing efficient routing algorithms for this application. We propose the novel metaheuristic method qGVNS which provides optimal or close to optimal solutions in a short period of time. Our implementation of routing functionality gives an overall comparative advantage compared the other implementations for two main reasons:

  1. We can compute efficient solutions in a short period of time. This makes the whole application efficient because we can compute and re-compute live and continuously many times for the same route and feed the results to the application.
  2. Our routing algorithm outperforms classic methods that are using for finding routes on GPS like Nearest Neighborhood and the solutions for every class of problem are near-optimal.

A real-time system like ours must be able to give prompt replies to such queries because in these situations the response time is of vital importance. By using efficient local search structures and the novel quantum inspired qGVNS for the problem tour, the metaheuristic algorithm can provide better results.

Most of GIS/GPS implementations are using either the Nearest Neighbor (NN) or some of its variations for computing the routes. NN is a well known and widely used construction heuristic in network designing problems. Initially in NN, an arbitrary node is inserted in the route as the starting node and then, iteratevely the nearest to the last added node, selected to insert in the route. The procedure is terminated when all the nodes have been added in the route. Therefore, because of the popularity of NN, when we test the performance of our algorithm, we compare its results both with the optimal solution and the solutions achieved by NN. This comparison demonstrates that qGVNS produces results that in all cases are near-optimal and significantly better than the Nearest Neighborhood algorithm, which is widely used by most GIS/GPS software. As a result of these better routing solutions, the total amount of fuel is drastically reduced. Thus our method is environmentally friendly, since fuel consumption has a direct impact on the environment.

5. Experimental Results

This section is devoted to the presentation of the experimental results that showcase the strengths of qGVNS. The experimental tests were implemented in Fortran 90 and ran on a laptop PC with an Intel Core i7-6700HQ Processor at 2.59GH and 16GB memory. The algorithms were tested on 48 benchmark instances from the TSPLIB. TSPLIB is a library that contains a collection of benchmarks for the TSP. These benchmarks are characterized by their diversity and their variation with respect to the dimension of the problem. It is precisely for these qualities that they are widely used by researchers for comparing results [33].

5.1. qGVNS versus Nearest Neighbor

In this work, we propose a novel method called quantum inspired GVNS (qGVNS), and we apply it to a garbage collection application which is an actual GPS-based problem. Below we present the computational results of our approach. As already mentioned, Nearest Neighbor and its variants are widely used in GPS/GIS applications in order to construct the tour of the underlying graph. In particular, we compare our method against the Nearest Neighbor (NN) heuristic and two of its most well-known variants: the Repeated NN and the Impoved NN. This section presents the comparative analysis among NN, qGVNS, and the Optimal Value, showing the results in a series of figures and tables. It is important to point out that we chose to examine the Nearest Neighbor because of its widespread use in GPS applications.

Table 1 contains the aggregated experimental results. Specifically, it contains the benchmark name, the Nearest Neighbor (NN) cost, the average for first improvement (FI) for each problem, the gap between qGVNS using FI vs. OV, the average for best improvement (BI) for each problem, the gap between qGVNS using BI vs. OV and the optimal value (OV). Given the outcome x, its gap from the optimal value OV is computed by the formula $\frac{OV-x}{OV}$. The gap is widely used in the field of Optimization to meausre how close to the optimal is a particular solution. The data demonstrate that qGVNS

- is consistently very close to the optimal value both with FI and BI, and
- outperforms Nearest Neighbor in all cases.

The shaded lines in Table 1 emphasize the superiority of qGVNS for certain instances of the benchmarks. Specifically we highlight these instances where qGVNS has 2% or less gap from the optimal value. For example, NN's gap from optimal value for *att48* is -0.2101, whereas qGVNS's BI search strategy is -0.0016 and qGVNS using FI is -0.0024. Moreover, qGVNS achieves the optimal value

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 November 2017 doi:10.20944/preprints201711.0128.v1

Table 1. The computational results demonstrate that qGVNS outperforms NN on all TSP instances.

Benchmark Name	NN	Gap NN vs OV	qGVNS BI	Gap qGVNS BI vs. OV	qGVNS FI	Gap qGVNS FI vs. OV	OV
a280	3157	-0.2241	2779	-0.0775	2766	-0.0725	2579
att48	12861	-0.2101	10645	-0.0016	10654	-0.0024	10628
bayg29	2005	-0.2453	1610	0.0000	1610	0.0000	1610
bays29	2258	-0.1178	2020	0.0000	2020	0.0000	2020
bier127	135737	-0.1475	121393	-0.0263	121551	-0.0276	118282
kroA100	27807	-0.3065	21664	-0.0179	21774	-0.0231	21282
burma14	4501	-0.3544	3454	-0.0394	3454	-0.0394	3323
ch130	7579	-0.2404	6342	-0.0380	6373	-0.0430	6110
ch150	8191	-0.2547	6849	-0.0492	6871	-0.0525	6528
d493	41666	-0.1903	37715	-0.0775	37882	-0.0823	35002
kroB100	29158	-0.3169	22514	-0.0168	22786	-0.0291	22141
kroC100	26227	-0.2640	21148	-0.0192	21245	-0.0239	20749
kroD100	26947	-0.2654	21768	-0.0223	21916	-0.0292	21294
kroE100	27460	-0.2443	22512	-0.0201	22709	-0.0290	22068
kroA150	33633	-0.2680	27641	-0.0421	27794	-0.0479	26524
kroB150	34499	-0.3202	27032	-0.0345	27274	-0.0438	26130
kroA200	35859	-0.2210	30900	-0.0522	31179	-0.0617	29368
kroB200	36980	-0.2562	31119	-0.0571	31387	-0.0662	29437
d198	18240	-0.1558	16196	-0.0264	16260	-0.0304	15780
brg180	69550	-34.6666	2026	-0.0390	2038	-0.0451	1950
berlin52	8980	-0.1906	7547	-0.0007	7590	-0.0064	7542
dantzig42	956	-0.3676	701	-0.0029	701	-0.0029	699
eil101	803	-0.2766	647	-0.0286	649	-0.0318	629
eil51	511	-0.1939	428	0.0000	429	-0.0023	428
eil76	642	-0.1933	548	-0.0186	549	-0.0204	538
fri26	1112	-0.1867	937	0.0000	937	0.0000	937
gil262	3208	-0.3490	2571	-0.0812	2558	-0.0757	2378
gr17	2187	-0.0489	2085	0.0000	2085	0.0000	2085
gr21	3333	-0.2312	2707	0.0000	2707	0.0000	2707
gr24	1553	-0.2209	1272	0.0000	1272	0.0000	1272
gr48	6098	-0.2084	5048	-0.0004	5054	-0.0016	5046
gr96	75065	-0.3596	56084	-0.0158	56133	-0.0167	55209
gr120	9351	-0.3470	7197	-0.0367	7199	-0.0370	6942
gr137	98720	-0.4132	72381	-0.0362	72536	-0.0384	69853
gr202	47080	-0.1723	42419	-0.0563	42287	-0.0530	40160
gr229	169715	-0.2608	141387	-0.0504	142175	-0.0563	134602
gr431	221402	-0.2916	184140	-0.0742	184993	-0.0792	171414
hk48	13181	-0.1500	11498	-0.0032	11531	-0.0061	11461
lin105	20356	-0.4156	14613	-0.0163	14607	-0.0159	14379
lin318	54019	-0.2852	45018	-0.0711	45179	-0.0749	42029
pcb442	61979	-0.2205	55416	-0.0913	55804	-0.0990	50778
pr76	153462	-0.4188	109103	-0.0087	109421	-0.0117	108159
pr107	46680	-0.0536	45183	-0.0199	45464	-0.0262	44303
pr124	69297	-0.1739	59705	-0.0114	59742	-0.0121	59030
pr136	120769	-0.2479	99622	-0.0295	100718	-0.0408	96772
pr144	61652	-0.0532	58776	-0.0041	58991	-0.0078	58537
pr152	85699	-0.1630	74703	-0.0139	74943	-0.0171	73682
pr226	94683	-0.1781	81379	-0.0126	81781	-0.0176	80369

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 November 2017 doi:10.20944/preprints201711.0128.v1

(0.0000 gap from OV) for *bayg29*, *bays29*, *eil51*, *fri26*, *gr17*, *gr21*, and *gr24*. Another characteristic case is the benchmark *lin105* where NN has a -0.4156 gap from the Optimal while qGVNS with BI strategy achieves -0.0163 and using FI -0.0159.

5.2. qGVNS versus Nearest Neighbor variants

In addition to the previous setup, we ran additional tests using the two most popular variants of the Nearest Neighbor heuristic. These are the *Repeated* NN and the *Improved* NN. Improved NN is a variant of the classic NN in which the starting pair of the route is the shortest edge in the distance matrix [34]. The remaining nodes are added to the route in a way identical to the simple NN method. Repeated or repetitive NN is another modification in which the NN algorithm is applied to every node. Finally, the route with the minimum total cost is selected as the best one.

We compared qGVNS with the Improved Nearest Neighbor and the Repeated Nearest Neighbor and the experimental results are contained in Tables 2 and 3. These tables show the Benchmark Name, the Improved NN tour cost, the Repeated NN tour cost, the qGVNS using BI tour cost, the qGVNS using FI tour cost, and the Optimal Value (OV). For consistency, we ran the same experiments as before, but instead of the simple NN, we used the Improved NN and the Repeated NN, and we compared the results with qGVNS and the Optimal value. Improved NN and Repeated NN achieve better results compared to the simple Nearest Neighbor for each benchmark. However, the fact remains that qGVNS both using FI and BI still outperforms these improved NN-based methods.

Table 2. The experimental results show that qGVNS outperforms the Improved and the Repeated NN on TSP instances (1/2).

Benchmark Name	Improved NN	Repeated NN	qGVNS BI	qGVNS FI	OV
a280	3171	3008	2779	2766	2579
att48	13447	12012	10645	10654	10628
bayg29	1938	1935	1610	1610	1610
bays29	2307	2134	2020	2020	2020
bier127	148330	133953	121393	121551	118282
kroA100	28244	24698	21664	21774	21282
burma14	4470	3822	3454	3454	3323
ch130	7342	7129	6342	6373	6110
ch150	7699	7113	6849	6871	6528
d493	41858	40189	37715	37882	35002
kroB100	28525	25884	22514	22786	22141
kroC100	25511	23660	21148	21245	20749
kroD100	29202	24852	21768	21916	21294
kroE100	28125	24782	22512	22709	22068
kroA150	32019	31479	27641	27794	26524
kroB150	37113	31611	27032	27274	26130
kroA200	36825	34543	30900	31179	29368
kroB200	38844	35389	31119	31387	29437
d198	18485	17620	16196	16260	15780
brg180	98460	59550	2026	2038	1950
berlin52	9156	8181	7547	7590	7542
dantzig42	957	864	701	701	699
eil101	823	746	647	649	629
eil51	555	482	428	429	428

Tables 2 and 3 confirm that qGVNS both using BI and FI outperforms the improved NN and the repeated NN. Let us consider for example the benchmark *eil101*, which consists of 101 nodes. The computational results show that that the Improved NN has a tour cost of 823, the Repeated NN has a tour cost of 746, qGVNS on first improvement has 649 and qGVNS on best improvement has 647, while optimal tour's cost Value is 629. It is clear that qGVNS is marginally close to the Optimal Value

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 November 2017 doi:10.20944/preprints201711.0128.v1

(OV) and the gap between both versions of qGVNS and both NN variants is relatively high. Other examples of benchmarks which highlight the superiority of qGVNS are *bayg29*, *d198*, and *ch130*.

Table 3. The experimental results show that qGVNS outperforms the Improved and the Repeated NN on TSP instances (2/2).

Benchmark Name	Improved NN	Repeated NN	qGVNS BI	qGVNS FI	OV
eil76	614	608	548	549	538
fri26	982	965	937	937	937
gil262	3027	2823	2571	2558	2378
gr17	2187	2178	2085	2085	2085
gr21	3265	3003	2707	2707	2707
gr24	1769	1553	1272	1272	1272
gr48	6287	5840	5048	5054	5046
gr96	70060	65416	56084	56133	55209
gr120	8791	8438	7197	7199	6942
gr137	93749	84376	72381	72536	69853
gr202	47460	45030	42419	42287	40160
gr229	167062	157745	141387	142175	134602
gr431	218383	197405	184140	184993	171414
hk48	13052	12137	11498	11531	11461
lin105	20359	16935	14613	14607	14379
lin318	54031	49201	45018	45179	42029
pcb442	64623	58950	55416	55804	50778
pr76	154708	130921	109103	109421	108159
pr107	46765	46680	45183	45464	44303
pr124	69154	67055	59705	59742	59030
pr136	127551	114553	99622	100718	96772
pr144	61904	60964	58776	58991	58537
pr152	85349	79564	74703	74943	73682
pr226	95573	92552	81379	81781	80369

Table 3 contains computational data that corroborate that qGVNS performs better than the Improved and the Repeated NN. Characteristic examples are *gr229*, *lin105*, and *pr136*, which are benchmark instances of 229, 105 and 136 nodes respectively. We can therefore conclude that qGVNS is always close to the optimal value and clearly it outperforms the NN variants.

5.3. qGVNS versus conventional GVNS

The data in Table 4 reveal that qGVNS in indeed an improvement over classic GVNS. From the data one may conclude that in all cases qGVNS is at least as good as GVNS, and in many cases outperforms GVNS. This is in accordance with the work in [27], which, through a comparative analysis, also showed that the quantum inspired GVNS achieves better results than the conventional GVNS.

Table 4. Comparison between qGVNS and the conventional GVNS [27].

Problem	Av. Value (qGVNS)	Best (GVNS)	Problem	Av. Value (qGVNS)	Best (GVNS)
bayg29	1610	1653	br17	39	39
bays29	2020	2069	ftv33	1357.8	1489
fri26	937	969	ftv35	1544.6	1791
gr17	2085	2085	ftv38	1616.6	1778
gr24	1272	1278	ftv44	1763.8	2014
ulysses16	6859	6859	p43	5554	5629
ulysses22	7013	7013	ry48p	14698.2	15134
gr48	5049.4	5325			
hk48	11508.6	11884			

Our aim in this paper was the implementation of a method based on well-known metaheuristics that would be efficient enough to solve real world problems, such as garbage collection studied here. We wanted to develop an algorithm that would be able to find near-optimal solutions in a relatively short period of time. To achieve our goals we chose to implement qGVNS, since we expected to outperform the conventional GVNS. This was indeed confirmed experimentally, as the results in Table 4 show. The bulk of our experiments were meant to determine how qGVNS fares against well-established methods that are widely applied for GPS, like the Nearest Neighborhood and its most important variations. Tables 1, 2, and 3 contain experimental evidence suggesting that qGVNS outperforms the Nearest Neighbor, the Repeated Nearest Neighbor, and the Improved Nearest Neighbor heuristics in all cases.

5.4. Graphical representation of the results

In this section, we present two different sets of figures with three different bar charts for each set. Each figure concerns a different subset of the total experiments, sorted according to the optimal value. In the first set (Figures 1, 2, and 3), each benchmark problem is represented with 4 different bars, one for each method. Specifically, one represents the Nearest Neighborhood algorithm, one the optimal value, and another two depict qGVNS (best and first improvement). It can be easily seen that our implementation achieves results very close to optimal, unlike the Nearest Neighborhood algorithm which does not provide efficient solutions.

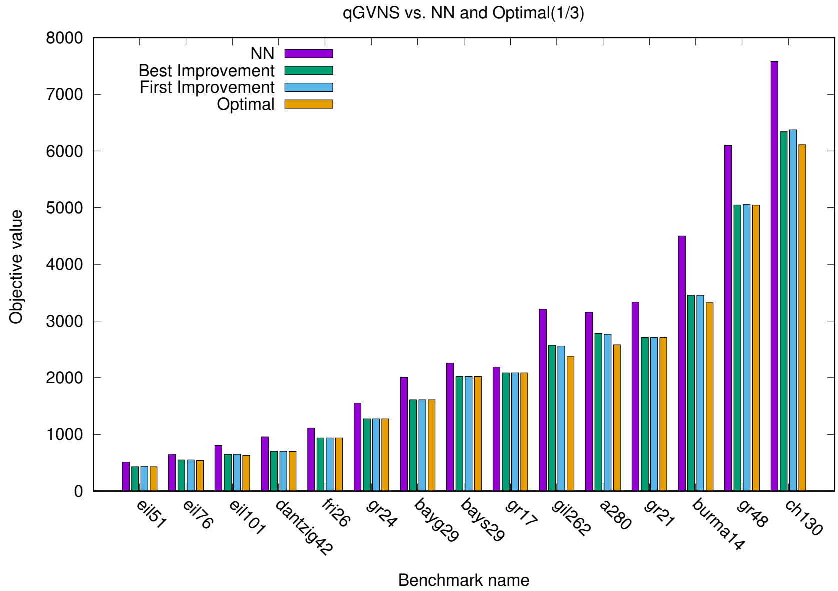

Figure 1. NN vs. qGVNS vs. OV (1/3).

A notable example of Figure 1 is *ch130*. We can see that qGVNS (both using FI and BI) is much closer to the optimal value, compared to NN.

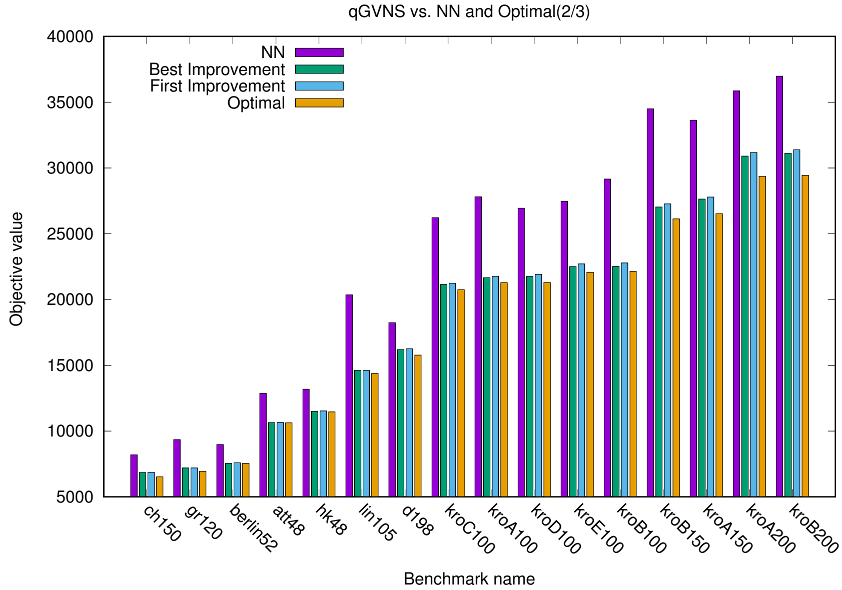

Figure 2. NN vs. qGVNS vs. OV (2/3).

It is clear from Figure 2 that in *kroC100*, *KroA100*, *KroD100*, *KroE100*, *KroB100*, *KroB150*, *KroA150*, *KroA200*, and *KroB200* the gap between the NN method and both variants of the qGVNS is too high. We can also notice some cases in Figure 2, where qGVNS achieves the optimal value. These are *berlin52*, *hk48*, and *att48*.

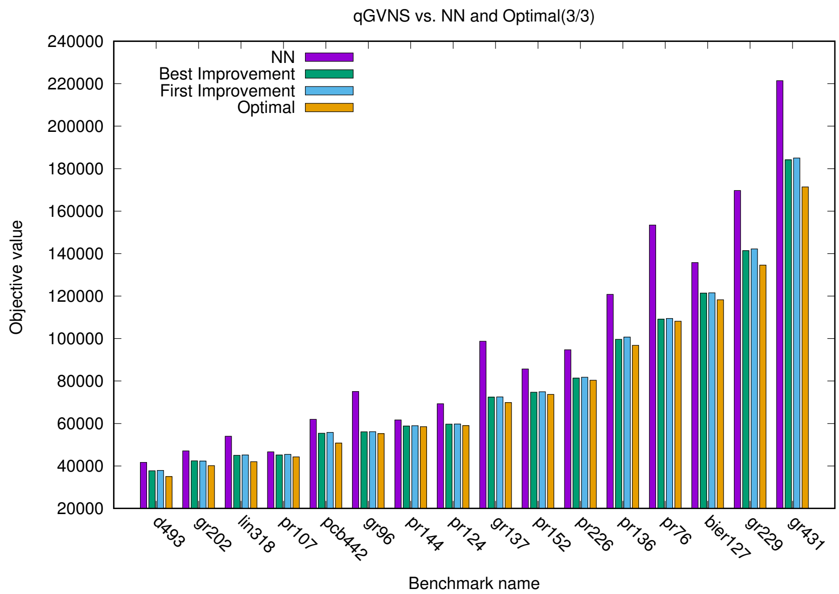

Figure 3. NN vs. qGVNS vs. OV (3/3).

We observe than in most benchmarks in Figure 3 qGVNS using FI and using BI comes close to the optimal value, when, in the same time, the NN bar seems to be far away from the optimal value. Such examples are: *lin318*, *pr124*, *gr137pr76*, *bier127*.

In the next set of figures (i.e., Figures 4, 5, and 6) each benchmark is represented with 5 different bars; one for the Improved Nearest Neighborhood algorithm, one for the Repeated Nearest Neighborhood algorithm, one for the optimal value, and another two for our implementation (best and first improvement). A close examination reveals that qGVNS is once again very close to optimal

 doi:10.20944/preprints201711.0128.v1

values, whereas the Improved and Repeated Nearest Neighborhood algorithms fail to provide equally good solutions.

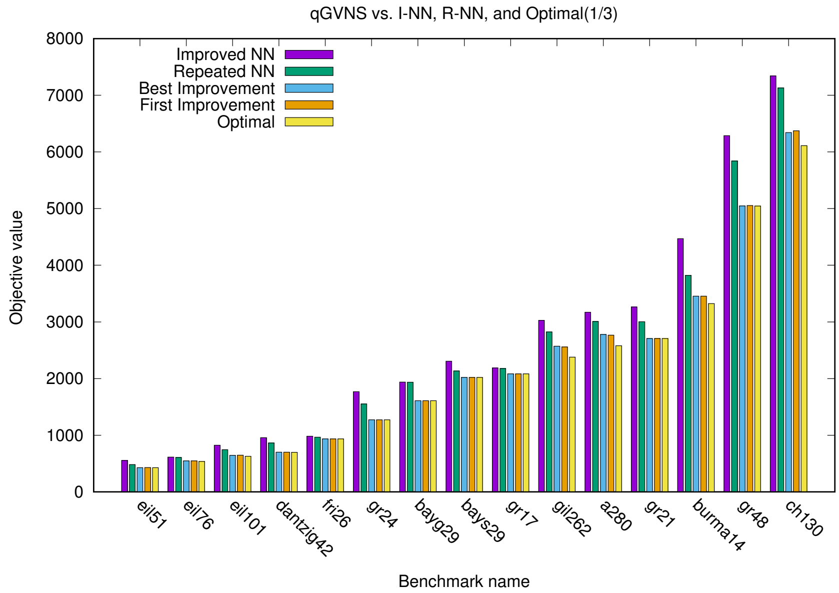

Figure 4. NN variants vs. qGVNS vs. OV (1/3).

Looking at Figure 4, we infer that when it comes for medium benchmark problems, qGVNS significantly approximates the optimal value, unlike both variants of NN which are far from the optimal. We can particularly observe this in *kroB100*, *lin105*, *kroB150* and *kroB200*.

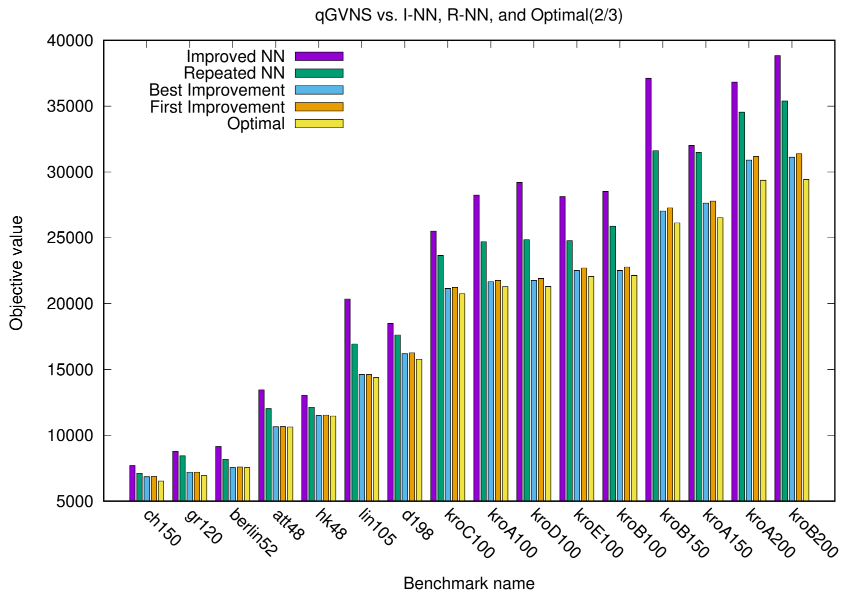

Figure 5. NN variants vs. qGVNS vs. OV (2/3).

In general, Figures 4, 5, and 6 show that Improved NN appears to be a better method than Repeated NN. However, in all cases both qGVNS's variants outperform the two NN variants.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 November 2017 doi:10.20944/preprints201711.0128.v1

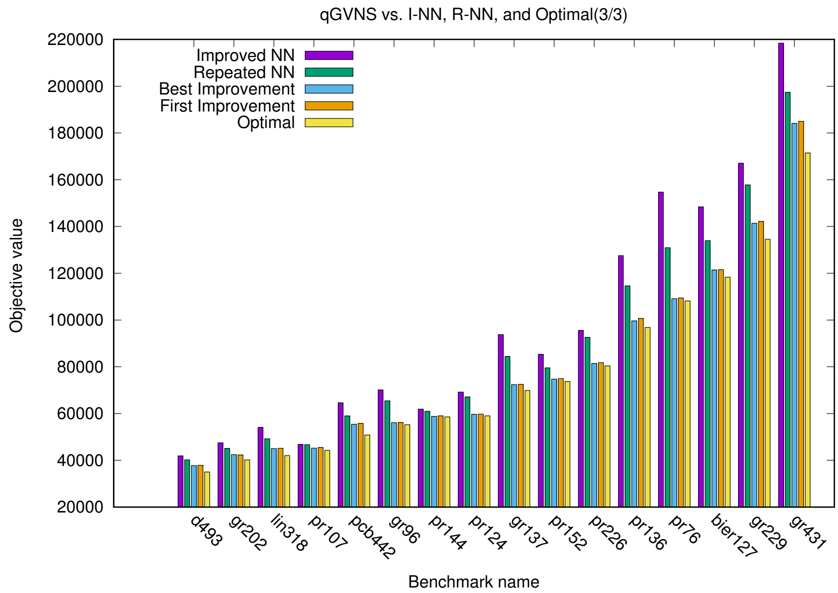

Figure 6. NN variants vs. qGVNS vs. OV (3/3).

Figure 6 demonstrates once again the superiority of qGVNS. Let us consider benchmark *pr76*, where qGVNS touches the optimal point, while both the Repeated and Improved NN have a much lower performance. In addition, in Figures 4, 5, and 6 we can see for a different set of benchmarks that again qGVNS outperforms the Nearest Neighborhood variants, getting significantly closer to the optimal value in each case.

To sum up, the graphical results allow us to conclude that qGVNS with best improvement and first improvement produces results that are quite near to the optimal for most benchmark tests, whereas the Nearest Neighborhood algorithm is far from being close (for every case). Previously, we provided the analytical results in tabular form. The graphical representation of these results makes it easy to see the efficiency of the proposed algorithm, since one can immediately can see that the divergence from optimal is almost negligible.

6. Conclusion and future work

This work studied an application of garbage collectors routing via a new metaheuristic method called qGVNS. We modeled the underlying problem as a TSP instance and went on to solve it using qGVNS. Our algorithm, i.e., qGVNS differs from conventional approaches due to its inspiration coming from principles of quantum computing. Our study was focused on quick and efficient transitions to different areas of the search space. This helped us to find efficient routes in a short period of time. Moreover, in order to assess the efficiency of our approach, we performed extensive experimental tests using well-known benchmarks. The results were quite encouraging, as they confirmed that qGVNS outperforms methods that widely used in practice, such as Nearest Neighborhood, Repeated Nearest Neighborhood and Repeated Nearest Neighborhood.

Future work may include the application of qGVNS to other real life routing optimization problems. Furthermore, different neighborhood structures and neighborhood change moves in VND (Variable Neighborhood Descent) could be investigated under the qGVNS framework. Finally, the current qGVNS scheme may be applied on many other TSP variants, depending on the chosen application.

Author Contributions: All of the authors have contributed extensively to this work. C. P. and P. K. conceived the initial algorithm and worked on the first prototypes. P. K., K. G., and C. P. thoroughly analyzed the current literature gathering all the necessary material. T. A. assisted C.P. in designing the methods used in the main part. T. A. and S. S. were responsible for supervising the construction of this work. S. S. was responsible for the

 doi:10.20944/preprints201711.0128.v1

interlinking between the theoretic model and the actual application. C. P. and K. G. contributed to the appropriate typing of the formal definitions and the maths that were used in the paper. All the authors contributed to the writing of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GIS	Geographic Information System
GPS	Global Positioning System
GVNS	General Variable Neighborhood Search
NN	Nearest Neighbor
OV	Optimal Value
qGVNS	quantum General Variable Neighborhood Search
qVNS	quantum Variable Neighborhood Search
TSP	Travelling Salesman Problem
VND	Variable Neighborhood Descent
VNS	Variable Neighborhood Search

References

1. Voigt, B.F. "Der Handlungsreisende, wie er sein soll und was er zu thun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiss zu zu sein". *Commis-Voageur, Ilmenau. Neu aufgelegt durch Verlag Schramm, Kiel* **1981**.
2. Lawler, E.L.; Lenstra, J.K.; Rinnooy Kan, A.H.G.; Shmoys, D.B. The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization **1985**. p. 476.
3. Rego, C.; Gamboa, D.; Glover, F.; Osterman, C. Traveling salesman problem heuristics: Leading methods, implementations and latest advances. *European Journal of Operational Research* **2011**, *211*, 427 – 441.
4. Papalitsas, Ch.; Giannakis, K.; Andronikos, Th.; Theotokis, D.; Sifaleras, A. Initialization methods for the TSP with Time Windows using Variable Neighborhood Search. IEEE Proc. of the 6th International Conference on Information, Intelligence, Systems and Applications (IISA 2015), 6-8 July, Corfu, Greece, 2015.
5. Silva, R.F.D.; Urrutia, S. A General VNS heuristic for the traveling salesman problem with time windows. *Discrete Optimization* **2010**, *7*, 203–211.
6. Mladenovic, N.; Todosijevic, R.; Urosevic, D. An efficient GVNS for solving Traveling Salesman Problem with Time Windows. *Electronic Notes in Discrete Mathematics* **2012**, *39*, 83–90.
7. Dey, S.; Bhattacharyya, S.; Maulik, U. New quantum inspired meta-heuristic techniques for multi-level colour image thresholding. *Applied Soft Computing* **2016**, *46*, 677 – 702.
8. Pavithr, R.; Gursaran. Quantum Inspired Social Evolution (QSE) algorithm for 0-1 knapsack problem. *Swarm and Evolutionary Computation* **2016**, *29*, 33–46.
9. Fang, W.; Sun, J.; Chen, H.; Wu, X. A decentralized quantum-inspired particle swarm optimization algorithm with cellular structured population. *Information Sciences* **2016**, *330*, 19–48.
10. Zheng, T.; Yamashiro, M. A novel hybrid quantum-inspired evolutionary algorithm for permutation flow-shop scheduling. *Journal of Statistics and Management Systems* **2009**, *12*, 1165–1182.
11. Lu, T.C.; Juang, J.C. Quantum-inspired space search algorithm (QSSA) for global numerical optimization. *Applied Mathematics and Computation* **2011**, *218*, 2516–2532.
12. Wu, Q.; Jiao, L.; Li, Y.; Deng, X. A novel quantum-inspired immune clonal algorithm with the evolutionary game approach. *Progress in Natural Science* **2009**, *19*, 1341–1347.
13. Giannakis, K.; Papalitsas, C.; Kastampolidou, K.; Singh, A.; Andronikos, T. Dominant Strategies of Quantum Games on Quantum Periodic Automata. *Computation* **2015**, *3*, 586–599.
14. Sze, J.F.; Salhi, S.; Wassan, N. A hybridisation of adaptive variable neighbourhood search and large neighbourhood search: Application to the vehicle routing problem. *Expert Systems with Applications* **2016**, *65*, 383–397.

15. Defryn, C.; Sörensen, K. A fast two-level variable neighborhood search for the clustered vehicle routing problem. *Computers & Operations Research* **2017**, *83*, 78–94.
16. Huber, S.; Geiger, M.J. Order matters–A Variable Neighborhood Search for the Swap-Body Vehicle Routing Problem. *European Journal of Operational Research* **2017**.
17. Curtin, K.M.; Voicu, G.; Rice, M.T.; Stefanidis, A. A comparative analysis of traveling salesman solutions from geographic information systems. *Transactions in GIS* **2014**, *18*, 286–301.
18. Papalitsas Christos, Karakostas Panayiotis, G.K.S.A.A.T. Initialization methods for the TSP with Time Windows using qGVNS. 2017.
19. Glover, F.; Kochenberger, G.A.; Glover, F.; Kochenberger, G.A. *Handbook of Metaheuristics*; Kluwer Academic Publishers, 2003; p. 557.
20. Goldberg, D. *Genetic algorithms in search, optimization, and machine learning*; Addison-wesley, 1989.
21. Mladenovic, N.; Hansen, P. Variable neighborhood search. *Computers & Operations Research* **1997**, *24*, 1097–1100.
22. Hansen, P.; Mladenovic, N.; Todosijevic, R.; Hanafi, S. Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization EURO J Comput Optim* **2016**.
23. Mladenovic, N.; Todosijevic, R.; Uroševic, D. Less is more: Basic variable neighborhood search for minimum differential dispersion problem. *Information Sciences* **2016**, *326*, 160 – 171.
24. Jarboui, B.; Derbel, H.; Hanafi, S.; Mladenovic, N. Variable neighborhood search for location routing. *Computers & Operations Research* **2013**, *40*, 47–57.
25. Sifaleras, A.; Konstantaras, I.; Mladenović, N. Variable neighborhood search for the economic lot sizing problem with product returns and recovery. *International Journal of Production Economics* **2015**, *160*, 133–143.
26. Sifaleras, A.; Konstantaras, I. General variable neighborhood search for the multi-product dynamic lot sizing problem in closed-loop supply chain. *Electronic Notes in Discrete Mathematics* **2015**, *47*, 69–76.
27. Papalitsas, C.; Karakostas, P.; Kastampolidou, K. A Quantum Inspired GVNS: Some preliminary results. GeNeDis 2016, Workshop on Natural, Unconventional, and Bio-inspired Algorithms and Computation Methods. Springer International Publishing, 2016., 2016.
28. Franklin, C. An Introduction to Geographic Information Systems: Linking Maps to Databases. *Database* **1992**, *15*, 12–21.
29. Muller, J.C. Guest editorial latest developments in GIS/LIS. *International Journal of Geographical Information Science* **1993**, *7*, 293–303.
30. Crossland, M.D.; Wynne, B.E.; Perkins, W.C. Spatial decision support systems: An overview of technology and a test of efficacy. *Decision support systems* **1995**, *14*, 219–235.
31. Keenan, P.B. Spatial decision support systems for vehicle routing. *Decision Support Systems* **1998**, *22*, 65–71.
32. Asimakopoulos, G.; Christodoulou, S.; Gizas, A.; Triantafillou, V.; Tzimas, G.; Gialelis, J.; Voyiatzis, A.; Karadimas, D.; Papalambrou, A. Architecture and Implementation Issues, Towards a Dynamic Waste Collection Management System. Proceedings of the 24th International Conference on World Wide Web; ACM: New York, NY, USA, 2015; WWW '15 Companion, pp. 1383–1388.
33. Reinelt, G. TSPLIB. http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/, 2013. [Online; accessed 25-September-2017].
34. Pimentel, F.G.S.L. Double-ended nearest and loneliest neighbour–a nearest neighbour heuristic variation for the travelling salesman problem. *Revista de Ciências da Computação* **2016**, *6*.