Quantum annealing algorithm for vehicle scheduling

Alan Crispin
School of Computing, Mathematics, Digital Technology
Manchester Metropolitan University
Manchester, UK
a.crispin@mmu.ac.uk

Alex Syrichas
School of Computing, Mathematics, Digital Technology
Manchester Metropolitan University
Manchester, UK
alex.syrichas@gmail.com

***Abstract*— In this paper we propose a new strategy for solving the Capacitated Vehicle Routing Problem using a quantum annealing algorithm. The Capacitated Vehicle Routing Problem is a variant of the Vehicle Routing Problem being characterized by capacitated vehicles which contain goods up to a certain maximum capacity. Quantum annealing is a metaheuristic that uses quantum tunneling in the annealing process. We discuss devising a spin encoding scheme for solving the Capacitated Vehicle Routing Problem with a quantum annealing algorithm and an empirical approach for tuning parameters. We study the effectiveness of quantum annealing in comparison with best known solutions for a range of benchmark instances.**

Keywords; Quantum Annealing, Soft Computing, Population Heuristics, Capacitated Vehicle Routing

I. INTRODUCTION

In this paper we describe a new approach to solve the Capacitated Vehicle Routing Problem (CVRP) using a quantum annealing algorithm. CVRP is a variant of the Vehicle Routing Problem (VRP) [1] in which all vehicles have the same capacity. It has important industrial applications as optimal solutions enable transportation costs to be reduced and customer service levels to be improved. The underlying structure of the CVRP is an undirected graph G=(V, E), where $V = \{v_0, v_1, \ldots v_n\}$ is a vertex set and $E = \{(v_i, v_j)/v_i , v_j \varepsilon V, i<j\}$ is an edge set. The restriction i<j ensures that the problem is symmetric i.e. the distance between two vertices is identical in both directions. Vertex v_0 is the depot from where a fleet of m ($m \geqslant 1$) identical vehicles, which may contain goods up to a certain maximum capacity Q, must serve exactly n customers represented by the set of n vertices $\{ v_1, v_2, \cdots v_n\}$. A non-negative cost d_{ij} can be defined in terms of a distance or time between vertices (customers) v_i and v_j. Each customer has a non-negative demand for goods q_i. The cost of the problem solution is the sum of the costs of its routes.

VRP heuristics can be divided into three classes known as construction heuristics, improvement heuristics and metaheuristics. Construction heuristics work by creating tours by inserting customers (vertices) into partial tours or combining sub-tours taking into consideration capacities and costs. The Clarke and Wright saving method [2] is an example of a construction heuristic which starts by assuming that a single vehicle services a single customer and then constructs tours by calculating the saving that can be obtained by merging customers and servicing them with a single vehicle. Improvement heuristics seek to iteratively enhance a feasible solution (often initially generated by a construction heuristic) by locally replacing/swapping a set of vertices (customers). A metaheuristic can be thought of as a top-level strategy which guides local improvement operators to find a global solution. Groër et al. [3] describe a library of local search heuristics for VRP. This paper describes a quantum annealing metaheuristic for solving the CVRP.

Rose and Macready [4] describe how quantum annealing (QA) can be used to solve optimization problems. In order to implement quantum annealing on a classical computer, it is necessary to use a stochastic variant such as Path-Integral Monte Carlo (PIMC), see Titiloye and Crispin [5]. This requires the development of a Hamiltonian consisting of the sum of two terms H_{pot} and H_{kin} where H_{pot} represents the classical potential energy of a given problem configuration and H_{kin} is a suitable kinetic energy term, devised for the problem, to provide quantum fluctuations during a Monte Carlo cycle. The kinetic energy term consists of a population of candidate solutions, encoded as spin matrices, with each member of the population known as a replica. In essence, QA is a population heuristic as it uses a number of current solutions (replicas) and combines them to generate new solutions. Consequently, Monte Carlo quantum annealing shares features with both simulated annealing (SA) and an evolutionary algorithm (EA). However, whereas SA attempts to overcome barriers thermally, Monte Carlo QA simulates quantum tunneling between the replicas to escape barriers to the ground state. The analogy with EA is drawn as both are based on the interaction of member solutions in a population.

The Path-Integral Monte Carlo Quantum Annealing algorithm has been applied with success to the Travelling Salesman Problem (TSP) [6]. An undirected TSP tour is represented in terms of Boolean variables by defining a connection matrix from which Ising spin variables are defined. A 2-opt move is used to attempt to improve a tour configuration in the Monte Carlo algorithm. A 2-opt move is a local improvement operator which tries to improve a tour configuration by removing two edges in the tour and replacing them with two new edges. In essence, the Travelling Salesman Problem (TSP) is a simplified variant of the VRP where the goal is to find the shortest possible tour, in terms of total distance, which visits each of the cities exactly once. This corresponds to a VRP where there is only one vehicle, no capacity restrictions, and the cost is only dependent on, and directly proportional to, the distance. The similarity between

978-1-4799-0652-9/13 $31.00 © 2013 IEEE
DOI 10.1109/SMC.2013.601

TSP and VRP suggests that a QA heuristic could be established for vehicle scheduling and [6] identifies the use of constrained Boolean variables to represent the problem as a key factor to success.

The paper is organized as follows. Section II describes our quantum annealing scheme for the CVRP. In section III we provide pseudo code for our PIMC quantum annealing algorithm and details of parameter tuning are given in section IV. In section V results from computational experiments on CVRP benchmark data sets are presented and compared against a simulated annealing algorithm. In section VI we analyze our QA algorithm. Concluding remarks are given in section VII.

II. QA SCHEME FOR CVRP

In the quantum Path-Integral Monte Carlo (PIMC) approach the statistical behavior of the quantum spin model is approximated to a classical simulation model by using a Suzuki-Trotter transformation [7]. With PIMC, the quantum system is mapped onto a classical model consisting of P ferromagnetically coupled spin matrices or replicas. The total Hamiltonian energy H can be written as [8]:

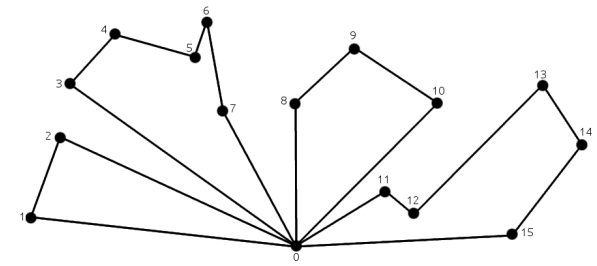
$$H = \frac{H_{pot}}{P} - J_{\Gamma}\Delta H_{kin} \quad (1)$$

Here H_{pot} is the potential energy summed over the P replicas so that H_{pot}/P is the average value and J_{Γ} is the transverse ferromagnetic coupling term given by:

$$J_{\Gamma} = \frac{-T}{2}\ln\tanh\left(\frac{\Gamma}{PT}\right) > 0 \quad (2)$$

In equation (2), T is the temperature and Γ the tunneling field strength parameter. The term PT is called the effective quantum temperature [8]. The change in kinetic energy term ΔH_{kin} provides a disturbance during a Monte Carlo iteration cycle for escaping local minima and is calculated from replica interactions. To do this a set of P spin matrices has to be defined with each matrix randomly initialized. The idea is that the change in kinetic energy ΔH_{kin} should decrease to zero over time as the replicas become similar so that H tends to the average potential energy value. The strength of J_{Γ} determines the importance of the ΔH_{kin} term relative to the potential energy term. Notice that J_{Γ} is positive for ferromagnetic interaction.

To apply QA to a problem requires finding an intuitive way to encode the problem in terms of a set of spin matrices or replicas. Here we define our spin matrix S_{ij} from a connectivity matrix which describes all routes in a VRP schedule. Fig.1 shows an example of our spin encoding scheme. In this example there are fifteen customers (vertices) which are served from a depot (v_0) by four routes, one for each vehicle. Rows i and columns j are labeled 0 to 15 and represent the vertices in the network. Each cell represents a connection between two vertices and we define S_{ij} =1 if vertex i is connected to vertex j in a route and S_{ij} = 0 otherwise. In general, the size *N* of the spin matrix is equal to the number of vertices in the problem and all diagonal elements are zero. As



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Hex
0	0	1	1	1	0	0	0	1	1	0	1	1	0	0	0	1	8d8d
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0005
2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0003
3	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0011
4	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0028
5	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0050
6	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	00A0
7	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0041
8	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0201
9	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0500
10	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0201
11	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1001
12	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	2800
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	5000
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	A000
15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4001

Figure 1. Spin matrix encoding showing routing between 15 customers

connections in the network are bi-directional the spin matrix is symmetric. The potential energy H_{pot} for the system can be defined as:

$$H_{pot} = \sum_{z} \sum_{<i,j>} d_{z,i,j} S_{z,i,j} \quad (3)$$

In equation (3), $d_{z,i,j}$ is the distance between vertex i and vertex j in the z^{th} replica and <i,j> is used to signify counting each connection only once [6]. In any one replica the classical potential energy is the total length of all routes. It is because the potential energy is calculated over a set of P replicas that we need to divide H_{pot} by P in equation (1) to determine the average value. The H_{pot}/P term represents the solution we seek provided vehicle capacity is treated as a constraint so that a customer cannot be added to a route if the vehicle capacity Q is exceeded. The solution space can be repeatedly sampled until a route configuration is found which does not violate vehicle capacity constraints and this is the approach we use in this work.

To calculate a kinetic energy each spin matrix (replica) needs to know the state of its neighbors. The kinetic energy of the entire system is calculated as:

$$H_{kin} = \sum_{z} \sum_{<i,j>} S_{z-1,i,j} S_{z,i,j} + \sum_{z} \sum_{<i,j>} S_{z,i,j} S_{z+1,i,j} \quad (4)$$

Here the first term of equation (4) is the sum of products of the current replica with its previous replica over all P replicas where z is the replica index. Likewise, the second term is the sum of products of the current replica and its next replica over all P replicas. We connect the first and last replicas together to

make a circular list so that the first and last replicas have previous and next neighbors. Like [6] we do not use single spin-flip moves but use local VRP neighborhood operators when attempting to improve a configuration to ensure that a valid network results. That is to say we are using a constrained Ising model [6].

III. QUANTUM ANNEALING ALGORITHM

The pseudo code for our Path-Integral Monte Carlo (PIMC) quantum annealing algorithm for CVPR is shown as Fig. 2. The quantum annealing CVRP algorithm (QACVRP) requires the following input parameters; the number of replicas P, the temperature T, gamma Γ, the rate of decrease of gamma ΔΓ, the maximum number of Monte Carlo steps M_c and a set of N neighborhood operators (insert, swap, 2-opt, cross, scramble, string-insert and 2-opt-Star). The insert operator moves one customer from one route to another route. The swap operator randomly chooses two customers from different routes and exchanges them. The 2-opt operator randomly selects two non-adjacent edges of a single route, and then reverses the connection order of the customers between the outer endpoints. The cross operator randomly chooses two sequences from different routes and swaps them. The scramble operator randomly chooses two customers from one route and shuffles the connection order between them. The string-insert operator removes a sequence of connections from one route and places it into a different route. The 2-opt-Star operator exchanges random length end portions of two routes, preserving the order of connections between customers. For a further discussion on VRP neighborhood operators see Bräysy and Gendreau [9].

On line 1 we compute S as an initial set of spin matrix replicas forming a circular list. The initial population of P spin replicas is generated using a random constructive algorithm so that a diversified set of spin matrix replicas is built. The random construction algorithm randomly chooses a number of routes and then randomly selects and inserts vertices, one at a time, into partial routes until a feasible solution is obtained taking capacity constraints into account. On line 2 we initialize the best spin matrix replica solution. On line 3 we start the Monte Carlo loop using the Monte Carlo maximum step value M_c. We calculate J_Γ on line 4 and on line 5 we set the initial value of the index z of the current replica to zero. On line 6 the replica loop is started. On line 7 we randomly choose a local operator from the set of operators (insert, swap, 2-opt, cross, scramble, string-insert and 2-opt-Star) and then create a modified replica solution s' using the chosen operator (line 8). On line 9 we compute a difference in potential energy using the modified replica solution s' and the unmodified replica S_z. On line 10 we calculate the difference in spin products, again using both the modified replica solution s' and the unmodified replica S_z to determine if the kinetic energy has changed. We calculate the total energy change on line 11.

On line 12 we check if either the change in potential energy ΔH_{pot} or the change in total energy ΔH has decreased

Variable definitions: P : number of replicas, T : temperature, Γ : coupling strength, $\Delta\Gamma$: decrement value of coupling strength, M_c : number of Monte Carlo steps, N : set of neighborhood operators, S : set of spin-matrix replicas, s_{best} : best solution

QACVRP(P, T, Γ, $\Delta\Gamma$, M_c, N)
1. S←circular list of P feasible randomised solutions
2. s_{best} ←S_0
3. while(M_c >= 0)
4. J_Γ←(–T/2) * ln(tanh(Γ/PT))
5. z←0
6. while(z < P)
7. Randomly choose $n \in N$
8. s'←n(S_z)
9. ΔH_{pot}←$H_{pot}(s')$ – $H_{pot}(S_z)$
10. ΔH_k←SpinProducts(S_{z-1}, s', S_{z+1}) – SpinProducts(S_{z-1}, S_z, S_{z+1})
11. ΔH←(ΔH_{pot}/P) + $J_\Gamma \Delta H_{kin}$
12. if (ΔH_{pot}<=0) OR (ΔH<=0)
13. then S_z←s'
14. else if exp(-ΔH/T) > random(0, 1)
15. then S_z←s'
16. if $H_{pot}(S_z) < H_{pot}(s_{best})$
17. then s_{best}←S_z
18. z←z+1
19. Γ←Γ – $\Delta\Gamma$
20. M_c←M_c – 1
21. return s_{best}

Figure 2. Pseudo code for QACVRP

and, if so, we assign the modified solution to S_z (line 13). If the cost value of the modified replica solution is worse than the previous value it can still be accepted (line 14) using the criteria: exp(-ΔH/T) > random(0, 1). That is, the value exp(-ΔH/T) is compared to a random number between 0 and 1 and, if greater, the modified solution s' is assigned to S_z. The acceptance of a worse solution (Metropolis criteria) is controlled by the change in total energy ΔH and the temperature, providing a mechanism for accepting worse solutions with the aim of escaping possible local minima.

On line 16 we calculate the potential energy of the current replica solution S_z and, if it is better than the current best potential energy, then S_z is assigned as the current best solution (line 17). Line 18 increments the current replica index z and ends the inner replica calculation loop. Line 19 provides an option to decrement gamma. Line 20 decrements the Monte Carlo step value ending the Monte Carlo step loop. Line 21 returns the best replica solution found.

Notice that our algorithm calculates changes in energy rather than calculating the Hamiltonian energy for the entire system at every iteration which would be computationally

Variable definitions: S_{z-1}, S_z, S_{z+1} : spin matrices

SpinProducts(S_{z-1}, S_z, S_{z+1})

1. $e = \sum_{i,j} S_{z-1,i,j} S_{z,i,j} + \sum_{i,j} S_{z,i,j} S_{z+1,i,j}$
2. return e

Figure 3. Pseudo code for the SpinProducts() function

expensive especially for non-trivial problem instances where P is large. We calculate ΔH_{pot} as simply the difference between the potential energy of the candidate s' and the previously calculated potential energy value. We calculate ΔH_k as the difference between the spin product of the candidate s' and the previously calculated spin product state. Fig. 3 shows the pseudo code for the SpinProducts(S_{z-1}, S_z, S_{z+1}) function used when calculating the kinetic energy difference.

The QACVRP algorithm described in Fig. 2 has been implemented in C++ using the Qt framework and compiled using the GNU GCC on a PC running a Linux operating system. Our computer hardware set up consisted of two cores and independent simulations were run on each of the cores to obtain the experimental results discussed in section V.

With the PIMC scheme the accuracy of the quantum simulation increases for larger values of P but using larger P requires more memory and computation [8]. Our VRP spin encoding system is extremely memory efficient allowing us to use a large value of P (e.g. P=40) in the simulations that follow. The chosen format of the spin matrix (see Fig.1) simplifies the memory storage for a VRP network. Since connections between customers can be represented in a single bit, memory usage can be reduced by storing each row of the matrix in a bit-stream represented as a hexadecimal number in Fig. 1. On a 64-bit based system each memory word can store up to 64 connections; so for example, a problem instance of 128 customers, with a matrix of (128x128) can be stored with 2x128 memory words. This allows the change in kinetic energy between matrices to be calculated efficiently. Sampling the bit-streams in whole memory words and using logical rather than arithmetic operations allow 64 spin products to be calculated together.

IV. PARAMETER TUNING

The quantum annealing algorithm in Fig. 2 requires that the parameters P, T, Γ and $\Delta\Gamma$ be tuned to ensure robust and high performance. To simplify tuning and to make a quantum annealing algorithm more robust, the rate of decrease of gamma can be set to zero, provided gamma itself is carefully chosen for the problem instance [10]. With this approach only three parameters (P, T, Γ) need to be tuned for good performance. In this work we have devised a methodology for tuning the QA parameters based on empirically measuring the

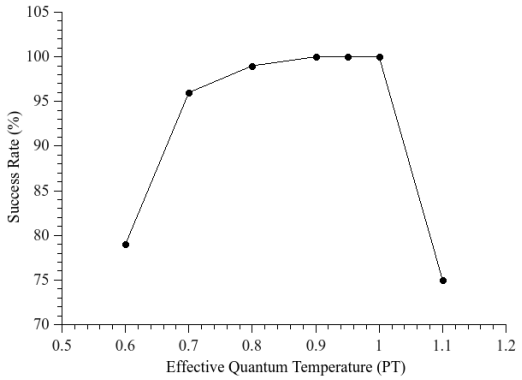


Figure 4. Parameter tuning

success rate of obtaining a known CVRP optimal value as the parameters are systematically changed. We chose the problem instance p-n101-k4 (101 nodes and 4 routes) as our tuning instance which has a known optimal value of 681:4 (best length of 681 with 4 routes). The objective was to determine a set of P, T, Γ values which result in a 100% success rate. A series of computational experiments was performed to enable us to plot the success rate against an effective quantum temperature PT for different values of P. That is, the effective quantum temperature was changed by keeping P fixed and increasing T. In these experiments the Monte Carlo maximum steps value M_c was set to $5x10^6$ and Γ =3. The computational experiments showed that the success rate of finding the best solution for p-n101-k4 increases with the number of replicas used in the simulation. For P=10 the success rate peaks at PT =0.9 with a value of 77%. For P=20 the success rate peaks at PT =0.9 with a value of 91%. When P=40, the success rate is 100% in the range $0.9<PT<1$, as shown in Fig. 4. In all cases the success rate steadily increases as PT is increased and after reaching a peak value it then declines steeply. It is clear by inspecting Fig. 4 that PT=0.9 results in 100% success rate and so P=40, T=22.5 $x10^{-3}$, Γ =3 with M_c = $5x10^6$ have been used as the base set of tuned parameters for this study.

V. EXPERIMENTAL RESULTS

Table 1 shows results of experiments for QACVRP for a set of well-known VRP benchmark instances. Column 1 shows the names of the problem instances with the best known solution (BKS) found using the branch and cut method [11] shown in brackets. Column 2 contains the results obtained from the QACVRP algorithm for 100 independent runs for statistical significance and is subdivided into the best value obtained by our QACVRP algorithm, the wall clock time and the percentage success rate of obtaining the best known solution for a particular instance. In all instances, except M-n121-k7 and F-n135-k7 (discussed below), we used the parameters P=40, T=22.5 $x10^{-3}$, Γ =3 with M_c = $5x10^6$. Column 3 contains the results obtained for a fixed temperature SA for 100 independent runs using the same local operators as QA on the same hardware configuration. To allow for a fair

comparison both algorithms were run for the same number of iterations (i.e. the SA was run for 200 x 10^6 iterations which is equivalent to QA with a 40 replica inner loop and $5x10^6$ outer Monte Carlo loop). The SA temperature was initially tuned for the p-n101-k4 instance (i.e. T=1) and then, if necessary, optimized for other instances to obtain the best possible results.

As can be seen from Table 1, our QACVRP algorithm matches or outperforms SA in terms of success rate for every case except three of the Augerat set B instances. The QACVRP results are also competitive compared to those reported in the literature for other population heuristics. For example with p-n50-k7 our QA finds the BKS with a success rate of 100% whereas the genetic algorithm result [12] differs from the BKS by up to 7.27%. For p-n101-k4 our QA again has a success rate of 100% whereas the result for the memetic algorithm [13] shows best and worst values.

With more difficult instances, for example M-n121-k7 and F-n135-k7, we found that it is necessary to re-tune the P, T and Γ parameters. The problem instance F-n135-k7, which represents a day of grocery deliveries, is particularly difficult for VRP algorithms [14]. With PT set to 0.3, (i.e. a temperature value of 0.006 with the replica number fixed at 50) and Γ set to 3, we were able to obtain the optimal value. The success rate of obtaining the optimal value was 5% when the number of Monte Carlo steps was set to $5x10^6$. Increasing the Monte Carlo steps to 7.5 $x10^6$ resulted in a success rate of 10%. The best success rate we were able to obtain was 11% with the Monte Carlo steps set to $10x10^6$. Increasing the maximum number of Monte Carlo steps extends the simulation time.

VI. ANALYSIS OF QA

Our QACVRP algorithm requires that a maximum number of Monte Carlo steps is defined. Consequently, we have investigated how the best solution averaged over 100 runs changes as the maximum number of Monte Carlo steps is increased. The results for p-n101-k4 with P=40, T=22.5 $x10^{-3}$ and Γ =3 are shown in Fig. 5 where the best solution equates to the sum of lengths over four routes. Increasing the maximum number of Monte Carlo steps results in the optimal solution (i.e. 681) being obtained when $M_c > 1x10^6$.

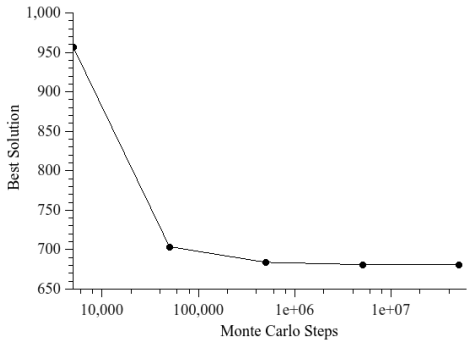


Figure 5. Monte Carlo Steps

Looking at the wall clock times for QAVRP for the benchmark instances shown in Table 1, we observe that in some instances this is larger than the SA wall clock time but in most cases this results in a higher success rate. It is important to note that because QAVRP obtains the BKS values in all cases, this implies that our choice of the spin encoding scheme to compute the kinetic energy term is effective in exploring the energy landscape of all these VRP instances. The artificially induced fluctuations provide an effective solution diversification strategy guiding the search to unexplored regions.

Using the parameters P=40, T=22.5 $x10^{-3}$, Γ =3 and M_c = $5x10^6$ we have calculated the acceptance ratio for p-n101-k4 by sampling solutions over all iterations. As shown in Fig. 6 the curve slowly declines over time and then stabilizes to a plateau level. In this case the effective quantum temperature (PT) is equal to 0.9 and as seen in Fig. 4, yields 100% success rate when finding the optimal solution. However, as Fig. 4 shows, changing the effective quantum temperature to PT=1.1 impacts the solution quality. The same acceptance ratio experiment was repeated with PT =1.1 (i.e. P=40 and T=27.5 $x10^{-3}$) and although the acceptance ratio curve has the same form, a plateau is reached earlier (see Fig. 6), often leading to premature convergence to a local optimum rather than the global value.

VII. CONCLUSION

To our knowledge this is the first study of the use of quantum annealing for solving the Capacitated Vehicle Routing Problem. An important step in applying QA to a problem is to find an intuitive way to encode the problem in terms of a set of spins, such that operators can be used by the QA algorithm to generate new valid solution configurations. The novelty of our encoding approach is that we have mapped VRP connectivity to the spin matrix and used VRP local neighborhood operators, when attempting to improve a configuration, as this ensures that a valid network is always created. Our encoding scheme allows the replicas to communicate between each other in a consistent manner. The performance of our quantum annealing algorithm has been measured by undertaking experiments using standard CVRP benchmarks and compared against a fixed temperature SA algorithm.

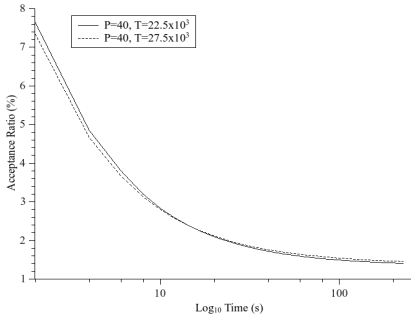


Figure 6. Acceptance ratio for two different effective quantum temperatures

TABLE I. COMPUTATIONAL RESULTS FOR BENCHMARKS

Instance (BKS)	QA Results			SA Results		
	Best value obtained	Time	Success rate %	Best value obtained	Time	Success rate %
P-n40-k5 (458:05)	458:05	00:00:50	100	458:05	00:00:26	100
P-n45-k5 (510:05)	510:05	00:01:53	100	510:05	00:36:51	93
P-n50-k7 (554:07)	554:07	00:09:14	100	554:07	00:04:45	100
P-n50-k10 (696:10)	696:10	08:27:14	63	696:10	03:57:49	28
P-n51-k10 (741:10)	741:10	01:12:35	100	741:10	02:39:49	50
P-n55-k7 (568:07)	568:07	01:11:40	100	568:07	00:30:58	99
P-n55-k10 (694:10)	694:10	10:14:24	35	694:10	02:26:43	31
P-n60-k10 (744:10)	744:10	01:07:38	100	744:10	01:27:26	73
P-n60-k15 (968:15)	968:15	08:48:46	79	968:15	03:12:11	79
P-n65-k10 (792:10)	792:10	01:01:47	100	792:10	00:30:04	100
P-n70-k10 (827:10)	827:10	13:23:50	78	827:10	05:25:02	61
P-n76-k4 (593:04)	593:04	14:50:56	52	593:04	03:50:16	44
P-n76-k5 (627:05)	627:05	10:16:25	87	627:05	04:38:44	22
P-n101-k4 (681:04)	681:04	07:56:20	100	681:04	00:46:35	97
B-n50-k8 (1312:08)	1312:08	01:34:54	100	1312:08	00:10:10	100
B-n52-k7 (747:07)	747:07	00:09:06	100	747:07	00:00:46	100
B-n56-k7 (707:07)	707:07	00:20:34	100	707:07	00:01:55	100
B-n57-k9 (1598:09)	1598:09	00:40:50	100	1598:09	00:04:31	100
B-n63-k10 (1496:10)	1496:10	12:49:30	26	1496:10	02:37:01	25
B-n64-k9 (861:09)	861:09	01:04:16	100	861:09	00:17:23	100
B-n66-k9 (1316:09)	1316:09	08:56:25	91	1316:09	02:45:03	44
B-n67-k10 (1032:10)	1032:10	15:42:00	42	1032:10	01:14:48	88
B-n68-k9 (1272:09)	1272:09	11:30:52	69	1272:09	01:29:32	87
B-n78-k10 (1221:10)	1221:10	07:10:29	97	1221:10	00:36:48	99
M-n121-k7[a] (1034:07)	1034:07	03:08:51	90	1034:07	04:57:44	76
F-n135-k7[b] (1162:07)	1162:07	00:45:25	11	1162:07	02:59:43	4

a. P=50, T=12 x10^{-3}, Γ =3, M_c = 5x10^6

b. P=50, T=6 x10^{-3}, Γ =3, M_c = 10x10^6

The primary conclusion is that our quantum annealing algorithm finds BKS values for all benchmarks used in this study and outperforms SA in terms of success rate in the majority of cases.

REFERENCES

[1] G. Dantzig and R. Ramer, "The truck dispatching problem", Management Science, 6, pp. 80-91, 1959.

[2] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points", Operations Research, 12, pp. 568–581, 1964.

[3] C. Groër, B. Golden, and E. Wasil, "A library of local search heuristics for the vehicle routing problem", Math. Prog. Comp. 2, pp. 79-101, 2010. doi:10.1007/s12532-010-0013-5.

[4] G. Rose, and W. G. Macready, "An introduction to quantum annealing", DWAVE Technical Document 0712, 2007. http://dwave.files.wordpress.com/2007/08/20070810_d-wave_quantum_annealing.pdf (Accessed 28th April 2013).

[5] O. Titiloye and A. Crispin, "Quantum annealing of the graph coloring problem". Discrete Optimization 8(2), pp. 376-384, 2011.

[6] R. Martonák, G. E. Santoro, E. Tosatti, "Quantum annealing of the traveling salesman problem", Phys. Rev., E 70, 057701, 2004.

[7] A. Das and B. K. Chakrabarti, "Quantum annealing and analog quantum computation", Rev. Mod. Phys., 80, pp. 1061–1081, 2008.

[8] D. A. Battaglia, G. E. Santoro, E. Tosatti, "Optimization by quantum annealing: lessons from hard satisfiability problems", Phys. Rev. E 71, 066707, 2005.

[9] O. Bräysy, and M. Gendreau, "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms", Transportation Science 39:1, pp. 104-118, 2005.

[10] O. Titiloye and A. Crispin, "Parameter tuning patterns for random graph coloring with quantum annealing", 2012. PLoS ONE 7(11): e50060. doi:10.1371/journal.pone.0050060.

[11] T. Ralphs, "Branch cut and price resource, vehicle routing data sets", 2003. http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm (accessed 24th April 2013).

[12] Á. S. Bjarnadóttir, "Solving the Vehicle Routing Problem with genetic algorithms", MSc Thesis, 72, Technical University of Denmark. 2004. http://etd.dtu.dk/thesis/154736/ (accessed 28th April 2013).

[13] I. Borgulya, "A memetic algorithm for the Capacitated Vehicle Routing Problem", Acta Agraria Kaposváriensis, 12, No. 2, pp. 59-69, 2008.

[14] A., Abraham, C. Grosan, and W. Pedrycz, "Engineering evolutionary intelligent systems", Springer, 2008. ISBN 978-3-540-75395-7.