Computers and Operations Research 87 (2017) 52–62

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Large-scale vehicle routing problems: Quantum Annealing, tunings and results

A. Syrichas*, A. Crispin

Manchester Metropolitan University, School of Computing, Mathematics and Digital Technology, John Dalton Building, Chester Street, Manchester, M15 5GD, United Kingdom

ARTICLE INFO

Article history:
Received 18 May 2016
Revised 8 May 2017
Accepted 23 May 2017
Available online 25 May 2017

Keywords:
Quantum
Annealing
CVRP
DCVRP
Tuning

ABSTRACT

Quantum Annealing was previously applied to the vehicle routing problem and the results were promising. For all benchmark instances in the study, optimal results were obtained. However, 100% success rate was not achieved in every case, and tuning the control parameters for larger instances proved cumbersome. This work addresses these remaining difficulties. An empirical approach is taken wherein measurements of run-time behaviour are exploited to transform existing good values of control parameters so that they can be used successfully for other problem instances. The course of this work shows a method which simplifies hand-tuning so that the heuristic performs successfully when applied to larger instances, and also demonstrates a tuning method which establishes control parameter values for instances which belong in broadly defined groupings. In addition, new best known solutions for large-scale instances, and initial results for the distance-constrained variant of the vehicle routing problem are presented.

1. Introduction

Previous research Crispin and Syrichas (2013) demonstrated the effectiveness of Quantum Annealing (QA) for solving many instances of the Capacitated Vehicle Routing Problem (CVRP). Optimal results were obtained for all benchmark instances by applying a single set of values for the algorithm's control parameters - values which were methodically determined to achieve the maximum success rate for a reference instance. The success rate is the percentage of the number of times the algorithm finds the best known score for a given instance over a number of runs.

Table 1 shows an excerpt of the instances for which this parameter set was unable to achieve 100% success rate. (An indication of the complexity of each instance can be inferred from the name. P-n101-k4 for example, has 101 nodes/customers served by 4 vehicles whereas M-n121-k7 has 121 served by 7.) Notably, the scores for smaller instances were much lower than for the reference instance. This is contrary to intuition, that one might expect parameters giving the best results for a larger instance would perform easily as well for smaller instances. (One may expect also that parameters for smaller instances will not work well for larger ones.) Given that the local search method is effective enough to allow the metaheuristic to find the optimal solution in at least 11% of the experiments, and that many of the instances appear less complex than the reference, one can conclude that the values of the control parameters are incorrect. If the temperature value is set too high or the magnetic field is too strong, convergence to a minimum is slowed down or inhibited completely. If set too low, the rate of convergence is high and entrapment at poor local minima is likely. If the population size is too small, the search of solution space covers only a reduced area which may not contain optimal solutions. It seems clear that the universal application of a single set of control parameters will not guarantee consistently good performance and the algorithm requires tuning on a case-by-case basis.

How then does one tune metaheuristic control parameters for best results? One could apply the tuning methodology for every instance, providing a specific set of control parameters for each. To save time, the processes of the methodology could be captured, encoded, and then left to a computer program to automatically decide the parameters. These approaches work because feedback can be derived from information known beforehand about the optimal result. Benchmark instances are often supplied with deterministically proven optimal solutions. However, for larger benchmarks, and in dynamic or industrial applications where problem instances are created in real-time, such information is limited or non-existent.

Additional tuning difficulties are presented by metaheuristics with two or more control parameters, each of which may be tightly interdependent. For example, the coupling term used in QA is a

* Corresponding author.
E-mail address: alex.syrichas@gmail.com (A. Syrichas).

http://dx.doi.org/10.1016/j.cor.2017.05.014

Table 1
Computational results excerpt.

Instance	QA Success %	SA Success %
P-n101-k4 (reference)	**100**	**100**
P-n50-k10	63	28
P-n55-k10	35	31
P-n60-k15	79	79
P-n70-k10	78	61
P-n76-k4	52	44
P-n76-k5	87	22
B-n63-k10	26	25
B-n66-k9	91	44
B-n67-k10	42	88
B-n68-k9	69	87
B-n78-k10	97	99
M-n121-k7	90	76
F-n135-k7	11	4

non-linear function of magnetic field strength and effective temperature. This term is extremely sensitive to variations in either parameter, and tuning is further complicated because the Metropolis criteria (Metropolis et al., 1953) is simultaneously dependent upon temperature. A Design of Experiments (DoE) method (Ridge and Kudenko, 2010) can be helpful in uncovering major dependencies between such variables, but a course of factorial experiments can be time-consuming, and predicting the ranges for numerous and sensitive variables is difficult without once again resorting to guesswork or serially hand-tuning.

It is for reasons like these that metaheuristics with fewer control parameters are attractive - they are simpler to tune. Late Acceptance Hill Climbing (Burke and Bykov, 2017) has a single parameter controlling the size of a fitness array which acts as a 'memory' of good solutions. Cuckoo Search is reported (Nie et al., 2014) to be superior to Genetic Algorithms in part because of having only two parameters - nest abandonment rate and population size. In QA, it has been shown (Titiloye and Crispin, 2011) that the number of parameters can be reduced by one, by setting the magnetic field value to be constant. This idea can be greatly extended by making the whole coupling term a constant, thereby removing the mutual dependence of the effective temperature and magnetic field parameters. With the key parameters uncoupled from one another, time is saved when determining their values by hand. Large-scale problems and instances of the Distance-constrained Capacitated Vehicle Routing Problem (DCVRP) may be tackled without tedium. Furthermore, if some means other than hand-tuning can predict the value of temperature, a single variable remains to be tuned - the replica count (population size).

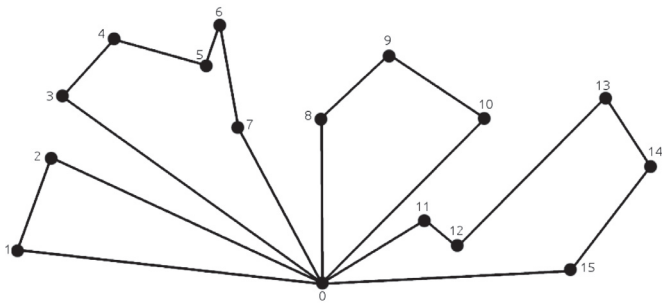
2. Quantum Annealing

Quantum Annealing is an energy-based metaheuristic which uses the Path-Integral Monte Carlo (PIMC) method (Battaglia et al., 2005) to approximate the ground state of the Ising Model. The fitness function is described (1) by the Hamiltonian

$$H = H_p + H_k \tag{1}$$

where the cost H is the sum of potential energy H_p and fluctuations in kinetic energy H_k. H_k is the term which represents the quantum mechanical phenomenon of tunnelling, where a particle trapped in a low energy state, can 'tunnel' through high potential barriers into a lower state. This effect can be simulated in a metaheuristic by using an Ising Model representation of the optimization problem. In simple terms, this is maintaining a population P of simultaneously evolving solutions called replicas, where H_k is calculated from an interaction between adjoining replicas.

When QA is applied to an optimization problem, H_p takes the role of the cost of a solution (for VRP, see (4)), while H_k is a



	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	8D8E
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	0	0	1	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

Fig. 1. Example of routes encoded as a spin matrix. Customer-customer connections are represented using single bits. The matrix is encoded row-wise to form hexadecimal (Hex) words which stored in memory as an array.

scaled sum of the spin interactions between P neighbouring solutions held in a circular list.

$$H_k = J_\Gamma \sum_P \sum_i \sigma_{P-1,i}\sigma_{P,i}\sigma_{P+1,i} \tag{2}$$

Each replica represents the solution as a spin matrix σ containing i elements which can assume values of $\{-1, +1\}$. The interaction energy between the spins of adjoining replicas is generated by the term, $\sigma_{P-1,i}\sigma_{P,i}\sigma_{P+1,i}$

J_Γ is the coupling term which is normally varied during the annealing process via adjustments to the magnetic field strength Γ, amplifying or attenuating the interactions between replicas.

$$J_\Gamma = \frac{-T}{2} \ln \tanh\left(\frac{\Gamma}{PT}\right) \tag{3}$$

Consequently, this contributes towards the acceptance of H in the Metropolis criteria.

2.1. QA for CVRP, and the PT tuning method

CVRP is a variant of VRP in which all vehicles are subject to the same capacity constraint Q. CVRP is an undirected graph $G = (V, E)$ consisting of the vertex set $V = \{v_0, v_1, \ldots v_n\}$ and edge set $E = \{(v_i, v_j)|v_i, v_j \in V, i < j\}$. The restriction $i < j$ ensures the distance between a pair of vertices is identical in both directions. The first vertex is usually considered to be the depot from which a fleet of trucks m serves n customers, whose locations are represented by a vertex set, and have varying demands for goods q_i . The goal is to minimize the number of routes and/or total distance travelled by the trucks d_{ij}. QA for CVRP (QACVRP) uses a two-dimensional spin matrix in which the elements represent customer-customer connections that form routes for each truck. A non-zero cell in

Variables: P: number of replicas, T: temperature, Γ: magnetic field, $\Delta\Gamma$: decrement value of magnetic field, M_C : number of Monte Carlo steps, N: set of neighbourhood operators, n: chosen operator, S: set of replicas, S_{best} : best solution, s': candidate solution, z: index of current replica.

```
1:  S ← circular list of feasible randomized solutions
2:  S_best ← S_0
3:  while M_C > 0 do
4:      J_Γ ← (−T/2)ln(tanh(Γ/PT))
5:      z ← 0
6:      while z < P do
7:          randomly choose n ∈ N
8:          s' ← n(S_z)
9:          ΔH_p ← H_p(s') − H_p(S_z)
10:         ΔH_k ← Interact(S_{z−1}, s', S_{z+1}) − Interact(S_{z−1}, S_z, S_{z+1})
11:         ΔH ← (ΔH_p/P) + J_Γ ΔH_k
12:         if ((ΔH_p <= 0) or (ΔH <= 0)) then
13:             S_z ← s'
14:         else if exp(−ΔH/T) > random(0, 1) then
15:             S_z ← s'
16:         end if
17:         if H_p(S_z) < H_p(S_best) then
18:             S_best ← S_z
19:         end if
20:         z ← z + 1
21:     end while
22:     Γ ← Γ+ΔΓ
23:     M_C ← M_C − 1
24: end while
25: return S_best
```

Fig. 2. Quantum Annealing for Capacitated Vehicle Routing Problems.

the matrix indicates a path between two vertices and because these connections are bi-directional, the spin matrix is symmetric. Fig. 1 shows this arrangement using an example of 15 customers serviced by 4 vehicles. The hexadecimal value shows how rows of connections, each expressed as a single bit, may be encoded into a single memory word. In any one replica, the classical potential energy is the total length of all routes and so H_p for the whole ensemble is given by (4).

$$H_p = \sum_p \sum_{i,j} d_{p,i,j}\sigma_{p,i,j} \tag{4}$$

The local search scheme used in QACVRP is comprised of a set of neighbourhood operators N={*Move, Swap, Move-string, Swap-string, 2-Opt, 2-Opt**} from which one is selected at random and applied repeatedly until a feasible configuration is found. The Move operator selects a customer and transplants randomly to another position. The Swap operator selects two customers at random and exchanges them. Array-based counterparts of these operators are Move-string and Swap-string which work with randomly sized sets of contiguous customers instead of individuals. The 2-Opt operator selects at random two non-adjacent edges of a single route and then reverses the connection order of the customers between the outer endpoints. 2-Opt* exchanges randomly-sized end portions of two routes, preserving the order of the connections between customers.

Fig. 2 shows the QACVRP implementation used previously by Crispin and Syrichas (2013). Lines 1 and 2 initialize the replicas with randomized feasible solutions, setting the current best solution from the first replica. Lines 3 and 24 define the outer loop which terminates when the number of remaining iterations (Monte Carlo steps) reaches zero in line 23. For each iteration of the outer loop, J_Γ is calculated from the current parameter values (line 4) and the replica index is reset (line 5).

Lines 6 and 21 define the inner loop which terminates when the replica index reaches the number of replicas in line 20. Line 7 selects an operator at random from the set of neighbourhood operators. On line 8 the solution belonging to the replica which is currently indexed is modified by the chosen operator until a feasible candidate is returned. The difference in potential energy is calculated on line 9 using the candidate and the replica solutions. On line 10 the difference in kinetic energy is calculated by computing the interaction terms (2) between the candidate and the replica solutions. The total change in energy is calculated on line 11. Line 12 checks if the difference in potential energy or total energy has decreased and if so, the candidate solution is assigned to the current replica (line 13). If the energy difference has increased (the solution cost is worse) the candidate solution can still be accepted (line 15) if the probabilistic check on line 14 is passed. The acceptance of a worse solution is controlled by the change in total energy and the temperature (Metropolis criteria). On line 17 the cost of the current replica solution is calculated and compared with the best recorded solution. If better, it is assigned as the best solution on line 18.

Once all replicas have been updated in this manner, the inner loop relinquishes control to the outer whereupon the magnetic

Table 2
Variable assignments for factorial experiments.

	T	G_0	G_1	P	M_C
Nominal	80	3000	1050	10	5,000,000
+1	2.5	500	500	4	500,000
−1	−2.5	−500	−500	−4	−500,000

Table 3
Results of factorial experiments using CVRP instance P-n101-k4.

T	P	G_0	G_1	M_C	Av. Score
−1	−1	−1	−1	−1	685.608
−1	−1	−1	−1	+1	684.863
−1	−1	−1	+1	−1	685.585
−1	−1	−1	+1	+1	685.485
−1	−1	+1	−1	−1	685.748
−1	−1	+1	−1	+1	685.580
−1	−1	+1	+1	−1	685.954
−1	−1	+1	+1	+1	684.804
−1	+1	−1	−1	−1	683.009
−1	+1	−1	−1	+1	682.610
−1	+1	−1	+1	-1	682.012
−1	+1	−1	+1	+1	681.913
−1	+1	+1	−1	−1	682.312
−1	+1	+1	−1	+1	681.997
−1	+1	+1	+1	−1	681.758
−1	+1	+1	+1	+1	681.579
+1	−1	−1	−1	−1	685.134
+1	−1	−1	−1	+1	684.793
+1	−1	−1	+1	−1	685.542
+1	−1	−1	+1	+1	685.055
+1	−1	+1	−1	−1	685.140
+1	−1	+1	−1	+1	684.523
+1	−1	+1	+1	−1	684.895
+1	−1	+1	+1	+1	684.945
+1	+1	−1	−1	−1	684.128
+1	+1	−1	−1	+1	683.732
+1	+1	−1	+1	−1	683.084
+1	+1	−1	+1	+1	682.725
+1	+1	+1	−1	−1	682.924
+1	+1	+1	−1	+1	682.609
+1	+1	+1	+1	−1	682.561
+1	+1	+1	+1	+1	682.224

field value is adjusted (line 22) for use in the next iteration. Once all outer loop iterations are complete, the best solution found is returned on line 25.

Previously, QACVRP was employed to solve CVRP instances with a single set of control parameters determined using a method entitled *PT Tuning*. A factorial DoE study was first undertaken to expose any dominance amongst control parameters for the chosen variable assignments (Table 2). Since there are five parameters and two adjustments (+1/−1) allowed to their nominal values, 2^5 experiments were performed, each consisting of 100 attempts to solve the same CVRP instance. Table 3 shows the effect of parameter interactions upon average solution cost. From this, contributions can be calculated for all possible interaction sets. For any chosen set, an average is taken from the scores where the product of the adjustments is equal to 1, and another is taken where the product is equal to −1. The two averages are used as endpoints to form a line, the steepness of which indicates the size of the contribution for the chosen set. The absolute gradient value for each parameter interaction set is presented in Fig. 4.

Although this shows that P alone makes the greatest single contribution, it is awkward to adjust P and maintain good success without also making compensatory adjustments to T and Γ. (This observation motivated subsequent research efforts to uncouple the parameters to ease tuning). However, the study supports the idea of tuning P and T as a single term with the evidence that PT makes the second greatest contribution to average solution cost. Monte

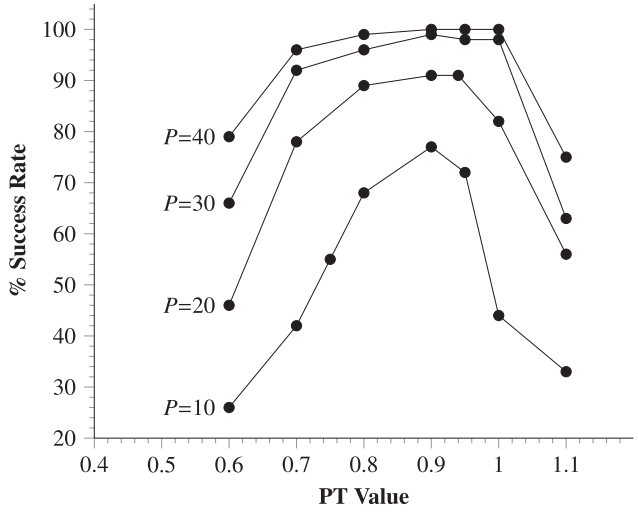


Fig. 3. *PT* tuning using instance P-n101-k4. Success improves with P.

Carlo steps M_C, and then magnetic field start and end values (G_0, G_1) are the next largest contributors. Naturally, larger values of M_C improve results by allowing longer search times. Temperature by itself has the lowest single contribution, with the effects of T likely being swamped by high values assigned to the magnetic field (3).

As shown in Titiloye and Crispin (2011), Γ was held constant during PT tuning whilst suitable values for temperature T and P were determined by experimentation. The effective quantum temperature PT was plotted (Fig. 3) against the success rate over a series of experiments using a reference instance. The reference instance was selected because it has the approximate median number of customers within the chosen benchmarks of Crispin and Syrichas (2013), which range from 50 to 262 customers. This was deemed to be of average complexity since there are 91 instances with fewer than 100 customers and only 10 instances with 100+ customers (Ralphs). While varying T against several fixed values of P, the value of PT which gave the best success rate was noted. The values of Γ, P, and T at this point formed the reference control parameter set $C_{ref} = \{\Gamma_{ref}, P_{ref}, T_{ref}\}$. It was then assumed that applying C_{ref} to any benchmark instance in the study would yield good success rates.

2.2. Simplifying tuning: parameters uncoupled

When attempting to tune the algorithm for use with larger or more complex instances, the difficulty of adjusting non-linearly codependent variables is compounded with the need to allow increased run times.

Effective quantum temperature and magnetic field strength are the variables used to calculate the coupling term J_Γ which scales the energy generated by interactions of spins amongst the replica ensemble. Consequently, the value of H_k is highly sensitive to small changes in P, T or Γ, and because T is also the variable which governs probabilistic acceptance in the Metropolis criteria, tuning is frustrating. This situation is exacerbated when larger instances are involved. The computational workload increases exponentially with problem size, and so results which can provide feedback to retune the parameters are delivered at longer intervals.

If J_Γ was established beforehand and kept constant whilst annealing, Γ can be ignored while T would have a role limited to governing thermal effects via the Metropolis criteria. In QACVRP, the Hamiltonian took the form

$$H = \frac{\Delta H_p}{P} - J_\Gamma \Delta H_k \tag{5}$$

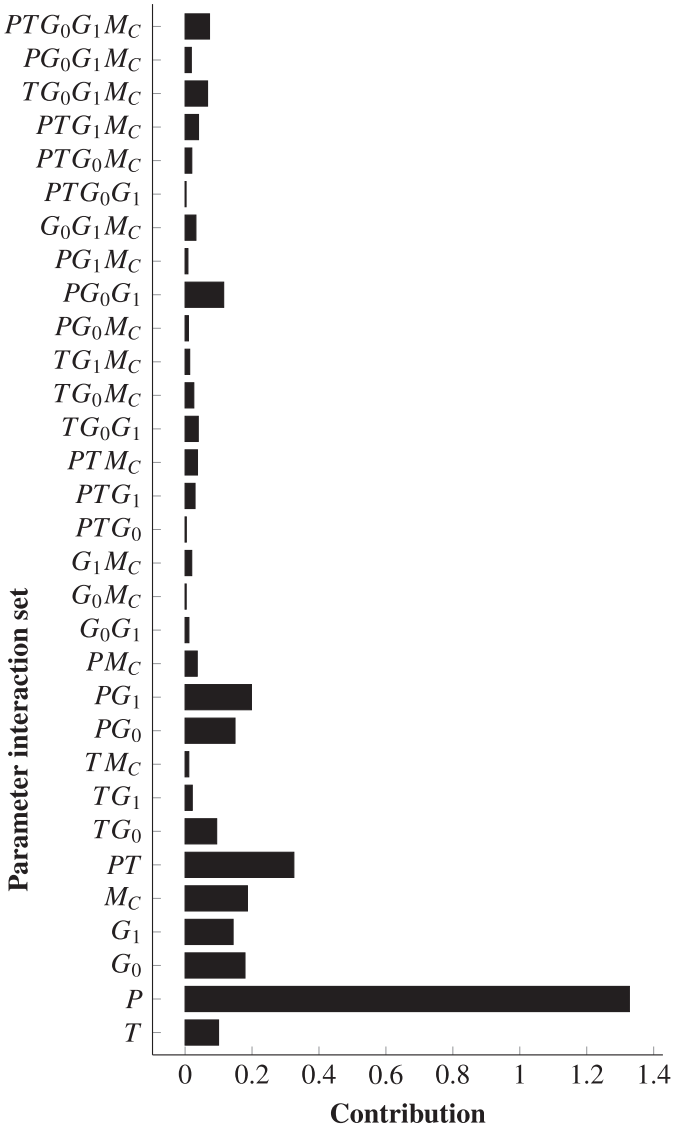


Fig. 4. Main and interaction effects of parameters P, $\Gamma(G_0, G_1)$, T and M_C upon solution cost.

Removing the averaging term P from (5) uncouples another control parameter and would further simplify tuning - P could be increased, regardless of temperature, to improve the success rate. The Hamiltonian for such an arrangement is shown in (6).

$$H = \Delta H_p - J_\Gamma \Delta H_k \tag{6}$$

The isolation of T requires that the annealing process be divided into two phases, each separately taking advantage of H_p and H_k. In the first phase, higher temperatures may be chosen so that H_p dominates H_k in (6). This 'thermal' phase results in a process akin to Simulated Annealing (SA) but which includes small, non-negligible quantum fluctuations. In the second phase, the temperature should be drastically lowered so that H_k dominates in H. In this 'quantum' phase, optimization relies almost exclusively upon the state of the replica ensemble and the interactions therein. In both phases, P may be adjusted without incurring the need to retune the other parameters. It may be increased to improve the accuracy of PIMC, with the expectation that a better result may be found at the expense of run time. Conversely, P may be decreased to speed up annealing at the expense of accuracy.

This two-phase approach may be employed to ease the tuning process, by repeating the thermal phase with small values for P and M_C to establish a good value for T. With T established, a final run of the first phase can be done with increased P and M_C. The resulting solution from this run may be supplied as a starting point for the quantum phase, in a similar fashion to a construction algorithm which supplies an improvement algorithm with an initial solution.

Fig. 5 shows the revised QA algorithm which uses a fixed value of J_Γ and on line 10, the altered Hamiltonian (6).

2.3. Determining temperature: energy-based scaled parameter tuning

With the parameters uncoupled from one another in FJ-QACVRP (*Fixed J_Γ Quantum Annealing for Capacitated Vehicle Routing Problems*), a systematic means of establishing T for the first annealing phase would be of benefit when tackling groups of problem instances, such as the benchmarks of Augerat et al. (1995) as attempted by QACVRP under the *PT* Tuning scheme. It would be ideal to identify and then reproduce for any instance, the runtime behaviour of the algorithm while it is successfully (and consistently) solving a reference instance (Figs. 6 and 7).

Even though success rates in Table 1 were inconsistent, there is promise in using C_{ref} since optimal results were found for every instance. To improve this, some factor needs to be discovered which relates a subject to a reference instance. This factor would have to reliably transform C_{ref} into a subject parameter set $C_{subj} = \{\Gamma_{subj}, P_{subj}, T_{subj}\}$. To find this factor, the fitness landscape Watson (2010) of the problem needs to be considered.

The fitness landscape for a problem instance is the abstract topology formed from the combination of all possible solutions, an objective function, and neighbourhood operators. For a combinatorial optimization problem like CVRP, it is impossible to visualize this landscape. Each point in the landscape represents a possible solution and is a vector composed of the solution cost and configuration, and must be connected to all possible neighbouring solutions. As a whole, this multidimensional topology is difficult to conceive and so must be greatly simplified if to be of any practical help in discovering relationships between instances. To assist with this discovery, the fitness landscape is reduced to an energy landscape. Measurements of potential energy H_p can be recorded and these are equivalent to the cost component of vertices in the landscape. Kinetic energy H_k can also be recorded, but is of less immediate use as a measure of similarity, it gives indirect evidence of how connections between vertices are grouped in the landscape. (In a single phase of FJ-QACVRP with high T, Hamiltonian energy H is imagined to form a similar topology to that formed by H_p, given that H_k makes a relatively tiny contribution to the total energy.) The energy landscape which is explored by a metaheuristic while it solves a problem can be visualized in the form of a scatterplot or Fitness Cloud (Collard et al., 2004). The geometric features of the plot give a visual indication of the run-time behaviour of the algorithm, and therefore may be regarded as a dynamic cost model (Watson, 2010). Fitness Clouds were originally invented to study how mutation and crossover operators caused solutions to evolve in Genetic Programming. They were further analysed to produce metrics for characterising (Vanneschi et al., 2006) and measuring the difficulty of optimization problems (Vanneschi et al., 2006). For this work, they provide visual substance to support the proposal that there are commonalities in the energy landscapes of different problem instances which may be exploited to assist with tuning. Several plots were made from the recordings of energy values as QACVRP solved selected CVRP instances. A cursory analysis of the plots show that in each case, there is quick convergence to, and an intensive search of a region around the optimal solution. More encouragingly, there is a good similarity in shape and structure, which lends support to the idea that one landscape may be approximately transformed into another through the use of scaling or

Variables: P: number of replicas, T: temperature, J_Γ: magnetic coupling strength, M_C : number of Monte Carlo steps, N: set of neighbourhood operators, n: chosen operator, S: set of replicas, S_{best} : best solution, s': candidate solution, z: index of current replica.

$S \leftarrow$ circular list of feasible randomized solutions
$S_{best} \leftarrow S_0$
while $M_C > 0$ **do**
  $z \leftarrow 0$
  while $z < P$ **do**
    randomly choose $n \in N$
    $s' \leftarrow n(S_z)$
    $\Delta H_p \leftarrow H_p(s') - H_p(S_z)$
    $\Delta H_k \leftarrow \text{Interact}(S_{z-1}, s', S_{z+1}) - \text{Interact}(S_{z-1}, S_z, S_{z+1})$
    $\Delta H \leftarrow \Delta H_p + J_\Gamma \Delta H_k$
    if $((\Delta H_p <= 0)$ **or** $(\Delta H <= 0))$ **then**
      $S_z \leftarrow s'$
    else if $\exp(-\Delta H/T) > \text{random}(0, 1)$ **then**
      $S_z \leftarrow s'$
    end if
    if $H_p(S_z) < H_p(S_{best})$ **then**
      $S_{best} \leftarrow S_z$
    end if
    $z \leftarrow z + 1$
  end while
  $M_C \leftarrow M_C - 1$
end while
return S_{best}

Fig. 5. Fixed J_Γ Quantum Annealing for Capacitated Vehicle Routing Problems.

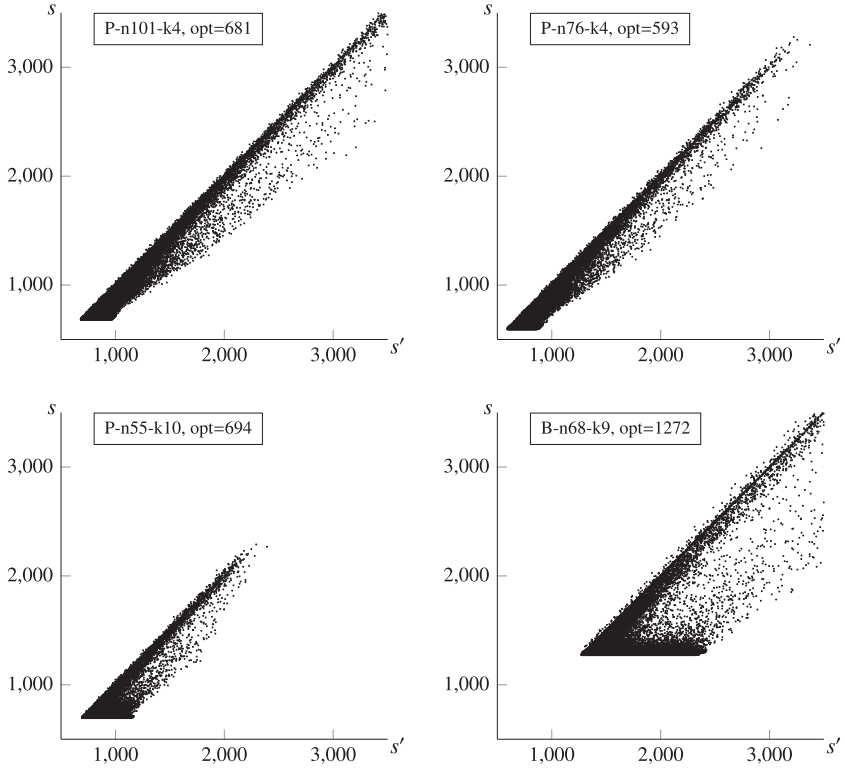


Fig. 6. Fitness Clouds of instances sampled from the benchmarks of Augerat et al. They were generated by QACVRP and are similar in structure.

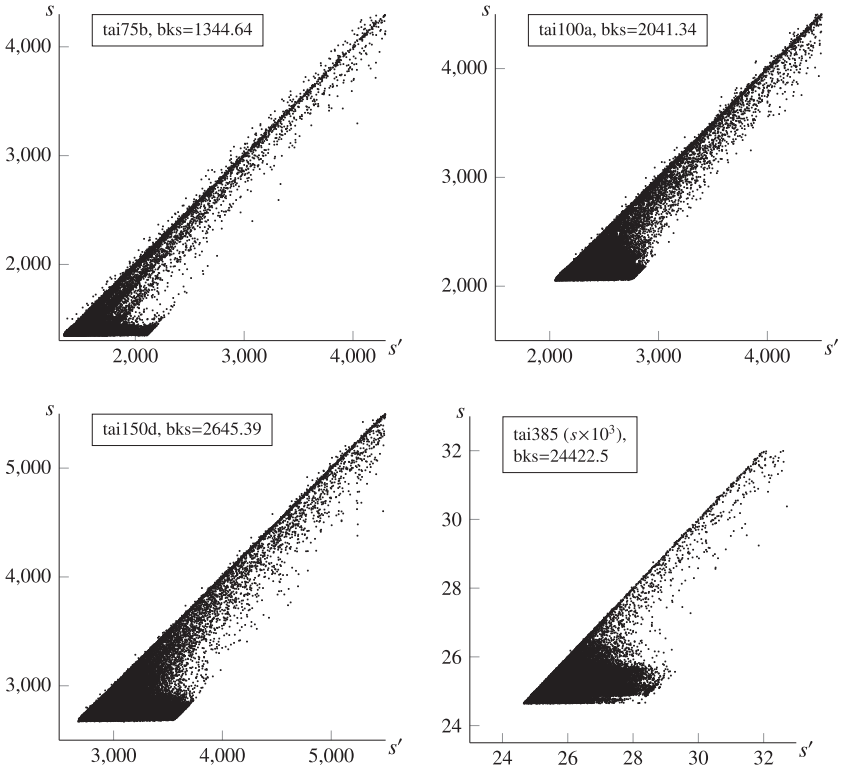


Fig. 7. Fitness Clouds sampled from the instances in the benchmarks of Taillard. They were generated by QACVRP and are similar in structure.

affine transformations. (In CVRP, the Fitness Cloud shows a distribution of the changes to the geometry of the routes, with ΔE_p essentially being a change in distance. Therefore, it should be possible to remap one instance to another.) With such transformations, the aim is to adapt a set of parameters for a reference instance for use in a subject.

For the first phase of FJ-QACVRP, we propose that a useful factor which correlates fitness landscapes is some measure or function of ΔH_p since it has the greatest impact (1) in the Metropolis criteria. If the ratio of such measures from subject and reference instances were known, and since the coupling term J_Γ (3) is now constant, then T for the subject instance can be calculated as the product of this ratio and the reference effective temperature $P_{ref}T_{ref}$. This would then remove the necessity of tuning the temperature parameter. Peak changes in accepted potential energy ΔH_p have already been sampled for several CVRP instances using QACVRP.

The coupling term J_Γ can be thought of as a factor which amplifies or attenuates similarities between population members (2). A good value for J_Γ was discovered in the course of determining the parameters of the reference instance, P-n101-k4 in Crispin and Syrichas (2013). The value of J_Γ which was successful for solving one instance may be used as a constant in solving another, if the temperature is suitably scaled.

When a subject instance is optimized by FJ-QACVRP, the peak change in accepted potential energy ΔE_{subj} is scaled by the normalization constant (7) to provide the value of temperature T_{subj} (8).

$$k = \frac{P_{ref}T_{ref}}{\Delta E_{ref}} \tag{7}$$

$$T_{subj} = k\Delta E_{subj} \tag{8}$$

The following is a brief, stepwise description of the new tuning method, tentatively entitled *Energy-based Scaled Parameter Tuning* (ESPT):

Table 4
Peak changes in accepted potential energy.

QACVRP	
Instance	**ΔE**
P-n101-k4(reference)	13
P-n50-k10	15
P-n55-k10	14
P-n60-k15	14
P-n76-k4	17
P-n63-k10	16
P-n67-k10	15
B-n68-k9	17

1. With QACVRP solving the reference instance, use the *PT* tuning method (or other) to identify the parameter set C_{ref} which gives the best possible success rate.
2. With QACVRP solving the reference instance using C_{ref}, record the peak change in accepted potential energy ΔE_{ref}. The constant k (7) can now be calculated.
3. With QACVRP solving the subject instance using C_{ref}, record the peak change in accepted potential energy ΔE_{subj}.
4. Use FJ-QACVRP to solve the subject instance using a constant value of J_Γ calculated with C_{ref}, $T_{subj} = k\Delta E_{subj}$, $P_{subj} = P_{ref} + p$ where p is a discretionary increase in the number of replicas.

Simple modifications were made to the acceptance function of the software in order to sample the changes in accepted potential energy. QACVRP then solved selected instances 20 times each with $C_{ref} = \{3, 40, 22.5 \times 10^{-}3\}$ determined in Crispin and Syrichas (2013), and with Monte Carlo steps $M_C = 10 \times 10^6$. Table 4 shows measurements made using steps 2 and 3 for several instances listed in Table 1.

Table 5
Computational results of FJ-QACVRP using CVRP instances of Taillard.

			Phase 1		Phase 2[a]		Total	
Problem	n	Best	Score	Time (s)	Score	Time (s)	Time	$M_C \times 10^6$
tai75a	75	1618.36[b]	1618.357	7	–	–	0:00:07	0.031
tai75b	75	1344.64[b]	**1344.619**	4	–	–	0:00:04	0.019
tai75c	75	1291.01[b]	1291.008	5	–	–	0:00:05	0.033
tai75d	75	1365.42[b]	1365.419	6	–	–	0:00:06	0.021
tai100a	100	2041.336[b]	2043.405	92	2041.337	3	0:01:35	0.213
tai100b	100	1939.9(Mester and Bräysy, 2005)	1939.904	1144	–	–	0:19:04	2.778
tai100c	100	1406.202[b]	1406.965	721	1406.202	2	0:12:03	1.487
tai100d	100	1581.25[b]	1582.112	156	**1580.458**	269	0:07:05	1.183
tai150a	150	3055.23[b]	3057.395	608	3055.232	366	0:16:14	2.538
tai150b	150	2727.67(Mester and Bräysy, 2005)	2730.304	958	2727.669	24,618	7:06:16	14.478
tai150c	150	2341.84[b]	2361.424	1754	2358.659	1989	1:02:23	12.898
tai150d	150	2645.39[b]	2647.713	28,146	2645.391	1	7:49:07	37.942
tai385	365	24422.5(De Franceschi et al., 2006)	24449.046	235,760	**24395.411**	84,561	88:58:41	138.535

[a] 50% replicas perturbed for 5 iterations before annealing.
[b] As reported by Alba and Dorronsoro (2006).

Table 6
Computational results of FJ-QACVRP using various CVRP instances.

			Phase 1		Phase 2[a]		Total	
Problem	n	Best	Score	Time (s)	Score	Time (s)	Time (s)	$M_C \times 10^6$
E-n101-k14	100	1067(De Franceschi et al., 2006)	1067	216	–	–	0:03:36	1.564
M-n151-k12	150	1015(De Franceschi et al., 2006)	1015	456	–	–	0:07:36	1.617
M-n200-k16	199	1274(Poggi and Uchoa, 2014)	1274	11,709	–	–	3:15:09	14.004
M-n200-k17	199	1275(Poggi and Uchoa, 2014)	1275	29,959	–	–	8:19:19	13.893
G-n262-k25	261	5530(Liu and Li, 2015)	5530	78,974	**5526**	20	21:56:34	46.060

[a] 50% replicas perturbed for 5 iterations before annealing.

3. Experimental results

Although multicore processor architectures allow for programs to be written to achieve parallelization of their workload, FJ-QACVRP was designed to take advantage in a different way. Rather than breaking the algorithm into parallelizable components, FJ-QACVRP was coded to place a whole experimental run on a separate thread so that it would be allocated to a single CPU core. This design allows many runs to occur simultaneously and so will output results much more quickly. FJ-QACVRP was coded in C++ using the Qt cross-platform application framework. The processor employed was the Intel Xeon E5-2683 v3, having 14 cores with a clock speed of 2 GHz, running the Linux operating system.

3.1. Large and very large scale results

For each of the moderately sized to very large-scale instances, a batch of runs equal in size to the number of CPU cores was performed simultaneously across a range of temperatures. For speed whilst tuning in this first phase, the number of replicas and Monte Carlo steps were kept low, with $P \leq 20$ and $M_C \simeq 20 \times 10^6$. When the batch was completed, the value of T from the run with the best score was considered the most useful and, if necessary, reused with higher values of P to seek further improvements.

The best solution from phase 1 was used to initialize each replica in phase 2. A percentage of the replica ensemble was perturbed by applying a randomized selection of the neighbourhood operators for a fixed number of iterations. A low temperature $T = 0.14$ and higher values of P (40, 60, 80, 160) and $M_C = 50 \times 10^6$ formed the parameter set for runs in the second phase of FJ-QACVRP.

The results of this method are shown in Tables 5–9 and any improvements to the best known scores in literature are shown in bold. Combined run times are shown together with the total number of iterations M_C taken for each instance.

Table 5 show the results of FJ-QACVRP on the CVRP instances in the benchmarks of Taillard (1993). In all but one instance (tai150c) the best known score was found or improved upon (tai75b and tai385), and the majority required both optimisation phases. In phase 1, temperature ranges were bounded by $1.18 \leq T \leq 1.82$ with intervals of 0.04. Combined run times ranged from 7s to 65 h.

Table 6 shows the results of FJ-QACVRP on the moderately sized CVRP instances in the benchmarks of Christofides and Eilon (1969), Gillett and Johnson (1976), and Christofides et al. (1981). In phase 1, temperature ranges were bounded by $0.6 \leq T \leq 1.4$ with intervals of 0.05. For all instances except G-n262-k35, the best known score is suspected to be optimal. Since no better results were achieved by significantly increasing the running time or population size of FJ-QACVRP ($M_C \geq 350 \times 10^6$, $P \geq 640$), and because no improvements have been published for over two years, the second phases were omitted. Combined run times ranged from 3.6 min to 21.9 h.

Tables 7 and 8 show the results of FJ-QACVRP on the benchmark of Golden et al. (1998), which include larger sized CVRP and DCVRP instances respectively. For the CVRP instances the temperature ranges for phase 1 were bounded by $0.26 \leq T \leq 1.4$ with intervals of 0.02. FJ-QACVRP performed well and delivered improvements over the best known score in all cases. For problems 9, 13, 15 and 16, scores from the second phase yielded no improvements over the first and are not shown. Combined run times for problems using both phases ranged from 11.2 to 144.8 h.

For the DCVRP instances, FJ-QACVRP matched the best known in 4 problems and approached to within a negligible fraction of a percent in 3 while new best scores were found for problems 1 and 3. For phase 1 the temperature ranges were within $1.8 \leq T \leq 3$ with intervals of 0.1. Combined run times for problems which required both phases ranged from 4.6 to 27.7 h.

Table 7
Computational results of FJ-QACVRP using Golden et al. CVRP instances.

			Phase 1		Phase 2[a]		Total	
Problem	n	Best	Score	Time (s)	Score	Time (s)	Time	$M_C \times 10^6$
9	255	583.39[b]	**579.713**	16,738	–	–	4:38:58	99.984
10	323	741.7(De Franceschi et al., 2006)	**737.512**	133,303	**737.412**	71	37:02:54	42.440
11	399	918.45[b]	**914.341**	84,969	**912.770**	141,287	62:50:56	23.084
12	483	1107.19[b]	**1102.542**	469,021	**1102.120**	52,312	144:48:53	88.583
13	252	859.11[b]	**857.189**	19,232	–	–	5:20:32	18.273
14	320	1081.31[b]	**1080.830**	106,996	**1080.553**	22	29:43:38	33.608
15	396	1345.23[b]	**1340.241**	140,262	–	–	38:57:42	38.508
16	480	1622.69[b]	**1611.503**	153,502	–	–	42:38:22	68.110
17	240	707.79[b]	707.992	39,557	**707.756**	192	11:02:29	24.406
18	300	998.73[b]	**998.354**	67,251	**995.984**	110	18:42:41	18.509
19	360	1366.8578(De Franceschi et al., 2006)	1368.291	61,744	**1366.591**	1187	17:28:51	189.301
20	420	1821.15[b]	1825.057	75,826	**1818.949**	23,950	27:42:56	140.475

[a] 50% replicas perturbed for 5 iterations before annealing.
[b] As reported by Alba and Dorronsoro (2006).

Table 8
Computational results of FJ-QACVRP using Golden et al. DCVRP instances.

			Phase 1		Phase 2[a]		Total	
Problem	n	Best	Score	Time (s)	Score	Time (s)	Time	$M_C \times 10^6$
1	240	5627.54[b]	**5624.119**	39,441	**5623.469**	43	10:58:04	33.042
2	320	8447.92(Tarantilis and Kiranoudis, 2002)	8464.495	12,007	8447.920	4599	4:36:46	11.112
3	400	11036.23(Tarantilis and Kiranoudis, 2002)	11047.007	56,584	**11036.223**	8085	17:57:49	110.986
4	480	13624.53(Tarantilis and Kiranoudis, 2002)	13632.913	51,109	13624.526	26,960	21:41:09	41.205
5	200	6460.98(Tarantilis and Kiranoudis, 2002)	6460.980	39,913	–	–	11:05:13	24.670
6	280	8412.88(Tarantilis and Kiranoudis, 2002)	8412.902	2966	–	–	0:49:26	1.143
7	360	10195.56(Tarantilis and Kiranoudis, 2002)	10200.543	5148	10195.587	2845	2:13:13	7.849
8	440	11663.55[b]	11681.035	29,603	11672.111	70,097	27:41:40	50.312

[a] 95% replicas perturbed for 30 iterations before annealing.
[b] As reported by Alba and Dorronsoro (2006).

Table 9
Computational results of FJ-QACVRP using Li et al. DCVRP instances.

			Phase 1		Phase 2[a]		Total	
Problem	n	Best(Li et al., 2005)	Score	Time (s)	Score	Time (s)	Time	$M_C \times 10^6$
21	560	16212.83	16232.347	12,550	16212.826	59,302	19:57:32	8.303
22	600	14641.64	**14632.952**	230,304	**14586.109**	363,701	165:00:05	26.302
23	640	18801.13	18824.837	37,464	18801.131	83,571	33:37:15	17.021
24	720	21389.43	21411.744	78,209	21389.432	12,344	25:09:13	23.895
25	760	17053.26	**16905.233**	57,344	**16851.976**	10,479	18:50:23	16.214
26	800	23977.74	23998.635	56,315	**23977.733**	16,977	20:21:32	43.214
27	840	17651.6	**17621.849**	266,076	**17508.388**	51,211	88:08:07	41.357
28	880	26566.04	26586.149	16,027	26566.035	69,905	23:52:12	3.474
29	960	29154.34	29176.634	54,894	29154.337	93,270	41:09:24	7.919
30	1040	31742.64	31774.909	100,519	31742.640	580,037	189:02:36	17.655
31	1120	34330.94	34361.727	287,648	34330.941	239,826	146:31:14	29.521
32	1200	36919.24	37352.460	304,073	37331.111	2139	85:03:32	25.061

[a] 95% replicas perturbed for 30 iterations before annealing.

Table 9 show the results of FJ-QACVRP on the benchmark of Li et al. which is comprised of procedurally-generated (Li et al., 2005) very large-scale DCVRP instances. In all cases but problem 32, FJ-QACVRP equals or improves best known scores. In phase 1 the temperature ranges were within $1.8 \leq T \leq 3.2$ with intervals of 0.1. All attempts required both phases to execute and their combined run times ranged from 18.8 to 189 h.

3.2. ESPT results

Table 10 shows the outcome of performing step 4 of the ESPT method, in which the temperature value for each instance was predicted for use in FJ-QACVRP. For a valid comparison with QACVRP, the chosen benchmarks were those as used in Crispin and Syrichas (2013) - sets B and P benchmarks of Augerat et al. (1995). Both sets contain instances which cluster the locations of customers and are likely to have similar geometric structures in their Fitness Clouds.

Using C_{ref} and ΔE_{ref}, k (7) and J_Γ (3) were calculated. For each subject instance, T_{subj} (8) was calculated using k and ΔE_{subj} (Table 4). With the parameter set $\{J_\Gamma, P_{ref} + p, T_{subj}\}$ and $p \in [-30, +120]$, FJ-QACVRP was used to solve each subject instance 100 times, after which the success rate was recorded.

In almost all cases, at $P \geq 20$, the success rate is increased beyond that achieved by QACVRP. At $P = 40$ there are significant increases for P-n76-k4, P-n76-k5, P-n55-k10 and B-n63-k10, instances that proved troublesome for QACVRP. Fig. 8 shows the effectiveness of FJ-QACVRP upon the latter three and with values of T determined beforehand, that P may alone be adjusted until a desired rate/time balance is obtained. Also encouraging for the ESPT method is that for $P \leq 20$ the results almost entirely improve over QACVRP, indicating the potential to improve running speed.

As expected, in all cases increasing P improves the success rate. The temperature component has been scaled using potential energy ratios and successfully applied to subject instances.

Table 10
Success rates after application of ESPT.

	Success %				
	QACVRP (Crispin and Syrichas, 2013)	FJ-QACVRP			
	P = 40 (time)	P = 10	P = 20	P = 30	P = 40 (time)
P-n101-k4	100 (07:56:20)	99	100	100	100 (00:14:24)
P-n40-k5	100 (00:00:50)	100	100	100	100 (00:00:04)
P-n45-k5	100 (00:01:53)	100	100	100	100 (00:00:05)
P-n50-k7	100 (00:09:14)	100	100	100	100 (00:00:17)
P-n50-k10	63 (08:27:14)	62	80	89	99 (00:26:40)
P-n51-k10	100 (01:12:35)	91	99	100	100 (00:04:15)
P-n55-k7	100 (01:11:40)	100	100	100	100 (00:02:42)
P-n55-k10[a]	35 (10:14:24)	55	87	93	97 (00:39:21)
P-n60-k10	100 (01:07:38)	100	100	100	100 (00:02:42)
P-n60-k15	79 (08:48:46)	86	97	98	100 (00:10:06)
P-n65-k10	100 (01:01:47)	100	100	100	100 (00:01:12)
P-n70-k10	78 (13:23:50)	90	100	100	100 (00:13:38)
P-n76-k4	52 (14:50:56)	91	100	100	100 (00:37:15)
P-n76-k5[a]	87 (10:16:25)	72	93	98	98 (00:57:48)
B-n50-k8	100 (01:34:54)	100	100	100	100 (00:03:34)
B-n52-k7	100 (00:09:06)	100	100	100	100 (00:00:18)
B-n56-k7	100 (00:20:34)	100	100	100	100 (00:01:26)
B-n57-k9	100 (00:40:50)	100	100	100	100 (00:01:06)
B-n63-k10[a]	26 (12:49:30)	25	36	68	68 (01:12:18)
B-n64-k9	100 (01:04:16)	100	100	100	100 (00:02:12)
B-n66-k9	91 (08:56:25)	96	100	100	100 (00:15:54)
B-n67-k10	42 (15:42:00)	100	100	100	100 (00:17:46)
B-n68-k9	69 (11:30:52)	93	99	100	100 (00:33:44)
B-n78-k10	97 (07:10:29)	100	100	100	100 (00:09:42)

[a] *P* value for 100% success shown in Fig. 8.

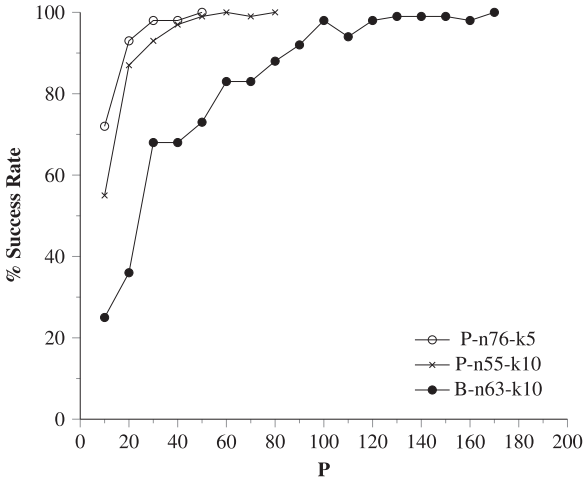


Fig. 8. ESPT - Success rate improves as *P* increases.

4. Conclusions

To our knowledge, this is the first study which reduces the number of control paramaters in QA to one (replica count) and which systematically establishes a constant value for the temperature through the use of scaling factors determined from the analysis of Fitness Clouds.

In principle, the methods and techniques presented could be extended to other vehicle routing problems such as those which use a hetergeneous fleet. A heterogeneous fleet can be represented in a spin matrix by increasing the dimensions by the number of differing vehicles. The additional cells would indicate which vehicle is assigned to each route. Further, the solution representation need not prevent a vehicle going back to the depot to collect more items.

It has been shown that with suitable adjustments to the Hamiltonian and by treating the term which scales interaction energy as a constant, QA is able to tackle very large VRP instances without incurring the need to fine tune all the control parameters. With the tuning much simplified, QA can be used to deliver results which are improvements over, or equal to the best-known scores in the greater majority of large instances of CVRP and DCVRP. The primary conclusion is that the modified QA heuristic is a good match for these kinds of vehicle routing problems.

A reduced number of tunable parameters in FJ-QACVRP presented the opportunity to focus subsequent tuning efforts upon establishing the temperature value through use of the ESPT method. This significantly improves the reliability (in terms of success rate) of QA when dealing with collections of instances which exhibit similar features in their fitness landscapes. For convenience, this study reused existing benchmarks containing instances with similar distributions of customer locations. Naturally, if this method was to be generalised, extending to arbitrary collections of instances, then some effort would need to be spent analysing their Fitness Clouds in order to group them sensibly prior to the application of ESPT. This would perhaps involve the use of more complicated affine transformations (e.g. skews, mirrors, rotations) rather than scaling factors.

References

Alba, E., Dorronsoro, B., 2006. Computing nine new best-so-far solutions for capacitated VRP with a cellular genetic algorithm. Inf. Process. Lett. 98 (6), 225–230.
Augerat, P., Belenguer, J.M., Benavent, E., Corberán, A., Naddef, D., Rinaldi, G., 1995. Computational Results With a Branch and Cut Code for the Capacitated Vehicle Routing Problem. Technical Report 949-M. Université Joseph Fourier, Grenoble, France.
Battaglia, D.A., Santoro, G.E., Tosatti, E., 2005. Optimization by quantum annealing: lessons from hard satisfiability problems. Phys. Rev. E 71 (6), 066707.
Burke, E.K., Bykov, Y., 2017. The late acceptance Hill-Climbing heuristic. Eur. J. Oper. Res. 258 (1), 70–78.
Christofides, N., Eilon, S., 1969. An algorithm for the vehicle-dispatching problem. Oper. Res. 309–318.
Christofides, N., Mingozzi, A., Toth, P., Sandi, C., 1981. Combinatorial optimization. Interfaces 11 (2), 115.
Collard, P., Vérel, S., Clergue, M., 2004. Local search heuristics: fitness cloud versus fitness landscape. In: Proceedings of the 16th Eureopean Conference on Artificial Intelligence (ECAI 2004), pp. 973–974.

Crispin, A., Syrichas, A., 2013. Quantum annealing algorithm for vehicle scheduling. In: Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on. IEEE, pp. 3523–3528.
De Franceschi, R., Fischetti, M., Toth, P., 2006. A new ILP-based refinement heuristic for vehicle routing problems. Math. Program. 105 (2–3), 471–499.
Gillett, B.E., Johnson, J.G., 1976. Multi-terminal vehicle-dispatch algorithm. Omega 4 (6), 711–718.
Golden, B.L., Wasil, E.A., Kelly, J.P., Chao, I.M., 1998. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In: Fleet Management and Logistics. Springer, pp. 33–56.
Li, F., Golden, B., Wasil, E., 2005. Very large-scale vehicle routing: new test problems, algorithms, and results. Comput. Oper. Res. 32 (5), 1165–1179.
Liu, W., Li, X., 2015. A problem-reduction evolutionary algorithm for solving the capacitated vehicle routing problem. Math. Probl. Eng. 2015.
Mester, D., Bräysy, O., 2005. Active guided evolution strategies for large-scale vehicle routing problems with time windows. Comput. Oper. Res. 32 (6), 1593–1614.
Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., 1953. Equation of state calculations by fast computing machines. J. Chem. Phys. 21 (6), 1087–1092.
Nie, H., Liu, B., Xie, P., Liu, Z., Yang, H., 2014. A cuckoo search algorithm for scheduling multiskilled workforce. J. Netw. 9 (5), 1346–1353.
Poggi, M., Uchoa, E., 2014. New exact algorithms for the capacitated vehicle routing problem. Vehicle Routing: Problems, Methods, and Applications 18, 59.
Ralphs, T., Branch and Cut for Vehicle Routing, Vehicle Routing Data Sets. https://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm. Accessed: 2017-05-24.
Ridge, E., Kudenko, D., 2010. Tuning an algorithm using design of experiments. In: Experimental Methods for the Analysis of Optimization Algorithms. Springer, pp. 265–286.
Taillard, É., 1993. Parallel iterative search methods for vehicle routing problems. Networks 23 (8), 661–673.
Tarantilis, C.D., Kiranoudis, C.T., 2002. Boneroute: an adaptive memory-based method for effective fleet management. Ann. Oper. Res. 115 (1–4), 227–241.
Titiloye, O., Crispin, A., 2011. Quantum annealing of the graph coloring problem. Discrete Optim. 8 (2), 376–384.
Vanneschi, L., Tomassini, M., Collard, P., Vérel, S., 2006. Negative slope coefficient: a measure to characterize genetic programming fitness landscapes. In: Genetic Programming. Springer, pp. 178–189.
Watson, J.P., 2010. An introduction to fitness landscape analysis and cost models for local search. In: Handbook of Metaheuristics. Springer, pp. 599–623.