

What Is the Optimal Annealing Schedule in Quantum Annealing

Oscar Galindo and Vladik Kreinovich

Department of Computer Science

University of Texas at El Paso

500 W. University

El Paso, TX 79968, USA

ogalindomo@miners.utep.edu, vladik@utep.edu

Abstract

In many real-life situations in engineering (and in other disciplines), we need to solve an optimization problem: we want an optimal design, we want an optimal control, etc. One of the main problems in optimization is avoiding local maxima (or minima). One of the techniques that helps with solving this problem is annealing: whenever we find ourselves in a possibly local maximum, we jump out with some probability and continue search for the true optimum. A natural way to organize such a probabilistic perturbation of the deterministic optimization is to use quantum effects. It turns out that often, quantum annealing works much better than non-quantum one. Quantum annealing is the main technique behind the only commercially available computational devices that use quantum effects – D-Wave computers. The efficiency of quantum annealing depends on the proper selection of the annealing schedule, i.e., schedule that describes how the perturbations decrease with time. Empirically, it has been found that two schedules work best: power law and exponential ones. In this paper, we provide a theoretical explanation for these empirical successes, by proving that these two schedules are indeed optimal (in some reasonable sense).

1 Formulation of the Problem

Optimization is important. In practical applications, whether it is applications to engineering or to economics, we want to find the best possible decision, i.e., the decision for which the appropriate objective function attains its largest possible value.

Local maxima are a big problem. There exist many numerical algorithms for optimization. How do we know that we have reached the maximum? One natural criterion is that we compare the quality of current alternative with

the qualities of several similar alternatives. If the current alternative is indeed optimal, then it is better than every other alternative – in particular, better than all nearby alternatives.

The problem is that often, we have a situation of local maximum, when an alternative is better than all nearby alternatives, but still worse than some other – distant – alternative.

Simulated annealing: an intelligent way to avoid local maxima. If increasing or decreasing one of the parameters increases the value of the objective function, then it is very clear what to do: change the value in this direction. But what if we are possibly at a local maximum? In this case, it is not clear in which direction we should go. All directions are, in this sense, equally good (or equally bad), so a reasonable idea is to pick each direction at random, so that each possible direction will have the same probability. The corresponding change should be significant – otherwise, we will not get out of the local maximum.

This is the main idea behind *simulated annealing*: we perform some optimization algorithm until we reach a possibly local maximum. Then, we perform a random jump, and apply the optimization algorithm again, hoping that this time it will lead us to a better alternative. If we get stuck in a (possibly local) maximum again, we jump again, etc.

We want this process to eventually stop. Thus, instead of jumping out every time, we jump out with some probability – and, for the process to stop, this probability decreases with time. How exactly this probability decreases with time is known as the *annealing schedule*.

Proper selection of annealing schedule is very important. The success – or not – of simulated annealing depends on annealing schedule. If we decrease the probability too fast, we will not give the algorithm enough time to jump out of a local maximum. On the other hand, if we decrease it too slowly, the algorithm will work forever and never stops.

Quantum annealing. Traditional (non-quantum) computations are deterministic. So, to perform simulated annealing, we need to artificially introduce randomness.

In contrast, quantum processes are, in general, probabilistic [7, 15, 19]. Thus, in quantum computing, to implement a random deviation, there is no need to artificially introduce randomness: it is sufficient to somewhat modify the system’s dynamics.

In general, the state of a quantum system is described by a complex-valued function $\psi(t)$ (known as the *wave function*), and the dynamics of a quantum system is described by Schroedinger’s equations

$$i \cdot \hbar \cdot \frac{\partial \psi}{\partial t} = H\psi, \tag{1}$$

where $i \stackrel{\text{def}}{=} \sqrt{-1}$ and H is a corresponding linear operator. In these terms, annealing-type modification means adding additional terms – decreasing with

time – to the operator H , i.e., replacing the original equation (1) with the modified equation

$$i \cdot \hbar \cdot \frac{\partial \psi}{\partial t} = H\psi + \gamma(t) \cdot H_0\psi, \quad (2)$$

where H_0 describes the deviation, and $\gamma(t)$ monotonically tends to 0 as t increases.

This method was first proposed in [8, 10] and has been actively explored since then.

Quantum annealing works. There are many theoretical and practical papers showing that with an appropriate choice of H_0 and $\gamma(t)$, quantum annealing indeed helps – and moreover, it often works more efficiently than non-quantum simulated annealing; see, e.g., [2, 3, 4, 6, 9, 11, 12, 14, 16, 17, 18, 20].

Quantum annealing is the main idea behind D-Wave Systems quantum computing devices – the only commercially available and practically useful computers that utilize quantum effects in computations; see, e.g., [2, 11, 18].

The efficiency of quantum annealing can be explained in the same way as efficiency of quantum computing in general: classical (non-quantum) processes can be viewed as a limit case (i.e., in effect, particular case) of quantum processes. Thus, when, in addition to non-quantum processes, we also allow quantum ones, we increase the space of possible choices – and, not surprisingly, this increase often leads to better selections.

Which annealing schedule works best: empirical facts. Similarly to the non-quantum case, the efficiency of quantum annealing strongly depends on the proper selection of the annealing schedule, i.e., the dependence of $\gamma(t)$ on time t . Empirically, depending on the specific optimization problem, two scheduled work the best:

- the power law annealing schedule $\gamma(t) = A \cdot t^a$, for some A and $a < 0$; see, e.g., [12, 13]; and
- the exponential annealing schedule $\gamma(t) = A \cdot \exp(a \cdot t)$ for some A and $a < 0$; see, e.g., [5, 13].

What we do in this paper. In this paper, we present a possible theoretical explanation for these empirical facts.

2 Analysis of the Problem

Physical meaning of annealing. The general idea of using simulating physical phenomena in optimization is that a physical systems tends to end up in a state with the smallest possible energy. For example, in a gravitational field, this means getting to as low a position as possible. In principle, we can place a ball on top of the mountain – which will constitute a local minimum of energy. However, if a strong wind blows and disturbs the ball, it will start falling down.

It may reach a few local minima along the way, but eventually it will reach the lowest possible position at the foot of the mountain.

So, a natural way to use simulated physical phenomena for optimization is to simulate a system for which, for all the values of the parameters, its energy is equal to the value of minimized objective function. In general, in Schroedinger equations, energy is represented by the operator H , so for quantum annealing, this operator must represent the desired objective function.

Need to select a family. The original optimization problem is usually not formulated in terms of energy. So, how we transform it into energy depends in our choice of units. We will get completely different results if we use a typical macro-world unit like Joule or a typical micro-world unit like MeV. If we select a different unit, this means that in original unit, instead of H , we will have $C \cdot H$, where C is the ratio of the units.

If for the original operator H the best schedule was $\gamma(t)$, then for the new operator $C \cdot H$, the best schedule is $C \cdot \gamma(t)$, since the corresponding equation

$$i \cdot \hbar \cdot \frac{\partial \psi}{\partial t} = C \cdot H \psi + C \cdot \gamma(t) \cdot H_0 \psi \quad (3)$$

is equivalent to the original equation (2) if we re-scale the time, i.e., consider t/C instead of the original time t .

Since, as we have mentioned, the choice of the energy unit is rather arbitrary, this means that we cannot select a single annealing schedule $\gamma(t)$: with each such optimal schedule, in different energy units, a schedule $C \cdot \gamma(t)$ is optimal. Thus, we can only select a *family* $\{C \cdot \gamma(t)\}_{C>0}$ of annealing functions, in which a function $\gamma(t)$ is fixed, and the parameter C can take any positive value.

What does optimal mean? Among all possible families, we want to select the best – the optimal one. What is the best depends on the situation. We may want to get the results as fast as possible; we may want to get the maximum as close to the global maximum as possible even if it takes more time; there are many other possible options.

In all these cases, we have a criterion for comparing families, i.e., saying when one family F_1 is better than family F_2 (we will denote it by $F_2 < F_1$), or F_2 is better than F_1 ($F_1 < F_2$), or F_1 and F_2 are of the same quality with respect to the given criterion; we will denote this by $F_1 \sim F_2$. Of course, if F_1 is better than F_2 and F_2 is better than F_3 , then F_1 should be better than F_3 .

An optimal family F_{opt} is a family which is better (or of the same quality) than every other family F , i.e., for which $F < F_{\text{opt}}$ or $F \sim F_{\text{opt}}$.

What if we have two different optimal families? This happens in real-life optimization problems: e.g., if we are minimizing time, we may have two different annealing schedules that takes the exact same time on given examples. In this case, a natural idea is to use this non-uniqueness to optimize something else: e.g., select a family with the more accurate approximation to the actual maximum. In effect, we thus have a more complex optimality criterion: F_1 is better than F_2 if either F_1 is faster, or if they take the same time but F_1 is more

accurate. If this new criterion still leaves several optimal families, we can optimize something else. We can continue thus modifying the optimality criterion until we get a *final* criterion, for which there is exactly one optimal family.

Need for re-scaling time. We are looking for the dependence of $\gamma(t)$ on time, but the numerical value of time also depends on the choice of the measuring unit. If we replace the original unit of time by a λ times smaller unit, then, for each moment of time, the original numerical value t is replaced by a new value $\lambda \cdot t$. For example, if we replace minutes by seconds, then 2 minutes becomes $60 \cdot 2 = 120$ seconds.

It is reasonable to require that the relative quality of a family not change if we simply change the unit for time: e.g., if $\{C \cdot \gamma_1(t)\}_{C>0} < \{C \cdot \gamma_2(t)\}_{C>0}$, then we should have $\{C \cdot \gamma_1(\lambda \cdot t)\}_{C>0} < \{C \cdot \gamma_2(\lambda \cdot t)\}_{C>0}$.

The numerical value of time also depends on the choice of the starting point. If we replace the original starting point with the one which is t_0 units earlier, then all numerical values t are replaced with shifted values $t + t_0$. It also makes sense to require that the relative quality of two families not depend on the choice of the starting point, i.e., that if $\{C \cdot \gamma_1(t)\}_{C>0} < \{C \cdot \gamma_2(t)\}_{C>0}$, then we should have $\{C \cdot \gamma_1(t + t_0)\}_{C>0} < \{C \cdot \gamma_2(t + t_0)\}_{C>0}$.

Now, we are ready to formulate our problem in precise terms.

3 Definitions and the Main Result

Definition 1. By a family, we mean a class of functions $\{C \cdot \gamma(t)\}_{C>0}$, where $\gamma(t)$ is a monotonic function that tends to 0 at infinity.

Definition 2. By an optimality criterion, we mean a pair of relations ($<$, $\sim$) on the class of all families for which the following properties hold:

- for every pair (F_1, F_2) , we have exactly one of the possible three relations $F_1 < F_2$, $F_2 < F_1$, and $F_1 \sim F_2$;
- if $F_1 < F_2$ and $F_2 < F_3$, then $F_1 < F_3$;
- if $F_1 < F_2$ and $F_2 \sim F_3$, then $F_1 < F_3$;
- if $F_1 \sim F_2$ and $F_2 < F_3$, then $F_1 < F_3$;
- if $F_1 \sim F_2$ and $F_2 \sim F_3$, then $F_1 \sim F_3$.

Definition 3. We say that a family F_{opt} is optimal with respect to the optimality criterion ($<$, $\sim$) if for every family F , we have $F < F_{\text{opt}}$ or $F \sim F_{\text{opt}}$.

Definition 4. We say that an optimality criterion is final if there is exactly one family which is optimal with respect to this criterion.

Definition 5.

- For every $\lambda > 0$, by a re-scaling $T_\lambda(F)$ of a family $F = \{C \cdot \gamma(t)\}_{C>0}$, we mean a family $\{C \cdot \gamma(\lambda \cdot t)\}_{C>0}$.

- For every t_0 , by a shift $S_{t_0}(F)$ of a family $F = \{C \cdot \gamma(t)\}_{C>0}$, we mean a family $\{C \cdot \gamma(t + t_0)\}_{C>0}$.
- We say that an optimality criterion is scale-invariant if for all F_i and $\lambda < 0$, $F_1 < F_2$ implies $T_\lambda(F_1) < T_\lambda(F_2)$ and $F_1 \sim F_2$ implies $T_\lambda(F_1) \sim T_\lambda(F_2)$.
- We say that an optimality criterion is shift-invariant if for all F_i and t_0 , $F_1 < F_2$ implies $S_{t_0}(F_1) < S_{t_0}(F_2)$ and $F_1 \sim F_2$ implies $S_{t_0}(F_1) \sim S_{t_0}(F_2)$.

It seems reasonable to require that the optimality criterion be both scale- and shift-invariant, but it turned out to be impossible:

Proposition 1. *No final optimality criterion is scale-and shift-invariant.*

Comment. For readers' convenience, all the proofs are placed in the special proofs section.

Good news is that when we require either scale-invariance or shift-invariance, we get exactly the empirically best annealing schedules.

Proposition 2. *For every scale-invariant optimality criterion, the optimal annealing schedule is $\gamma(t) = A \cdot t^a$.*

Proposition 3. *For every shift-invariant optimality criterion, the optimal annealing schedule is $\gamma(t) = A \cdot \exp(a \cdot t)$.*

4 Proofs

Proof of Proposition 1 follows from Propositions 2 and 3: if an optimality criterion is scale-invariant, then the corresponding annealing schedule has the form $\gamma(t) = A \cdot t^a$ and is, thus, according to Proposition 2, *not* shift-invariant.

Proof of Proposition 2. Let $F_{\text{opt}} = \{C \cdot \gamma_{\text{opt}}(t)\}_{C>0}$ be the family which is optimal with respect to the scale-invariant optimality criterion.

1°. Let us first prove that this family is itself scale-invariant, i.e., that we have $T_\lambda(F_{\text{opt}}) = F_{\text{opt}}$ for all $\lambda > 0$.

Indeed, since the family F_{opt} is optimal, we have $F < F_{\text{opt}}$ or $F \sim F_{\text{opt}}$ for every family F . In particular, we have $T_{\lambda^{-1}}(F) < F_{\text{opt}}$ or $T_{\lambda^{-1}}(F) \sim F_{\text{opt}}$.

One can easily see that $T_\lambda(T_{\lambda^{-1}}(F)) = F$. Thus, due to scale-invariance, for every family F , we have $F < T_\lambda(F_{\text{opt}})$ or $F \sim T_\lambda(F_{\text{opt}})$. By definition of optimality, this means that $T_\lambda(F_{\text{opt}})$ is also an optimal family. However, our optimality criterion is final, which means that there is only one optimal family. Thus indeed $T_\lambda(F_{\text{opt}}) = F_{\text{opt}}$.

2°. By definition of scaling $T_\lambda(F)$, the function $\gamma(\lambda \cdot t)$ belongs to the family $T_\lambda(F_{\text{opt}})$. Since this family is equal to the original family $F_{\text{opt}} = \{C \cdot \gamma_{\text{opt}}(t)\}_{C>0}$,

this means that the function $\gamma(\lambda \cdot t)$ belongs to this original family as well, i.e., that

$$\gamma(\lambda \cdot t) = C(\lambda) \cdot \gamma(t), \quad (4)$$

for some C depending on λ .

It is known – see, e.g. [1] – that all monotonic solutions of the functional equation (4) have the desired form $A \cdot x^a$. The proposition is proven.

Comment. The cited result about the functional equation (4) is easy to prove if we additionally assume that the function $\gamma(t)$ is differentiable. Then, due to (4), the function $C(\lambda)$ is also differentiable – as a ratio of two differentiable functions. Thus, we can differentiate both sides of (4) with respect to λ and take $\lambda = 1$; we will get

$$t \cdot \frac{d\gamma}{dt} = a \cdot \gamma,$$

where we denoted $a \stackrel{\text{def}}{=} C'(1)$. By moving all the terms containing γ to one side and all terms containing t to another side, we get

$$\frac{d\gamma}{\gamma} = a \cdot \frac{dt}{t}.$$

Integrating, we get $\ln(\gamma) = a \cdot \ln(t) + C_0$, where C_0 is the integration constant. By applying $\exp(z)$ to both sides, we get the desired formula $\gamma = A \cdot t^a$, for $A = \exp(C_0)$.

Proof of Proposition 3. Let $F_{\text{opt}} = \{C \cdot \gamma_{\text{opt}}(t)\}_{C>0}$ be the family which is optimal with respect to the shift-invariant optimality criterion.

1°. Let us first prove that this family is itself scale-invariant, i.e., that we have $S_{t_0}(F_{\text{opt}}) = F_{\text{opt}}$ for all $\lambda > 0$.

Indeed, since the family F_{opt} is optimal, we have $F < F_{\text{opt}}$ or $F \sim F_{\text{opt}}$ for every family F . In particular, we have $S_{-t_0}(F) < F_{\text{opt}}$ or $S_{-t_0}(F) \sim F_{\text{opt}}$.

One can easily see that $S_{t_0}(S_{-t_0}(F)) = F$. Thus, due to scale-invariance, for every family F , we have $F < S_{t_0}(F_{\text{opt}})$ or $F \sim S_{t_0}(F_{\text{opt}})$. By definition of optimality, this means that $S_{t_0}(F_{\text{opt}})$ is also an optimal family. However, out optimality criterion is final, which means that there is only one optimal family. Thus indeed $S_{t_0}(F_{\text{opt}}) = F_{\text{opt}}$.

2°. By definition of a shift $S_{t_0}(F)$, the function $\gamma(t + t_0)$ belongs to the family $S_{t_0}(F_{\text{opt}})$. Since this family is equal to the original family $F_{\text{opt}} = \{C \cdot \gamma_{\text{opt}}(t)\}_{C>0}$, this means that the function $\gamma(t + t_0)$ belongs to this original family as well, i.e., that

$$\gamma(t + t_0) = C(t_0) \cdot \gamma(t), \quad (5)$$

for some C depending on t_0 .

It is known – see, e.g. [1] – that all monotonic solutions of the functional equation (5) have the desired form $A \cdot \exp(a \cdot t)$. The proposition is proven.

Comment. The cited result about the functional equation (5) is easy to prove if we additionally assume that the function $\gamma(t)$ is differentiable. Then, due to (5), the function $C(t_0)$ is also differentiable – as a ratio of two differentiable functions. Thus, we can differentiate both sides of (5) with respect to t_0 and take $t_0 = 0$; we will get

$$\frac{d\gamma}{dt} = a \cdot \gamma,$$

where we denoted $a \stackrel{\text{def}}{=} C'(0)$. By moving all the terms containing γ to one side and all terms containing t to another side, we get

$$\frac{d\gamma}{\gamma} = a \cdot t.$$

Integrating, we get $\ln(\gamma) = a \cdot t + C_0$, where C_0 is the integration constant. By applying $\exp(z)$ to both sides, we get the desired formula $\gamma = A \cdot \exp(a \cdot t)$, for $A = \exp(C_0)$.

Acknowledgments

This work was supported in part by the US National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science) and HRD-1242122 (Cyber-ShARE Center of Excellence).

References

- [1] J. Aczél and J. Dhombres, *Functional Equations in Several Variables*, Cambridge University Press, 2008.
- [2] S. Boixo, T. F. Ronnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, “Evidence for quantum annealing with more than one hundred qubits”, *Nature Physics*, 2014, Vol. 10, pp. 218–224.
- [3] J. Brooke, D. Bitko, T. F. Rosenbaum and G. Aeppli, “Quantum annealing of a disordered magnet”, *Science*, 1999, Vol. 284, pp. 779–781.
- [4] E. Crosson, E. Farhi, C. Y.-Y. Lin, H.-H. Lin, and P. Shor, *Different Strategies for Optimization Using the Quantum Adiabatic Algorithm*, 2014, arXiv:1401.7320.
- [5] A. Das, B. K. Chakrabarti, and R. B. Stinchcombe, “Quantum Annealing in a Kinetically Constrained System”, *Physical Review Letters*, 2005, Vol. 72, Paper 026701.
- [6] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Ludgren, and D. Preda, “A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem”, *Science*, 2001, Vol. 292, pp. 472–475.

- [7] R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.
- [8] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, “Quantum annealing: A new method for minimizing multidimensional functions”, *Chemical Physics Letters*, 1994, Vol. 219, No. 5–6, pp. 343–348.
- [9] B. Heim, T. F. Rønnow, S. V. Isakov and M. Troyer, “Quantum versus classical annealing of Ising spin glasses”, *Science*, 2015, Vol. 348, pp. 215–217.
- [10] T. Kadowaki and H. Nishimori, “Quantum annealing in the transverse Ising model”. *Physical Reviews E*, 1998, Vol. 58, Paper 5355.
- [11] T. Lanting, A. J. Przybysz, A. Yu. Smirnov, F. M. Spedalieri, M. H. Amin, A. J. Berkley, R. Harris, F. Altomare, S. Boixo, P. Bunyk, N. Dickson, C. Enderud, J. P. Hilton, E. Hoskinson, M. W. Johnson, E. Ladizinsky, N. Ladizinsky, R. Neufeld, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, S. Uchaikin, A. B. Wilson, and G. Rose, “Entanglement in a quantum annealing processor”, *Physical Review X*, 2014, Vol. 4, No. 2, Paper 021041.
- [12] S. Morita and H. Nishimori, “Mathematical Foundation of Quantum Annealing”, *Journal of Mathematical Physics*, 2008, Vol. 49, Paper 125210.
- [13] S. Mukherjee and B. K. Chakrabarti, “Multivariable optimization: Quantum annealing and computation”, *The European Physical Journal Special Topics*, 2015, Vol. 224, pp. 17–24.
- [14] D. Muthukrishnan, T. Albash, and D. A. Lidar, *When Diabatic Trumps Adiabatic in Quantum Optimization*, 2015, arXiv:1505.01249.
- [15] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, U.K., 2000.
- [16] G. E. Santoro and E. Tosatti, “Optimization using quantum mechanics: quantum annealing through adiabatic evolution”, *Journal of Physics A*, 2006, Vol. 39, paper R393.
- [17] V. N. Smelyanskiy, E. G. Rieffel, S. I. Knysh, C. P. Williams, M. W. Johnson, M. C. Thom, W. G. Macready, and K. L. Pudenz, *A Near-Term Quantum Computing Approach for Hard Computational Problems in Space Exploration*, 2012, arXiv:1204.2821.
- [18] S. Steiger, B. Heim, T. Ronnow, and M. Troyer, “Performance of quantum annealing hardware”, In: D. A. Huckridge, R. Ebert, M. T. Gruneisen, M. Dusek, and J. G. Rarity (eds.), *Electro-Optical and Infrared Systems: Technology and Applications XII; and Quantum Information Science and Technology*, SPIE Proceedings, 2015, Vol. 9648, Paper 964816.

- [19] K. S. Thorne and R. D. Blandford, *Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics*, Princeton University Press, Princeton, New Jersey, 2017.
- [20] S. E. Venegas-Andraca, W. Cruz-Santos, C. McGeoch, and M. Lanzagorta, “A cross-disciplinary introduction to quantum annealing-based algorithms”, *Contemporary Physics*, 2018, Vol. 59, No. 2, pp. 174–196.