REFERENCES.

[1] E. Nelson, *Quantum Fluctuations*, Princeton University Press, Princeton, N.J. (1985), and previous references quoted there.

[2] D. Dohrn and F. Guerra, Phys. Rev. D31, 2531 (1985).

[3] E. Aldrovandi, D. Dohrn and F. Guerra, Stochastic Action of Dynamical Systems on Curved Manifolds. The Geodesic Interpolation, Preprint, University of Rome, 1988.

[4] F. Guerra and L.M. Morato, Phys. Rev. D27, 1774 (1983).

[5] E.Nelson, *Dynamical Theories of Brownian Motion*, Princeton University Press, Princeton, N.J. (1967).

[6] M.Pavon, Applied Mathematics and Optimization, 14, 265 (1986).

[7] F.Guerra and M.Pavon, Stochastic Variational Principles and Free Energy for Dissipative Processes, in: *Analysis and Control ofNonlinear Systems*, C.I. Byrnes, C.F. Martin, R.E. Saeks, eds, North Holland, Amsterdam (1988).

[8] E. Aldrovandi, D. Dohrn and F. Guerra, in preparation.

[9] B.S. DeWitt, Rev. Mod. Phys. 29, 337 (1957).

[10] W. Pauli, *Pauli Lectures on Physics*, vol. 6, M.I.T. Press, Cambridge, Mass., (1973).

[11] N. Ikeda, S. Watanabe, *Stochastic Differential Equations and Diffusion Processes*, North Holland, Amsterdam, (1981).

[12] C. DeWitt–Morette, K.D. Elworthy, B.L. Nelson and G.S. Sammelman, Ann. Inst. Henri Poincarè, 32, 327 (1980).

[13] D. Dohrn and F. Guerra, Lett. Nuovo Cimento, 22, 121 (1978).

[14] K. Itô, The Brownian Motion and Tensor Fields on Riemannian Manifold, Proc. Int. Congress Math. (Stockholm), 356, (1962).

[15] E. Dynkin, Sov. Math. Doklady, 9, 532, (1968).

[16] T.G. Dankel, Jr., Archive for Rat. Mech. and Anal., 37, 192 (1970).

[17] E. Aldrovandi, Principi variazionali stocastici per sistemi dinamici su varietà, Thesis, University of Rome, 1986.

A Numerical Implementation of "Quantum Annealing"

BRUNO APOLLONI, NICOLÒ CESA-BIANCHI, DIEGO DE FALCO

Dip. di Scienze dell'Informazione, Università degli Studi di Milano,
via Moretto da Brescia 9, I-20133 Milano (ITALY).

Abstract. "Quantum Annealing", a variant of the Simulated Annealing algorithm, simulates the quantum tunnel effect to find approximate solutions to the problem of minimizing a real function of boolean variables. More precisely, the procedure carries out a Markov chain whose equilibrium distribution is associated with the ground state wave function of a particular Schrödinger Hamiltonian. A numerical implementation of the algorithm is presented and some applications are discussed. Experimental results show that Quantum Annealing and Simulated Annealing are comparable in terms of their performances even though significative differences have been found in their detailed behaviour.

KEYWORDS: stochastic optimization / Schrödinger-Nelson dynamics / quantum tunnelling / simulated annealing

1 Introduction

We present here an implementation of a recently proposed [1] stochastic method for finding approximate solutions of global combinatorial optimization problems, namely problems concerning the search of global minima of an objective function $V : \mathcal{R} \rightarrow \mathbb{R}$ defined on a finite set $\mathcal{R}$.

Roughly speaking, stochastic optimization algorithms set up a stochastic process whose probability distribution concentrates on the global minima of the objective function. However, in all finite-time implementations, these methods usually provide only approximate solutions. In this case we may be interested in finding the strategy producing the best results when compared to other methods in the same period of time.

Acknowledgements: This research was supported in part by Consiglio Nazionale delle Ricerche (through grant 88.03556.12). Support by the Research Center Bielefeld-Bochum-Stochastics is also gratefully acknowledged.

A stochastic method which has been extensively applied is the Simulated Annealing (SA) algorithm ([2], [3] and [4]). We briefly recall that SA relies on Monte Carlo techniques to generate a Gibbs ensemble for a given physical system at positive temperature. Interpreting the objective function V as the "energy" of such a system, in this Gibbs ensemble the equilibrium probability for the occurrence of a given $r \in \mathcal{R}$ is:

$$p_c(r) = \frac{1}{Z(c)} \exp\left(-\frac{V(r)}{c}\right), \tag{1.1}$$

where $Z(c)$ is a normalization factor and c is a control parameter proportional to the temperature. This technique allows (at least in principle) the algorithm to overcome the problem of being trapped in local minima of the function to be minimized.

In [5] Hajek proved that the sequence of random variables $X(t)$ associated to the Metropolis chain of SA is such that

$$\lim_{t\to\infty} \Pr\{X(t) \in \mathcal{R}^*\} = 1 \tag{1.2}$$

(where $\mathcal{R}^*$ is the set of the global minima of V) if and only if

$$c(t) \geq \frac{d^*}{\log(t+1)}. \tag{1.3}$$

where d^* is a constant factor depending on V. Similar results have also been obtained by Holley and Stroock [6].

Quantum Annealing (QA) is a variant of the above algorithm based on the observation that the quantum ground state of many Schrödinger Hamiltonians of interest can be described by a Gibbs ensemble in path space. More specifically, QA carries out a Markov process whose stationary distribution is associated with the ground state wave function of a particular Schrödinger Hamiltonian. Unlike SA, the resulting transition rule is not local i.e., before taking a transition to a neighbouring point, the QA procedure estimates a suitable average of the objective function on a region of the search space which is randomly spread around the current point.

In Section 2 the domain of the problem is stated and the stochastic process simulated by QA is defined. In Section 3 the physical motivations of the approach are briefly recalled (for a detailed analysis of this point we refer to the companion paper [1]). In Section 4 the algorithmical formulation is described togheter with a brief discussion about the tuning of the parameters involved. In Section 5 we compare the results obtained applying QA to hard combinatorial problems with those obtained by SA. In particular, the test bed problems considered are Graph Partitioning in the formulation of Johnson *et al.* [7] and the problem of generating binary strings with low autocorrelation, as described in [8]. Section 6 is devoted to conclusions and outlook.

2 Stochastic Formulation

We focus our attention on the search of global minima of a real-valued function V of n boolean variables $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n) \in Z_2^n \equiv \{-1, 1\}^n$. A notion of neighbourhood in Z_2^n can be easily formalized introducing, for $i = 1, 2, \ldots, n$, the operators $S_i : Z_2^n \to Z_2^n$ such that

$$S_i(\varepsilon_1, \ldots, \varepsilon_i, \ldots, \varepsilon_n) := (\varepsilon_1, \ldots, \varepsilon_{i-1}, -\varepsilon_i, \varepsilon_{i+1}, \ldots, \varepsilon_n) \tag{2.1}$$

and defining the neighbourhood of any $\varepsilon \in Z_2^n$ as the set:

$$\mathcal{R}_\varepsilon := \bigcup_{i=1}^{n} S_i \varepsilon. \tag{2.2}$$

The continous time, time homogeneous Markov process simulated by the Quantum Annealing algorithm has a transition function of the general form

$$q_{\varepsilon,\varepsilon'}(t) = \sum_{r=0}^{\infty} e^{-\mu t} \frac{(\mu t)^r}{r!} (M^r)_{\varepsilon,\varepsilon'}, \tag{2.3}$$

the stochastic matrix M having elements of the form

$$m_{\varepsilon,\varepsilon'} := \begin{cases} \frac{\nu}{\mu} c_j(\varepsilon) \frac{\psi_\nu(\varepsilon')}{\psi_\nu(\varepsilon)} & \text{if } \exists\, j \text{ s.t.} \varepsilon' = S_j \varepsilon, \\ 1 - \frac{\nu}{\mu} \sum_{k=1}^{n} c_k(\varepsilon) \frac{\psi_\nu(S_k \varepsilon)}{\psi_\nu(\varepsilon)} & \text{if } \varepsilon' = \varepsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

for suitable positive parameters μ and ν. The coefficients $c_j(\varepsilon)$ $(j = 1, 2 \ldots, n)$ satisfy the following conditions:

$$c_j(\varepsilon) \geq 0, \qquad c_j(\varepsilon) = c_j(S_j(\varepsilon)), \qquad \sum_{k=1}^{n} c_k(\varepsilon) = 1. \tag{2.5}$$

For the sake of simplicity, we will henceforth choose

$$c_j(\varepsilon) := |\mathcal{R}_\varepsilon| = n^{-1} \tag{2.6}$$

for every $\varepsilon \in Z_2^n$.

The function $\psi_\nu : Z_2^n \to \mathbb{R}$ has the following definition:

$$\psi_\nu(\varepsilon_0) := \lim_{t\to\infty} \frac{1}{C_t} \mathrm{E}\left\{\exp\left(-\int_0^t V(\varepsilon(\tau))d\tau\right) \,\Big|\, \varepsilon(0) = \varepsilon_0\right\}, \tag{2.7}$$

where C_t is a normalization factor.

The expectation E is calculated over the random paths $\varepsilon(\tau)$ having ε_0 (the point where ψ_ν is being evaluated) as their initial state. The Markov process generating these random paths on Z_2^n has transition probabilities of the form

$$\Pr\{\varepsilon(\tau_0+\tau)=\varepsilon' \mid \varepsilon(\tau_0)=\varepsilon\} = p_{\varepsilon,\varepsilon'}(\tau) = \sum_{r=0}^{\infty} e^{-\nu\tau}\frac{(\nu\tau)^r}{r!}\left(K^r\right)_{\varepsilon,\varepsilon'}. \tag{2.8}$$

with

$$k_{\varepsilon,\varepsilon'} = \begin{cases} c_j(\varepsilon) & \text{if } \varepsilon' \in \mathcal{R}_\varepsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

For M to be a stochastic matrix the following condition on the parameters μ and ν is sufficient:

$$\mu \geq \nu + (V_{max} - V_{min}) \qquad \nu > 0 \tag{2.10}$$

where V_{max} and V_{min} are, obviously, the maximum and minimum of V.

We remark that each transition of the QA process is determined by drawing long ***prospection chains*** from the current configuration and comparing suitable averages of V along these prospections. In this respect, while in the SA strategy a transition is decided ***locally***, by choosing randomly a neighbour and carrying out a simple Monte Carlo computation, here we explore many configurations before actually performing a transition, so the strategy is inherently ***non local***.

3 Relation with Simulated Annealing

As discussed in [1], equations (2.3) and (2.4) describe the ground state process [9] associated with a "Schrödinger Hamiltonian" H_ν and equation (2.7) describes, by a "Feyman-Kac formula", its ground state wave function ψ_ν.

More precisely, having introduced the Hilbert space $\mathcal{H}$ of complex functions on Z_2^n with scalar product

$$\langle \phi \mid \psi \rangle = \sum_{\varepsilon \in Z_2^n} \overline{\phi(\varepsilon)}\psi(\varepsilon) \tag{3.1}$$

and the Pauli spin 1/2 operators

$$\begin{aligned} (\mathbb{1}\psi)(\varepsilon) &= \psi(\varepsilon), \\ (\sigma_1(j)\psi)(\varepsilon) &= \psi(S_j(\varepsilon)), \\ (\sigma_2(j)\psi)(\varepsilon) &= -i\varepsilon_j\psi(S_j(\varepsilon)), \\ (\sigma_3(j)\psi)(\varepsilon) &= \varepsilon_j\psi(\varepsilon), \end{aligned} \tag{3.2}$$

and having set

$$K_\nu := \frac{\nu}{n}\sum_{k=1}^{n}(1-\sigma_1(k)), \tag{3.3}$$

H_ν is defined by

$$H_\nu : \mathcal{H} \to \mathcal{H},\ \psi \mapsto H_\nu\psi := K_\nu\psi + V\psi. \tag{3.4}$$

QA amounts therefore to substituting the original problem of minimizing V with the auxiliary problem of minimizing the quadratic form $\langle \psi \mid H_\nu\psi \rangle$, under the condition $\langle \psi \mid \psi \rangle = 1$.

The relation between QA and the original optimization problem is described by the inequalities:

$$E_\nu - \nu \leq V_{min} \leq E_\nu, \tag{3.5}$$

$$\forall k > 0 : \quad P_\nu\Big(\{\varepsilon \in Z_2^n : V(\varepsilon) - V_{min} \geq k\}\Big) < \frac{\nu}{k}, \tag{3.6}$$

where P_ν is the probability measure

$$P_\nu(A) = \sum_{\varepsilon \in A}\psi_\nu(\varepsilon)^2, \tag{3.7}$$

for every subset A of Z_2^n and E_ν the lowest eigenvalue of H_ν.

The comparison with SA is most easily carried out through the continous time description of the Metropolis chain leading, in SA, to the eqilibrium distribution

$$\rho_\beta(\varepsilon) = \frac{1}{Z(\beta)}\exp(-\beta V(\varepsilon)). \tag{3.8}$$

This Metropolis chain is described by the transition function

$$\tau_{\varepsilon,\varepsilon'}(t) = \exp\Big(-tL(\beta)\Big)_{\varepsilon,\varepsilon'} \tag{3.9}$$

where:

$$(L(\beta))_{\varepsilon,\varepsilon'} = \begin{cases} -n^{-1}\exp\Big(-\beta(V(\varepsilon') - V(\varepsilon))^+\Big) & \text{if } \varepsilon' \in \mathcal{R}_\varepsilon, \\ n^{-1}\sum_{k=1}^{n}\exp\Big(-\beta(V(S_k\varepsilon) - V(\varepsilon))^+\Big) & \text{if } \varepsilon' = \varepsilon, \\ 0 & \text{otherwise;} \end{cases} \tag{3.10}$$

$^+$ denoting positive part.

Because of the detailed balance condition

$$\rho_\beta(\varepsilon)\Big(L(\beta)\Big)_{\varepsilon,\varepsilon'} = \rho_\beta(\varepsilon')\Big(L(\beta)\Big)_{\varepsilon',\varepsilon} \tag{3.11}$$

an intrinsic characterization in the sense of [9] of $\rho_\beta(\varepsilon)$ is that $\rho_\beta^{1/2}(\varepsilon)$ (to be compared with ψ_ν) is the eigenvector, normalized to 1 in $\mathcal{H}$, belonging to the lowest eigenvalue of the Hermitian operator on $\mathcal{H}$

$$\begin{aligned} A_\beta =& \frac{1}{n}\sum_{k=1}^{n}\Big(e^{-\beta\left(V(S_k\varepsilon)-V(\varepsilon)\right)^+}\,\mathbb{1} + \\ & - e^{-\frac{\beta}{2}|V(S_k\varepsilon)-V(\varepsilon)|}\,\sigma_1(k)\Big), \end{aligned} \tag{3.12}$$

to be compared with

$$\frac{H_\nu}{\nu} = \frac{1}{n}\sum_{k=1}^{n}\Big(1-\sigma_1(k)\Big) + \frac{1}{\nu}V. \tag{3.13}$$

Notice that, while both A_β and H_ν/ν reduce, in the trivial case $\nu = 0$, to

$$\frac{1}{n}\sum_{k=1}^{n}\Big(1-\sigma_1(k)\Big), \tag{3.14}$$

the deceivingly simpler form of H_ν/ν corresponds via (2.7) to a much more extensive prospection of the graph of V before accepting proposed transitions. In spite of the formal analogies stressed here, the two approaches correspond in fact to adopting quite different physical metaphores for the optimization process: thermal noise for SA, "quantum noise" for QA.

4 Implementation

Achieving a *finite-time* implementation of the QA algorithm requires the specification and the tuning of a set of parameters governing the convergence of the process. A possible set of such parameters is:

- the number z of prospection chains drawn from each configuration,
- the mean length m of each chain,
- the value of the control parameter ν,
- the value of the control parameter μ,
- the stop criterion.

Relying on the analogy between the c parameter in SA and the ν parameter in QA, one could wonder whether an empirical scheduling of ν would lead to more accurate approximations. Actually, our schedule starts from eq. (3.5) and from the idea of obtaining a feasible run-time task, which consists in stating as target of the optimization process a value ρ% lower than the current minimum V_m. This accounts for a dynamical scaling of V in $V' = (V-a)/b$ where $a = V_m(100-\rho)\%$ and $b = V_m\rho\%$. Obviously, this scaling does not affect the position of minima.

Once the value of ψ_ν has been estimated for each neighbour $S_j(\varepsilon)$ of the current configuration, the procedure must choose if and where to make a transition. Since the lack of transitions is undoubtely a loss of time, we avoided the possibility of remaining in the current configuration by dynamically choosing $\mu = \mu(\varepsilon)$ so that, at every transition, $m_{\varepsilon,\varepsilon}$ becomes 0.

Other improvements have been derived by empirical considerations. First, note that the computation of an element of the matrix M defined in (2.4) requires the drawing of z prospection chains from *each* neighbour of the current configuration ε. Since in most pratical applications the neighbourhood is huge, we selected each time only a random subset of the neighbours of ε. A second improvement can be achieved by a more effective exploitation of the quantum *tunnel effect* simulated by the the random motion originating the prospection chains. When the value of V_m remains unchanged for more than a fixed numbers of steps a tunnel is drilled; namely, the current configuration is substituted with that found at the end of the last prospection chain drawn.

Finally, we obtained better results when QA was interleaved with a local descent optimization procedure (LO). The modified algorithm first invokes LO reaching a local minimum, then applies QA to improve that minimum. If no improvements of V_m are reported after a given amount of time, LO takes place again starting from the tail of the last chain.

A further remark is about the stop criterion. While SA progressively reduces the rate of transitions by decreasing the control parameter, so that we can stop when the activity is almost ceased, QA makes a transition at every step. Hence, no stop criterion is safe since relevant improvements in the current solution have been often observed after a period of idleness. Therefore, the only criterion we adopted is to stop the algorithm after a given amount of time corresponding to the computational resources allocated for the task. Summarizing, an instance of QA is specified by the following parameters:

L: mean number of steps along a prospection chain.
S: number of the sampled prospection chains.
P: specifies the percentage of the neighbours of the current configuration that will be selected as starting points of prospection chains.
I: maximum number of transitions not affecting V_m before invoking the **Local_Optimization** procedure.
T: maximum number of transitions not affecting V_m before drilling a tunnel.

R: the factor $\rho\%$ which dynamically fixes the scale of V.

Figures 1–3 show a pseudo-Pascal coding of the QA algorithm. Note that the value of current V_m could change also while the subroutine **Quantum_Transition** is running if, as a prospection chain is being drawn, a configuration with cost lower than V_m is met.

5 Numerical Results

The algorithm has been implemented in Pascal and run on a HP 9000/840. The first test bed we considered is the Graph Partitioning problem in the formulation given by Johnson *et al.* [7]. The formal statement of this problem is the following:
Given a symmetrical $n \times n$ matrix A with elements $a_{i,j} \in \{0, 1\}$ and given a positive constant α, find $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n) \in Z_2^n$ minimizing the function:

$$V(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n) = \sum_{1 \leq i < j \leq n} \left(\frac{a_{i,j}}{4} (\varepsilon_i - \varepsilon_j)^2 \right) + \alpha \left(\sum_{i=1}^{n} \varepsilon_i \right)^2 \qquad (5.1)$$

For n even, interpreting $a_{i,j} = 1$ ($a_{i,j} = 0$) as the presence (absence) of an arc between the vertices i and j of a graph, the minimum of V is reached when the vertices are partitioned into two almost equally sized subsets such that the number of arcs between them is minimal.

One hundred random istances of the problem have been generated, each of them being a graph of 500 vertices with an arc drawn between two vertices with probability $p = 0.01$. The parameter α in (5.1) was set to 0.05. For each istance, QA and SA have been started from the same randomly chosen initial configuration ε_0 and run for the same fixed amount of time. The histogram in Figure 4 shows the distribution of the costs of all the starting configurations.

Table 1 reports the parameters specifying both optimization processes. The *cooling schedule* of SA was done according to the criteria of ref. [7] based on extensive experimental work on Graph Partitioning. However, we abandoned their stop rule and tuned instead the schedule so that the algorithm becomes "frozen" almost at the same time of QA.

Figure 5 summarizes the results of SA and QA on the 100 istances. Observe that SA on average performs better than QA and the points where it is beaten by QA are not concentrated on any particular region. On the other hand, the performances of QA are only slightly worse than those of SA and, as it turns out comparing Figures 6 and 7, QA is faster in reaching a good approximation.

As second test bed, the problem of generating pseudo-random binary strings has been considered. Following the formulation presented in [10] we define

$$R_k(\varepsilon) = \sum_{i=1}^{n-k} \varepsilon_i \varepsilon_{i+k}, \qquad V(\varepsilon) = \sum_{k=1}^{n} R_k(\varepsilon)^2 \qquad (5.2)$$

and look for $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n) \in Z_2^n$ maximizing the function:

$$F(\varepsilon) = \frac{n^2}{2V(\varepsilon)}. \qquad (5.3)$$

indicating the *merit factor* for the string being evaluated. As in the previous case, we applied QA 100 times, requiring each time the generation of a 100 binary digits string.

Table 2 synthetizes the results of the test. We compare directly our results with those reported in [10] for 100 runs of SA. In this case QA beats SA both as average and best values. We remark that at present there are no theorems giving the value of the global maximum even if it is conjectured in [10] to be in the range 6–12.

6 Conclusions

Coherently with the fact that QA has asymptotical properties different from those exhibited by SA, experimental tests show that the two strategies behave differently on the same problem instances. In the two test beds we considered, both strategies reported one victory and one defeat each. The analysis of the convergence rate showed that QA converges very fast to a good approximation that is improved only slightly in the remaining time. On the contrary, the convergence of SA is slower and more regular.

It has been experimentally verified that the first local optimization phase in the QA strategy is not totally responsible for the very steep descent in the first 100 time steps. Hence, QA *is* globally more effective than SA, though only in the early phase of the optimization process. This kind of behaviour leads to the observation that QA seems to be more suitable than SA at least when we are given a small amount of time. The basic question of selecting problems where QA is definitively better than SA is probably related to structural properties of the objective functions and will be matter of further studies.

The last point we would like to discuss is the development of parallel implementations of the QA algorithm. The purely local transition rule of SA allows implementations on distributed, fine-grained architectures like Boltzmann Machines [11]. In these models, each binary variable ε_i is associated with a small processor element that can decide, with a relatively small computational cost, if the variable has to be changed or not. On the other hand, the true sequential nature of SA (which assures its convergence properties) hampers parallel implementations on coarse-grained computers as has been pointed out in [4] and [12].

For the QA algorithm the situation appears to be specular. The need of exploring the region around the current configuration makes the transition rule complex and non local. This, in turn, makes difficult to charge a single small processor with the burden of deciding if the current configuration has to be changed. However, increasing the resources of each processor element, so that it becomes able to carry out a whole prospection chain, makes more realistic the perspective of achieving a parallel coarse-grained implementation of QA without any corruption of its attractive mathematical properties.

REFERENCES

[1] Apolloni, B., Carvalho, C., and de Falco, D., *Quantum Stochastic Optimization*, to appear in: Stochastic Processes and their Applications, 1989.

[2] Černy, V., *Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm*, Journal of Optimization Theory and Applications, Vol. 45, pp. 45–51, 1985.

[3] Kirkpatrick, S., Gelatt, C.D. Jr. and Vecchi, M.P., *Optimization by Simulated Annealing*, IBM Research Report RC 9355, 1982.

[4] van Laarhoven, P.J.M., and Aarts, E.H.L., *Simulated Annealing: Theory and Applications*, Reidel, Dordrecht, 1987.

[5] Hajek, B., *Cooling Schedules for Optimal Annealing*, Mathematics of Operations Research, Vol. 13, pp. 311–329, 1988.

[6] Holley, R. and Stroock, D., *Simulated Annealing and Sobolev Inequalities*, Comm. Math. Phys. Vol. 115, pp. 553–569, 1988.

[7] Johnson, D.S., Aragon, C.R., Mc Geoch, L.A., and Schevon, C., *Optimization by Simulated Annealing: an Experimental Evaluation (part 1)*, AT&T Bell Laboratories preprint, 1986.

[8] Bernasconi, J., *Statistical Physics and Optimization: Binary Sequences with Small Autocorrelations*, Stochastic Processes in Physics and Engineering, Edited by S. Albeverio, P. Blanchard, M. Hazewinkel and L. Streit, Reidel Publishing Company, pp. 1–16, 1988.

[9] Albeverio, S., Hoegh-Krohn, R. and Streit, L., *Energy Forms, Hamiltonians and Distorted Brownian Paths*, Journ. Math. Phys., Vol. 18, 907–917, 1977.

[10] Golay, M.J.E., IEEE Transactions on Information Theory, Vol. IT 28, No. 543, 1982.

[11] Hinton, G.E., Sejnowski, T.J., and Ackley, D.H., *Boltzmann Machines: Constraint Satisfaction Machines that Learn*, Dept. of Computer Science Technical Report CMU-CS-84-119, Carnegie Mellon University, Pittsburgh, 1984.

[12] Aarts, E., and Korst, J., *Simulated Annealing and Boltzmann Machines*, J. Wiley, New York, 1989.

FIGURES

```
Procedure: Quantum_Annealing.
Begin
   Choose a random initial configuration $\varepsilon$;
   $V_m := V(\varepsilon)$;
   Repeat
      Apply Local_Optimization to the current configuration $\varepsilon$;
      $i = 0$; $j = 0$;
      Repeat
         Generate $\varepsilon'$ by applying Quantum_Transition to $\varepsilon$.
         If $V(\varepsilon') < V_m$ Then
         Begin
            $V_m := V(\varepsilon')$;
            $i := 0$; j:=0;
         End;
         Else
         Begin
            $i := i + 1$;
            $j := j + 1$;
         End;
         If $j > T$ Then
         Begin
            Drill a tunnel from $\varepsilon$ to $\varepsilon'$;
            $j := 0$;
         End;
      Until $i > I$;
   Until time expires;
   Output $V_m$;
End;
```

Figure 1. Main procedure of the QA algorithm. It calls alternatively the two subroutines **Local_Optimization** and **Quantum_Transition**.

```
Procedure: Local_Optimization.
Input parameters: $\varepsilon$;
Begin
    While there is an untested neighbour of $\varepsilon$ Do
    Begin
        Choose an untested neighbour $\varepsilon'$ of $\varepsilon$;
        If $V(\varepsilon') < V(\varepsilon)$ Then $\varepsilon := \varepsilon'$;
    End
    Return $\varepsilon$;
End
```

Figure 2. Subroutine of QA carrying out a local descent from the current configuration.

```
Procedure: Quantum_Transition.
Input parameters: $\varepsilon$;
Begin
    Compute $p_k(\varepsilon) = \psi_\nu(S_k(\varepsilon))$ with $k$ varying over a percentage $P$ of elements
    in the set $\{1, 2, \ldots, n\}$;
    Cumulate all $p_k(\varepsilon)$ on the unitary segment;
    Choose a random number $x \in (0,1)$;
    $\varepsilon' := S_j(\varepsilon)$ where $j$ is the least $k$ such that $x < p_k(\varepsilon)$;
    Return $\varepsilon'$;
End
```

Figure 3. Subroutine of QA generating a transition from the current configuration after having inspected a region around it by means of prospection chains.

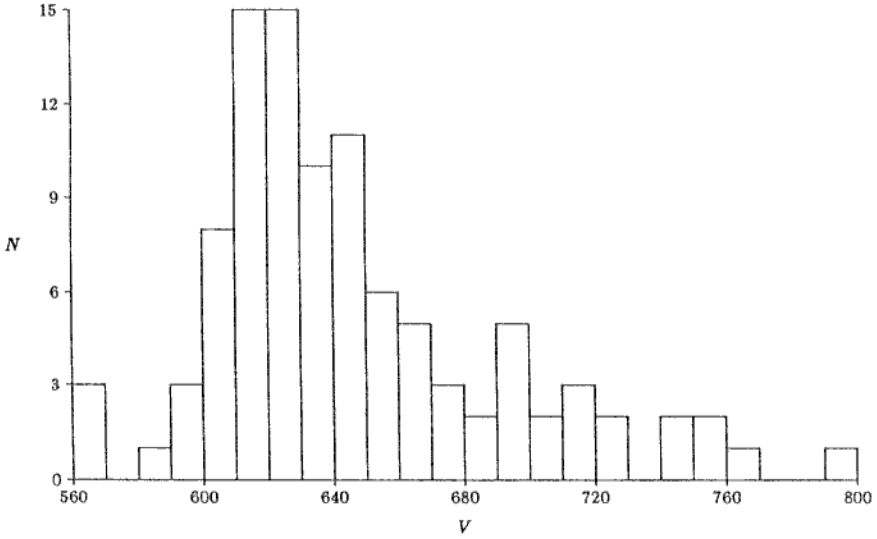

Figure 4. Histogram of the 100 initial configuration values for the Graph Partitioning problem.

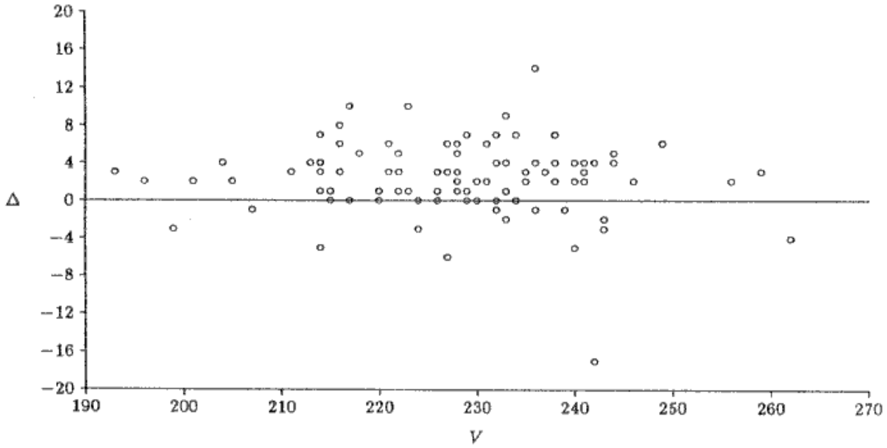

Figure 5. The small circles indicate the performance of QA w.r.t. SA on each instance of the Graph Partitioning problem. In particular, for each point the x-axis denotes the outcome of the algorithm which performed the best on that instance while the y-axis denotes the difference between the output of QA and the output of SA.

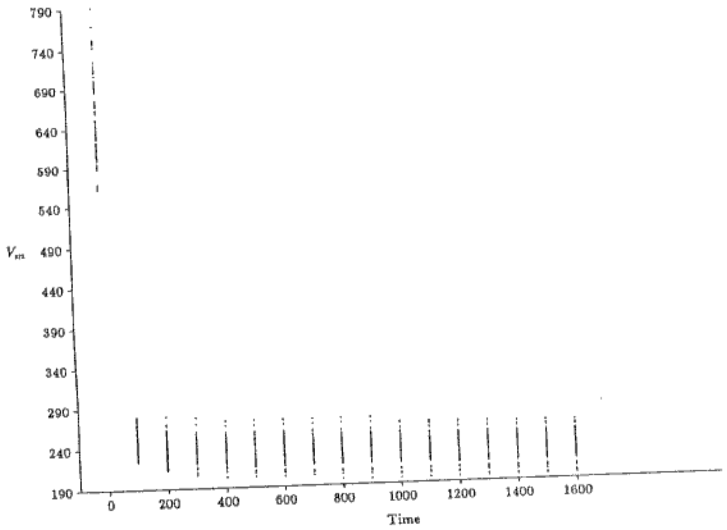

Figure 6. Values of V_m reached by QA on each instance of Graph Partitioning after every 100 time steps.

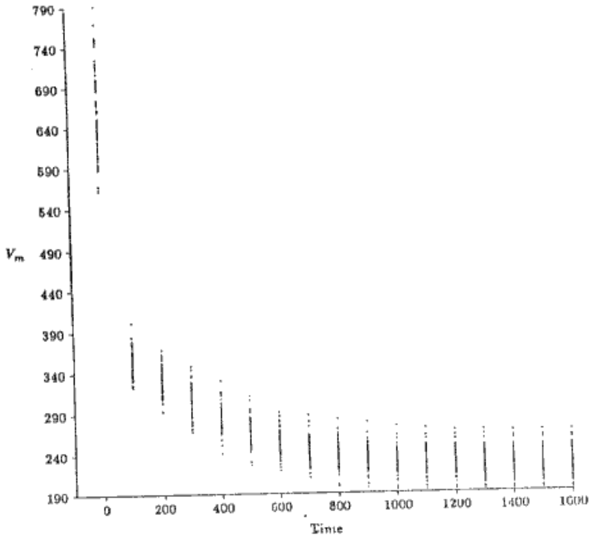

Figure 7. Values of V_m reached by SA on each instance of Graph Partitioning after every 100 time steps.

	Graph Partitioning	String Generation
L	200	50
S	4	2
ν	0.05	0.05
P	0.05	0.05
I	10	10
T	5	5
R	1%	1%

Table 1. Choice of parameter values for QA for each test bed.

	Average	Best
LO	3.7	4.5
SA	4.8	5.4
QA	5.0	5.7

Table 2. Merit factors for binary sequences of length 100 obtained during 100 runs each of LO, SA and QA. For each algorithm the average value and the best value over the 100 runs are reported.