Exercizes on combinatoric optimisation

Bert Kappen

December 6, 2010

Run the program `makedata.m` to generate an instance of the following combinatoric optimisation problem:

$$E = -\frac{1}{2}x^t w x,$$

with w an $n \times n$ symmetric matrix with zero diagonal and $x = (x_1, \ldots, x_n)$ a binary vector: $x_i = \pm 1$. Finding the minium of E is intractable in general because x is binary (what is the solution when x is real and $\|x\| = 1$?).

However, for specific choices of w, the problem can be significantly more or less difficult. For instance, if all elements w_{ij} are positive or zero, there are two optimal solutions:

$$x = \pm(1, \ldots, 1)$$

(show this result). This solution minimizes the cost for each interaction term separately. These systems are called ferro-magnetic.

Instead, when w_{ij} has arbitrary sign, there is typically no global solution x that minimizes each term $w_{ij}x_ix_j$. Because not all terms can be satisfied simultaneously, these systems are called frustrated. A simple example is the interaction matrix

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{pmatrix}$$

the global minimum is the best compromise for all interaction terms taken together.

We will study several methods to approximately solve this problem.

1 Iterative Improvement

The iterative improvement is the simplest method for discrete optimization. It consists of the following ingredients:

Initialization We start with a random initialization of x. Compute the cost $E(x)$.

Definition of neighborhood The iterative improvement algorithm compares the cost of x with the cost of neighboring states x'. If the new cost is lower than the old cost, x is replaced by x':

$$E(x') < E(x), \rightarrow x := x'$$

Otherwise, x' is rejected. Clearly, the larger the neighborhoud, the more time that is needed for convergence and the better the solution that is obtained.

Termination When no further improvement is obtained for any state in the neighborhood of x, the algorithm terminates.

Excersizes

Use the program `optimizer.m` to apply the interative improvement method to the combinatoric optimization problem.

- Compare the ferro-magnetic and frustrated problems. How many restarts are needed for reproducible results?
- For the frustrated problem, study the influence of the neighborhood size on the quality of the solution and the cpu time required.

2 Simulated annealing

Simulated annealing is an advanced method for combinatoric optimization. The idea is to convert the optimization problem to a probability estimation problem by defining the probability distribution

$$p(x) = \frac{\exp(-\beta E(x))}{Z} \tag{1}$$

Z is a normalization constant and β is an adjustable parameter, in physics referred to as the inverse temperature. For small β, $p(x)$ looks like an inverted version of $E(x)$. For large β, $p(x)$ becomes peaked around the global minimum of $E(x)$. See fig. 1.

This suggests the following algorithm for finding the minimum:

Initialization Choose a random initialization of x. Choose the initial value of β_{init} such that the sampling will reach all parts of the x-space with high probability, independent of the particular initial value of x.

Cooling schedule Choose a increasing sequence of β values $\beta_1, \ldots, \beta_{T2}$. Then for each β_i:

Markov chain Use the Metropolis method to sample $T1$ samples from the distribution $p(x) \propto \exp(-\beta_i E(x))$. Estimate

$$\langle E \rangle = \frac{1}{T1} \sum_{i=1}^{T1} E(x_i), \quad \sigma^2 = \frac{1}{T1} \sum_{i=1}^{T1} (E(x_i) - \langle E \rangle)^2$$

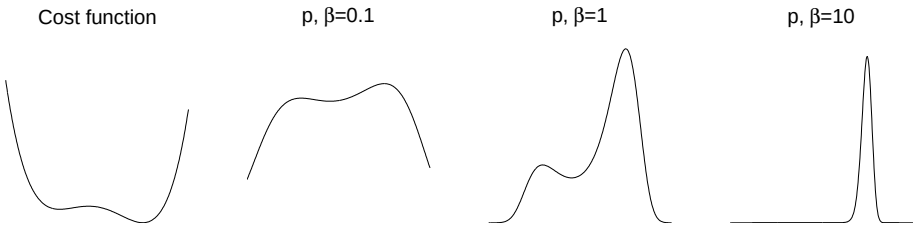

Figure 1: Simulated annealing. The algorithm samples the distribution Eq. 1 for increasing values of β. If this is done carefully, the global minimum of $E(x)$ is obtained.

Termination For increasing β, the spread in values of E that are obtained in the Markov chain decreases. The algorithm terminates when the spread is zero.

An illustration is given in Fig. 2.

2.1 Excersizes

Use the frustrated problem with $n = 50$.

- Reproduce parts of figure 31.11 of MacKay a ferromagnetic system of $n = 50$ spins, ie. estimate the mean energy and the standard deviation of the energy. Repeat this for a frustrated system by choosing random couplings.
- Study the effect of initial β and the cooling schedule (`factor`) and the length of the Markov Chain $T1$ on the performance and reproducibility of the SA result. Estimate the critical temperature in both cases. Use a larger n to get more accurate results if your computer or patience allows.
- Which method (SA or Iter) has the best performance in terms of speed and quality?
- Put $n = 200$ and make an instance with the random seed fixed (rand('state',0)). Try to find the best solution and compare with your fellow students.

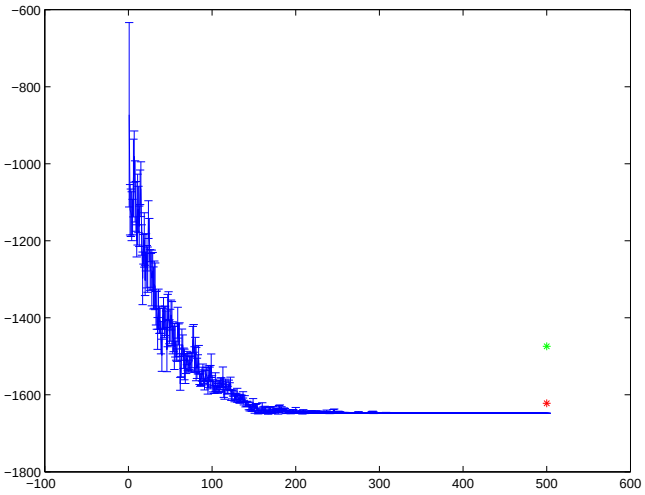

Figure 2: Blue curve: A typical SA run. $n = 200$ frustrated problem. Initial $\beta_1 = 0.0486$, $\beta_{i+1} = 1.01 * \beta_i$. Markov chain length is $T1 = 1000$, nearest neigbors only. Green dot: best out of 100 iterative improvement runs with nearest neigbors. Red dot: best out of 10 iterative improvement runs with pair flips.