Cooperative Behavior in Coupled Simulated Annealing Processes with Variance Control

Samuel Xavier-de-Souza†, Johan A.K. Suykens†, Joos Vandewalle†, and Désiré Bollé‡

†K.U.Leuven, ESAT/SCD-SISTA
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
Email: [samuel.xavierdesouza, johan.suykens, joos.vandewalle]@esat.kuleuven.be

‡K.U.Leuven, Institute for Theoretical Physics
Celestijnenlaan 200D, B-3001 Leuven, Belgium
Email: desire.bolle@fys.kuleuven.be

Abstract—In this paper we describe the use of coupling to interconnect different Simulated Annealing (SA) processes. The objective is to allow cooperative behavior among the processes in order to improve performance. Coupled Simulated Annealing (CSA) permits a high degree of parallelization while delivering much better results than a typical Parallel SA (PSA) algorithm. This is possible due to the introduction of coupling in the acceptance probability functions. Moreover, the coupling also allows controling the variance of the acceptance probabilities. This is especially important because it reduces the sensitivity to initial parameters, while guiding optimization to quasi-optimal runs. It was observed that the solutions generated by CSA are more concentrated around the global optimum, while PSA has concentrations of solutions often in unfavorable regions of the cost function. Also, the number of iterations per process necessary to reach a given minimum energy tolerance decreases exponentially when the number of optimizers is increased.

1. Introduction

Coupling has been applied in recent years to many different engineering applications and methods. It has proven to be helpful for synchronizing chaotic dynamical systems [6], where two identical systems can synchronize by coupling one of their state variables. Such approaches have attracted the interest of many researchers which found applications in many different fields, including communication and robotics.

Coupling has also been applied to develop the concept of Cellular Neural Networks (CNNs) [2], which features many dynamical systems, or cells, disposed in a regular grid with coupling between neighbor cells. Nowadays, CNNs gave the researchers in the field, another paradigm for information processing which can take advantages of massively parallel VLSI implementations to process data at extraordinary speed. This is only possible due to the clever yet simple coupling applied to cells.

In optimization, coupling is used to aid gradient based methods to escape from local minima [8]. In that approach, coupling of local optimization processes outperforms multi-start methods by minimizing the average cost of all coupled processes, subject to synchronization constraints between the solutions of the individual processes.

In this paper, we analyze the performance and behavior of an algorithm called Coupled Simulated Annealing (CSA). Basically, it features several SA processes running in parallel, coupled by their acceptance probabilities. The coupling provides CSA with the ability to perform optimization which has a good balance between localized search and global exploration for multi-modal problems having several local minima. Moreover, the structure of CSA permits the variance control of the acceptance probabilities via the acceptance temperature. This not only improves the performance, but also reduces the sensitivity of the algorithm to the initial acceptance temperature.

This paper is organized as follows. In Section 2, we describe the CSA algorithm and its characteristics, and we present the variance control of the acceptance probabilities. The results of our experiments are presented in Section 3, followed by the conclusion.

2. Coupled Simulated Annealing

While in classical Simulated Annealing [4] the acceptance probability of an uphill move is often given by the Metropolis rule, which depends only on the current and the probing solution, in CSA the decision of accepting such a move takes into account other current solutions. Namely, this probability depends also on the costs of the solutions in a set $\Theta \in \Omega$, where Ω is the set of all possible solutions. This dependence is given by the coupling term γ which is generally a function of the costs of the solutions in Θ. In CSA, the acceptance probability A_Θ and the coupling term γ are given by

$$A_\Theta(\gamma, x_i \rightarrow y_i) = \exp\left(\frac{E(x_i) - \max_{z\in\Theta}(E(z))}{T_k^{ac}}\right)/\gamma, \quad (1)$$

with

$$\gamma = \sum_{\forall x\in\Theta} \exp\left(\frac{E(x) - \max_{z\in\Theta}(E(z))}{T_k^{ac}}\right). \quad (2)$$

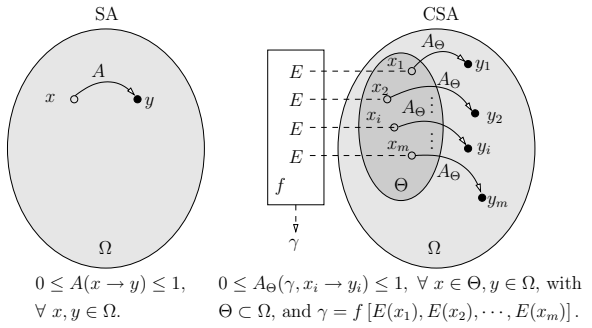

Figure 1: The general difference between SA and CSA lies in the acceptance process. While SA only considers the current solution x for the acceptance decision of the probing state y, CSA considers many current states in the set Θ, which is a subset of all possible solutions Ω, and accepts each probing state y_i based not only on the corresponding current state x_i but by considering also the coupling term γ, which depends on the energy of all other elements of Θ.

Here T_k^{ac} is the acceptance temperature and x_i and y_i denote an individual solution of Θ and its corresponding probing solution, respectively. These two equations define A_Θ as a probability. The sum of probabilities of *leaving* any of the current states equals 1. Fig. 1 depicts the main differences between SA and CSA.

Functionally, CSA differs from an ensemble of SA processes [5, 7] because of two aspects. The first is the coupling which modifies the acceptance probability of each process according to the energy of the current solutions of all processes. The other aspect is *blind acceptance*. While downhill moves are always accepted during the optimization process, in CSA, the decision to accept an uphill move does not depend on the destination of the move, or target solution. In other words, $A_\Theta(\gamma, x_i \to y_i)$ is not a function of y_i. At first sight, this property may not seem helpful because the target solution may be much worse than the original one. Moreover, it may be argued that an excellent solution can easily be lost with such an approach. However, there are good reasons to use blind acceptance in CSA. The first one is that blind acceptance of uphill moves improves exploration of the energy surface. This is an essential property to solve hard multi-modal optimization problems. Additionally, due to the coupling of the acceptance functions of the different processes, uphill moves are much more likely to happen in processes that hold poor solutions than with processes holding good solutions. Therefore, the best solutions are not easily lost, whereas the poorest are fastly discarded.

The acceptance temperature in CSA is not responsible for weighting the difference between the energy of the probe and current solutions as it is in classical SA, but rather it is responsible for weighting the proportion that each acceptance probability has to the overall sum of the probabilities, which in any case must be equal to 1. This temperature can then be used to control the variance of the probabilities regardless of the current energies. Although the ideal variance value is unknown to us, our experiments with different cost functions show that values in the neighborhood of the maximum variance deliver the best results. Typically, we recommend 99% of the maximum variance value. A very simple control rule can be used to steer this variance to the desired value. It can be done in the following manner:

$$\begin{aligned} &\text{if } \sigma^2 < \sigma_D^2 \text{ then, } T_k^{ac} = T_{k-1}^{ac}\,(1-\alpha)\,, \\ &\text{if } \sigma^2 > \sigma_D^2 \text{ then, } T_k^{ac} = T_{k-1}^{ac}\,(1+\alpha)\,, \end{aligned}$$

where σ_D^2 is the desired variance value and α is the rate for the increase or decrease of the temperature, typically in the range of $(0, 0.1]$. If the value of the acceptance variance is below its desired value, the acceptance temperature is decreased by a factor of $1-\alpha$, otherwise, it is increased by a factor of $1+\alpha$. Such simple variance control can be applied only due to the coupling in the acceptance probability function. It substitutes a schedule for the acceptance temperature and more importantly, it works for any initial acceptance temperature. This is important because the setup of initial parameters in SA is most of the time a very cautious work. With this approach, we eliminate two initialization aspects at once, which are the choices for an acceptance schedule and an initial acceptance temperature. In return, two other parameters are introduced, α and σ_D^2, but these have a well defined operating range and are much less dependent on the optimization problem at hand.

3. Experiments and Results

We have tested CSA and the variance control explained above in a set of multi-modal functions with dense local minima. For comparison, we have also performed experiments using CSA without Variance Control (CSAwoVC) and using a Parallel SA (PSA) [1] algorithm. This parallel version of SA features several sequential SA processes running in parallel and sharing the best current solution. As soon as one of the parallel instances finds a better solution, all the others are informed about the new current solution and proceed to the next generation step.

3.1. Test Problems

We have used a set of four D-dimensional functions as test problems for the algorithms under analysis. All four functions share the property of being multi-modal with dense and uniform sets of local minima and are described by the following equations.

$$\begin{aligned} f_1(x) &= 1 - \textstyle\prod_{i=1}^{D} \operatorname{sign}\left(\frac{\sin x_i}{x_i}\right)\left|\left(\frac{\sin x_i}{x_i}\right)^{\frac{1}{4}}\right|. \\ f_2(x) &= \textstyle\sum_{i=1}^{D}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]. \\ f_3(x) &= -20\exp\left(-0.2\sqrt{\tfrac{1}{D}\textstyle\sum_{i=i}^{D} x_i^2}\right) \\ &\quad - \exp\left(\tfrac{1}{D} * \textstyle\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e. \\ f_4(x) &= \tfrac{1}{4000}\textstyle\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1. \end{aligned}$$

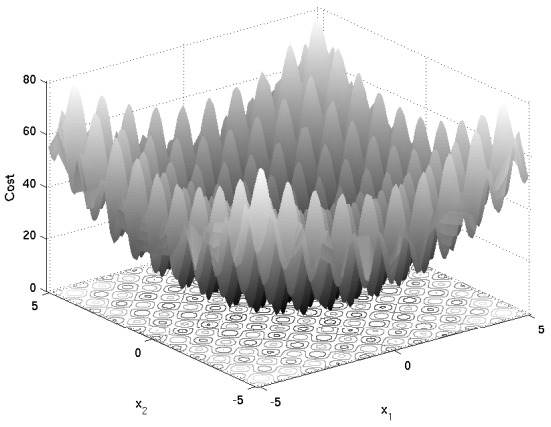

Figure 2: Surface plot of a 2D version of test problem $f_2(x)$.

The ranges used for the values of x in dimension D were $[-50, 50]$, $[-5.12, 5.12]$, $[-32, 32]$, and $[-600, 600]$, respectively for functions f_{1-4}. In Fig. 2 a 2-dimensional instance of the second problem is depicted. The problems under study here are those where the global optimum is hidden behind many local optima. Therefore, the challenge faced by the proposed CSA algorithm is to find the global optimum at all, avoiding faster convergence that may lead to local optimum traps.

3.2. Results and Discussion

We have performed a variety of experiments in order to assess the performance of CSA and analyze its behavior. For all performed experiments, we have used the schedule in [9] for the generation procedure of all tested algorithms. For the acceptance procedure, we used the schedule in [3] for the PSA and CSAwoVC algorithms. For CSA, the temperature was used to perform the control of the acceptance probabilities.

The coupling in CSA has the objective to increase cooperation among optimizers and provide better acceptance decisions. This ensures that a process in a higher energy region concentrates on exploration rather than on localized search, as opposed to a process in a lower energy region. Such effect does not exist in PSA, where the search is concentrated in isolated spots of the energy surface. This concentration only changes its focus when a better solution is found. Fig. 3 has illustrative samples of typical PSA and CSA runs. Every sub-plot in the figure depicts the spread of the solutions visited by each algorithm. The global optimum lies exactly in the center of each plot. It can be seen that while PSA features many concentrated sets of solutions sparsely distributed, which are not necessarily around the global optimum, CSA features mesh-like concentrations which are around the center of the plot. It can also be seen that although each plot has the same number of visited solutions, CSA plots seems to fill better the solution space. We have observed the same results for other test problems. These results suggest that CSA has very good exploration characteristics, especially when qualitatively compared with PSA.

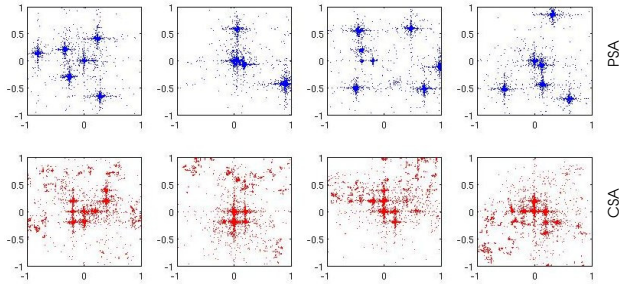

Figure 3: Plots of the position of each visited solution for test problem $f_2(x)$ in dimension $D = 2$. The top row holds the results for PSA while the second row holds those for CSA. CSA has a wider spread while concentrating on the global optimum region, while PSA has more isolated sets not necessarily in the region of the global optimum.

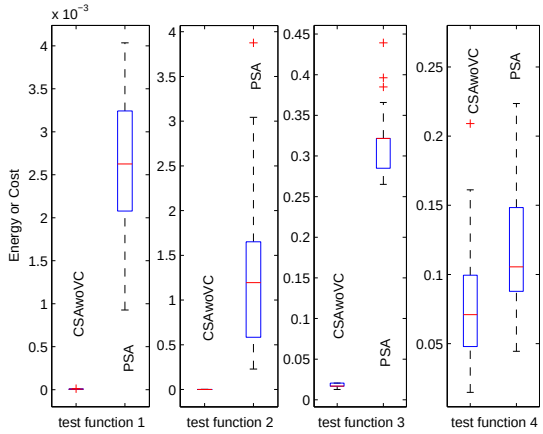

Figure 4: Box-plots of 50 optimization runs of the CSAwoVC and PSA algorithms, for all four test problems. For all algorithms and functions, $m = 8$ parallel processes were used with dimension $D = 8$, 500 steps per fixed temperature, the same cooling schedules and initial temperatures, and a maximum number of cost-function evaluations equal to 20,000 per processing node.

Fig. 4 shows the performance of CSAwoVC versus PSA for our four test problems. Both algorithms had their initial temperature obtained by an exhaustive search. It can be seen that CSAwoVC has a much better performance, except for test function f_4.

We tested the effect of the variance control in the performance of CSA by running an experiment that compares it with CSAwoVC, and two PSA setups for test function f_2, with $D = 5$ and $m = 5$. The experiment consists of 1000 runs of each algorithm with 1000 iterations, which is not much but it is enough to show how the algorithms behave in the early stages of the optimization. The acceptance temperature varied from 0.01 at iteration 1 until twice the mean energy at iteration 1000, except in one of the PSA setups, which had fixed $T_0^{ac} = 0.1$. Box-plots of the results are

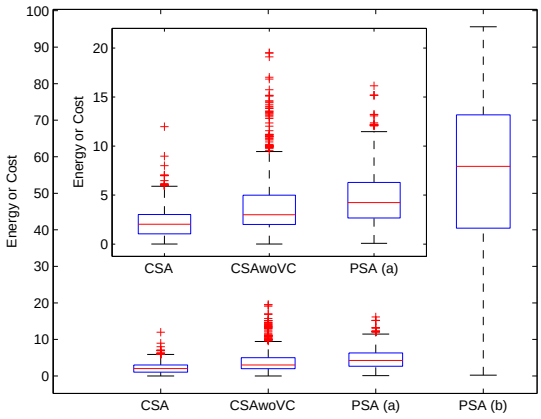

Figure 5: Box-plots over 1000 runs of the CSA, CSAwoVC, and PSA (a) for f_2. T_0^{ac} for CSA, CSAwoVC, and PSA (b), varied from 0.01 to 185 along the 1000 runs. The value of T_0 was found by exhaustive search and it was $T_0 = 2$ for CSA algorithms, and $T_0 = 0.6$ for both PSA. The maximum number of function evaluations was set to 1000 per optimizer.

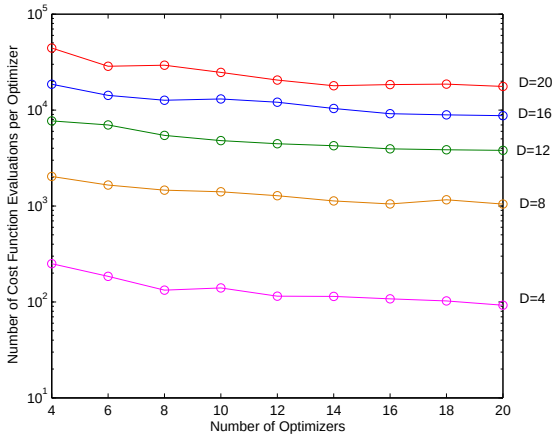

Figure 6: Plots of performance curves of CSA for different dimensions D of f_2. Every curve was obtained using the same initial temperatures. The vertical axis represents the number of necessary cost function evaluations per optimizer in order to reach a minimum energy tolerance. For all dimensions, this number decreases approximately exponentially with the number of optimizers. Note that the values in the vertical axis are on a logarithmic scale.

shown in Fig. 5. The reader can observe that for the PSA setup in which we used the variation in the initial acceptance temperature, the performance was much poorer than the one of the other algorithms. This happens because the performance of PSA, as of many other SA algorithms, depends substantially on the initialization parameters. For the PSA with fixed initial acceptance temperature, the results improved much. Nevertheless, both CSA algorithms performed better. Besides the superior performance of CSA with the variance control w.r.t. all other algorithms, it presented also the smallest variance in the results. A zoomed version of the box-plots for the first 3 algorithms can be seen in the inner plot of Fig. 5.

At last, we performed experiments with CSA to check the scaling of the necessary number of iterations to reach a given minimum energy tolerance, with an increase in the number of optimizers. These tests were executed for function f_2 with several different values for D. The results can be seen in Fig. 6, which is presented with a logarithmic scale in the vertical axis for better visualization. This figure suggests that an increase in the number of optimizers decreases exponentially the number of necessary iterations to reach a given energy tolerance, regardless of the dimension D of the problem.

4. Conclusion

We have described the algorithm Coupled Simulated Annealing (CSA). In this algorithm, several Simulated Annealing (SA) processes are coupled by their acceptance probabilities. Additionally, we have presented a straightforward control rule for the variance of the acceptance probabilities among the different processes. The results confirmed the positive effect of the cooperation introduced by the coupling in the search capabilities of the CSA algorithm.

Acknowledgments Research supported by: • Research Council KUL: GOA-Mefisto 666, GOA-AMBioRICS, BOF OT/03/12, Center of Excellence Optimization in Engineering; • Flemish Government: ◦ FWO: PhD/postdoc grants, G.0407.02, G.0080.01, G.0211.05, G.0499.04, G.0226.06, research communities (ICCoS, ANMMM); ◦ Tournesol 2005 • Belgian Federal Science Policy Office IUAP P5/22. J. Suykens is an associate professor, D. Bollé and J. Vandewalle are full professors, all with K.U. Leuven Belgium. DB thanks Marcus Müller for useful correspondence, and SXS thanks João Ramos and Müştak E. Yalçın for insightful discussions.

References

[1] E. H. L. Aarts and J. H. M. Korst, *Simulated Annealing and Boltzmann Machines*. New York: Wiley (Interscience), 1989.

[2] L. O. Chua and T. Roska, *Cellular neural networks and visual computing: foundations and appl.* Cambridge Univ. Press, 2002.

[3] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, Nov 1984.

[4] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[5] S. Lee and K. Lee, "Synchronous and asynchronous parallel simulated annealing with multiple Markov chains," *IEEE Trans. on Parallel and Distributed Systems*, vol. 7, no. 10, pp. 993–1008, Oct 1996.

[6] L. Pecora and T. Carroll, "Synchronization in chaotic systems," *Phys. Rev. Lett.*, vol. 64, pp. 821–824, 1990.

[7] G. Ruppeiner, J. M. Pedersen, and P. Salamon, "Ensemble approach to simulated annealing," *J. Phys. I*, vol. 1, pp. 455–470, 1991.

[8] J. A. K. Suykens, J. Vandewalle, and B. De Moor, "Intelligence and cooperative search by coupled local minimizers," *Int. J. of Bifurcation and Chaos*, vol. 11, no. 8, pp. 2133–2144, Aug 2001.

[9] H. H. Szu and R. L. Hartley, "Fast Simulated Annealing," *Physics Letters A*, vol. 122, pp. 157–162, 1987.