On the convergence of Parallel Simulated Annealing

Christian Meise
Universität Bielefeld

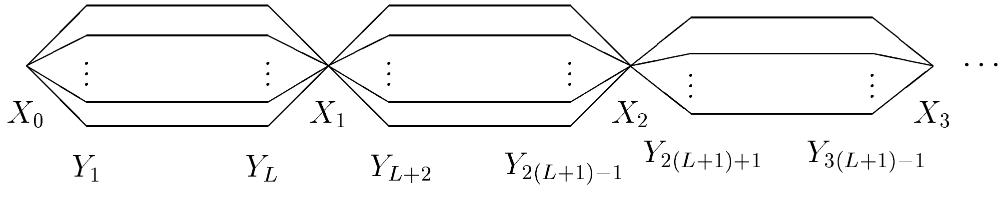
Abstract. Using the Matrix-Tree Theorem and coupling methods the convergence of the Parallel Chain (PC) algorithm to the set of global minima is established for various selection functions. It is illustrated that there may be convergence to a set of non-global minima when selecting one of the best states (Best-Wins strategy). In the latter case the convergence of the homogeneous PC algorithm is proved for sufficiently large length of the parallel Markov chains and number of parallel Markov chains. For any selection function we show that if the PC algorithm is convergent, then the cooling schedule must satisfy $\sum_{n=0}^{\infty} \exp(-\beta(n)\theta) = \infty$, $(\theta > 0)$.

Keywords: Simulated Annealing, Weak ergodicity, Strong ergodicity, Coupling

1 Introduction and Results

Let X be a finite set and let $U : X \to \mathbb{R}_+$ be a non-constant function that should be minimized. We denote by $X_{\min}$ the set of global minima of U. Suppose that we have $p \geq 1$ processors. Choose any starting point $x_0 \in X$ and let each processor simulate a Markov chain of length $L \geq 1$ starting in x_0. After L transitions the simulation is stopped and a state x_1 is selected from the p states according to a selection strategy. Then each processor simulates a Markov chain of length L starting in x_1. Again at the end of the simulation a state x_2 is selected from the p states and the next simulation starts from x_2, etc.

Figure 1.1 *Illustration of the algorithm's working method.*



Of course the number of parallel Markov chains that are simulated need not be fixed. We denote by $r(n) \geq 1$ the number of parallel Markov chains starting at time $n(L+1)$ and suppose that $(r(n))_n$ is monotone decreasing. Moreover we assume that $\lim_{n\to\infty} r(n) =: p \in \{1, \ldots, r(0)\}$ exists.

Remark 1.2 *In our model the Markov chain transitions should depend on U: If only worst states, e.g. states with maximal costs, are selected, there will be no convergence to the set of global minima, when using Markov chain transitions independent of U. In comparison to the model of genetic algorithms in [C94], minimization of U in the PC algorithm is achieved via the Simulated Annealing transition kernel and not by appropriate selection functions. Recall that the operations such as crossover and mutation used in the model of Cérf do not depend on U. Therefore optimization in the model of Cérf can only be achieved by using appropriate selection functions.*

In the sequel we specify the transition probabilities for each of the $r(n)$ Markov chains starting at time $n(L+1)$. Let q be an irreducible exploration kernel on X, that also satisfies the following reversibility condition

$$q(x,y) > 0 \quad \Longleftrightarrow \quad q(y,x) > 0 \text{ for all } x, y \in X. \tag{1}$$

Moreover let $\beta(n,i)$ $(n \in \mathbb{N}_0,\ 0 \leq i < L)$ denote a sequence satisfying the following properties:

1. $\beta(n) \equiv \beta(n,0) =\leq \beta(n,i) \leq \beta(n, L-1) \leq \beta(n+1,0)$, for $n \in \mathbb{N}$ and $0 \leq i < L$
2. $\lim_{n\to\infty} \beta(n) = \infty$.
3. $\lim_{n\to\infty} \beta(n, L-1) - \beta(n) = 0$

The transition probabilities for the $r(n)$ Markov chains, being simulated at time $n(L+1)$, are given by the classical Simulated Annealing transition kernel, e.g. at time $n(L+1)+i$, $(0 \leq i \leq L-1)$ the transition probability for the j-th Markov chain, $j \in \{1, \ldots, r(n)\}$, is given by

$$P\left(Y^j_{n(L+1)+i+1} = z \mid Y^j_{n(L+1)+i} = y\right) = p_{\beta(n,i)}(y,z) \quad \text{for } y, z \in X, \tag{2}$$

where

$$p_\beta(x,y) = \begin{cases} \exp\left(-\beta\,(U(y) - U(x))^+\right) q(x,y) & : \quad x \neq y \\ 1 - \sum_{z\neq x} p_\beta(x,z) & : \quad x = y. \end{cases} \tag{3}$$

The selection behavior is described by selection functions.

Definition 1.3 *A function $F : \{1, \ldots, s\} \times X \times X^s \to [0,1]$ is called a selection function (of degree s), if*

$$\textstyle\sum_{m=1}^s F(m, x, y_1, \ldots, y_s) = 1 \text{ for } (x, y_1, \ldots, y_s) \in X \times X^s.$$