PROBABILISTIC HILL CLIMBING ALGORITHMS: PROPERTIES AND APPLICATIONS

by

F. Romeo and A. Sangiovanni-Vincentelli

Memorandum No. UCB/ERL M84/34

13 March 1984

PROBABILISTIC HILL CLIMBING ALGORTIHMS: PROPERTIES AND APPLICATIONS

by

F. Romeo and A. Sangiovanni-Vincentelli

Memorandum No. UCB/ERL M84/34

13 March 1984

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Probabilistic Hill Climbing Algorithms: Properties and Applications

Fabio Romeo[1] and Alberto Sangiovanni-Vincentelli

Department of EECS

University of California

Berkeley, California 94720

Abstract

Simulated annealing proposed by Kirckpatrick et al. has proven to be an effective technique to solve general combinatorial optimization problems. Its derivation was based heavily on the analogy between combinatorial optimization problems and the annealing process in statistical physics.

A mathematical model of the operations of Simulated Annealing is needed to understand the essential features which guarantee the algorithm to perform efficiently and to improve the speed of execution. Markov chains are proposed as mathematical models of the Simulated Annealing algorithm. Using these models, it has been possible to prove that under certain assumptions on the rules used by the algorithm to generate the configurations of the problem and on the time spent at each temperature, the Simulated Annealing algorithm generates a global optimum solution with probability one.

This result has made possible the definition of a general class of algorithms with the same statistical properties: the class of probabilistic hill-climbing methods. The mathematical properties of this class are presented and rules on the selection of annealing schedules are obtained from these properties.

---

[1] On leave of absence from Honeywell Information Systems Italy

The theoretical work on Simulated Annealing has been done in parallel with an experimental study aimed at building a general package for the layout of integrated circuits using Simulated Annealing, Timber Wolf, developed at the University of California, Berkeley by C. Sechen.

1. Introduction

Many combinatorial optimization problems belong to a class of problems which are difficult to solve, i.e., the class of NP-complete problems[GAR79]. For these problems, there is no known algorithm whose worst-case complexity is bounded by a polynomial in the size of the input. Heuristic algorithms are used to solve NP-complete problems approximately, i.e. to find "good" solutions which are "close" to the optimum. These algorithms explore a discrete space of admissible configurations, S, in a deterministic fashion. Starting from an initial configuration j_0, a sequence of configurations is selected until a satisfactory one is found. The rules according to which a configuration is generated and the algorithm terminates, specify the algorithm. Often the search terminates with a local minimum, i.e. with a configuration $\hat{j}$ such that if we denote by $c(j)$ the cost of j and by $S(j)$ the set of configurations that can be generated from j by the algorithm in one step, $c(\hat{j}) \leqslant c(j), \forall\, j \in S(\hat{j})$. This is often due to the fact that heuristic algorithms are "greedy", i.e., only moves which reduce "maximally" the cost are accepted.

To avoid this behavior, randomizing algorithms (e.g. [SCH80]) can be devised which generate the next configuration randomly. The configuration is recorded as a new temporary solution if its cost is lower than the present temporary solution. The algorithm terminates after a certain number of moves. Randomizing algorithms perform well if the number of optimal solutions is fairly high, since the probability of stopping at an optimum is proportional to the ratio between the number of optimal configurations and the number of total configurations. Note that randomizing algorithms can "climb hills", i.e., moves that generate configurations of higher cost than the present one are accepted.

Simulated Annealing as proposed by Kirckpatrick et al. [KIR83], allows "hill climbing" moves but these moves are accepted according to a certain criterion which takes the cost into consideration and not blindly as randomizing algorithms. The controlling mechanism is based on the observation that combinatorial optimization problems with a large configuration space exhibit properties similar to physical processes with many degrees of freedom.

In particular, bringing a fluid into a low energy state such as growing a crystal, has been considered in [KIR83] similar to the process of finding an optimum solution of a combinatorial optimization problem. Annealing is a well-known process to grow crystals. It consists in melting the fluid and then lowering the temperature slowly until the crystal is formed. The rate of decrease of temperature has to be very low around the freezing temperature. The Metropolis Monte Carlo method [MET53, BIN78] can be used to simulate the annealing process. It has been proposed as an effective method for finding global minima of combinatorial optimization problems. This method when applied to combinatorial optimization generates moves randomly and checks whether the cost of the new configuration satisfies an acceptance criterion based on temperature. If the cost decreases, the move is accepted. If the cost increases, then a random number between zero and one is generated and compared with $f\ (\Delta c_{ij}, T) = \exp(\frac{-\Delta c_{ij}}{T})$ where Δc_{ij} is the change in cost obtained by moving from configuration i to j and T is temperature, the controlling parameter. If the random number is larger than f, the move is accepted, otherwise the move is discarded. Note that the higher the temperature is, the more likely it is that a "hill climbing" move is accepted. Note also that "hill climbing" moves are less and less probable as the temperature is decreased. A certain number of moves are generated and checked before a decrease in temperature is allowed. The initial temperature, the number of moves generated at each temperature and the rate of decrease of temperature are all important parameters that affect the speed of the algorithm and the quality of the final configuration. Experimental results [KIR83, VEC83, SEC84, JOH84] show that Simulated Annealing produces very good results when compared to other techniques for the solution of combinatorial optimization problems such as those arising from the layout of integrated circuits, at the expense of large computer time (a 1,500 standard cell placement problem can take as much as 24 hours of a VAX 11/780 [SEC84]).

A mathematical analysis of the algorithm is very important to understand the essential features which make the algorithm work well and to suggest techniques for controlling its operation. Markov chains [KAR73, FEL70, DOO51, FRI71] can be used as a mathematical

model of Simulated Annealing. We proved that under certain assumptions on the number of moves generated by the algorithm at each temperature, Simulated Annealing produces asymptotically the optimum solution of combinatorial optimization problems with probability 1. A similar proof was independently proposed by Lundy and Mees [LUN84]. The proof has underlined the essential properties of the algorithm, so that we have been able to derive a class of "probabilistic hill-climbing algorithms" that have the same asymptotic properties of Simulated Annealing.

The paper is organized as follows. In Section 2, the class of Probabilistic Hill-Climbing (PHC) algorithms is formally defined and its mathematical representation in terms of Markov chains introduced. In Section 3, basic definitions and theorems specifying the properties of Markov chains relevant to the analysis of PHC algorithms are reviewed. In Section 4, the assumptions on the parameters of the PHC algorithms needed to guarantee the optimality properties are introduced and the convergence theorems are proved. In Section 5, additional results which give insight on how to select the parameters of PHC algorithms are presented and some experimental results are provided. In Section 6, new research directions and concluding remarks are given.

2. Probabilistic Hill Climbing Algorithms

Given a combinatorial optimization problem specified by a finite set of configurations or states S and by a cost function c defined on all the states $j \in S$, Probabilistic Hill Climbing (PHC) algorithms are characterized by a rule to generate randomly a new state or configuration with a certain probability, and by a random acceptance rule according to which the new configuration is accepted or rejected. A parameter T controls the acceptance rule. We assume that $T \geqslant 0$ and that an updating rule generates a monotonically decreasing sequence $\{T_m\}$, $m = 1,2,\ldots$ with limit zero. PHC algorithms define a random variable X which takes values on the states generated and accepted by the algorithm. Their structure is shown below.

PHC Algorithm Structure (j_0, T_0)

```
{
/* Given an initial state j0 and an initial value for the parameter T , T0 . */
T = T0;
X = j0;
while( "stopping criterion" is not satisfied )
     {
     while( "inner loop criterion" is not satisfied )
          {
          j = generate (X )
          if( accept( c ( j ), c (X ), T )
               X = j ;
          }
          T =update (T )
     }
}
```

The acceptance of a new state j is determined by **accept**, whose structure is shown below.

```
accept( c(j), c(i), T )

    {

    /*
     returns 1 if the cost
     variation passes a test.
     T is the control parameter.
    */

    Δc_ij = c(j) - c(i);
    y = f (Δc_ij, T );
    r = random (0,1);

    /*
     random is a function which returns a pseudo
     random number uniformly distributed on the
     interval [ 0 , 1 ] .
    */

    if (r < y )
        return(1);
    else
        return (0);
    }
```

The acceptance strategy is essentially represented by the function f which takes values on the interval [0,1]. It is possible to vary its shape by adjusting the parameter T.

Remark 2.1. Simulated annealing [KIR82], belongs to the PHC class. In Simulated Annealing, the control parameter T, called "temperature", is updated by means of the following law

$$T_{new} = \alpha T_{old} \qquad 0 \leqslant \alpha \leqslant 1$$

and hence it satisfies the property that $T \geqslant 0$ and that the sequence of updates converges to zero. The acceptance function for the Simulated Annealing is

$$f\ (\Delta c_{ij}, T\ ) = e^{-\frac{c(j) - c(i)}{T}} \quad . \tag{2.1.a}$$

for $\Delta c_{ij} > 0$ and

$$f(\Delta c_{ij}, T) = 1 \qquad \text{(2.1.b)}$$

otherwise. Hence, it takes values on the interval [0,1].

PHC algorithms applied to a combinatorial optimization problem can be represented by Markov chains [KAR73, FEL57, FRI71]. We derive this model on a simple example: a linear placement problem.

Suppose that 3 interconnected modules $\{a, b, c\}$ have to be placed on a monodimensional grid, so that the global length of interconnections is minimized. The state space S consists of 6 configurations, all the possible placements of the three modules (3!), i.e., $S = \{1, \ldots, 6\}$

$$1 = \{a, b, c\}$$
$$2 = \{c, a, b\}$$
$$3 = \{b, c, a\}$$
$$4 = \{a, c, b\}$$
$$5 = \{c, b, a\}$$
$$6 = \{b, a, c\}.$$

We assume that the generation of new configurations is done by PHC algorithms applied to this problem by exchanging the positions of two elements. In this case, the set $S(1)$ is equal to $\{4,5,6\}$. All the other $S(i)$'s can be obtained easily. Note that for all $i \in S$, $|S(i)|=3$.

For each placement i, we denote by $S(i) \subseteq S$ the set of all the states $j \in S$ such that there is an arc (i, j) in the graph. This is the set of all the configurations that can be generated by a PHC algorithm with the generating rule introduced before in one step starting from i.

The graph shown in fig. (2.1) is a schematic representation of the generating rule. Each node of the graph represents one of the configurations; there is an arc (i, j) in the graph if j can be obtained from i by interchanging the positions of two modules.

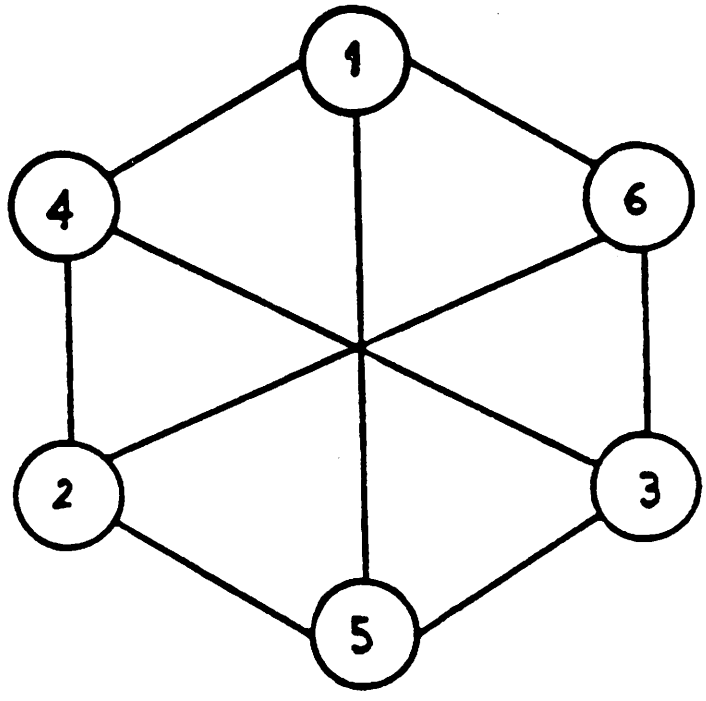

Fig. 2.1. Schematic representation of the generating rule.

For the sake of simplicity, assume that the configurations have been numbered so that

$$c(i) \leqslant c(j) \qquad \forall\, i \leqslant j\ ,\ i,j = 1,\ldots,6\ .$$

We can now append to the arcs of the graph the probability that the transitions between two configurations occurs when a PHC algorithm is applied. For example, if Simulated Annealing is applied, there are certain transitions which are independent of T, i.e., the transitions corresponding to a decrease in cost, and others which are dependent on T, e.g., the "hill climbing" transitions. In the example shown in fig (2.2), dashed arcs represent T-dependent transitions, while solid arcs represent T-independent transitions.

In general, the application of a PHC algorithm to a combinatorial optimization problem can be represented by a graph whose nodes are configurations, whose arcs represent

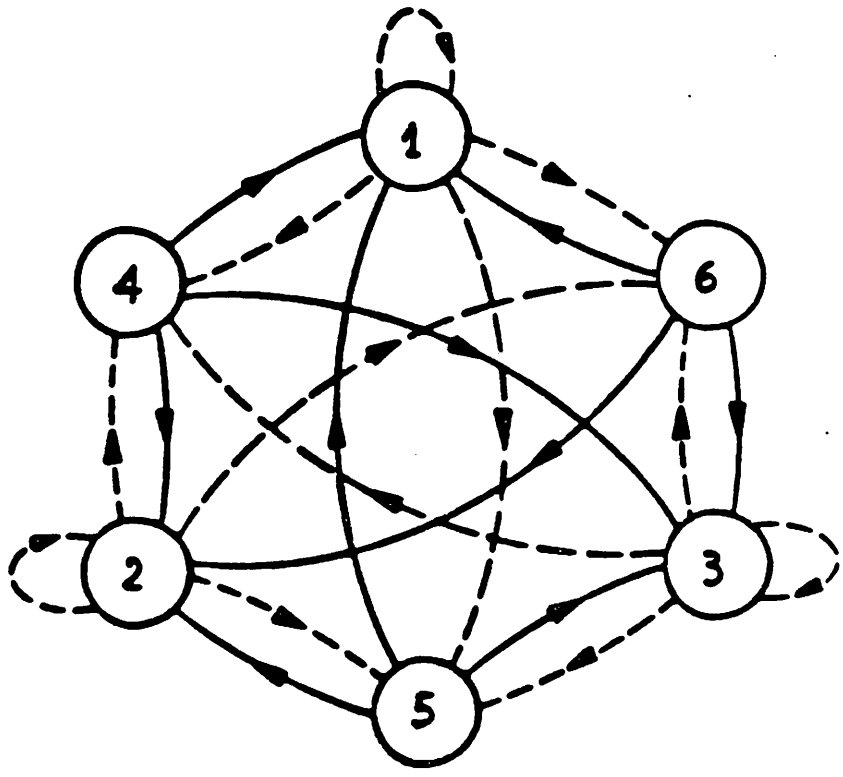

Fig. 2.2. Transition graph. The dashed arcs represent
T -dependent transitions.
The solid arcs represent T -independent transitions

configurations which can be obtained by the generation rule of the PHC algorithm and whose arc labels represent the probability that the corresponding transition is generated and accepted by the algorithm.

In general, the probability that the configuration selected by a PHC algorithm at the $(k+1)$—st iteration be j given that at the k —th iteration was i, is defined by

$$Prob\{ X_{k+1} = j \mid X_k = i \} \triangleq P_{ij}(T) \ , \tag{2.2}$$

where

$$P_{ij}(T) = G_{ij}(T) \bullet f(\Delta c_{ij}, T) \quad \forall\ j \in S(i) \ , \tag{2.3}$$

where $G_{ij}(T)$ is the probability of *generating* state j being in state i possibly dependent on the parameter T [1], $\Delta c_{ij} = c(j) - c(i)$, and X_k is the value taken on by the random variable X, representing the generic solution given by the algorithm, at the $k-th$ iteration. Since $G_{ij}(T)$ must be a probability, the following relation must hold

$$\sum_{j \in S(i)} G_{ij}(T) = 1 \tag{2.4}$$

Eq. (2.3) is the product of two different terms : $G_{ij}(T)$ is the probability that j is generated by the algorithm, the second term is the probability that the new configuration is accepted. An example of $G_{ij}(T)$ is given by

$$G_{ij}(T) = \begin{cases} 1/|S(i)| & \forall\, j \in S(i) \\ 0 & otherwise \end{cases}$$

where the probability of generating all the states that can be generated is uniform and independent of T. In our example, we assume that the generation probability is uniform and independent of T, and since $|S(i)|=3$, $\forall i \in S$,

$$G_{ij}(T) = \begin{cases} 1/3 & \forall\, j \in S(i) \\ 0 & \forall\, j \notin S(i) \end{cases}$$

We assume also that f is given by (2.1).

Note that since f is not in general identically equal to one, there is a finite probability that the algorithm will remain in configuration i. The following equation determines this probability:

$$P_{ii}(T) = 1 - \sum_{j \in S(i)} P_{ij}(T). \tag{2.5}$$

The stochastic process represented by the evolution of the random variable X produced by PHC algorithms is a *Markov process*. In fact, eqs. (2.3) imply that, given the value X_k, the value of X_{k+1} depend only on the value of X_k, i.e., the probability of any particular future behavior of the process, when its present state is known exactly, is not altered by

[1] Note that by definition $G_{ij}(T) = 0$ for the states that are not in $S(i)$.

additional knowledge concerning its past behavior. Since the configuration space of combinatorial optimization problems is a countable and in general finite set, the process is a *discrete time Markov chain with a finite state space.*

3. Basic Definitions and Results on Markov Chains.

In this section, a few basic definitions and theorems on Markov chain which are relevant to the discussion of PHC algorithms are reviewed. All the theorems are presented without proofs since they can be found in [KAR73, FEL70, FRE71].

Eq. (2.2) can be easily extended to describe an n *−step* transition of the Markov chain. The probability of going from state i to state j in n iterations is given by

$$Prob\{ X_{k+n} = j \mid X_k = i \} \triangleq P_{ij}^{(n)}(T) \quad .$$

Definition 3.1. State j is said to be *accessible* from state i if for some integer $n \geq 0$, $P_{ij}^{n} > 0$. Two states i and j, accessible to each other, are said to *communicate* .

■

The relation induced by this definition is an equivalence relation. The equivalence classes induced by this relation consist of all those states for which there exist a probability greater than zero to go from one state to the other in both directions in a finite number of steps.

Definition 3.2. A Markov chain is said to be *irreducible* if the equivalence relation induces a unique class.

■

Example 3.1. The Markov chain represented by the following probability transition matrix

$$P = \begin{vmatrix} 0 & P_2 \\ P_1 & 0 \end{vmatrix} ,$$

where the elements of P_i, $i = 1,2$ are all non zero, is irreducible. The Markov chain represented by

$$P = \begin{vmatrix} P_1 & 0 \\ 0 & P_2 \end{vmatrix} ,$$

where P_1 and P_2 are two matrices with elements not all zero, is not irreducible.

■

Definition 3.3. The period of the state x_i is said to be the greatest common divisor of all integers $n \geq 1$ such that

$$P_{ii}^{(n)} > 0 \quad .$$

■

Theorem 3.1 If the Markov chain is irreducible and there exists a state, i, such that

$$P_{ii} > 0 \quad .$$

then all states have period one.

■

Definition 3.4. A Markov chain in which each state has period one is said to be *aperiodic.*

■

Example 3.2. The Markov chain represented by the following probability transition matrix

$$P = \begin{bmatrix} 0 & P_2 \\ P_1 & 0 \end{bmatrix} ,$$

where P_1 and P_2 are two matrices which elements are not all zero, is periodic with period two.

■

It is easy to show that periodicity is a *class property* i.e., all the states in an equivalence class have the same period.

Definition 3.5 Let $h_{ii}^{(n)}$ be the probability that starting from state i, the first return to state i occurs at the n-th transition, i.e.,

$$h_{ii}^{(n)} = Prob\{X_n = i, X_j \neq i, j = 1,2,\ldots,n-1, \mid X_0 = i\}.$$

Then a state i is recurrent if $\sum_{n=1}^{\infty} h_{ii}^{(n)} = 1$.

■

This definition says that a state i is recurrent if, starting from state i, the probability of returning to state i after some finite length of time is one.

Example 3.3. The following probability transition matrix

$$P_{ij} = \begin{cases} p & \text{if } i = j+1 \\ q & \text{if } i = j-1 \\ 0 & \text{otherwise} \end{cases} \qquad (3.1)$$

represents the Markov process known as one-dimensional random walk on the positive and negative integers where, at each transition, a particle moves with probability q one unit to the right and with probability p one unit to the left with ($p + q = 1$). If the process starts from the origin, if $p \neq q$, there is a non zero probability that a particle initially at origin will drift to $+\infty$ if $q > p$ ($-\infty$ in the other case) without ever coming back to the origin. Hence the origin is not recurrent.

The following condition

$$p = q = \frac{1}{2} \ . \qquad (3.2)$$

is necessary to ensure the recurrence of the Markov chain determined by eq. (3.1). In [KAR73] a formal proof of condition (3.2) is given.

■

Recurrence as periodicity is a class property, i.e., all the states in an equivalence class are either recurrent or non recurrent.

The properties that have been introduced above define a large class of Markov chains for which an ergodic theory has been developed. The two main results of this theory are recalled below.

Theorem 3.2. Let i be the initial state and let $P_{ii}^{(0)} \triangleq 1$. If a Markov chain is irreducible, aperiodic and recurrent, then

a) the following limit exists

$$\lim_{n \to \infty} P_{ii}^{(n)} = \frac{1}{\sum_{n=0}^{\infty} n \; h_{ii}^{(n)}} \quad ,$$

where $h_{ii}^{(n)}$ can be defined recursively by

$$P_{ii}^{(n)} - \sum_{k=0}^{n} h_{ii}^{(n-k)} P_{ii}^{(k)} = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n > 0 \end{cases} .$$

b) For each j,

$$\lim_{n \to \infty} P_{ji}^{(n)} = \lim_{n \to \infty} P_{ii}^{(n)} = \pi_i \, . \tag{3.3}$$

■

An intuitive explanation of Theorem 3.2 is as follows. If k is large enough, P_{ji}^{k}, the probability of being at the k-th iteration in state i, starting from state j, depends only on the state itself and is totally independent on the initial state j. Note that π_i defined in Theorem 3.2 is always larger than or equal to zero. If it is larger than zero, then the following important result holds.

Theorem 3.3. If π_i of eq. (3.3) is greater than zero $\forall \; i$ and the Markov chain is recurrent, irreducible and aperiodic then

$$\lim_{n \to \infty} P_{ii}^{(n)} = \pi_i = \sum_{j=1}^{|S|} \pi_j P_{ji} \quad ,$$

and the π_i's are uniquely determined by the following set of equations

$$\sum_{i=1}^{|S|} \pi_i = 1 \quad , \tag{3.4.a}$$

$$\sum_{i=1}^{|S|} \pi_i \, P_{ij} = \pi_j \quad , \tag{3.4.b}$$

$$\pi_i \geq 0 \qquad \forall \, i \quad . \tag{3.4.c}$$

■

The set $\{\pi_i, i = 1,\ldots,|S|\}$ determined by Theorem 3.2 is called the *stationary probability distribution* of the Markov chain.

The stationary probability distribution is very important since it completely characterizes the asymptotic behavior of a Markov chain.

4. Asymptotic Properties of Probabilistic Hill Climbing Algorithms

Results quoted in Section 3 cannot be applied directly to the Markov chains representing the stochastic processes induced by PHC algorithms. In fact, these results are in general valid for *stationary* transition probabilities, i.e., for transition probabilities that are independent of time. Note that the transition probabilities defined in Section 2 depend on the parameter T which is updated during the evolution of the algorithms and hence are dependent on time. However, when the parameter is kept constant, i.e., in the inner loop of PHC algorithms, the transition probabilities are constant and the associated Markov chain stationary.

Our strategy to prove the properties of PHC algorithms is to determine first which conditions PHC algorithms must satisfy so that a stationary probability distribution exists at a given temperature. Then we will introduce a function $\pi_i(T)$, defined on all the set of configurations, with the property that, when T approaches zero, $\pi_i(T) \neq 0$ only for those configurations that are global optima for the combinatorial optimization problem. Finally conditions on the acceptance function f will be determined such that $\pi_i(T)$ is the stationary probability distribution of the Markov chain describing the PHC Algorithm. If we can prove that, given a function $\pi_i(T)$ with the above mentioned properties and a suitable generation function $G_{ij}(T)$, it is possible to determine an acceptance function f with the features outlined in section 2, then a PHC algorithm is obtained which generates with probability one, asymptotically, a global optimal solution for the combinatorial optimization problem.

The first assumption on the PHC algorithms is related to the generation rule and the acceptance rule. The rules must be such that for all T different from zero, the Markov chain induced by the algorithm is irreducible. This means that for each pair of configurations, say i, j there must be integers $m, n \geq 0$, so that $P_{ij}^{(n)} \neq 0$ and $P_{ji}^{(m)} \neq 0$. In other words, the graph obtained by representing the generation rule with directed arcs as in Section 2, must be strongly connected. In addition, we must be sure that the acceptance rule does not eliminate arcs of the graph by assigning them probability zero which cause the labeled graph obtained by removing the arcs with zero weight not to be strongly connected. Note that the acceptance

rule specified by simulated annealing assigns a non zero probability to all the arcs of the graph corresponding to the generation rule and hence, for this PHC algorithm we only need to verify that the generation rule produces a strongly connected graph in the configuration space.

The next condition is related to the aperiodicity of the Markov chain.

Proposition 4.1. Let the Markov chain corresponding to a PHC algorithm be irreducible for all $T \geq 0$. If the acceptance function f of the PHC algorithm is such that there exists at least a pair of states i and j for which

$$0 < f(\Delta c_{ij}, T) < 1, \qquad \forall T > 0, \tag{4.1}$$

then the Markov chain is aperiodic for all $T > 0$.

Proof. If (4.1) holds, then according to (2.3), $P_{ii}(T) > 0$, $\forall T > 0$ and the proof follows from Theorem 3.1 and the irreducibility of the Markov chain.

■

The condition of Proposition 4.1 is always satisfied by Simulated Annealing, since there is at least one state for which (4.1) holds: the global optimum.

The acceptance functions which satisfy the condition of Proposition 4.1 and which take values in the closed set [0,1] are said to be *admissible*.

Note that the Markov chain associated to PHC algorithms is obviously recurrent since it is finite.

According to Theorem 3.3, the Markov chain associates with a PHC algorithm which satisfies the conditions of Proposition 4.1, has a stationary probability distribution.

Now we look for a form of the stationary probability distribution which, when T goes to zero, is different from zero only in global minima of c.

To this end, we have to find under which conditions a stationary probability distribution $\pi_i(T)$ is different from zero, as T goes to zero, only if the i-th configuration is the global optimal solution.

Theorem 4.1. Let $\pi_i(T)$, $\forall i \in S$ be defined by

$$\pi_i(T) = \Lambda(T)\, g(c(i),T) \tag{4.2.a}$$

where $\Lambda(T)$ is a normalizing factor such that

$$\sum_{i \in S} \pi_i(T) = 1 \tag{4.2.b}$$

and g is such that

$$g(c(i),T) > 0 \qquad \forall\, T > 0, \forall\, c. \tag{4.3.a}$$

$$\lim_{T \downarrow 0} g(c,T) = \begin{cases} 0 & \text{if } c \geq 0 \\ \infty & \text{if } c < 0 \end{cases} \tag{4.3.b}$$

$$\frac{g(c_1,T)}{g(c_2,T)} = g(c_1 - c_2, T) \tag{4.3.c}$$

$$g(0,T) = 1, \quad \forall T > 0, \tag{4.3.d}$$

then

$$\lim_{T \downarrow 0} \pi_i(T) = \begin{cases} 1/|M| & \text{if } i \in M \\ 0 & \text{if } i \in S - M \end{cases} \tag{4.4.a}$$

where the set M is defined by

$$M = \{ i \mid c(i) \leq c(j),\ \forall\ j \in S \}\ . \tag{4.4.b}$$

■

Proof. The proof of (4.4) is straightforward because of the properties of the function g. In fact $\forall i \in M$, by (4.3.c-d), eqs. (4.2) can be rewritten as

$$\pi_i(T) = \frac{1}{|M| + \sum_{j \in (S-M)} g(c(j) - c(i), T)}$$

but since

$$c(j) - c(i) > 0, \quad \forall\, i \in M\ ,\ \forall\ j \in (S - M)$$

then by (4.3.b),

$$\lim_{T \downarrow 0} \pi_i(T) = \frac{1}{|M|} \qquad \forall\, i \in M \quad .$$

If the same reasoning is applied for an $i \in (S - M)$ then at least one element of the sum goes to ∞ as T goes to 0. This completes the proof of the Theorem.

■

Now, we have to specify under which conditions on f and G a $\pi_i(T)$ of the form described above is indeed the stationary probability function of the Markov chain associated to the PHC algorithm.

Theorem 4.2. Let $S'(j)$ be defined by

$$S'(j) = \{i : j \in S(i)\},$$

if f is admissible and

$$\sum_{i \in S'(j)} g(c(i),T)\, G_{ij}(T)\, f(\Delta c_{ij},T) = \tag{4.5}$$

$$= g(c(j),T) \sum_{i \in S(j)} G_{ji}(T)\, f(\Delta c_{ji},T)$$

then $\pi_i(T)$ defined by eqs. (4.2) is the stationary probability distribution of the Markov chain whose one-step transition probability is given by eqs. (2.3-2.5).

Proof. The proof of Theorem is carried out by verifying that g, G and f satisfy the conditions of Theorem 3.3. In view of the assumptions made on function G and of Propositions 4.1 and 4.2, the Markov chain defined by eq. (2.3, 2.5) is irreducible, aperiodic and positive recurrent. It is now immediate to see that $\pi_i(T)$ defined by eq. (4.2.a) satisfies eq. (3.6.a) because of (4.2.b) and (3.4.c) because of (4.3.a). Finally it takes just a little thinking to see that (3.4.b) is satisfied automatically once f is chosen as specified by (4.5).

■

Theorem 4.2 is important since it suggest a way to construct a PHC algorithm with guaranteed convergence properties. In fact one first selects a function $\pi_i(T)$ that ensures the convergence to the global optima (Theorem 4.1) and then selects an acceptance function f and a generation rule G such that Propositions 4.1, 4.2 and Theorem 4.2 are satisfied.

Up to now we placed no assumptions on $G_{ij}(T)$ and obtained a general result. We now assume that the rule to generate new states is such that the existence of $G_{ij}(T)$ implies the existence of $G_{ji}(T)$. Under this assumptions we can prove the following

Corollary 4.1. If the function $G_{ij}(T)$ is such that

$$G_{ij}(T)\,G_{ji}(T) \neq 0 \tag{4.6}$$

then an admissible function f defined by

$$\frac{f(\Delta c_{ij},T)}{f(\Delta c_{ji},T)} = \frac{G_{ji}(T)}{G_{ij}(T)} g(c(j)-c(i)),$$

satisfies eq. (4.6).

■

The proof of the Corollary follows directly from eqs. (4.2) (4.5) and (2.5).

Remark 4.1. Simulated Annealing as proposed by Kirkpatrick [KIR83] has g, G and f defined as follows

$$g(c(i),T) = e^{-\frac{c(i)}{T}} \tag{4.7}$$

$$G_{ij}(T) = \begin{cases} \dfrac{1}{|S(i)|} & \forall\, j \in S(i) \\ 0 & \forall\, j \notin S(i) \end{cases}$$

$$f(\Delta c_{ij},T) = \min[1, e^{-\frac{c(j)-c(i)}{T}}] \quad \forall\, j \in S\ .$$

These functions satisfy the conditions of Corollary 4.1 and then, *a fortiori*, of Theorem 4.2. Hence the configurations generated by Simulated Annealing asymptotically converges to the global optimum.

Another PHC algorithm can be generated just by replacing the acceptance function of Simulated Annealing with the following one

$$f(\Delta c_{ij},T) = \frac{e^{-\frac{c(j)-c(i)}{T}}}{1+e^{-\frac{c(j)-c(i)}{T}}} \tag{4.8}$$

and leaving functions G and g the same. Obviously the two algorithms exhibit an identical

asymptotic behavior.

A slightly different PHC algorithm related to Simulated Annealing has been used in a package for standard cell placement [SEC84]. In this algorithm, the acceptance function is the same as Simulated Annealing and $G_{ij}(T)$ satisfy the following relations:

$$G_{ij}(T) = G_{ji}(T)$$

but are not uniform. This algorithm is described by a Markov chain with the same stationary probability distribution as Simulated Annealing.

Note that when the control parameter T, approaches zero, the acceptance functions f given by (2.1) and (4.8) become a unitary step function that assigns probability one only to those transitions which improve the cost function and probability zero to the others. Hence, when T is set to zero, they degenerate into the usual greedy strategy and select, among all the new configuration that are generated, the ones with lower cost than the present configuration only.

Unfortunately Theorems 4.1 requires the algorithm to perform an infinite number of iterations every time the parameter T is updated. It is clear that a strategy of this kind is practically inapplicable. In fact, a PHC algorithm performing an infinite number of iterations for each value of the parameter T is a conceptual, non implementable algorithm [POL71], in the sense that an internal loop is never exited.

If we assume that the function g is continuous in its second argument, then also $\pi_i(T)$ is a continuos function.

The continuity of $\pi_i(T)$ implies that the stationary probability distribution for a particular value of the controlling parameter, say $\hat{T}$, is a good approximation for the stationary probability distribution for all the values of T sufficiently close to $\hat{T}$.

This result suggests a strategy for the control of T: start with a value of T for which the stationary probability distribution is easy to estimate, and update T so that only a few iterations are needed to obtain a good approximation to the new stationary probability distribution.

Remark 4.2. Simulated annealing as proposed in [KIR83], follows the strategy outlined above. In fact, starting with a "high" temperature guarantees an easy estimation of $\pi_i(T)$. The acceptance function (2.2) implies that for T sufficiently large, the probability of accepting a move is close to one. Hence, if the generation probability is uniform and the associated Markov chain irreducible, all the states are equally likely and the stationary probability distribution trivial to estimate.

In Simulated Annealing, the temperature is slowly decreased. Since in this case, the function g is given by (4.7) which is obviously continuous in T, we can interpret the control strategy as a direct application of Proposition 4.3.

In addition, the updating rule of Simulated Annealing is

$$T_{m+1} = \alpha(T_m)T_m$$

where α can be a constant or a function of T but always less than one and larger than zero. Values of α that yielded good results are in general ~ .9-.99, which forces the updating to become slower and slower as the algorithm approaches $T = 0$.

■

All the results presented in this section are asymptotic and hence they can only help deciding the control strategy and how long the inner loop of PHC algorithms should be run. However, no sharp bound is given on the actual choice of the number of steps to be taken in the inner loop.

5. Control Strategies and Experimental Results

In the previous section we have presented theoretical results which can be used to explain the success of PHC algorithms and to derive qualitative reasoning on the strategy for controlling the parameter T. In this section, we present results which can be used to estimate how many steps should be attempted for each value of T.

PHC algorithms can be used to compute the actual stationary probability distribution in the following way. Store how many times each of the states has been visited by the algorithm during its evolution. The ratio between this number and the total number of iterations gives an estimate on the $\pi_i(T)$'s. Actually, when the number of iterations goes to infinity, the result of the calculation converges to the $\pi_i(T)$'s.

The ideal approach to the determination of the number of iterations to take, would be to detect when the approximation is within a specified distance from the stationary probability distribution. Unfortunately, we have not been able to obtain such a result yet. Thus we have to resort to another technique. This technique has been obtained by observing that, in order to obtain a good final result, the PHC algorithms have to be able to leave local minima where they could end up during the computation. Thanks to the properties of Markov chains, it is indeed possible to estimate the number of iterations needed to get out of a local minima at a given value of T with probability $1-\epsilon$, $\epsilon > 0$.

Proposition 5.1. Given a state , e.g., i, such that $P_{ii}(T) \leqslant 1$, then a PHC algorithm has probability $1 - \epsilon$ to leave i if at least

$$\hat{N}_i = \frac{\ln \epsilon}{\ln P_{ii}(T)} \tag{5.1}$$

iterations are performed by the algorithm.

Proof. For the sake of notational simplicity, in this proof and in the proof of Proposition 5.2. the dependence of P on T will be implicit. Let

$$Q_i^N = \sum_{n=1}^{N} P_{ii}^{n-1}(1 - P_{ii}) \tag{5.2}$$

be the probability of leaving the i after at most N iterations. Taking the sum of the series, (5.2) becomes

$$Q_i^N = \frac{(1 - P_{ii})(1 - P_{ii}^N)}{1 - P_{ii}} .$$

Now if a number of iterations $\hat{N}_i$, given by (5.1) are performed, then

$$Q_i^{\hat{N}_i} = 1 - P_{ii}^{\hat{N}_i} =$$

$$= 1 - P_{ii}^{\frac{\ln \epsilon}{\ln P_{ii}}} =$$

$$= 1 - P_{ii}^{\frac{\log_{P_{ii}} \epsilon}{\log_{P_{ii}} P_{ii}}} = 1 - \epsilon.$$

■

The evaluation of $P_{ii}(T)$, $\forall\ i \in S$ requires to know the values taken by the cost function on the configuration space, which is obviously out of the question. Assuming that the acceptance function is monotonic decreasing in the first argument as is the case of the acceptance functions given in (2.1) and (4.8), there are a number of possible techniques to estimate $P_{ii}(T)$. For the result stated by Proposition 5.1 to hold, we need a conservative estimate of $P_{ii}(T)$. Unfortunately, the techniques we have been able to discover cannot be guaranteed to obtain an upper bound on $P_{ii}(T)$. The most conservative bound is obtained by assuming that $\Delta c_{ij} \quad \forall\ j \in S(i), \forall\ i \in S, i \neq j$ is constant and equal to $\Delta c_{\hat{i}\hat{j}}$ where $\hat{j}$ is the worst configuration and $\hat{i}$ is the best configuration found so far. To use this estimate, we have to insert a step in the PHC algorithm to record the best and worst configuration.

Proposition 5.1 is a "worst case" result; a similar approach is useful to determine the *expected value* of the number of iterations necessary to leave i.

Proposition 5.2. Let i be a state such that $P_{ii}(T) < 1$ and let R_i^N be the probability that a PHC algorithm leave i after N iterations. The expected value of the number of iterations required to leave i, $\tilde{N}$, is given by

$$\tilde{N}_i = \frac{1}{1 - P_{ii}(T)}$$

Proof. By the definition of expected value

$$\tilde{N}_i = \sum_{n=0}^{\infty} n \; R_i^n \tag{5.3}$$

Substituting in (5.3) the expression for R_i^N given by

$$R_i^N = P_{ii}^{N-1}(1 - P_{ii})$$

we obtain

$$\tilde{N}_i = \sum_{n=0}^{\infty} n \; P_{ii}^{\,n-1}(1 - P_{ii}) =$$

$$= (1 - P_{ii}) \sum_{n=0}^{\infty} n \; P_{ii}^{\,n-1} =$$

$$= (1 - P_{ii}) \frac{d}{dP_{ii}} \sum_{n=0}^{\infty} P_{ii}^n =$$

$$= (1 - P_{ii}) \frac{d}{dP_{ii}} \frac{1}{1 - P_{ii}} =$$

$$= \frac{1}{1 - P_{ii}}$$

■

As in the previous case, P_{ii} must be estimated. The approximations introduced above are also needed to estimate $\tilde{N}_i$.

Note that a Markov chain represents a stochastic dynamical system. The time (or number of iterations, since the configuration space is finite and discrete) necessary to leave a

particular state plays a role which is similar to the time constant in a linear dynamical system. If a linear dynamical system is controlled by a piece-wise constant function, a time as long as a few time constants will bring the system to a new steady state condition. A similar reasoning can be applied here if we assume that a number of iterations which is between 3 and 5 times $\tilde{N}_i$ is needed to reach a stationary probability distribution.

$\hat{N}$ and $\tilde{N}$ defined as the the largest values of of $\hat{N}_i$ and $\tilde{N}_i$ respectively determine two different estimates of the number of iterations necessary for the algorithm to obtain a good estimate of the stationary probability distribution.

It is important to note that both $\hat{N}$ and $\tilde{N}$ increase as T approaches zero. Once more we find an agreement with the qualitative result introduced in Section 4: more time has to be spent at lower values of T.

Some of the ideas on the control strategy have been tested on a particular problem whose optimal configurations were known *a priori*. The test case consists of a two-dimensional placement problem where 24 modules have to be placed on a 5-by-5 grid. A total of 25! configurations are possible. Due to the particular structure of the example, we know that 16 configurations are optimal. The value of the cost function at the optimal configurations is 38.

Five different sets of experiments have been carried out changing the updating factor for the parameter T and the function f. For all the experiments the same function G_{ij} has been selected, i.e.

$$G_{ij} = \frac{2}{n\ (n-1)}$$

with $n = 25$, the number of available positions on the grid.

The parameter T is updated according to

$$T_{m+1} = \alpha(T_m)\, T_m$$

starting from an initial condition of $T_0 = 40$

The function f and the value of α selected for each experiment are shown in Table 5.1, where f_1 and f_2 are given by (2.1.a) and (4.8) respectively. In experiment 5, α has been changed dynamically with the range indicated and hence is dependent on T.

In each experiment a fixed number of iterations was used for each value of T. In particular, for high values of T, 15 iterations were performed while as T decreases, the number of iterations was increased up to 30.

Each experiment was repeated 40 times. A few statistical data about the quality of the results obtained are shown in Table 5.2.

The analysis of the results shown in Table 5.2 demonstrates that the updating factor has a great influence on the quality of the final solution obtained by the algorithm. This is the experimental verification of the qualitative remarks introduced in Section 4 and in this section: the slower is the updating of the parameter T, the better is the solution determined by the algorithm. In fact, since in both cases the number of iterations performed for each value of T is the same, when the updating factor is closer to one, better convergence to the stationary probability distribution is obtained.

Exp. #	f	updating factor
Exp. #1	f_1	.95
Exp. #2	f_2	.95
Exp. #3	f_1	.99
Exp. #4	f_2	.99
Exp. #5	f_2	.90-.95

Table 5.1 Experiment summary

Exp. #	# of trials	av. value	σ	worst sol.
Exp. #1	40	39.60	1.26	44
Exp. #2	40	40.63	1.69	45
Exp. #3	40	38.75	.90	41
Exp. #4	40	38.85	.82	41
Exp. #5	40	40.63	1.71	47

Table 5.2. Experimental results.

The use of different acceptance functions does not alter significantly the quality of the solution obtained even if the results of Table 5.2 show that an acceptance function of the form (2.1) gives results that are always slightly better then those obtained when (4.8) is used.

Finally, an adaptive control strategy for T was attempted but the results obtained were not competitive with the ones in which α was constant. However, we do not suggest that this strategy should be rejected on the basis of this experiment. The rule used to switch the values of α was very simple-minded: α was increased when the cost function had large variations from iteration to iteration, it was decreased again when the cost function showed little improvement from iteration to iteration. These experiments were designed before we had many of the results presented in the previous sections of the paper. A new set of experiments is being designed to illustrate as many aspects of the theory as possible.

6. Conclusions

A theory of a class of algorithms for the solution of combinatorial optimization problems inspired by the technique known as Simulated Annealing has been developed. The class of algorithms has the characteristic of being probabilistic and of being able to climb "hills", i.e. to accept intermediate solutions which increase the cost. For this reason, these algorithms have been called Probabilistic Hill Climbing (PHC) algorithms. The mathematical model used in the study of their properties is a Markov chain with finite state space.

Based on the key results on stationary probability distributions of Markov chains, we have derived conditions on the parameters of PHC algorithms to guarantee that an optimum configuration is found with probability one. The theory requires that an infinite number of iterations be performed as intermediate steps.

Some guidelines in the selection of the number of iterations used in the intermediate steps have been given. Finally some experimental results have been analyzed in view of the theoretical results developed.

Geman and Geman [GEM84] have shown that the number of iterations can be limited in the intermediate steps, provided that the annealing schedule varies logarithmically with the number of iterations. This schedule guarantees that the algorithm reaches the optimum with probability one, but it is also asymptotic since when approaching zero the temperature has to be lowered infinitely slow.

Much work remains to be done to exploit fully the mathematical model. We are exploring new control strategies and techniques which we hope will give tight bounds on the number of iterations needed to maintain a given level of confidence in the optimality of the results. In particular, we are looking at the theory of nonhomogeneous Markov chains to be able to find stronger results than the ones so far obtained. In addition, a set of placement and routing packages is being developed by C. Sechen which incorporates the control strategies suggested by the theory.

7. Acknowledgements

The authors wish to thank Carl Sechen for the collaboration in running the experimental results and for the many lively arguments on the practical aspects of Simulated Annealing. Early discussions on Simulated Annealing and its analysis with Professor Karp are gratefully acknowledged. Professors Sastry and Walrand have been very helpful discussing with the authors various theoretical aspects of stochastic processes.

This research has been sponsored in part by SRC-82-11-008, the Italian National Research Council and Honeywell Information Systems Italy.

References

[BIN78] K. Binder, *Monte Carlo Methods in Statistical Physic*, Springer-Verlag, 1978

[DOO51] J.L. Doob, *Stochastic Processes*, J.Wiley, 1951

[FEL70] W. Feller, *An Introduction to Probability Theory and Applications*, J.Wiley, 3rd Edition, 1970

[FRE71] D. Freedman, *Markov Chains*, Holden-Day, 1971

[GAR79] M.R. Garey, D.S. Johnson,*Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, 1979

[GEM84] S. Geman, D. Geman, Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, unpublished manuscript

[JON84] D.S. Johnson, Simulated Annealing Performance Studies, presented at Simulated Annealing Workshop, Yorktown Heights, April 1984.

[KAR73] S. Karlin, *A First Course in Stochastic Processes*, Academic Press, 1973

[KIR83] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by Simulated Annealing, *Science*, Vol. 220, N. 4598, pp. 671-680, 13 May 1983

[LUN84] M. Lundy, A. Mees, Convergence of the Annealing Algorithm, presented at Simulated Annealing Workshop, Yorktown Heights, April 1984

[MET53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, *J. Chem. Phys.*, Vol. 21, p. 1087, 1953

[POL71] E. Polak, *Computational Methods in Optimization a Unified Approach*, Academic Press, 1971

[SEC84] C. Sechen, A. Sangiovanni Vincentelli, The TimberWolf Placement and Routing Package, *Proc. 1984 Custom Integrated Circuit Conference*, Rochester, May 1984.

[SCH80] J.T. Schwartz, Fast Probabilistic Algorithms for Verification of Polynomial Identities, *Journal of ACM*, Vol 27, No. 4, Oct 1980

[VEC83] M.P. Vecchi, S. Kirkpatrick, Global Wiring by Simulated Annealing, *IEEE Transactions on Computer-Aided Design*, Vol. CAD-2, No. 4, Oct 1983