SEQUENCING DNA FRAGMENTS BY USING A SIMULATED ANNEALING ALGORITHM

Abstract

We use a simulated annealing algorithm to solve an ordering problem which consists in sequencing fragment lenghts to recover the structure of DNA. This is known as the multiple digest problem in molecular biology. The problem of molteplicity of solutions and the problem of measurement error are considered.
The algorithm allows to solve large mapping problems. For istance,a solution to a double digest problem of size 7.9×10^{525} is found running the algorithm on a VAX computer spending a little more than 4 days of CPU computer time.

1.Introduction

In this paper we consider a mathematical approach to some combinatorial ordering problems which naturally arise in several different contexts. Particularly,we treat an ordering problem which finds an important application in molecular biology.

Indeed,we consider in the following the so called multiple digest mapping problem.
Let us consider a linear segment of DNA ;this segment is cut at all occurrences of a specific short pattern by restriction enzymes.DNA sequences can be viewed as finite sequences over the four aminoacid bases A (= amine) ,C (= cytosine) ,G (= guanine) , T (= timine) .
Each restriction enzyme cuts the double stranded DNA at a short pattern specific to that enzyme ; for istance,the restriction enzyme EcoRI cuts at GAATTC .
Some different enzymes can be used singly or in combination and by means of gel electrophoresis techniques the resulting fragment lenghts are observed,and then recorded in an arbitrary order (thus,the order in which the fragments are disposed onto the considered segment of DNA is unknown).
The multiple digest problem consists in finding the right permutation of elements of the set of restriction fragments to show the locations of cleavage sites. Such a problem is referred to as constructing a restriction map of DNA.

The multiple digest problem is nothing but a combinatorial problem; obviously it finds application not only in molecular biology, but also in a lot of different contexts. For istance,we can consider the following problem. Let $Q_1, ...Q_n$ be n distinct telephonic

devices and let the times of arrivals of a call to each device be distibuted according e.g. to a Poisson process.We observe the above devices over a time interval $[0, T]$;let $T_1^i, T_2^i, ...T_r^i$ be the times of arrivals of a call to the device Q_i,$i = 1, ...n$, and let $\delta_j^i = T_{j+1}^i - T_j^i, j = 1, ...r-1$ be the lenghts of time intervals between two successive arrivals to the device Q_i ,where $\sum_{j=1}^{r-1} \delta_j^i = T$.
We can consider one or more devices simultaneously ;for fixing ideas suppose to consider two devices (for istance,two telephone sets in a room,with independent lines).
Now,let us observe the calls arriving to the device Q_1 and record the lenghts of time intervals between two successive calls to Q_1 ;do the same thing for the device Q_2 . Then,let us observe the calls arriving to the cluster (Q_1, Q_2) and record the lenghts of time intervals between successive calls to one or the other of devices Q_1, Q_2 ,without distinction.
The problem is to reconstruct exatly the times of arrivals of calls to the devices Q_1 and Q_2,respectively. This is nothing but a "double digest problem".

Several authors ([3],[4],[10],[13],[14],[15]) considered computer algorithms to resolve the above problem in the context of molecular biology. In [11] Pearson proposed solving the double digest problem in exhaustive way by considering all permutations of two single digests. Naturally,this method is succesfull (with respect to computation time) only when the number of sites where enzymes cut is small enough.

In this paper ,we are concerned with a simulated annealing algorithm to solve the multiple digest problem and we present computer results showing its efficacy. The problem of nonuniqueness of solutions is considered,and also the problem of measurement errors in the lenght of observed fragments is discussed.
An analogous approach can be found in [6],where the case of the usual small mapping problems of molecular biology is emphasized.Here,we present in details a simulated annealing algorithm adapted to the actual problem ;this algorithm allows us to solve also large mapping problems.
For istance,a solution to a double digest problem of size 7.9×10^{525} was found running our simulated annealing algorithm on a VAX computer spending about 4 days of CPU computer time.

Acknowledgement

I wish to thank Dr. G. Pepe for having furnished me with the exact data from the human β-globine gene.

2. A solution to the multiple digest problem via simulated annealing.

2.1 The SA algorithm.

We review briefly the theory of the simulated annealing algorithm .

The simulated annealing algorithm (SAA) is a successfull tool in many optimization problem in which one wishes to find a global minimum of a function f defined on a finite space Ω(the state space or the space of configurations).

The SAA is a strategy obtained by means of a stochastic relaxation method;the statistical mechanics interpretation is that of a physical system on which one consider an energy function f and a Gibbs distribution at temperature T . The Gibbs distribution gives the probability of finding the system in a particular configuration at same given temperature .

The algorithm consists of letting the system evolve according to the Gibbs distribution and cooling the system slowly enough,so that the limit (as $T \rightarrow 0$) distribution is the uniform distribution over the states of minimum energy. So,if the system is not cooled too rapidly,one can construct a suitable Markov chain $\underline{X}_n$ whose transition probabilities satisfy the balance equation and such that ,as T is decreased to 0 and $n \rightarrow \infty, X_n$ converges towards a state of minimum energy (see e.g. [5]) .

The invariant measure of the Markov process is the uniform probability measure on global minima ([1]) and the convergence of the algorithm is guaranteeed by the property of Markov process.

Let the state space Ω be a finite set whose elements are the configurations σ,and let $f : \Omega \longrightarrow R$ be a real valued function on the state space.
The SAA concerns the problem of finding an element $\sigma_{opt} \in \Omega$ such that

$$\min_{\sigma \in \Omega} f(\sigma) = f(\sigma_{opt}).$$

Generally,such a configuration σ_{opt} is not unique.
The set of optimal configurations will be denoted $\Omega_{opt} = \{\sigma_{opt} \in \Omega\}$.
For any $T > 0$,let Π_T be the Gibbs distribution over Ω given by:

$$\Pi_T(\sigma) = exp(-f(\sigma)/T)[\sum_{\sigma \in \Omega} \Pi_T(\sigma)]^{-1} \tag{2.1}$$

One can observe that for large values of T the distribution tends to be uniform over Ω,while for small values of temperature T the favorable elements of Ω,that is those elements σ of Ω for which $f(\sigma)$ is small,are weighted with large probability.Thus a probabilistic solution to the above optimization problem is given by sampling from the distribution Π_T for small value of $T > 0$.

This can be achieved by simulating a Markov chain $\{X_n\}_{n \geq 0}$ with state space Ω that has Π_T as its stationary distribution, and letting it to approach equilibrium.

First,it is necessary to determine for each $\sigma \in \Omega$ a set of neighbors $\Omega_\sigma \subset \Omega$ where transitions from σ are allowed in such a way that the resulting Markov chain is irreducible. It is supposed that $\sigma \in \Omega'_\sigma \Longleftrightarrow \sigma' \in \Omega_\sigma$ and for every $\sigma \in \Omega \mid \Omega_\sigma \mid= \Theta = constant$.

Now we define the transition probabilities

$$p_T(\sigma, \sigma') = P_T(X_{n+1} = \sigma' \mid X_n = \sigma\}$$

$$p_T(\sigma, \sigma') = \begin{cases} 0 & \text{if } \sigma' \notin \Omega_\sigma \\ exp\{-(f(\sigma') - f(\sigma))^+/T\}/\Theta & \text{if } \sigma' \in \Omega_\sigma \end{cases} \tag{2.2}$$

with the condition that $\sum_{\sigma' \in \Omega_\sigma} p_T(\sigma, \sigma') = 1$.
It is easy to see that Π_T satisfies the balance equation : (see [1])

$$p_T(\sigma, \sigma')\Pi_T(\sigma) = p_T(\sigma', \sigma)\Pi_T(\sigma') \tag{2.3}$$

which is enough to guaarantee that Π_T is the unique stationary distribution of the chain X_n .
In practice,the Markov chain is simulated in the following way : when the system stays at σ a neighbor σ' of σ is selected from Ω_σ uniformly and $f(\sigma')$ is computed. Then,the transition from σ to σ' is accepted with probability $p = exp\{-(f(\sigma') - f(\sigma))^+/T\}$ and the new state of the chain is σ',else the transition is rejected and the system remains in the state σ .

If the system is cooled slowly enough ,that is if the temperature $T = T_n$ is lowered at a rate $\sim const./log(n)$, $n \rightarrow \infty$, Geman and Geman ([5]) showed that,if at stage n in the algorithm one uses the transition probabilities given by (2.2) with $T = T_n$, the limit distribution $\Pi_0 = lim_{T \rightarrow 0^+} \Pi_T$ is the invariant distribution of the chain X_n and it is the uniform measure over the states of minimum energy f.

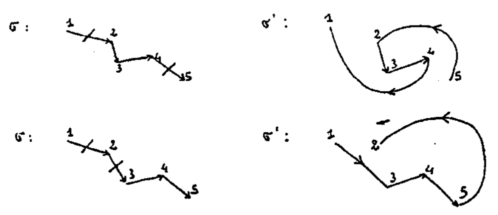
The algorithm is a general tool to resolve combinatorial optimization problems; we note,however,that the choices of the neighborhood structure on Ω and of the energy function f are of peculiar importance to implement the algorithm succesfully.

For a given finite discrete set Ω of configurations, a possible good choice of neighborhood structure is as in the travelling salesman problem. It consists in finding an optimal tour (that is of minimal lenght) to be taken by a salesman who has to visit each of n cities, we say $a_1, a_2, ...a_n$,and then return home.
Here Ω is the set of all permutations of the indexes $\{1, 2, ...n\}$; each permutation $\sigma \in \Omega$ is in one-to-one corrispondence with a tour taken in the order given by σ . The energy function is taken to be the total lenght of the tour.

Now,we give a description of the so called **2-change neighbor structure** that many authors (see e.g.[2,9]) have considered specifically to the travelling salesman problem. The last problem has a lot of affinities with the multiple digest problem.

A configuration (a tour) $\sigma = (a_1, a_2, ...a_n)$consists of an oriented connected graph where for each node a_i ,$i = 1, ...n$ there exist exatly one preceding edge a_{i-1} and one successive edge a_{i+1} and oriented paths (or links) which connect a_{i-1} to a_i and a_i to a_{i+1}.
For a given configuration σ we say that the configuration σ' is a **2-change neighbor** of σ if σ' is obtained from σ by breaking 2 links and substituting them with other two links in such a way σ' is still a configuration connecting all the nodes $a_1, a_2, ...a_n$ (in the sense given above).

Examples :



If σ is any configuration identified with a permutation $a_1,...a_n$,we extract randomly two links $l_1, l_2 \in \{1,2,...n-1\}$; then breaking these two links causes the graph to have at most 3 connected components which remain fixed.
We denote $[a_l, a_{l+k}]$ a path connecting successively $a_l, a_{l+1}, ...a_{l+k}$,in the order.
Then,our **2-change** mechanism implies one of the following new configurations : $(n > 2)$

a) 1 connected component remains fixed :

(a_1) : $l_1 = 1$, $l_2 = 2$:
$[a_2, a_1, w_1]$
$[a_1, w_1, a_2]$,where $w_1 = [a_3, ...a_n]$
$[w_1, a_2, a_1]$

(a_2) : $l_1 = n-2$, $l_2 = n-1$:
$[w_1, a_n, a_{l_2}]$
$[a_{l_2}, w_1, a_n]$,where $w_1 = [a_1, a_{l_1}]$
$[a_n, a_{l_2}, w_1]$

(a_3) : $l_1 = 1$, $l_2 = n-1$:
$[a_1, a_n, w_1]$
$[a_n, w_1, a_1]$,where $w_1 = [a_2, a_{n-1}]$
$[w_1, a_1, a_n]$

b) 2 connected components remain fixed

(b_1) : $l_1 = 1$,$l_2 > 2$, $l_2 < n-1$:
$[w_1, a_1, w_2]$
$[w_2, w_1, a_1]$,where $w_1 = [a_2, a_{l_2}]$; $w_2 = [a_{l_2+1}, a_n]$
$[a_1, w_2, w_1]$

(b_2) : $n-2 > l_1 \geq 2$; $l_2 = l_1 + 1$
$[w_1, w_2, a_{l_2}]$
$[a_{l_2}, w_1, w_2]$,where $w_1 = [a_1, a_{l_1}]$; $w_2 = [a_{l_2+1}, a_n]$
$[w_2, a_{l_2}, w_1]$

(b_3) : $l_1 \geq 2$; $l_2 = n-1$
$[w_1, a_n, w_2]$
$[w_2, w_1, a_n]$,where $w_1 = [a_1, a_{l_1}]$; $w_2 = [a_{l_1+1}, a_{l_2}]$
$[a_n, w_2, w_1]$

c) 3 connected components remain fixed :
$l_1 \geq 2$;$l_2 > l_1 + 1$
$[w_1, w_3, w_2]$
$[w_2, w_1, w_3]$
$[w_3, w_2, w_1]$,
where $w_1 = [a_1, a_{l_1}]$, $w_2 = [a_{l_1+1}, a_{l_2}]$, $w_3 = [a_{l_2+1}, a_n]$

We observe that the presented **2-change** mechanism is a little different from analogue other ones ([2],[9]) ; indeed,in the case when $l_2 = l_1 + 1$ and $l_1 = 1$ or $l_1 = n-2$,a new configuration that reverses order of the broken link $[a_l, a_{l+1}]$, $l = l_1, l_2$ is permitted, that is the link $[a_{l+1}, a_l]$ is allowed in the new configuration.
As easily seen,the size of the neighbor of a configuration σ ,according to the 2-change above, is constant (independent of σ) and it is :(*)

$$|\Omega_\sigma| = 3 \times \binom{n-1}{2} \tag{2.4}$$

$$= \frac{3}{2} \times (n-1)(n-2).$$

Moreover,any configuration σ_1 can be obtained by any other configuration σ_2 through a finite sequence of permutations of nodes $a_1, ...a_n$ which are obtained by means of the 2-change mechanism described above.
Thus,the notion of neighbourhood implied by the 2-change mechanism yields an irreducible Markov chain ,to be used in the SAA.

(*) In the usual 2-change neighbourhood structure,for the travelling salesman problem one has $\Theta = (n-1)(n-2)$.

2.2 The multiple digest problem

In this subsection we describe the multiple digest problem of molecular biology. In particular,we treat here the case of two digests,without considering for the moment measurement errors.This problem is called 'double digest problem'.

Let us consider a linear tract of DNA of lenght L_d and two restriction enzymes used singly or in combination. Each of restriction enzymes cuts the DNA at all occurrences of a short specific pattern.

We apply the first enzyme and record the lenghts of resulting fragments; we do the same thing with the second enzyme.
Thus,we obtain two data lists of fragments lenght where each enzyme is used singly :

$$A = \{a_1, a_2, ...a_n\}, \sum_{i=1}^{n} a_i = L_d$$

$$B = \{b_1, b_2, ...b_m\}, \sum_{j=1}^{m} b_j = L_d$$

Using the enzymes in combination,the DNA is cut at all occurrences specific to both patterns ,so we obtain an other data list :

$$C = AB = \{c_1, c_2, ...c_{n_C}\}, \sum_{k=1}^{n_C} c_k = L_d$$

(Note that we have recorded only lenght of fragments.)
Generally, A , B , C are multisets,in the sense that there may be values of fragment lenghts that occur more than once.
We adopt the convention that the sets A , B , C are ordered ,that is $a_i \leq a_j$, for $i \leq j$ and likewise for the sets B and C .
Our problem consists in finding orderings for the set A and B by using the above data in such a way the double digest implied by these ordering coincides exatly with C . In other words,a solution to the double digest problem consists in localizing the (possible) sites of cleavage of each enzyme.

The mathematical statement of the problem is as follows.
Consider a permutation $\pi = \{a_{\pi_1}, ...a_{\pi_n}\}$ of the elements of A and a permutation $\mu = \{b_{\mu_1}, ...b_{\mu_n}\}$ of the elements of B , we call $\sigma = (\pi, \mu)$ a configuration.
Now set :

$$\Omega'_\pi = \{a_{\pi_1}, a_{\pi_1} + a_{\pi_2}, ...a_{\pi_1} + ... + a_{\pi_n}\}$$
$$\Omega''_\mu = \{b_{\mu_1}, b_{\mu_1} + b_{\mu_2}, ...b_{\mu_1} + ... + b_{\mu_m}\}$$

and

$$\Omega = \Omega'_\pi \cup \Omega''_\mu = \{\sigma_0, \sigma_1, ...\sigma_{n_{12}}\}$$

where the elements in Ω are listed in ascending order.
We observe that now Ω is not a multiset.
In the corrispondence of the configuration $\sigma = (\pi, \mu)$ we put

$$C_\sigma = C_{\pi\mu} = \{c : c = \sigma_j - \sigma_{j-1}\}$$

that is

$$C_\sigma = \{\sigma_0, \sigma_1 - \sigma_0, \sigma_2 - \sigma_1, ...\sigma_{n_{12}} - \sigma_{n_{12}-1}\}.$$

The problem is to find a configuration $\sigma = (\pi, \mu)$ such that $C_\sigma = C$.

Remark

Here n_{12} ,the number of elements of the set C_σ ,may be different from n_C ,the number of elements of C .
In order to implement the SAA as described in subsection 2.1 we are need an energy function (to minimize) and a neighborhood structure.
Several choises are available for the energy function f ; here we consider two possible functions :

(2.5) $$f(\sigma) = f(\pi, \mu) = \frac{\sum_{i=1}^{min(n_C, n_{12})}(c_i(\pi, \mu) - c_i)^2 + (n_{12} - n_c)^2}{\sum_{i=1}^{n_C} c_i^2}$$

or

(2.6) $$f(\sigma) = f(\pi, \mu) = \frac{\sum_{i=1}^{min(n_C, n_{12})}(c_i(\pi, \mu) - c_i)^2)}{c_i} + \frac{(n_{12} - n_C)^2}{n_C}$$

The function in (2.5) measures the norm of the difference between $C(\pi, \mu)$ and C divided by the norm of C ; the function in (2.6) is ispired to a chi-squared like criterion.
Note that the above functions reach their global minimum value zero when $C_\sigma = C(\pi, \mu) \equiv C$;the quantity $(n_{12} - n_C)^2$ has been introduced to increase cost of configurations (π, μ) for which the cardinality n_{12} of the set $C(\pi, \mu)$ is different from the given cardinality n_C of the set C .
Thus,in the SAA, configurations $\sigma = (\pi, \mu)$ for which $(n_{12} - n_C)$ is 'small' are weighted with larger probability then any other configuration for which $(n_{12} - n_C)$ is 'large' ,also for a fixed value of $\| C(\pi, \mu) - C \|$.

We define the set of neighbors of a configuration $\sigma = (\pi, \mu)$ by :

$$\Omega_\sigma = \{\sigma' = (\eta, \mu), \eta \in \Omega_\pi\} \cup \{\sigma'' = (\pi, \gamma), \gamma \in \Omega_\mu\}$$

where Ω_ρ are the neighbors of ρ given by the 2-change mechanism described in section 2.1 .
With the ingredients described above,we runned the SAA by using either the functions (2.5) and (2.6) ;when the function (2.6) was used ,the algorithm resulted to be more efficient and rapidly convergent.

We have used a simple cooling schedule in which the decrement of the control parameter was given by $T_{k+1} = \alpha T_k$, $\alpha = 0.9$. (We have not lowered temperature at the rate *const./logn* ,for the sake of rapidity.)
The initial value of temperature has been choosen in such a way that the initial acceptance ratio $\chi_0 = \chi(T_0)$ was close to 1 ;here

$$\chi(T) = \frac{(the\ number\ of\ accepted\ transitions)}{(the\ number\ of\ proposed\ transitions)} |_T$$

In practice,this has been achieved by starting off at small positive value of T_0 and multiplying it with a constant factor ,larger than 1 ,untill the corresponding value of χ_0 ,calculated from the generated transitions, was greater than 0.8 .
Typical values of T_0 used in our computation were $T_0 = 10 \div 20$.

We remark that the success of the algorithm depends very strongly on the choice of the initial temperature T_0 ;indeed, a too small value of the initial temperature T_0 does not allow to accept virtually all proposed transitions ,when starting,and this may cause the failure of the algorithm .
The lenght of Markov chains was choosen independent from the temperature T_k and equal to the size of neighbor of configurations $\sigma = (\pi, \mu)$,that is : $L(T_k) = L = 2\Theta$,where Θ is given by (2.4) .
The initial configuration (σ_0, μ_0) was taken in such a way the fragment lenghts in σ_0 and μ_0 were listed in increasing order.

3. On the molteplicity of solutions in the multiple digest problem

For given multiple digest ,the solution to the multiple digest problem is not unique.
For istance,in the case of double digest, we can consider the exact sequence taken from Human β-Globin gene sequence (see sect. 4).

1st digest	$\{556, 15, 760, 11, 57, 49, 130, 217, 66, 128, 63\}$
2nd digest	$\{664, 262, 132, 994\}$
double digest	$\{556, 8, 7, 255, 132, 373, 11, 57, 49, 130, 217, 66, 128, 63\}$

Now,we say respectively A,B,C the digests obtained from the above ones by taking the fragment lenghts in increasing order.
By choosing different initial values of the random numbers generator , by means of SAA ,we found several solutions to the above problem of size $11! \times 4! = 958003200$. Two of these solving orderings are :

$$(I) \quad A = \{63, 217, 11, 760, 15, 556, 66, 130, 57, 128, 49\}$$

$$B = \{664, 132, 262, 994\}$$

$$(II) \quad A = \{49, 57, 556, 66, 130, 128, 15, 760, 11, 63, 217\}$$

$$B = \{994, 262, 132, 664\}.$$

Here ,we will discuss the problem of nonuniqueness of the solution to the multiple digest problem. To this end,we recall the following Kingman subadditive ergodic theorem [8] :

Subadditive ergodic theorem.

For any two integers $n \leq m$,let $X_{n,m}$ be a collection of random variables such that :

(i) $n < k < m \Rightarrow X_{n,m} \leq X_{n,k} + X_{k,m}$;
(ii) the joint distributions of $\{X_{n+1,m+1}\}$ are the same as those of $\{X_{n,m}\}$;
(iii) $E(X_0, n)$ exists and satisfies $E(X_0, n) \geq -Cn$, for some constant C and all $n > 1$.

Then :
there exists the finite limit :

$$\lim_{n \to \infty} \frac{X_{0,n}}{n} = \lambda$$

with probability one and in mean .

Let us consider now the following probability model for the multiple digest problem.
Take a segment of DNA of lenght L_d with sites labeled $1, 2, \ldots L_d$ and consider N restriction enzymes .Assume any one of the restriction enzyme used in a single digest cuts at site l independently with probability $p_i \in (0,1)$,$i = 1, ..N$.
Indeed,a segment of DNA can be viewed as a string of independent, identically distributed random variables with values in a four alphabet.
The number of cutting by restriction enzymes,for example,relatively to the bacteriophage λ gene sequence,varies from relatively small value of order unity (1 as for ApaI , 5 as for EcoRI enzyme) to higher values (50 as for RsaI) (see [12]) .

We define a coincidence to be the event that a site is cut by all the N enzymes ; the probability of the occurrence of such an event is

$$P = \prod_{i=1}^{N} p_i \, .$$

(At the ends sites it occurs,by definition).
For $n \leq m$ we may consider the multiple digest problem for only that segment located between the n^{th} and m^{th} coincidence, we say this segment $[n, m]$. We set $Y_{n,m}$ the number of solutions to the multiple digest problem for this segment ; that is $Y_{n,m}$ is the number of orderings of sets $A_1, \ldots A_N$ which produce the segment $[n, m]$ of the given set C of fragment lenghts obtained when all the enzymes are used simultaneously.
It easily follows that $Y_{n,m} \geq Y_{n,k} \cdot Y_{k,m}$,if $n < k < m$.
So,if we define $X_{n,m} = -log Y_{n,m}$,we obtain
$X_{n,m} \leq X_{n,k} + X_{k,m}$,whenever $n < k < m$,and condition (i) of the sub. erg. theorem is satisfied.

The condition (ii) follows by the assumption that the cuts are independently distributed in each digest, and they are independent of the sites of cutting.
In order to verify condition (iii) holds, we put n_i ,$i = 1, 2, \ldots$, the lenght of the segment between the $(i-1)^{st}$ and i^{th} coincidence ;the r.v. n_i are independent and identically distributed with mean $E(n_i) = P^{-1}$.
The lenght of the segment from the start until the k^{th}coincidence is $m(k) = n_1 + n_2 + \ldots + n_k$.

There are exatly $2^{(m(k)-k-1)}$ ways for each of the N restriction enzymes to cut the remaining $m(k)-k-1$ sites between 0 and $m(k)$; thus the total number of N-ple of orderings of $A^i_{[o,k]}$, $i=1,\dots N$ is exatly $2^{N[m(k)-k-1]}$.
Therefore $Y_{0,k} \le (2^N)^{m(k)-(k+1)}$ that is

$$X_{0,k} \ge -(log2^N)[m(k)-(k+1)] = -(Nlog2)[m(k)-(k+1)].$$

So

$$E(X_{0,k}) \ge -Nlog2[E(m(k))-(k+1)] =$$

$$= -Nlog2(\frac{k}{P}-k-1) \ge -kNlog2(\frac{1}{P}-1) \ge$$

$$\ge -Ck\ ,$$

where $C = \frac{Nlog2}{P} > 0$.

Thus the limit $\lim_{k\to\infty} \frac{X_{0,k}}{k} = \lambda$ exists with probability one ,thanking to the subadditive ergodic theorem ; moreover $E(\lambda) = E(\lim_{k\to\infty} \frac{X_{0,k}}{k}) = \lambda$.

Now,we show that the constant λ is negative .
Indeed,by the inequality $Y_{n,m} \ge Y_{n,k}\cdot Y_{k,m}$ we obtain $Y_{0,k} \ge \prod_{i=1}^k Y_{i-1,i}$ that implies :

$$\frac{E(logY_{0,k})}{k} \ge \frac{1}{k}E(\sum_{i=1}^k logY_{i-1,i}) =$$

$$= \frac{1}{k}\sum_{i=1}^k E(logY_{i-1,i}) =^{(*)} \frac{1}{k}\sum_{i=1}^k E(logY_{0,1}) =$$

$$= E(logY_{0,1})\ ,$$

where in the equality $(*)$ we have used the fact that $Y_{i+1,j+1}$ has the same distribution as $Y_{i,j}$.
We observe that :

$$Pr\{Y_{0,1} > 1\} = \sum_{i=1}^{L_d} Pr\{Y_{0,1} > 1/n_1 = i\}\cdot Pr\{n_1 = i\} =$$

$$= 0 + Pr\{n_1 = 2\} + ... \ge Pr\{n_1 = 2\} = (1-\prod_{i=1}^N p_i)\cdot\prod_{i=1}^N p_i =$$

$$= P\cdot(1-P)\ .$$

We used the facts that $Pr\{Y_{0,1} > 1/n_1 = 1\} = 0$ and $Pr\{Y_{0,1} > 1/n_1 = 2\} = 1$,as it is easy to see .
Then :

$$E(logY_{0,1}) = \sum_y logy\ Pr\{Y_{0,1} = y\} =$$

$$= \sum_{y\ge2} logy\ Pr\{Y_{0,1} = y\} \ge log2 \sum_{y\ge2} Pr\{Y_{0,1} = y\} =$$

$$= log2\cdot Pr\{Y_{0,1} > 1\} \ge (log2)\cdot(1-P)\cdot P\ .$$

On the other hand :

$$E(logY_{0,1}) \le logL_d \cdot \sum_{y\ge2} Pr\{Y_{0,1} = y\} =$$

$$= logL_d\cdot Pr\{Y_{0,1} > 1\}$$

and therefore :

$$log2\cdot P(1-P) \le \frac{E(logY_{0,k})}{k} \le logL_d\cdot Pr\{Y_{0,1} > 1\}\ ;$$

letting k go to infinity :

$$-\lambda = \lim_{k\to\infty} -\frac{E(X_{0,k})}{k} = \lim_{k\to\infty}\frac{E(logY_{0,k})}{k} = \bar{\gamma}$$

where :

$$0 < P(1-P)\cdot log2 \le \bar{\gamma} \le logL_d\cdot Pr\{Y_{0,1} > 1\}\ .$$

Now, let $Z_{m(k)}$ be the number of solutions for the segment of lenght $m(k)$ beginning at 0 ; by definition $Z_{m(k)} = Y_{0,k}$ and so

$$\lim_{k\to\infty}\frac{logZ_{m(k)}}{m(k)} = \lim_{k\to\infty}\frac{logY_{0,k}}{k}\cdot\lim_{k\to\infty}\frac{k}{m(k)}\ ;$$

by the strong law of large numbers we have :

$$\frac{m(k)}{k} = \frac{n_1 + ...n_k}{k} \longrightarrow (with\ prob.\ 1) \longrightarrow E(n_1) = \frac{1}{P}\ .$$

Therefore :

$$\lim_{k\to\infty}\frac{E(logZ_{m(k)})}{m(k)} = P\cdot\bar{\gamma}$$

Then,for a segment of large lenght m ,we obtain the approximation :

$$Z_m \approx e^{\gamma m}\ ,$$

where $\gamma = P \cdot \bar{\gamma}$. From this formula,it follows that the number of solutions to the multiple digest problem increases exponentially fast as a function of the lenght of the segment ; moreover,such a number depends on the quantity $P = \prod_{i=1}^{N} p_i$; if $\bar{p} = max\{p_k \ , k = 1, ...N\}$ we have

$$P = e^{\sum_{k=1}^{N} logp_k} \leq e^{N log \bar{p}} = e^{-N(log \frac{1}{\bar{p}})}$$

and so $P \to 0 \ , as N \to \infty$.

Therefore,we have exatly only one solution to the multiple digest problem in the limit case when the number of digest is equal to infinity .
Really,it is enough to suppose that $[e^{P\bar{\gamma}m}] = 1$,that is $P\bar{\gamma}m < log2$.
In practice,for a given short enough segment of DNA ,the multiple digest problem in molecular biology is solved by using a great enough number of restriction enzymes.

Therefore,molteplicity of solutions in the multiple digest problem depends not only on the lenght of segment,but also on the frequency of cutting sites by restriction enzymes (the number of solutions increases with the value of $P = \prod p_i \ , N, m$ fixed).

In a simulated double digest with $L_d = 1000$,and $p_1 = 0.3$ $p_2 = 0.4$, (see section 4.) the number of solution is about :

$$Z_{1000} \approx e^{120\bar{\gamma}}$$

where

$$0.073 \leq \bar{\gamma} \leq const. \cdot log1000 \ .$$

Thus Z_{1000} is at least of order 10^4 .

4. Computer results

We have tested the algorithm on exact known data from the bacteriophage λ gene sequence with restriction enzymes Baml and Pst1 ([11]) yelding a problem of size
$| A |! \cdot | B |! = 3! \cdot 4! = 144$ and on data from the Human β-Globine gene ([7]) with restriction enzymes Alu and RsaI ,yelding a problem of size $| A |! \cdot | B |! = 11! \cdot 4! = 958.0032 \cdot 10^6$. Solutions have been obtained spending a few seconds of CPU computer time (*) ,after 3 and 880 iterations ,respectively, for the first and second digest.

(*) Computations have been executed by means of VAX 8250 computer at the Department of Mathematics of the Universita' "Tor Vergata" ,Roma.

The initial configuration (σ_0, μ_0) was taken in such a way the fragment lenghts in σ_0 and μ_0 was listed in increasing order.
We tested our algorithm also on simulated data constructed as described in section 3 .
For example,we have considered a simulated segment of DNA of lenght $L_d = 100$ with $p_1 = 0.3$, $p_2 = 0.4$ corresponding to a double digest problem where $| A |= 29$, $| B |= 36$, $| C |= 56$,and the size of neighbors ,according to (2.4), is $\Theta = 2919$.
We performed several trials and studied how the choices of the initial temperature, the number of temperature stages allowed,and of the lenght of Markov chains ,affected the succes and rapidity of convergenge of the algorithm.

We have taken the lenght of Markov chains $L \leq 2919$ and we observed that for great values of L the algorithm found a solution at the 4th temperature stage ($T_0 = 10$, $\alpha = 0.9$) within 1 minute of CPU time .

For $L = 50$,a solution to the above simulated double digest problem was found in only 22 seconds of CPU time ,altough the final temperature was obtained after 20 decrements of temperature (the solution was located after 1000 iterations from the initial configuration).

If $L < 50$, the lenght of Markov chains is too small if compared with the size of neighbors $\Theta = 2919$,so the algorithm has not large enough probability of visiting at least a major part of the neighbourhood of a given solution ,and this causes failure of the procedure.

Since the size of neighbors Θ increases factorially fast as the size of the single digests,giving to L the value of the entire size of neighbors is very time consuming,when running the algorithm on a computer.
Thus, one has to do a compromise between the lenght of Markov chains and the fact that the algorithm needs to visit a great enough part of neighbors with large probability ,to avoid failures.

We have considered also simulated segments of DNA of larger lenght
($L_d = 200$, 300 , $400, ...1000$) .
In a trial with $L_d = 200$, $| A |= 64$, $| B |= 77$, $| C |= 119$, $\Theta = 14,409$,we found a solution to the double digest problem,taking $L = 250$,spending about 20 minutes of CPU time (the final temperature was obtained after 16 decrements of temperature,$T_0 = 10$; the solution was located after 4000 iterationd from the initial configuration).

In another trial with $L_d = 300$, $| A |= 100$, $| B |= 112$, $| C |= 177$, $\Theta = 32,868$,we found a solution taking $L = 600$,spending only 8 minutes of CPU time (the final temperature was obtained after 7 decrements of temperature ,$T_0 = 10$; the solution was located after 4200 iterations from the initial configuration).

Well, in these trials we have taken $L \cong \frac{\Theta}{50}$.

Of course, since the algorithm is based on a stochastic relaxation method, the consumed computing time is not simply a (deterministic) increasing function of sizes of single digests.

Finally,we considered a simulated segment of DNA of lenght $L_d = 1000$,with $| A |= 300$, $| B |= 406$, $| C |= 586$, $\Theta = 379083$,yelding a double digest problem of size

$$\frac{300!}{89! \times 69! \times 32! \times 40! \times 22! \times 16! \times 8! \times 7! \times 9! \times 4! \times 6! \times 2!} \cdot$$

$$\cdot \frac{406!}{148! \times 113! \times 65! \times 30! \times 22! \times 12! \times 9! \times 3! \times 2!} =$$

$$= 7.9 \times 10^{525} \ .$$

A solution was found ,taking $L = 50,000$,spending 4 days and 22 hours of CPU time (the final temperature was obtained after 12 decrements of temperature ,at $T = 3.13$, $T_0 = 10$;the solution was located after $600,000$ iterations from the initial configuration) .

All the trials described above have been performed using both functions f defined in (2.5) ,(2.6) ; the reported results refer to the second function which appeared to make the algorithm more efficient.

Of course,the presence of multiple solutions to the double digest problem confuses the efficiency of the algoriyhm,since the number of solutions increases exponentially fast as a function of the size of digests and of the number of cuttings by restriction enzymes in the single digests . So,for large size of single digests and if the number of cleavings is high,a solution (among the many ones) may be found with a minor effort.

We remark that the usual ,small mapping problem of molecular biology involves short segments of DNA in which the number of cuts by restriction enzymes is relatively small ,if compared with data used in our computer runs.
On the other hand ,large segments of DNA can be broken into several shorter segments by means of wellknown techniques in molecular biology ;terefore one can reduce to study segments which are cleaved a few times only , by restriction enzymes.

Thus the presented SA algorithm is also relevant to large mapping project.

5. Measurement errors

In this section we are concerned with the problem of measurement errors.

In real data ,the lenghts of fragments cannot be measured precisely . Indeed, measurement error is approximately proportional to fragment lenght.

Thus ,we have to process sets of unordered fragment lenghts data such that the sum of fragment lenghts from set to set may no longer even agree.

In order to consider the problem of measurement errors,we have taken exact data from a double digest and we have introduced errors in these data.
Errors have been introduced in all the digests A,B,C ,and the algorithm was succesfull to find a solution to the double digest problem when errors were of order 2% of fragment lenghts.
In fig. 1 a simulated double digest is shown ,together with a solution found by the algorithm (in the last case,the digests A and B are ordered according to the obtained solution,while C is reported in increasing order) .
In figures 2 , 3 , 4 ,we show different runs which refer to sequences data obtained perturbing independently the error free data of digests A,B,C ,corresponding to the same double digest of fig. 1 .

Fig. 1

Error free data from a double digest

$A = \{6,9,12,13,27,33\}$,$n = 6$,$\sum_{i=1}^{6} a_i = 100$
$B = \{29,30,41\}$,$m = 3$,$\sum_{j=1}^{3} b_j = 100$
$C = \{5,6,9,12,13,13,20,22\}$,$n_C = 8$,$\sum_{j=1}^{8} c_j = 100$

A solution found by the algorithm

$A = \{9,33,6,27,13,13\}$
$B = \{29,41,30\}$
$C = \{5,6,9,12,13,13,20,22\}$

Fig.2

Data affected by errors obtained perturbating data of fig.1

$A = \{6.1,9.1,12.1,13.1,27.1,33.1\}$,$\sum_{i=1}^{6} a_i = 100.6$
$B = \{29.2,30.2,41.2\}$,$\sum_{j=1}^{3} b_j = 100.6$
$C = \{5.,6.,9.,12.,13.,13.,20.,22.\}$,$\sum_{j=1}^{8} c_j = 100$

A solution found by the algorithm

$A = \{13.1,6.1,27.1,12.1,33.1,9.1\}$,$\sum_{i=1}^{6} = 100.6$
$B = \{41.2,30.2,29.2\}$,$\sum_{j=1}^{3} b_j = 100.6$
$C = \{5.1,6.1,9.1,12.1,13.,13.1,20.1,22.\}$,$\sum_{j=1}^{8} c_j = 100.6$

(Compare the row C with the above one).

Fig. 3

Data affected by errors obtained perturbating data of fig.1

$A = \{6.1,9.1,12.1,13.1,27.1,33.1\}$,$\sum_{i=1}^{6} a_i = 100.6$
$B = \{29.2,30.2,41.2\}$,$\sum_{j=1}^{3} b_j = 100.6$
$C = \{5.1,6.,9.05,12.05,13.1,13.1,20.1,22.1\}$,$\sum_{j=1}^{8} c_j = 100.6$

A solution found by the algorithm

$A = \{6.1, 13.1, 27.1, 12.1, 33.1, 9.1\}$, $\sum_{i=1}^{6} a_i = 100.6$
$B = \{41.2, 30.2, 29.2\}$, $\sum_{j=1}^{3} b_j = 100.6$
$C = \{5.1, 6.1, 9.1, 12.1, 13.0, 13.1, 20.1, 22.\}$, $\sum_{j=1}^{8} c_j = 100.6$

(Compare the row C with the above one).

Fig. 4

Data affected by errors obtained perturbating data of fig.1

$A = \{6.2, 9.2, 12.1, 13.1, 27.1, 33.12\}$, $\sum_{i=1}^{6} a_i = 100.82$
$B = \{29.2, 30.3, 41.32\}$, $\sum_{j=1}^{3} b_j = 100.82$
$C = \{5.1, 6.02, 9.15, 12.05, 13.10, 13.10, 20.20, 22.10\}$, $\sum_{j=1}^{8} c_j = 100.82$

Error $\sim 2\%$

A solution found by the algorithm

$A = \{9.2, 33.1, 12.1, 27.1, 6.2, 13.1\}$, $\sum_{i=1}^{6} a_i = 100.82$
$B = \{29.2, 30.3, 41.32\}$, $\sum_{j=1}^{6} b_j = 100.82$
$C = \{5.07999, 6.1999, 9.20, 12.10, 13.10, 13.12, 20.0, 22.02\}$, $\sum_{j=1}^{8} c_j = 100.8$

(Compare the row C with the above one).

6. Concluding Remarks .

We have described a stochastic relaxation method,the so called simulated annealing algorithm,to solve multiple digest mapping problems for DNA ,with the aid of a computer.
The procedure was written in Fortran language and implemented on a VAX computer.
Indeed,we wrote a rather crude sequential program to execute the procedure, so surely it may be improved to reduce execution time.
Moreover,one can consider the opportunity of modifying the program for implementing it on a parallel computer ,so the execution time may be further reduced.

With the above improvements,it is realistic to use in the SAA a logarithmical decrement of temperature as requested to do the algorithm converge.
Indeed,the faster exponential decrement of temperature causes failure of the algorithm in some cases ,for some choices of initial temperature (the system ends in metastable states),altough our state space is discrete and therefore we have not to observe really relaxation,but,in the succesfull cases,equilibrium is reached exatly in a finite number of transitions.

Finally,increasing the rapidity of execution of the program,makes it more realistic for large mapping projects,that is for problems of size larger than the size of the double digest problem considered in section 4.(it was of order 10^{525} and required about 4 days of CPU computer time to find a solution).

We remark that the choices of the initial temperature,of the neighboorhod structure and of the cost function f are peculiar to make more efficient the algorithm.

References

[1] : E. Aarts,J. Korst :" Simulated Annealing and Boltzmann Machines". John Wiley and Sons (1989).

[2] : E. Bonomi,J. L. Lutton :"The N-city travelling salesman problem: statistical mechanics and the Metropolis algorithm" SIAM Rev. 26 (1984) 551-568

[3] : R. Durand,F. Bregerere:"An efficient program to construct restriction maps from experimental data with realistic error levels" Nucleic Acids Res. 12 (1985),703,716 .

[4] : W. M. Fitch,T. F. Smith ,W. W. Ralph :"Mapping the order of DNA restriction fragments" Gene 22 (1983),19-29

[5] : S. Geman,D. Geman :"Stochastic relaxation,Gibbs distribution,and the Bayesian restoration of images" IEEE Trans. Pattern and Mach.intell. 6 (1984) ,721-741.

[6] : L. Goldstein,M. S. Waterman :"Mapping DNA by stochastic relaxation" Adv. in Appl. Math. 8 (1987) ,194-207.

[7] : Human β-Globine Gene sequence : data kindly furnished by Dr. G. Pepe of the Depart. of Biology ,the University "Tor Vergata",Rome.

[8] : J. F. C. Kingman :"Subadditive ergodic theory" Ann. Probab. 1 (1973), 883-909 .

[9] : S. Lin :"Computer solutions of the travelling salesman problem" Bell System Tech. J. 44 (1965) ,2245-2269.

[10] : C. Nolan,G. P. Mairna,A.A. Szalay :"Plasmid mapping computer program" Nucleic Acids Res. 12 (1984),717-729.

[11] : W. Pearson :"Automatic construction of restriction site maps" Nucleic Acids Res. 10 (1982),217-227 .

[12] : Promega : Reference Restriction enzymes.

[13] : M. Stefik :"Inferring DNA structure from segmentation data" Artificial Intell. 11 (1978) ,85-114 .

[14] : M. S. Waterman,J. R. Griggs :"Interval graphs and maps of DNA" Bull. Math. Biol. 48 No 2 (1986) ,189-195 .

[15] : M. Wulkan,T.J. Lott :"Computer aided construction of nucleic acid restriction maps using defined vectors" Comput. Appl. Biosci. 1,(1985) ,235-239 .