



MODELS MATEMÀTICS I DE SIMULACIÓ DEL POLIMORFISME CROMOSÒMIC

PRODUÏT PER INVERSIONS

Memòria presentada per
optar al grau de Doctor
en Biologia per la Univer-
sitat de Barcelona per
en

JORDI OCAÑA I REBULL

Vist i Plau
El Director de la Tesi

Dr. ANTONI PREVOSTI I PELEGRIN
Catedràtic de Genètica
Director del Departament de Genètica
Facultat de Biologia
Universitat de Barcelona

Barcelona, Setembre de 1981.



APENDIXS

Apèndix 1. Càlcul de la probabilitat de supervivència proposada
a l'apartat 3.2.

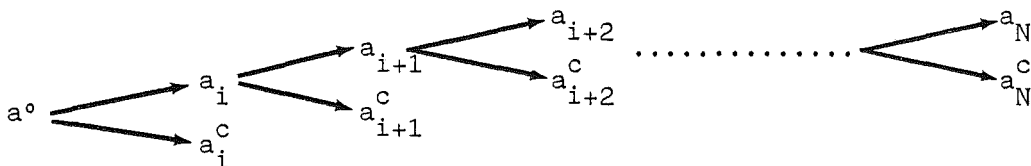
Sigui una població molt gran (infinita) de zigots constituïda per dos tipus A i B (per exemple dos fenotips associats a un locus amb dos al·lels i amb dominància), en proporcions P i $Q=1-P$ respectivament. Competeixen per K "unitats de medi", cada una de les quals pot allotjar un zigot. Un nombre N de zigots es reparteixen seqüencialment i a l'atzar per les K caselles, si dos zigots coincideixen a una mateixa casella "compe-teixen" per ella i un desplaça a l'altre, el "guanyador" pot continuar el seu desenvolupament i el "perdedor" mor. La regla de competició és: A sempre "guanya" a B, si hi ha dos tipus iguals es fa una elecció aleatòria. Es suposa que l'única força selectiva és aquesta competició. Els N zigots que arriben al medi, als instants $t_1 < t_2 < \dots < t_N$, són una mostra dels molts més produïts; per exemple són les llavors, d'entre les moltes produïdes, que troben un lloc adequat per créixer, o són els zigots que finalment arriben a formar-se a partir d'una molt més gran població de gamets.

Sigui a un zigot concret de la classe A. Aquest zigot estarà en una de les K caselles, diguem-ne c. Representem per a_i l'esdeveniment "el zigot a és present a l'instant t". a_N indica que ha arribat a adult, la probabilitat $\text{Prob}(a_N)$ és la proba és la probabilitat de supervivència d'un zigot concret a de la

classe A. $\text{Prob}(a_N) = w_A$. Quelcom semblant es podria definir per l'altre classe. Representem com a_i^o l'esdeveniment consistent en l'arribada de a a l'instant t_i . $\text{Prob}(a_N/a_i^o)$ és la probabilitat de supervivència d'un zigot arribat a l'instant t_i . Podem escriure, per la fórmula de les probabilitats totals:

$$\text{Prob}(a_N) = \sum_{i=1}^N \text{Prob}(a_i^o) \text{Prob}(a_N/a_i^o) \quad (\text{A.1.1})$$

Tenim que $\text{Prob}(a_i^o) = 1/N$, ja que la sembra de a es pot produir en un qualsevol dels N moments. Per altra banda l'esdeveniment a_N/a_i^o "sobreviu encara a l'instant t_N si havia entrat a t_i " és una realització d'un procés estocàstic que es pot representar en forma d'arbre:



la branca de dalt representa aquest esdeveniment, el pas $a_i^o \rightarrow a_i$ representa la implantació de a a la casella on ha anat a parar, $a_i^o \rightarrow a_i^c$ (a_i^c és el complementari o negació de a_i) la no implantació (si ja n'hi havia un altre de A present i l'elecció aleatòria no li ha resultat favorable), $a_i \rightarrow a_{i+1}$ vol dir que a hi és a l'instant següent, etc. Per tant:

$$\begin{aligned} \text{Prob}(a_N/a_i^o) &= \text{Prob}(a_i/a_i^o) \cdot \text{Prob}(a_{i+1}/a_i) \cdot \text{Prob}(a_{i+2}/a_{i+1}) \cdots \\ &\cdots \text{Prob}(a_N/a_{N-1}) \end{aligned} \quad (\text{A.1.2})$$

La probabilitat d'implantació $\text{Prob}(a_i/a_i^0)$ es pot descomposar, també emprant la fórmula de les probabilitats totals, en funció que a la casella on va ja hi hagi un zigot del tipus A (esdeveniment A_i), del tipus B (B_i) o res, sigui buida, (esdeveniment $(A_i \cup B_i)^C$). Si ja hi havia A, competeixen i pot implantar-se amb probabilitat $\frac{1}{2}$, si hi havia B o res, aquesta probabilitat és 1, per tant:

$$\begin{aligned} \text{Prob}(a_i/a_i^0) &= \text{Prob}(A_i) \cdot \text{Prob}(a_i/a_i^0 \times A_i) + \text{Prob}(B_i) \cdot \text{Prob}(a_i/a_i^0 \times B_i) + \\ &+ \text{Prob}((A_i \cup B_i)^C) \cdot \text{Prob}(a_i/a_i^0 \times (A_i \cup B_i)^C) = \\ &= \text{Prob}(A_i) \cdot \frac{1}{2} + \text{Prob}(B_i) \cdot 1 + (1 - \text{Prob}(A_i) - \text{Prob}(B_i)) \cdot 1 = \\ &= 1 - \frac{1}{2} \text{Prob}(A_i) \end{aligned} \quad (\text{A.1.3})$$

Anàlogament, la probabilitat de permanència $\text{Prob}(a_{j+1}/a_j)$ depèn de les noves arribades a l'instant $j+1$ en aquella casella, pot arribar A (esdeveniment A_{j+1}^0), B (B_{j+1}^0) o res ($(A_{j+1}^0 \cup B_{j+1}^0)^C$). Si arriba A, competeixen i pot quedar a amb probabilitat $\frac{1}{2}$, als altres dos casos aquesta probabilitat és 1 (o guanya o no li cal competir), per tant

$$\text{Prob}(a_{j+1}/a_j) = 1 - \frac{1}{2} \text{Prob}(A_{j+1}^0) \quad (\text{A.1.4})$$

A_{j+1}^0 vol dir que s'ha produït una arribada de A a l'instant $j+1$ i precisament al lloc c d'entre els K que n'hi ha, per tant

$$\text{Prob}(A_{j+1}^0) = P \cdot \frac{1}{K} = \frac{P}{K} \quad (\text{A.1.5})$$

i aquesta probabilitat és sempre la mateixa per a tot j , per tant, substituint a (A.1.2):

$$\text{Prob}(a_N/a_i) = (1 - \frac{1}{2}\text{Prob}(A_i)) \left(1 - \frac{P}{2K}\right)^{N-i} \quad (\text{A.1.6})$$

i a (A.1.1):

$$\text{Prob}(a_N) = \frac{1}{N} \sum_{i=1}^N (1 - \frac{1}{2}\text{Prob}(A_i)) \left(1 - \frac{P}{2K}\right)^{N-i} \quad (\text{A.1.7})$$

Caldria ara determinar $\text{Prob}(A_i)$. A_i vol dir que A és present a c a l'instant t_i . Es pot definir un procés markovià amb matriu de transició

$$\begin{array}{cc} & \begin{array}{c} A_{i+1} \\ A_{i+1}^c \end{array} \\ \begin{array}{c} A_i \\ A_i^c \end{array} & \left(\begin{array}{cc} p_{ii+1} & 1-p_{ii+1} \\ q_{ii+1} & 1-q_{ii+1} \end{array} \right) \end{array}$$

Si A és present (A_i) ja no es perd: $p_{i,i+1}=1$, $1-p_{i,i+1}=0$, si A és absent (A_i^c) la transició $A_i^c \rightarrow A_{i+1}$ implica únicament un esdeveniment d'arribada de A a c , aleshores segun que restarà A a c , per tant $q_{ii+1} = \text{Prob}(A_{j+1}^c) = \frac{P}{K}$, de manera que la matriu de transició queda:

$$M = \left(\begin{array}{cc} 1 & 0 \\ \frac{P}{K} & 1 - \frac{P}{K} \end{array} \right) \quad (\text{A.1.8})$$

de manera que les probabilitats de permanència i absència de A a l'instant i , diguem-ne π_i i $1-\pi_i$ i $P_i=(\pi_i, 1-\pi_i)$, seran en notació matricial

$$P_i = P_1 M^{i-1} \quad (\text{A.1.9})$$

amb $P_1 = (\frac{P}{K}, 1 - \frac{P}{K})$. Si operem, obtenim

$$\begin{aligned} \pi_i &= \text{Prob}(A_i) = 1 - \left(1 - \frac{P}{K}\right)^i \\ \text{Prob}(A_i^c) &= \left(1 - \frac{P}{K}\right)^i \end{aligned} \quad (\text{A.1.10})$$

I per tant, finalment en substituir a (A.1.6) s'obté:

$$\text{Prob}(a_N) = \frac{1}{N} \sum_{i=1}^N \left[\left(1 - \frac{1}{2} \left(1 - \left(1 - \frac{P}{K}\right)^i\right)\right) \left(1 - \frac{P}{2K}\right)^{N-i} \right]$$

és a dir

$$\begin{aligned} W_A &= \text{Prob}(a_N) = \frac{1}{2N} \sum_{i=1}^N \left\{ \left(1 + \left(1 - \frac{P}{K}\right)^i\right) \left(1 - \frac{P}{2K}\right)^{N-i} \right\} = \quad (\text{A.1.11}) \\ &= \frac{1}{2N} \sum_{i=1}^N \left\{ \left(1 - \frac{P}{2K}\right)^{N-i} + \left(1 - \frac{P}{K}\right)^i \left(1 - \frac{P}{2K}\right)^{N-i} \right\} \end{aligned}$$

i en sumar ambdues sèries geomètriques finites, s'obté finalment

$$W_A = \frac{K}{NP} \left[1 - \frac{P}{K} \left(1 - \frac{P}{2K}\right)^N - \left(1 - \frac{P}{K}\right)^{N+1} \right] \quad (\text{A.1.11)-bis}$$

Per altra banda i seguint la mateixa nomenclatura per a B, raonaments completament anàlegs ens portarien a

$$\text{Prob}(b_N) = \sum_{i=1}^N \text{Prob}(b_1^o) \text{Prob}(b_N/b_i^o)$$

$$\text{Prob}(b_N/b_i^o) = \text{Prob}(b_i/b_i^o) \text{Prob}(b_{i+1}/b_i) \dots \text{Prob}(b_N/b_{N-1})$$

$$\text{Prob}(b_i^o) = \frac{1}{N}, \quad \text{Prob}(B_i^o) = \frac{Q}{K} = \frac{1-P}{K}$$

$$\text{Prob}(b_{i+1}/b_i) = 1 - \frac{1}{2} \text{Prob}(B_i^o) - \text{Prob}(A_i^o) = 1 - \frac{1+P}{2K}$$

$$\text{Prob}(b_i/b_i^o) = 1 - \frac{1}{2} \text{Prob}(B_i) - \text{Prob}(A_i) \quad (\text{A.1.12})$$

Anàlogament es pot definir el procés markovià amb matriu de transició

$$\begin{array}{c} A_{i+1} \quad B_{i+1} \quad (A_{i+1} \cup B_{i+1})^c \\ \begin{array}{c} A_i \\ B_i \\ (A_i \cup B_i)^c \end{array} \end{array} \begin{pmatrix} 1 & 0 & 0 \\ \frac{P}{K} & 1 - \frac{P}{K} & 0 \\ \frac{P}{K} & \frac{Q}{K} & 1 - \frac{1}{K} \end{pmatrix} \quad (\text{A.1.13})$$

tenint en compte que l'estat inicial és

$$\left(\frac{P}{K}, \frac{Q}{K}, 1 - \frac{1}{K} \right) = (\text{Prob}(A_1), \text{Prob}(B_1), \text{Prob}((A_1 \cup B_1)^c))$$

es dedueix fàcilment que

$$\text{Prob}(B_i) = \left(1 - \frac{P}{K}\right)^i - \left(1 - \frac{1}{K}\right)^i$$

i per tant

$$\begin{aligned} \text{Prob}(b_i/b_i^o) &= 1 - \frac{1}{2} \left[\left(1 - \frac{P}{K}\right)^i - \left(1 - \frac{1}{K}\right)^i \right] - 1 + \left(1 - \frac{P}{K}\right)^i = \\ &= \frac{1}{2} \left[\left(1 - \frac{P}{K}\right)^i + \left(1 - \frac{1}{K}\right)^i \right] \end{aligned} \quad (\text{A.1.14})$$

d'on

$$W_B = \text{Prob}(b_N) = \frac{1}{2N} \sum_{i=1}^N \{ ([1-\frac{P}{K}]^i + [1-\frac{1}{K}]^i) (1-\frac{1+P}{2K})^{N-i} \} \quad (\text{A.1.15})$$

que si sumem les sèries geomètriques finites corresponents es converteix en

$$W_B = \frac{K}{N(1-P)} \left[\left(1-\frac{P}{K}\right)^{N+1} - \left(1-\frac{1}{K}\right)^{N+1} - \frac{(1-P)}{K} \left(1-\frac{1+P}{2K}\right)^N \right] \quad (\text{A.1.15})\text{-bis}$$

Les eficàcies de A i B, W_A i W_B , que són respectivament $\text{Prob}(a_N)$ i $\text{Prob}(b_N)$ depenen evidentment de P i Q (fixats K i N) és a dir de la freqüència relativa, i de N o K, la densitat.

En particular és interessant estudiar aquestes probabilitats quan $K \rightarrow \infty$ per N constant, a la pràctica quan els recursos del medi són molt superiors al nombre de zigots produïts, és a dir, que no hi ha pràcticament competició. Aleshores, a partir de (A.1.11) i de (A.1.15) es veu fàcilment que

$$\begin{aligned} \text{Prob}(a_N) &\longrightarrow 1 \\ \text{Prob}(b_N) &\longrightarrow 1, \quad \text{quan } K \rightarrow \infty \end{aligned}$$

Efectivament,

$$\lim_{K \rightarrow \infty} \left(1-\frac{P}{K}\right) = \lim_{K \rightarrow \infty} \left(1-\frac{P}{2K}\right) = \lim_{K \rightarrow \infty} \left(1-\frac{1}{K}\right) = \lim_{K \rightarrow \infty} \left(1-\frac{1+P}{2K}\right) = 1$$

d'on

$$\lim_{K \rightarrow \infty} W_A = \frac{1}{2N} \sum_{i=1}^N \{(1+1) \cdot 1\} = \frac{2N}{2N} = 1$$

$$\lim_{K \rightarrow \infty} W_B = \frac{1}{2N} \sum_{i=1}^N \{(1+1) \cdot 1\} = \frac{2N}{2N} = 1$$

De (A.1.15) es dedueix que per $P=1$ es té

$$W_B = \frac{1}{2N} \sum_{i=1}^N \{ ([1-\frac{1}{K}]^i + [1-\frac{1}{K}]^i) (1-\frac{1}{K})^{N-i} \} = \frac{1}{2N} \sum_{i=1}^N 2(1-\frac{1}{K})^N = (1-\frac{1}{K})^N$$

i de (A.1.15)-bis podem deduir que, quan $P=0$,

$$W_B = \frac{1}{N} [K(1-(1-\frac{1}{K})^{N+1}) - (1-\frac{1}{2K})^N]$$

mentre que de (A.1.11) i (A.1.11)-bis s'èn dedueix respectivament que

$$W_A = \frac{K}{N} [1-\frac{1}{K} (1-\frac{1}{2K})^N - (1-\frac{1}{K})^{N+1}] \quad \text{quan } P=1$$

i

$$W_A = 1 \quad \text{quan } P = 0 .$$

Apèndix 2: Deducció de les expressions que permeten calcular les viabilitats dels diferents genotips a les simulacions finites descrites a la secció 6.

Sigui G el conjunt de tots els genotips que es poden formar a partir dels gamets produïts per una població diploide a la qual s'estudien L ($L \geq 1$) loci. A cada $g \in G$ li correspon un fenotip mètric $X(g)$ (que pot ser multivariable $X=(X_1, \dots, \dots, X_c)$) corresponent al fenotip mitjà de tots els individus portadors del genotip g . Cada fenotip té associat un valor adaptatiu, que es pot caracteritzar pel valor pres per una funció $a: \mathbb{R}^c \longrightarrow \mathbb{R}$, $a(X(g))$. Per tant es pot parlar del valor adaptatiu de genotip g : $S(g) = a(X(g))$. En particular, a pot ser certa funció f , decreixent, de la distància entre el fenotip $X(g)$ i cert fenotip hipotètic òptim 0 : $a = f(d(X(g), 0))$. Aquest és el cas de les simulacions finites descrites en aquesta memòria.

Si L és finit, el nombre de genotips diploides possibles és també finit i igual a $m=2^L \times (2^L + 1)/2$ i per tant el recorregut $S(G)$ de la funció S també serà finit. Sigui $S(G) = \{s_1, \dots, \dots, s_n\}$ amb $n < m$. Considerem aquests valors ordenats de més gran a més petit $s_1 \geq s_2 > \dots > s_n$. G està dividit en n classes excloents A_1, \dots, A_n , amb $A_l = S^{-1}(\{s_l\})$, ($l=1, \dots, n$) de genotips segons el seu valor adaptatiu. Sigui p_l la proporció de genotips formats de la classe A_l : $p_l = P(A_l)$.

Volem deduir una expressió general per calcular la viabilitat d'un genotip qualsevol $g \in A_1$, és a dir, la probabilitat que un individu portador de g arribi a la fase adulta com a individu reproductor. Considerem que es compleixen les condicions de les simulacions finites realitzades, les quals es poden resumir en els següents punts: 1) el medi és dividit en K "unitats de medi" que poden ser ocupades per un i solament un preadult, 2) els preadults són generats amb genotip pertanyent a la classe i segons una probabilitat $p_i = P(A_i)$, constant a cada generació (depèn només de les freqüències genotípiques de la generació anterior), 3) es generen N preadults amb total independència i un després de l'altre, i es "col·loquen", un cop generats, en una de les K unitats de medi, totalment a l'atzar, és a dir, amb probabilitat $1/K$ per a cada "unitat de medi" 4) si la unitat de medi on ha arribat el nou preadult (de genotip g) és ja ocupada per un altre (de genotip g') ambdós competeixen segons l'esquema: si $S(g) > S(g')$ el nou substitueix a l'anterior ocupant, el qual desapareix, en cas contrari roman l'anterior ocupant i es perd el nou. Aquesta competició és l'única força selectiva present.

Considerem un concret dels N preadults produïts, el qual pertany a la classe l . Introduïrem la notació següent: a_{li} indicarà l'esdeveniment "el preadult concret (dels N produïts) de la classe l ($l=1, \dots, n$) segueix present a la població des-

en compte que l'arribada de nous individus a una unitat de medi és independent de la presència o absència d'altres, podem escriure

$$P(a_{1i}/a_{1i}^{\circ}) = \sum_{r=1}^n P(A_{ri})P(a_{1i}/a_{1i}^{\circ} \cap A_{ri}) + P(\bar{A}_i)P(a_{1i}/a_{1i}^{\circ} \cap \bar{A}_i) \quad (\text{A.2.3})$$

amb

$$P(a_{1i}/a_{1i}^{\circ} \cap A_{ri}) = \begin{cases} 0 & \text{si } r \leq i \\ 1 & \text{si } r > i \end{cases}$$

i

$$P(a_{1i}/a_{1i}^{\circ} \cap \bar{A}_i) = 1$$

donat l'esquema segons el qual es realitza la competició. Com que $P(\bar{A}_i) = 1 - \sum_{r=1}^n P(A_{ri})$, obtenim immediatament, a partir de (A.2.3), l'expressió:

$$P(a_{1i}/a_{1i}^{\circ}) = \sum_{r=1}^n P(A_{ri}) + 1 - \sum_{r=1}^n P(A_{ri}) = 1 - \sum_{r=1}^i P(A_{ri}) \quad (\text{A.2.4})$$

Igualment, si indiquem com A_{rj}° l'arribada (a la unitat de medi ocupada pel preadult estudiat) d'un preadult de la classe r a l'instant j i com \bar{A}_j° l'absència d'arribada de cap competidor a l'instant j , podem escriure

$$P(a_{1j+1}/a_{1j}) = \sum_{r=1}^n P(A_{rj}^{\circ})P(a_{1j+1}/a_{1j} \cap A_{rj}^{\circ}) + P(\bar{A}_j^{\circ})P(a_{1j+1}/a_{1j} \cap \bar{A}_j^{\circ}) \quad (\text{A.2.5})$$

on

$$P(a_{1j+1}/a_{1j} \cap A_{rj}^{\circ}) = \begin{cases} 0 & \text{si } < 1 \\ 1 & \text{si } \geq 1 \end{cases}$$

$$P(a_{1j+1}/a_{1j} \cap \bar{A}_j^{\circ}) = 1$$

$$i \quad P(\bar{A}_h^{\circ}) = 1 - \sum_{r=1}^n P(A_{rj}^{\circ})$$

de manera que, en substituir a (A.2.5) obtenim

$$P(a_{1j+1}/a_{1j}) = 1 - \sum_{r=1}^{l-1} P(A_{rj}^{\circ}) \quad (\text{A.2.6})$$

A_{rj}° implica la ocurrència conjunta de dos esdeveniments independents: l'arribada del nou individu produït a l'instant j precisament a una unitat de medi concreta (amb probabilitat $1/K$) i que aquest individu sigui de tipus r (amb probabilitat p_r), de manera que $P(A_{rj}^{\circ}) = \frac{1}{K} p_r = \frac{p_r}{K}$

$$P(a_{1j+1}/a_{1j}) = 1 - \frac{1}{K} \sum_{r=1}^{l-1} p_r = 1 - \frac{Q_{l-1}}{K} \quad (\text{A.2.7})$$

amb $Q_1 = \sum_{r=1}^l p_r$, constant per tota arribada j ($j=i+1, \dots, N$).

Si substituïm (a.2.4) i (A.2.7) a (A.2.1) obtenim

$$W_1 = \left(1 - \sum_{r=1}^l P(A_{ri})\right) \left(1 - \frac{Q_{l-1}}{K}\right)^{N-1} \quad (\text{A.2.8})$$

Cal conèixer la probabilitat $P(A_{ri})$ de presència d'un preadult de la classe r a l'instant i , a la unitat de medi consi-

derada. Les probabilitats $(P(A_{1i}), P(A_{2i}), \dots, P(A_{ni}), P(\bar{A}_i))$ es poden considerar les probabilitats d'estat a l'instant i d'un procés markovià amb matriu de transició

$$\begin{matrix}
 & A_{1j+1} & A_{2j+1} & A_{3j+1} & \dots & A_{nj} & \bar{A}_j \\
 A_{ij} & 1 & 0 & 0 & \dots & 0 & 0 \\
 A_{2j} & P(A_{1j+1}^{\circ}) & 1-P(A_{1j+1}^{\circ}) & 0 & \dots & 0 & 0 \\
 A_{3j} & P(A_{1j+1}^{\circ}) & P(A_{2j+1}^{\circ}) & 1-(P(A_{1j+1}^{\circ})+P(A_{2j+1}^{\circ})) & \dots & 0 & 0 \\
 \vdots & \vdots & & & \dots & & \vdots \\
 A_{nj} & P(A_{1j+1}^{\circ}) & P(A_{2j+1}^{\circ}) & P(A_{3j+1}^{\circ}) & \dots & 1-\sum_{r=1}^{n-1} P(A_{rj+1}^{\circ}) & 0 \\
 A_j & P(A_{1j+1}^{\circ}) & P(A_{2j+1}^{\circ}) & P(A_{3j+1}^{\circ}) & \dots & P(A_{nj+1}^{\circ}) & 1-\sum_{r=1}^n P(A_{rj+1}^{\circ})
 \end{matrix}$$

i estat inicial $P_0 = (0, 0, \dots, 0, 1)$ ja que en principi el medi és buit. Efectivament, suposant els tipus de genotips ordenats de superior a inferior valor adaptatiu, la transició $A_{rj} \rightarrow A_{sj+1}$ només es produirà si al l'instant $j+1$ arriba un $s > r$ i $A_{rj} \rightarrow A_{rj+1}$ si no n'arriba cap de nou o un d'inferior. Substituint pels valors de $P(A_{rj+1}^{\circ})$ obtenim la matriu de transició, constant per tot instant:

$$\begin{matrix}
 & A_{1j+1} & A_{2j+1} & A_{3j+1} & \dots & A_{nj+1} & \bar{A}_{j+1} \\
 A_{1j} & 1 & 0 & 0 & \dots & 0 & 0 \\
 A_{2j} & \frac{p_1}{K} & 1-\frac{p_1}{K} & 0 & \dots & 0 & 0 \\
 A_{3j} & \frac{p_1}{K} & \frac{p_2}{K} & 1-(\frac{p_1}{K} + \frac{p_2}{K}) & \dots & 0 & 0 \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 A_{nj} & \frac{p_1}{K} & \frac{p_2}{K} & \frac{p_3}{K} & \dots & 1-\sum_{r=1}^{n-1} \frac{p_r}{K} & 0 \\
 \bar{A}_j & \frac{p_1}{K} & \frac{p_2}{K} & \frac{p_3}{K} & \dots & \frac{p_n}{K} & 1-\frac{1}{K}
 \end{matrix} = M \tag{A.2.9}$$

de manera que el vector de probabilitats

$$P_i = (P(A_{ri}))_{r=1, \dots, n}$$

es pot calcular com $P_i = P_0 M^i$.

Per inducció s'obté fàcilment que per qualsevol $i=1, \dots, N$:

$$\begin{aligned} P(A_{1i}) &= 1 - \left(1 - \frac{p_1}{K}\right)^i \\ P(A_{ri}) &= \left(1 - \frac{1}{K} \sum_{h=1}^{r-1} p_h\right)^i - \left(1 - \frac{1}{K} \sum_{h=1}^r p_h\right)^i \quad r = 1, \dots, n \\ P(\bar{A}_i) &= \left(1 - \frac{1}{K}\right)^i \end{aligned}$$

i si fem $Q_r = \sum_{h=1}^r p_h$, $Q_0 = 0$, $Q_n = \sum_{h=1}^n p_h = 1$

$$\begin{aligned} P(A_{ri}) &= \left(1 - \frac{Q_{r-1}}{K}\right)^i - \left(1 - \frac{Q_r}{K}\right)^i, \quad r = 1, \dots, n \\ P(\bar{A}_i) &= \left(1 - \frac{Q_{n-1}}{K}\right)^i - \left(1 - \frac{Q_n}{K}\right)^i \end{aligned} \quad (\text{A.2.10})$$

En substituir (A.2.10) a (A.2.8) obtenim

$$\begin{aligned} w_1 &= \frac{1}{N} \sum_{i=1}^N \left\{ \left(1 - \sum_{r=1}^1 \left[\left(1 - \frac{Q_{r-1}}{K}\right) - \left(1 - \frac{Q_r}{K}\right) \right] \right) \left(1 - \frac{Q_1}{K}\right)^{n-i} \right\} = \\ &= \frac{1}{N} \left\{ \sum_{i=1}^N \left(1 - \frac{Q_1}{K}\right)^{N-i} - \left(1 - \frac{Q_1}{N}\right)^N \sum_{r=1}^1 \sum_{i=1}^N \left[\left(1 - \frac{Q_{r-1}}{K}\right) - \left(1 - \frac{Q_r}{K}\right) \right]^i \right\} = \\ &= \frac{1}{N} \left\{ \frac{1 - \left(1 - \frac{Q_1}{K}\right)^N}{\frac{Q_1}{K}} - \left(1 - \frac{Q_1}{K}\right)^N \left[\frac{1 - \left(1 - \frac{Q_1}{K}\right)^N}{1 - \frac{1}{K}} - 1 - N \right] \right\} \end{aligned}$$

i després d'algunes simplificacions

$$W_1 = \frac{K}{NQ_1} \left[1 - \left(1 - \frac{Q_1}{K}\right)^2 \right] + \left(1 - \frac{Q_1}{K}\right)^N \quad (\text{A.2.11})$$

L'expressió (A.2.11) indica clarament que la viabilitat de cada tipus genotípic depèn de les freqüències (terme $Q_1 = \sum_{r=1}^1 p_r$) de la classe 1 i de totes les classes superiors i de la densitat (N/K) on K és la capacitat del medi i N el nombre de preadults produïts.

És interessant l'expressió assintòtica per K i N molt grans: si suposem que $K \rightarrow \infty$, $N \rightarrow \infty$ (el medi és gran i el nombre de preadults produïts també) i que $N/K \rightarrow \lambda$ on λ és certa constant positiva podem escriure

$$W_1 = \frac{K}{NQ_1} \left[1 - \left(1 - \frac{Q_1}{K}\right)^2 \right] + \left(1 + \frac{1}{-\frac{K}{Q_1}}\right)^{\frac{K}{Q_1} \cdot \frac{-N}{K} \cdot Q_1}$$

de manera que

$$\lim_{\substack{N \rightarrow \infty \\ K \rightarrow \infty \\ \frac{N}{K} \rightarrow \lambda}} W_1 = \frac{1}{\lambda Q_1} \cdot [1 - 1] + e^{-\lambda Q_1} = e^{-\lambda Q_1}$$

a les simulacions finites realitzades $K=400$, $N=1000$ $N/K=2.5$ (densitat) de manera que molt aproximadament podem suposar $W_1 = e^{-2.5Q_1}$, és a dir, que la viabilitat és una funció exponen-

cial negativa de la densitat (2.5 a totes les simulacions realitzades) i de Q_1 que es pot considerar la funció de distribució de la variable S o llast genètic (p.e. allunyament de l'òptim) de cada genotip,

$$Q_1 = \sum_{r=1}^l p_r = P [S \leq s_1] = F(s_1)$$

Apèndix 3. Distribució de la variable aleatòria R="nombre d'esdeveniments de recombinació observables" a la producció dels gamets del model de la secció 6.

A l'apartat 6.1 suposem que la variable aleatòria $NQ =$ 'nombre de quiasmes' a la producció d'un gamet segueix una distribució binomial, $NQ \sim B(4, 15/16)$ per un gamet de 16 loci. La funció de densitat o llei de probabilitat d'aquesta variable és, per tant

$$P_{NQ}(x) = \binom{4}{x} \cdot (15/16)^x (1/16)^{4-x}, \quad x=0,1,2,3,4$$

i 0 en cas contrari

x	$\binom{4}{x} (15/16)^x (1/16)^{4-x}$
0	1.51587×10^{-5}
1	9.15527×10^{-4}
2	0.0205993
3	0.2059936
4	0.7724762

La llei de probabilitat de la variable aleatòria R, nombre d'esdeveniments de recombinació observables, la podem indicar

$$p_R(r) = P[R=r] = \sum_{x=0}^4 p_{NQ}(x) P[R=r/x]$$

on $P[R=r/x]$ indica la probabilitat condicionada que es produeixin r recombinacions si s'han produït x quiasmes.

La probabilitat d'absència total de recombinació és:

$$P[R=0] = p_{NQ}(0) \cdot 1 + p_{NQ}(1) \cdot 0 + p_{NQ}(2) \cdot P[2] + p_{NQ}(3) \cdot 0 + \\ + p_{NQ}(4) \{P[4] + P[2,2]\}$$

on $P[2]$ indica la probabilitat de tots dos quiasmes produïts afectin al mateix lligam entre loci veïns (i per tant no hi hagi recombinació), $P[4]$ la probabilitat que tots quatre quiasmes afectin la mateixa unió i $P[2,2]$ la probabilitat que dos quiasmes afectin una mateixa unió i dos més una altra, situacions que impliquen una absència final de recombinació observable.

Podem afirmar que $P[2] = \frac{15!}{1!14!} \times \left(\frac{1}{15}\right)^2$ ja que quan s'han produït dos quiasmes la probabilitat d'una qualsevol de les maneres de repartir-se, a l'atzar, d'aquests dos quiasmes és $(1/15)^2$ i hi ha $\frac{15!}{1!14!} = 15$ configuracions possibles a les quals el dos quiasmes afecten una de les 15 unions. Igualment $P[4] = \frac{15!}{1!14!} \left(\frac{1}{15}\right)^4$. $P[2,2]$ es pot indicar $\frac{15!}{2!13!} \cdot \frac{4!}{2!2!} \cdot \left(\frac{1}{15}\right)^4$ ja que una configuració concreta, donat que hi ha 4 quiasmes, té probabilitat $(1/15)^4$, fixades dues unions hi ha $\frac{4!}{2!2!}$ maneres possibles de repartir quatre quiasmes de manera que en cor-

responguin dos a cada una i , finalmen, hi ha $\frac{15!}{2!13!}$ configuracions de repartiment de quiasmes a les quals dos loci en presenten 2 i la resta cap.

En general, si indiquem $P[i_1, i_2, \dots, i_K]$ la probabilitat de l'esdeveniment " i_1 quiasmes a una mateixa unió de loci, i_2 a una altra, etc., fins a i_K quiasmes a una altra unió, amb $0 < i_j \leq x$, $\sum_{j=1}^K i_j = x$ on x és el nombre total de quiasmes", podem afirmar que

$$P[i_1, i_2, \dots, i_K] = \frac{15!}{h_1! \dots h_m!} \times \frac{x!}{i_1! i_2! \dots i_K!} \times \left(\frac{1}{15}\right)^x$$

on h_s , per $s = m$, és el nombre de vegades que cada valor possible de i_j és repetit a i_1, \dots, i_K , i h_m és $15 - (h_1 + h_2 + \dots + h_{m-1})$. Per exemple $P[2, 1, 1] = \frac{15!}{1!2!12!} \cdot \frac{4!}{2!1!1!} \left(\frac{1}{15}\right)^4$, amb $h_1=1$ ja que el valor 2 (dos quiasmes en una mateixa unió) està repetit un cop, $h_2=2$ ja que el valor 1 (un quiasma en una unió) està repetit dues vegades i $h_3=15-(1+2)=12$, perquè hi ha 12 unions no afectades per quiasmes.

Efectivament, $\frac{x!}{i_1! i_2! \dots i_K!} \left(\frac{1}{15}\right)^x$ és la probabilitat que, fixat un nombre de x quiasmes, aquests afectin i_1 vegades determinada unió de loci, i_2 una altra, etc., ja que cada configuració de repartiment de x quiasmes (fixat x) té probabilitat $(1/15)^x$ i n'hi ha $\frac{x!}{i_1! \dots i_K!}$ que afectin amb freqüèn-

cies i_1, \dots, i_K ($\sum i_j = x$), K unions determinades. $\frac{15!}{h_1! \dots h_m!}$ indica les maneres possibles en què, donades 15 unions de loci veïns (corresponents a 16 loci), s'en poden escollir h_1 amb un determinat nombre de quiasmes, h_2 amb un altre, etc. fins a h_m sense quiasmes.

Aplicant les fórmules anteriors obtenim:

$$\begin{aligned}
 p_R(0) = P[R=0] &= p_{NQ}(0) \cdot 1 + p_{NQ}(1) \cdot 0 + p_{NQ}(2)P[2] + p_{NQ}(3) \cdot 0 + \\
 &+ p_{NQ}(4) \cdot \{P[4] + P[2,2]\} = \\
 &= 1.52587 \times 10^{-5} + 0.0205993 \times \left\{ \frac{15!}{1!14!} \cdot \frac{2!}{2!} \cdot \left(\frac{1}{15}\right)^2 \right\} + \\
 &+ 0.7724762 \times \left\{ \frac{15!}{1!14!} \cdot \frac{4!}{4!} \cdot \left(\frac{1}{15}\right)^4 + \frac{15!}{2!13!} \cdot \frac{4!}{2!2!} \cdot \left(\frac{1}{15}\right)^4 \right\} = \\
 &= 0.0112304
 \end{aligned}$$

igualment

$$\begin{aligned}
 p_R(1) = P[R=1] &= p_{NQ}(0) \cdot 0 + p_{NQ}(1) \cdot 1 + p_{NQ}(2) \cdot 0 + p_{NQ}(3)P[2,1] + \\
 &+ p_{NQ}(4) \cdot 0 = \\
 &= 9.15527 \times 10^{-4} + 0.2059936 \times \left\{ \frac{15!}{1!11!13!} \cdot \frac{3!}{2!1!} \cdot \left(\frac{1}{15}\right)^3 \right\} = \\
 &= 0.0393676
 \end{aligned}$$

$$\begin{aligned}
 p_R(2) = P[R=2] &= p_{NQ}(2) \cdot P[1,1] + p_{NQ}(4) \cdot P[2,1,1] = \\
 &= 0.0205993 \times \left\{ \frac{15!}{2!13!} \cdot \frac{2!}{1!1!} \cdot \left(\frac{1}{15}\right)^2 \right\} + \\
 &+ 0.7724762 \times \left\{ \frac{15!}{1!2!12!} \cdot \frac{4!}{2!1!1!} \cdot \left(\frac{1}{15}\right)^4 \right\} = \\
 &= 0.2691649
 \end{aligned}$$

$$p_R(3) = p_{NQ}(3) P[1,1,1] = 0.2059936 \times \left\{ \frac{15!}{3!12!} \cdot \frac{3!}{1!1!1!} \cdot \left(\frac{1}{15}\right)^3 \right\} =$$

$$= 0.1666259$$

$$p_R(4) = p_{NQ}(4) \cdot P[1,1,1,1] = 0.7724762 \times \left\{ \frac{15!}{4!11!} \cdot \frac{4!}{1!1!1!1!} \cdot \left(\frac{1}{15}\right)^4 \right\} =$$

$$= 0.4998779$$

A partir d'aquests valors és immediat el càlcul de l'esperança de R o en nombre mitjà de recombinacions per gamet produït.

$$ER = \sum_{r=0}^4 r p_R(r) = 3.1$$

que coincideix amb la que s'obtidria si es calcula la probabilitat o taxa de recombinació entre dos loci veïns, corresponent a la presència d'un nombre senar de quiasmes

$$\binom{4}{1} \left(\frac{1}{16}\right)^1 \left(\frac{15}{16}\right)^3 + \binom{4}{3} \left(\frac{1}{16}\right)^3 \left(\frac{15}{16}\right)^1 = 0.2069$$

i se la multiplica per 15, càlcul vàlid suposant independència entre esdeveniments de recombinació, suposició que no es pot fer segons el mètode emprat per generar-los al present model.

La variància de R es:

$$\sigma^2(R) = ER^2 - (ER)^2 = \sum_{r=0}^4 r^2 p_R(r) - 31^2 = 1.15$$

que no coincideix amb la que esperaríem en condicions de no in-

terferència entre quiasmes: $15 \times 0.2069 \times (1 - 0.2069) = 2.46$.

Pels 8 primers loci la variable aleatòria NQ='nombre de quiasmes' segueix igualment una binomial però de paràmetres 4 i 7/16. Raonat igual que abans podem obtenir la llei de probabilitat de la variable aleatòria R='nombre d'esdeveniments efectius de recombinació'

$$p_R(r) = \sum_{x=0}^4 p_{NQ}(x) P[R=r/x]$$

la qual, tabulada, és:

r	$P_R(r)$
0	0.1539459
1	0.3806762
2	0.3306884
3	0.1153564
4	0.0128173

de manera que el nombre mitjà de recombinacions pels 8 primers loci és:

$$ER = \sum_{r=0}^4 r p_R(r) = 1.44$$

i la variància de R

$$\sigma^2(R) = \sum_{r=0}^4 r^2 p_R(r) - 1.44^2 = 0.87$$

Apèndix 4. Descripció del programa "LEVENE DERIVA"

Les simulacions descrites a la secció 5.4 han estat realitzades mitjançant el programa "LEVENE DERIVA", redactat en BASIC. Aquest programa pot treballar immediatament en qualsevol microordinador COMMODORE de les sèries 2000 i 3000 que disposi d'una unitat de cinta i d'un mínim de 16 kbytes de memòria. El caire conversacional del programa, bassat al caràcter conversacional del llenguatge BASIC el fa summament senzill d'utilitzar. Creiem que una breu discussió de les entrades i sortides del programa és suficient per aprendre a fer-lo servir.

Utilització del programa:

Un cop carregat el programa en memòria (LOAD "LEVENE DERIVA") es pot executar (RUN). Immediatament s'esborra tot el contingut de la pantalla i apareix el missatge:

NOM DEL PROCES ?

podeu contestar amb qualsevol nom i **RETURN** (per exemple: SIMULACIO PRIMERA **RETURN**)

NOMBRE DE LOCI, DE GENERACIONS, DE NINXOLS I EFICACIA MAXIMA ?

separats per comes i en el mateix ordre, heu d'indicar:

nombre de loci estudiats (variable L)

nombre de generacions de durada del procés (NG)

nombre de nínxols o clapes en què està subdividit l'ambient (NX)

eficàcia màxima, és la constant N_m de la fórmula d'avaluació de les eficàcies (WM)

(per exemple: 5,100,2,81 **RETURN** farà que internament $L=5$, $NG=100$, $NX=2$, $WM=81$)

Seguidament apareixerà NX vegades el missatge

OPTIM I CAPACITAT DE LA CLAPA IX

on IX valdrà successivament 1,2,...,NX. Cada vegada cal contestar amb:

valor òptim del caràcter mètric a la clapa IX ($O(IX)$)

capacitat de la clapa IX ($C(IX)$)

(per exemple si **OPTIM I CAPACITAT DE LA CLAPA 1** es contesta: 1, 400 **RETURN**, internament $O(1)=1$, $C(1)=400$, això voldrà dir que per la primera clapa l'òptim O_1 del caràcter mètric avaluat additivament a partir del genotip serà $O_1=1$ i que aquesta clapa sustenta $c_1=400$ gamets).

Si $C(1)$, $C(2)$... les capacitats de cada clapa són totes elles menors que 1 i sumen 1 aleshores tenen un altre sentit: proporció de gamets procedents de cada clapa. (Per exemple po-

dríem haver indicat 1, 0.5 **RETURN** , aleshores $O_1 = 1$, $C_1 =$
 $= 0.5$).

Si $\sum_{i=1}^{NX} C(i) > 1$ es considera que el procés es realitza
 en condicions de deriva. En tal cas el programa demana

NOMBRE DE REPLIQUES ?

cal respondre amb el nombre de repeticions del procés de NG ge-
 neracions que es considera adient realitzar (per exemple: 10
RETURN).

Hi ha dues versions del programa. Una d'elles inicia nous
 processos, en tal cas no cal fer res d'especial, el programa
 genera unes freqüències gamètiques estàndard inicials totes
 iguals i avalua a partir d'elles i de la informació proporciona-
 da, una sèrie de paràmetres necessaris. L'altra versió llegeix
 l'estat final d'un procés anterior per continuar-lo (possible-
 ment amb paràmetres canviats). En aquesta darrera versió cal
 una cinta, en aquest moment, a la corresponent unitat amb la
 informació adient, procedent de la sortida d'un procés anterior.
 Quan ha acabat la lectura, apareix el missatge

STOP AL CASSET I PREMEU QUALSEVOL TECLA

i s'atura el procés, cal enretirar el caset anterior; si es prem
 qualsevol tecla de dades es reemprén el procés.

dríem haver indicat 1, 0.5 **RETURN** , aleshores $O_1 = 1$, $C_1 =$
 $= 0.5$).

Si $\sum_{i=1}^{NX} C(i) > 1$ es considera que el procés es realitza
 en condicions de deriva. En tal cas el programa demana

NOMBRE DE REPLIQUES ?

cal respondre amb el nombre de repeticions del procés de NG ge-
 neracions que es considera adient realitzar (per exemple: 10
RETURN).

Hi ha dues versions del programa. Una d'elles inicia nous
 processos, en tal cas no cal fer res d'especial, el programa
 genera unes freqüències gamètiques estàndard inicials totes
 iguals i avalua a partir d'elles i de la informació proporciona-
 da, una sèrie de paràmetres necessaris. L'altra versió llegeix
 l'estat final d'un procés anterior per continuar-lo (possible-
 ment amb paràmetres canviats). En aquesta darrera versió cal
 una cinta, en aquest moment, a la corresponent unitat amb la
 informació adient, procedent de la sortida d'un procés anterior.
 Quan ha acabat la lectura, apareix el missatge

STOP AL CASSET I PREMEU QUALSEVOL TECLA

i s'atura el procés, cal enretirar el caset anterior; si es prem
 qualsevol tecla de dades es reemprén el procés.

A continuació apareixen els següents missatges:

PERIODICITAT DE SORTIDA DE DADES RESUMIDES ?

cal respondre amb un nombre enter (variable PE). Si IG és el número de la generació actual en un noment donat, si IG és múltiple de PE es genera una sortida de dades resumides per pantalla. Si, a més, $PE > 0$ la mateixa informació treta per pantalla es graba en una cinta, en tal cas cal que a la unitat de cinta n'hi hagi una de disponible. La informació proporcionada és: número de la generació, summa de freqüències gèniques (de l'al·lel 1), summa de desequilibris gamètics, valor d'eficàcia mitjana a cada subpoblació o subdivisió del medi i freqüència del gamet invertit

PERIODICITAT DE SORTIDA EXTENSA ?

cal respondre amb un número enter (PD). El sentit és el mateix que el de PE llevat que la informació proporcionada és més detallada: cada gamet amb la seva freqüència, vector de freqüències gèniques (de l'al·lel 1) a cada locus i summa de freqüències gèniques, matriu de desequilibris gamètics i summa de desequilibris i freqüència de la inversió.

PERIODICITAT DE GRABACIO DE L'ESTAT ?

cal respondre un nombre enter (variable GR) amb el mateix sentit que els anteriors. Especifica cada quantes generacions es graba en cinta l'estat de la simulació per ser, eventualment, recomençada en aquell punt si molt convé.

INVERSIÓ A ALGUN GAMET ? (S,N)

I el procés s'atura novament. Tant sols prement la tecla **S** s'especifica que es produeix una nova inversió, qualsevol altra tecla implica que no hi ha una nova inversió. En tot cas el procés recomença en haver tocat una tecla de dades.

Si la tecla fou **S** apareixen el missatges:

GENERACIÓ INICIAL ?

Cal respondre un nombre enter (NN), per si es vol canviar la numeració de les generacions.

NÚMERO DE GAMET I FREQUÈNCIA INICIAL DE LA INVERSIÓ ?

Cal respondre amb un número enter de 0 a $2^L - 1$ (inclosos), el número del tipus gamètic sobre el qual es produeix la inversió (IN) i amb un número real entre 0 i 1 (inclosos) (QO) que és la freqüència del nou gamet invertit (per exemple 3, 0.001 **RETURN** especifica que la inversió s'ha pro-

duit sobre el gamet 3 (00011) i té una freqüència inicial 0.001).

Si $PO(IN)$, la freqüència inicial del tipus estàndard sobre el que es produeix la inversió, és menor que PO aleshores és impossible aquesta freqüència inicial i es produeix el missatge:

**LA FREQUENCIA INICIAL D'AQUESTA INVERSIO NO POT SUPERAR:
<valor de PO(IN)>**

i es retorna al punt en què es produí el missatge:

INVERSIO A ALGUN GAMET ? (S,N)

Si la tecla fou N o qualsevol altra diferent de S es passa directament a

TAXA DE RECOMBINACIO ENTRE LOCI A:

I des de 1 fins a L-1 es repeteix

1 DE DISTANCIA ?

cal respondre amb un nombre real entre 0 i 1 (inclosos) que indica la probabilitat de recombinació entre loci separats per 1 espai interloci (per exemple 0.15 RETURN)

2 DE DISTANCIA ?

idem per 2 espais interloci.
etc.

Aleshores el programa inicial la simulació propiament dita,
per totes les generacions i rèpliques especificades.

Apèndix 5. Model de simulació SGEN

A.5.1. Consideracions generals

El sistema SGEN és un conjunt de subprogrames redactats en llenguatge PL/I i orientats a facilitar la programació de models de simulació genètica finits i amb generacions no superposades.

Un programa que simuli un model genètic concret estarà format per uns quants d'aquests subprogrames i (eventualment) altres de programats per l'usuari.

Els subprogrames simulen processos implicats en diferents nivells de la modelització dels sistemes genètics. El de més alt nivell ("procedure" principal en PL/I) és CONTROL el qual dirigeix la seqüència lògica dels processos fonamentals en el canvi de les freqüències (gèniques, gamètiques...) de la població, així com la recollida periòdica de resultats CONTROL crida altres subprogrames que simulen processos més específics com ara la migració d'adults des de l'exterior de la població (subrutina MIGRA) o la producció de noves inversions (subrutina INVER, la qual a la seva vegada crida la subrutina NBRI -que dona el nombre d'inversions produïdes a una generació- i POSI - que retorna la localització d'aquestes noves inversions (individu, gamet, i posició concrets). POSI activa la funció RAN, generadora de nombres aleatoris.

Els generadors de nombres aleatoris (RAN) i de distribucions estadístiques (POISS, GAUSS) o les subrutines encarregades de calcular estadístics concrets (COMPUTG) formen els subprogrammes més bàsics. A continuació hi ha els subprogrammes que simulen processos que a nivell genètic també els considerariem bàsics, com ara la gametogènesi (GAMT) o la influència de l'estructura cromosòmica sobre la recombinació (STCR). Al nivell més alt hi ha subprogrammes com REPRO que dirigeix tota la seqüència d'operacions que impliquen la substitució d'una generació per la següent o com CONTROL que controla a REPRO i altres subprogrammes d'alt nivell.

Hom pot fer servir tots els subprogrammes ja existents per construir els propis models. Si la seqüència lògica de processos desitjada no s'adapta a les possibilitats del subprogrammes de SGEN, l'usuari en pot programar una d'adequada pel seu compte i utilitzar els subprogrammes més bàsics ja programats a SGEN. Contràriament pot ser que hi siguim útils els subprogrammes d'alt nivell però que desitgi canviar els processos més bàsics (per exemple l'efecte de l'estructura cromosòmica sobre la recombinació), aleshores bastarà que programi de nou la simulació d'aquests processos.

El sistema té un cert grau de flexibilitat, aconseguida de la manera següent: cada subprograma té un nom, el qual consta a la seva primera instrucció PL/I (la sentència PROCEDURE). Per

exemple hi ha un subprograma encapçalat amb el nom ELOV encarregat de triar aleatòriament la subdivisió del medi on seran "ovipositats" un o més "òvuls fecundats". Usualment ELOV és cridat des de REPRO. Tot bloc o subprograma tal que es vulgui que simuli l'elecció del lloc d'oviposició i que sigui activat des de REPRO ha de ser encapçalat amb el nom ELOV. Però segons el model que vulguem simular aquesta elecció és possible que s'hagi de realitzar de diverses maneres. Pot ser, per exemple, que es vulgui escollir la subdivisió totalment a l'atzar (si hi ha M subdivisions, amb probabilitat $1/M$ per cada una), o que es vulgui fer segons un vector de probabilitats qualsevol (p_1, \dots, \dots, p_M) , $\sum_{i=1}^M p_i = 1$, $p_i > 0$, etc. Una solució seria fer el subprograma ELOV suficientment general per preveure un conjunt prou ampli de possibilitats. Això complicaria molt la lògica dels subprogrames i requeriria l'ocupació de molta memòria central.

Hem preferit definir blocs més curts, que individualment no preveguin totes les possibilitats, però dels quals n'hi hagi diferents versions totes elles encapçalades amb un mateix nom (per exemple ELOV) però enregistrats en una biblioteca (al cas del sistema IBM 360/20 en una àrea de disc magnètic) amb diferents noms (ELOV1002, ELOV2002). Si hom vol que a un programa l'elecció sigui aleatòria, ha d'incloure entre els blocs del programa a ELOV1002, si cal que sigui segons un vector de probabilitats qualsevol, ELOV2002. Aquesta inclusió es fa mitjançant alguna instrucció pròpia del sistema operatiu de l'ordinador utilitzat.

En resum, cada nom lògic de bloc o subprograma pot estar representat per un o més "mòduls" etiquetats amb diferents noms. Naturalment han de tenir tots (els mòduls versió d'un mateix subprograma) els mateixos paràmetres els quals han de coincidir amb els arguments amb què s'els crida des dels subprogrames de nivell més alt.

El subprogrames presents actualment a la biblioteca són indicats, ordenats alfabèticament, a la taula A.5.1 amb una indicació del subapartat d'aquest apèndix al qual s'els descriu més detalladament.

L'esquema jeràrquic és indicat a la taula A.5.2, on s'indica cada nom de subprograma, a continuació el nom dels mòduls on hi ha una versió d'eix subprograma (entre parèntesis) i a sota el nom dels subprogrames utilitzats per aquella versió. Per exemple, del subprograma DESCR, encarregat de descriure l'estat general de la població, n'hi ha dues versions disponibles que corresponen als mòduls DESCRO02 i DESCRTOT. La primera és una curta subrutina que compta el nombre de cariotips presents a tota la població i les seves freqüències, donant finalment l'índex de diversitat de Shanon. Aquesta subrutina no en crida cap altra. La segona fa un inventari detallat dels cariotips presents a cada subdivisió del medi, en determina la seqüència i la genealogia d'inversions produïdes fins a arribar a aquell cariotip, compta les freqüències dels diferents cariotips i determina el contingut al·lèlic d'aquests, comptant també les fre-

qüències dels genotips, finalment dóna el nombre mitjà de loci heterozigòtics per individu i la variància, dins de cada subdivisió i a la població global. Aquesta versió crida les subrutines DECOD (mòdul DECODTOT), que decodifica l'estructura cromosòmica i dóna la seqüència de loci, la genealogia, i COMPUTG (mòdul COMPUTG) que calcula el grau d'heterozigosi.

Els subprogrames marcats amb un asterisc (*) encara no tenen cap mòdul que els representi.

Es important remarcar que si un subprograma en crida d'altres i aquests a la seva vegada d'altres, etc..., si s'inclou el primer en l'ordre jeràrquic, s'hauran d'incloure, per evitar errors d'execució, tots els que són per sota d'ell (els cridats per ell, els cridats pels cridats per ell, etc.) llevat d'aquells que no s'executaran mai en el procés a realitzar i tots els que depenguin d'ells (per exemple si els paràmetres que defineixen un procés de simulació indiquen que no s'han de produir noves inversions, no cal incloure INVER(INVER002) ni els activats des d'ell: NBRI(NBRI002) i POSI(POSI0002), ni tampoc RAN(JORRANMO), cridat per POSI, si és que RAN no s'utilitza a cap lloc més). La inclusió d'un mòdul no utilitzat no representa cap problema, exepete l'ocupació de memòria addicional, i té l'avantatge de fer más generals els programes construïts amb ells, en fer-los vàlids per altres processos on sí que s'utilitzin els subprogrames representats en aquells mòduls.

CLRV	la descripció del qual és a l'apartat A.5.2.27
COMPUTG	idem A.5.2.8
CONTROL	idem A.5.2.1
DECOD	idem A.5.2.7
DESCR	idem A.5.2.6
DSTF	idem A.5.2.18
DSTQ	IDEM A.5.2.22
ELOV	idem A.5.2.21
EMRG	idem A.5.2.28
FIREP	idem A.5.2.16
GAMT	idem A.5.2.15
GAUSS	idem A.5.2.30
GENERA	idem A.5.2.2
GRABA	idem A.5.2.5
INIC	idem A.5.2.13
INVER	idem A.5.2.9
LOQ	idem A.5.2.23
MODIF	idem A.5.2.32
NVRI	idem A.5.2.10
OVPS	idem A.5.2.26
PARAM	idem A.5.2.3
POISS	idem A.5.2.29
POSI	idem A.5.2.11
POVP	idem A.5.2.20

Taula A.5.1. continua

PROG	Idem A.5.2.17
RAN	idem A.5.2.31
RECFUP	idem A.5.2.4.
REPRO	idem A.5.2.12
SIPOS	idem A.5.2.19
SIREP	idem A.5.2.15
STCR	idem A.5.2.24

Taula A.5.1. continuació

```

- CONTROL (CONTROL 2)
  -GENERA      (GENERA02)
                -RAN (JORRANMO)
  -PARAM      (PARAM002)
  -RECUP      (RECUPO02)
  -GRABA      (GRABA002)
  -DESCR      (DESCRO02)
                -DECOD (DECODTOT)
                (DESCRTOT)
                -DECOD (DECODTOT)
                -COMPUTG (COMPUTG)
                (DESCRTO2)
                les mateixes que DESCRTOT
  -MIGRA      (*)
  -MUTAC      (*)
  -INVER      (INVER002)
                -NBRI (NBRI0002)
                -POSI (POSI0002)
                -RAN (JORRANMO)
  -REPRO      (REPRO002)
                -INIC (INIC0002)
                -WAPR (*)
                -PRNX (*)
                -SIREP
                -FIREP
                -SIPOS
                -ELNX (*)
                -PROG (PROG0002)
                -FERT (*)
                -DSTF (DSTF01)
                    GAUSS (GAUSS01)
                    RAN (JORDANMO)
  -POVP      (POVP1002)
                (POVP2002)
  -ELOV      (ELOV1002)
                -RAN (JORRANMO)
                (ELOV2002)
                -RAN (JORRANMO)
  - RECO      (*)
  - DSTQ      (DSTQS01)
                POISS (POISS01)
                RAN (JORRANMO)
  -LOCQ      (LOCQ0002)
                -RAN (JORRANMO)
                -STCR (STCR01)
  -GAMT      (GAMT0002)
  -OVPS      (OVPS0002)
                -CLRV (CLRVO002)
                -RAN (JORRANMO)
  -EMRG      (EMRG0002)
  -MODIF      (MODIFICA)

```


Alguns dels subprogrames són d'una mena tan poc generalitzable (per exemple PARAM, vegeu la secció següent) que allò més usual serà que s'hagin de redactar expressament per cada cas. Si al processos que realitzen calen paràmetres que no estan previstos a la llista d'arguments de les instruccions des d'on es criden, es poden definir com a variables externes (atribut EXTERNAL) dins d'aquestes versions dels subprogrames. L'atribut EXTERNAL farà que puguin ser compartits per diferents subprogrames.

Tots l'anterior és l'esquema general d'organització dels subprogrames del sistema SGEN i de la manera d'utilitzar-los. Un altre aspecte a discutir, abans de passar a la descripció detallada a cada subprograma, és la codificació de les entitats fonamentals sobre les que treballen els programes: els gamets, els genotips i les poblacions.

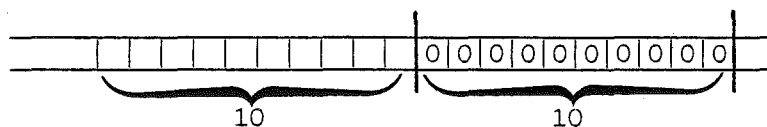
Si estudiem L loci, tot gamet estarà representat per una variable del llenguatge PL/I amb l'atribut BIT(L*(NST+1)). Això vol dir que tot gamet estarà codificat per una tira de $L \times (1 + NST) = L + L \times NST$ llocs de la memòria central de l'ordinador, que poden presentar dos estats (i que com és habitual indiquem 0 i 1) i a la que ens podem referir emprant el nom de la variable. Per exemple, la següent declaració de variable en llenguatge PL/I permetria codificar un gamet:

```
DCL GAMET BIT (L*(NST+1));
```

Els L primers bits codifiquen el contingut gènic, de la manera habitual (un al·lel 0, l'altre 1). Sempre en el mateix ordre, cada posició de la tira de L (primers) bits codifica el contingut al·lèlic d'un locus, sempre el mateix encara que, per exemple, hi hagi inversions.

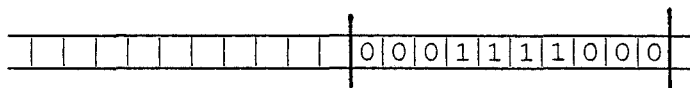
Els NSTxL bits restants codifiquen l'estructura cromosòmica del gamet. NST és una variable entera, $NST \geq 0$, que indica el nivell admissible d'imbricació de les inversions cromosòmiques. Si $NST=0$, no hi pot haver cap codificació de les inversions, la codificació del gamet es redueix als L primers bits que codifiquen el contingut gènic.

Si $NST=1$, $L=10$, i els loci s'anomenen A,B,C,D,E,F,G,H,I i J; un gamet estàndard tindria una seqüència de loci ABCDEFGHIJ i estaria codificat per una tira (si $NST=1$) de 20 bits

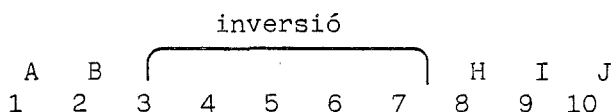


amb els 10 primers representant el contingut gènic i els 10 darrers tots 0. L'absència d'inversions (a partir d'una seqüència considerada arbitràriament com estàndard) sempre es codificada per una tira de zeros. Imaginem ara que volem codificar un gamet portador d'una inversió des del locus 3 al locus 7, (i de sola-

ment aquesta inversió), és a dir ABGFEDCHIJ. Com abans, els deu primers bits codificaran el contingut gènic, cada posició codificarà el mateix locus que amb estructura estàndard. Els deu darrers seran:

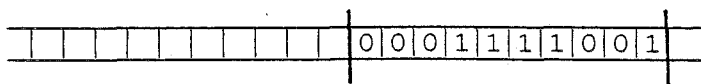


indicant que hi ha una inversió que afecta les unions quarta, cinquena, sisena i setena entre loci, considerant que hi ha les següents unions:

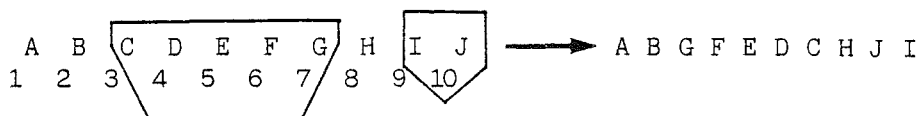


En general, tota inversió que afecta els loci l_1 a l_2 amb $1 \leq l_1 < l_2 \leq L$, es codifica per una tira de L bits $\zeta_1 \zeta_2 \dots \zeta_L$, amb $\zeta_K = 0$ si $K \leq l_1$ ò $K > l_2$ i $\zeta_K = 1$ si $l_1 < K \leq l_2$.

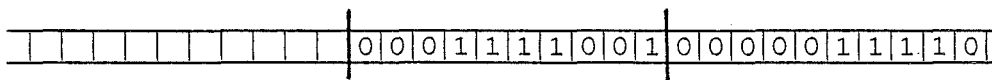
Qualsevol altra ordenació de loci definida per dues o més inversions disjunctes podria ésser codificada igualment, amb $NST=1$, és a dir, amb una sola tira addicional de bits. Per exemple si, a part de la inversió dels loci 3 al 7 n'hi hagués una altra pels loci 9 al 10, la codificació seria



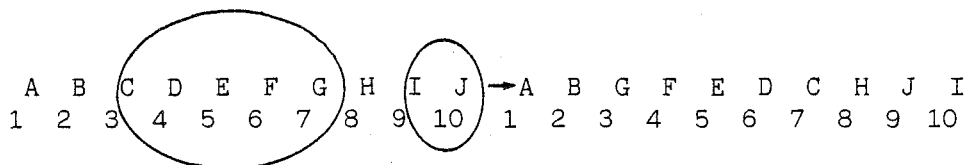
és a dir, dues inversions, la primera afecta els lligams 4,5,6 i 7 i la segona el 10:



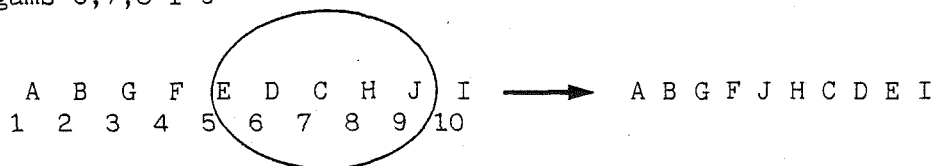
La presència d'inversions produïdes en part o totalment sobre d'altres ja existents requereix la presència de tires addicionals de L bits. És per això que a la variable NST l'hem anomenada "nivell admissible d'imbricació". Per exemple si pel cas anterior, sobre l'ordenació definida per les inversions 3 a 7 i 9 a 10 hi hagués una inversió des del locus 5 al locus 9 (cinquè i novè segons la nova ordenació, no segons l'estàndard), caldrien 30 bits per codificar el gamet (NST=2), els 20 darrers codificant l'estructura:



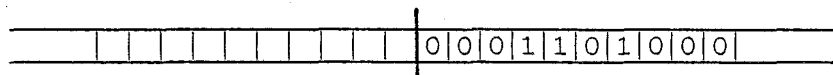
ja que sobre estàndard ABCDEFGHIJ s'haurien produït dues inversions que afectarien els lligams 4,5,6,7 i 10



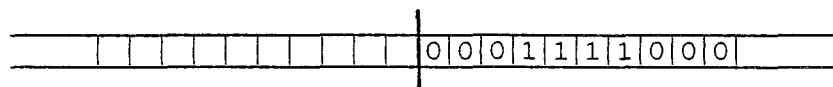
i sobre aquesta ordenació, una altra inversió que afectaria els lligams 6,7,8 i 9:



Notem que no hi ha confusió entre la codificació de dues inversions veïnes (però disjunctes) i la d'una única inversió que afecti els mateixos loci. Per exemple, no es condificarien igual dues inversions, dels loci 3 al 5 i del 6 al 7:



que una única inversió del 3 a 7:



Aquesta manera de codificar l'estructura cromosòmica produïda per inversions permet deixar inalterada, respecte d'estàndard, la posició de la codificació de l'al·lel de cada locus. Això és avantatjós quan l'expressió fenotípic d'un genotip xy , amb $x=x_1x_2\dots x_L$, $y=y_1,\dots,y_L$, és certa funció $f(x_1,\dots,x_L,y_1,\dots,y_L)$ per a l'avaluació de la qual és important l'ordre dels arguments. Per exemple si els K primers ($K < L$) loci codifiquen un caràcter C_1 i els restants loci un altre caràcter C_2 , és convenient no alterar l'ordre dels loci per a una programació fàcil de la funció $f: (C_1, C_2) = f(x_1\dots x_L, y_1\dots y_L)$. També permet representar fàcilment l'efecte de l'heterozigosi estructural sobre la recombinació (vegeu la descripció de la subrutina STCR, apartat A.5.2.24). Finalment, cal dir que és més econòmica quant a utilització de memòria central que altres possibilitats, de les que hem considerat, de representació de l'estructura.

Potser el principal inconvenient és el caràcter poc clar de la codificació, quant a una identificació ràpida de les ordenacions codificades. Aquest problema el soluciona el subprograma DECOD (apartat A.5.2.7).

La representació dels genotips (o zigots, o individus) no té cap dificultat. Un genotip està format per dos gamets, per tant el codificarem mitjançant l'ús d'una variable, que serà un vector o agregat de dues tires de bits, cada una representant un gamet. Per exemple, la següent declaració de variable del llenguatge PL/I podria codificar un genotip:

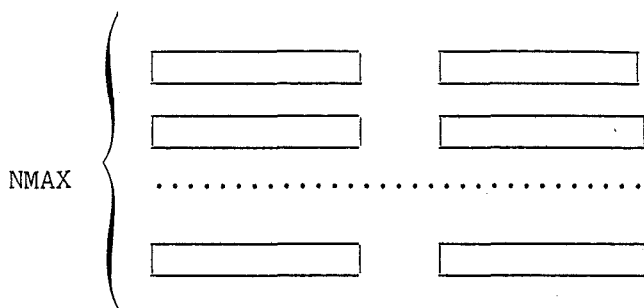
```
DCL ZIGOT(2) BIT(L*(NST+1));
```

ZIGOT(1) és la primera component del vector ZIGOT i és una tira de $L \times (NST+1)$ bits que pot representar un gamet de la manera usual. ZIGOT(2) representa, de la mateixa manera, l'altre gamet que ha produït el genotip.

Una població és un conjunt finit de N genotips. Aquesta població la suposem ocupant un medi amb capacitat per a $NMAX$ ($N \leq NMAX$) individus. Això ho representen mitjançant un agregat (de gamets) de dimensió $NMAX \times 2$. Per exemple

```
DCL POP(NMAX,2) BIT(L*(NST+1));
```

defineix una variable POP que és un agregat de tires de $L \times (NST+1)$ bits (gamets)



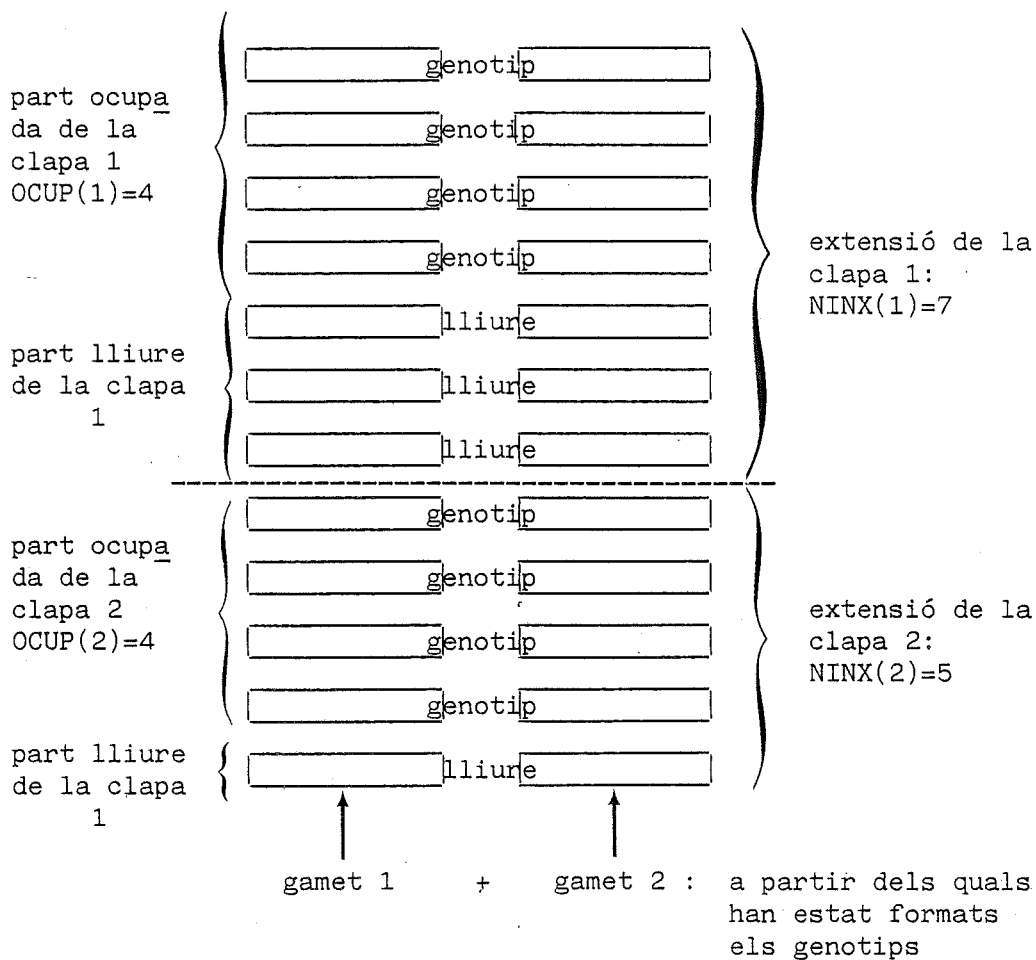
cada fila (2 gamets) representa un genotip, si aquelles posicions de l'agregat POP estan ocupades, o bé un lloc buit, que més endavant pot estar ocupat per un genotip.

Al sistema SGEN, la informació proporcionada per POP o qualsevol altra variable que codifiqui la població, sempre és complementada per tres vectors de nombres enters: NINX, PRIMER i OCUP tots ells de dimensió $NNX \geq 1$. NNX és una variable entera que indica el nombre de clapes, subdivisions o subpoblacions o qualsevol altre terme apropiat, en què està subdividit el medi ocupat per la població (i de passada la població a un instant donat). Evidentment si $NNX=1$ no hi ha cap mena de subdivisió. NINX(j), per $j=1, \dots, NNX$, indica la capacitat (el nombre de files de POP) de la clapa j. Per tant $\sum_{j=1}^{NNX} NINX(j) = NMAX$. PRIMER(j), per $j=1, \dots, NNX$, indica la fila de POP on comença la clapa j. Per tant $1 \leq PRIMER(1) < PRIMER(2) < \dots < PRIMER(NNX) \leq NMAX$. D'una manera estàndard es pren $PRIMER(1)=1$.

OCUP(j), per $j=1, \dots, NNX$, indica el nombre de files de la clapa j que estan realment ocupades per genotips. $OCUP(j) \leq NINX(j)$. $NINX(j) - OCUP(j)$ indica el nombre de files lliures.

$\sum_{j=1}^{NNX} OCUP(j) = N$. Als subprogrames SGEN sempre es fa que per cada clapa les OCUP(j) files ocupades siguin les primeres, des de PRIMER(j) a PRIMER(j)+OCUP(j)-1 i les NINX(j)-OCUP(j) files lliures les darreres, des de PRIMER(j)+OCUP(j) a PRIMER(j+1)-1.

El següent esquema, per NNX=2, NINX(1)=7, NINX(2)=5, OCUP(1)=4, OCUP(2)=4, PRIMER(1)=1, PRIMER(2)=8, il·lustra gràficament les relacions entre aquestes variables:



$$NMAX = 12 = NINX(1) + NINX(2)$$

$$N = 8 = OCUP(1) + OCUP(2)$$

Aquesta manera de representar les poblacions i el medi que ocupen, així com la subdivisió d'aquest medi, permet definir models de l'estil del de Dickinson i Antonovics (1973b) per un locus amb dos al·lels, però més generals ($L \geq 1$ loci, inversions etc.)

A.5.2. Descripcions dels subprogrames

A continuació es descriuen els subprogrames, adjuntem també el llistat en PL/I i un diagrama de flux en aquells casos on pugui ajudar a comprendre la lògica.

A.5.2.1. Subprograma de control CONTROL.

Només hi ha una versió d'aquest programa, amb el nom de mòdul CONTROL2.

Es tracta d'un bloc PL/I principal (MAIN) encarregat de controlar la seqüència d'operacions principals o esdeveniments a realitzar en una simulació, concretament:

a) Lectura de dades, paràmetres que caracteritzaran al procés de simulació, tant els usuals del sistema com els definits per l'usuari.

b) Processos fonamentals en el canvi de les freqüències gèniques o cromosòmiques: migració, mutació, selecció, canvis estructurals als cromosomes, deriva (implícita en el tipus de model).

c) Descripció de l'estat de la població.

d) Enregistrament en memòria auxiliar de l'estat de la població, per si és necessari cancel·lar el procés i recomençar-lo posteriorment.

Les variables definides al subprograma són:

Variables escalars "caràcter" (alfabètiques, literals).

NOM tira de 80 caràcters per identificar el procés en els llistats del programa.

Aquesta variable s'inicialitza en tota execució de CONTROL, amb lectura del seu valor a partir de fitxa perforada. Es sempre la primera dada demanada pel subprograma.

Variables escalars enteres de 15 dígit binaris.

NMAX Grandària màxima de la població simulada

L Nombre de loci estudiats.

NST Nivell d'estructura cromosòmica. NST=0 no es pot representar cap mena de mutació estructural. Si NST=1 es poden representar inversions produïdes sobre una estructura estàndard. Si NST=2 es poden representar inversions produïdes sobre estàndard i imbricades o ensolapades amb les primeres. I així successivament

NNX	Nombre de compartiments en què està dividit l'espai ocupat per la població.
MINZ	Nombre mínim de zigots produïts en cada fecundació.
MAXZ	Idem, nombre màxim.
MINQ	Nombre mínim de quiasmes en la producció d'un gamet.
MAXQ	Idem, nombre màxim.
NFREC	$(0.5)^{NFREC}$ és la freqüència d'al·lels 1 en la població inicial generada aleatòriament (subprograma GENERA). Si NFREC=0 no es genera aleatòriament i la població inicial es llegeix en fitxa.
IGEN	Número de la generació inicial. Si IGEN=0 s'inicia el procés (lectura en fitxa de llavors aleatòries, grandària i grau d'ocupació de les clapes, nom dels loci, generació o lectura en fitxa de la població inicial). Si IGEN > 0 es recupera l'estat del procés d'algun dispositiu de memòria auxiliar (i en particular es recupera el valor de IGEN quan es va interrompre, que prevaldrà sobretot altre valor).
NGEN	Nombre de generacions que ha de durar el procés (evidentment, si no es cancel·la abans d'acabar).
IGRA	Periodicitat de l'enregistrament de l'estat del procés (cada quantes generacions).
IDES	Idem, de la descripció.
NCOD	Nombre màxim de caràcters emprats per anomenar un locus. (P.e. si s'anomenen A,B,C,... aleshores NCOD=1 si ADH, PGM-1, ... aleshores NCOD=5).

Totes aquestes variables, i en aquest ordre, sempre s'inicialitzen a partir de dades presentades en fitxa. Es llegeixen després de NOM. Altres variables enteres, internes del programa són:

I,IND,INX	Indexos en diverses generacions repetitives.
N	Nombre d'adults presents una generació concreta. ($N \geq NMAX$)
NB	Nombre de bits de memòria ocupats per un cromosoma. $NB=Lx(NST+1)$ arrodonit perquè sigui múltiple de 8.

Variables escalars lògiques (bit)

BMGR	Si val 1 ("SI" o "CERT") hi ha migració des de l' <u>ex</u> terior (subrutina MIGRA).
BMUT	Si val 1 hi ha mutació al·lèlica (Subr. MUTAC).
BINV	Si val 1 es produeixen noves inversions (Subr. INVER)
BDES	Si val 1 es descriu periòdicament l'estat de la població. (Subr. DESCR).
BDSC	Si val 1 s'enregistra periòdicament l'estat del procés (Subr. GRABA).
BCONTROL	Tira de 16 bits sobre la qual (les 5 primeres posicions) en el mateix ordre indicat, estan definides les anteriors. Conté altres paràmetres que s'expliquen en les subrutines on actuen. Es llegeix (en execució de CONTROL) de fitxa perforada i a continuació de les variables enteres escalars.

Variables vectorials "caràcter" (la dimensió va entre parèntesis a continuació del nom de la variable).

ABC(L)	Vector de caràcters, tots de llargada NCOD. Identificacions dels loci. Es llegeix de fitxa, després de LLAVOR, NINX i OCUP, si IGEN=0.
SI (0:1)	Conté els caràcters 'NO' i 'SI', SI(0)='NO', SI(1)='SI'.

Variables vectoriales enteres de 15 bits

(La dimensió és entre parèntesis a continuació del nom de la variable):

LLAVOR(15) (externa) LLAVOR(1) és IGEN
 LLAVOR(j), j=2,3,...,15 són les llavors
 aleatòries, separades pels diferents tipus
 de tries aleatòries a realitzar, per asse-
 gurar en part la comparabilitat entre dos
 processos de simulació en els quals s'hagi
 canviat alguna cosa (*):
 LLAVOR(2) elecció del lloc d'aparellament
 LLAVOR(3) elecció dels progenitors
 LLAVOR(4) elecció del lloc d'oviposició
 LLAVOR(5) nombre de quiasmes
 LLAVOR(6) lloc dels quiasmes
 LLAVOR(7) nombre d'inversions
 LLAVOR(8) lloc de les inversions
 LLAVOR(9) generació de la població inicial
 LLAVOR(10) nombre de zigots produïts en
 un aparellament.
 LLAVOR(11) migració externa
 LLAVOR(12) mutació al·lèlica
 LLAVOR(13) competència larvària
 (LLAVOR(14,15) lliures)

NINX(NNX) NINX(i), i=1,...,NNX, és la grandària (nom-
 bre d'individus que hi caben) del comparti-
 ment del medi. $NMAX = \sum_{i=1}^{NNX} NINX(i)$

OCUP(NNX) OCUP(i) és el grau d'ocupació del comparti-
 ment i. $OCUP(i) \leq NINX(i)$, $N = \sum_{i=1}^{NNX} OCUP(i)$.

Si IGEN=0, aquests tres vectors s'inicialitzen a partir de
 dades presentades en fitxa, en el mateix ordre. Si IGEN > 0 es
 recuperen d'un dispositiu de memòria auxiliar.

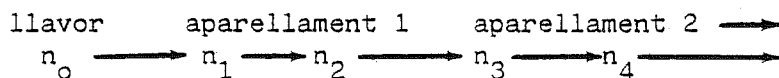
PRIMER(NNX) PRIMER(i) és la posició dins POP (vegeu
 més endavant) on comença el compartiment
 i.

(*) Els generadors congruencials de nombres pseudoaleatoris (com
 la funció RAN) utilitzen el darrer nombre generat com a llavor
 del proper, en un procés totalment determinista. Si, per exem-
 ple, en una simulació l'únic esdeveniment aleatori que hi ha
 és l'elecció de parella per a la reproducció, les successives
 eleccions implicaran altres tantes generacions de nombre pseudoa-
 leatoris n_i :

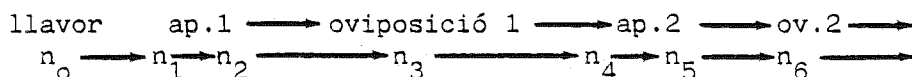
Variabls matriciales bit

POP(NMAX,2) Matriu de NMAX files i 2 columnes on cada element és una tira de NB bits de memòria, que representa un gamet. Cada fila representa un individu (o més correctament, un genotip) si aquell espai està ocupat. Si IGEN=0 i NFREC=0 es llegeix de fitxa perforada després de ABC. Si IGEN=0 i NFREC > 0 es genera aleatòriament (subrutina GENERA). Si IGEN > 0 es recupera d'un dispositiu de memòria auxiliar (subrutina RECUP).

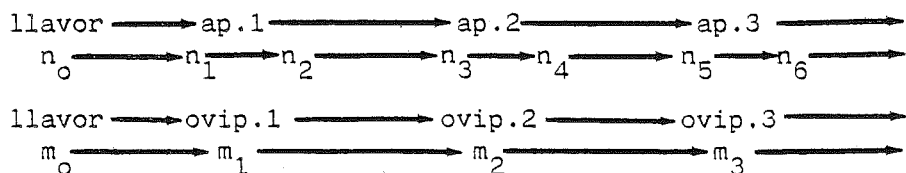
Cada gamet ocupa $NB=Lx(NST+1)$ bits. Els L primers codifiquen els gens estudiats, sempre en el mateix ordre. Els $L*NST$ restants bits codifiquen l'estructura cromosòmica per inversions, segons les especificacions de l'apartat anterior.



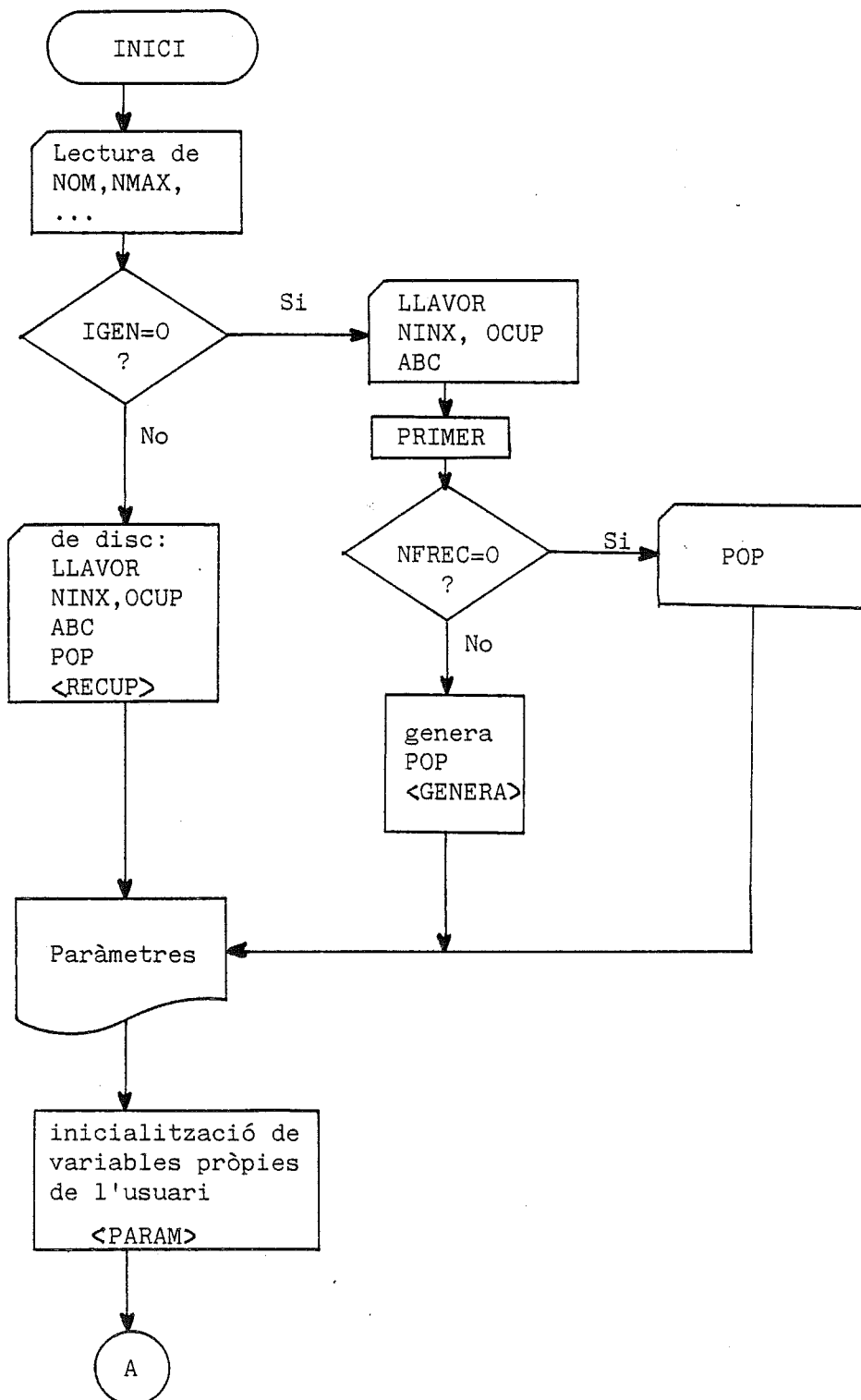
si, iniciant el procés amb la mateixa llavor, n_0 , en una nova simulació s'inclou una tria aleatòria de lloc on òvipositar, amb cada aparellament seguit de tal tria, com la seqüència de nombres aleatoris serà la mateixa, els aparellaments produïts seran diferents als de la primera simulació ja que els corresponderan uns altres nombres aleatoris:

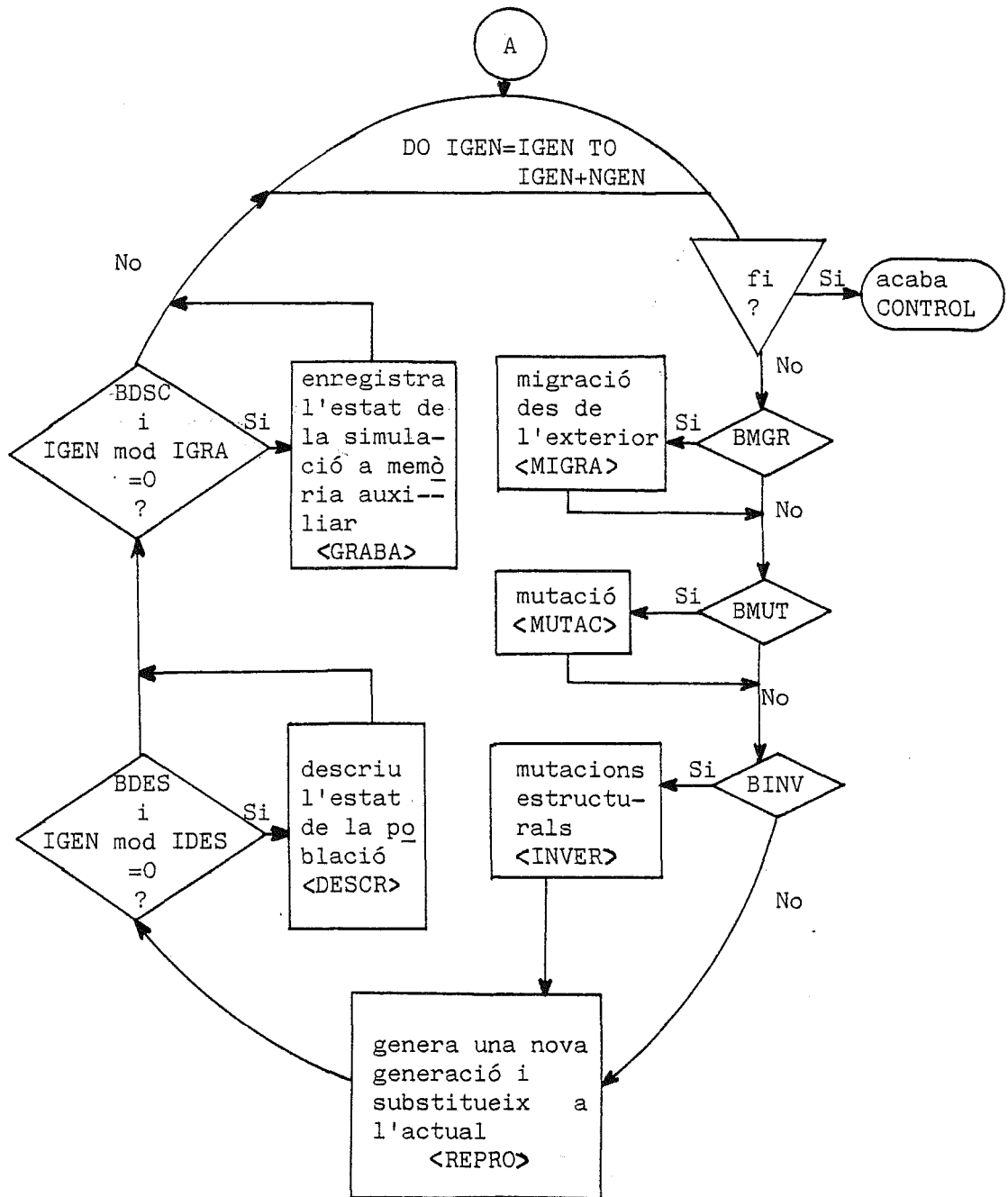


Això s'elimina en part si hi ha dues seqüències independents de nombres aleatoris, una per cada tipus de tria:



sempre que allò que passi a una seqüència no impliqui un atur o una major producció de nombres aleatoris a l'altra.

DIAGRAMA DE FLUX DE CONTROL



A.5.2.2. Subrutina GENERA.

Està representada pel mòdul GENERAO2.

Genera una població inicial a l'atzar, amb unes freqüències gèniques especificades als paràmetres i amb estructura cromosòmica estàndard.

Paràmetres: POP, N, L, NB, NST, NINX, PRIMER, OCUP, NNX, NMAX, BCONTROL, NFREC

tots ells han estat descrits a la secció dedicada a CONTROL (A.5.2.1). N, NST, NINX, NMAX i BCONTROL no s'utilitzen en aquest mòdul.

Altres variables:

LLAVOR(15) Ja descrita a A.5.2.1. L'element mové LLAVOR(9), definit amb el nom LLGENR, és la llavor aleatòria utilitzada per generar la població.

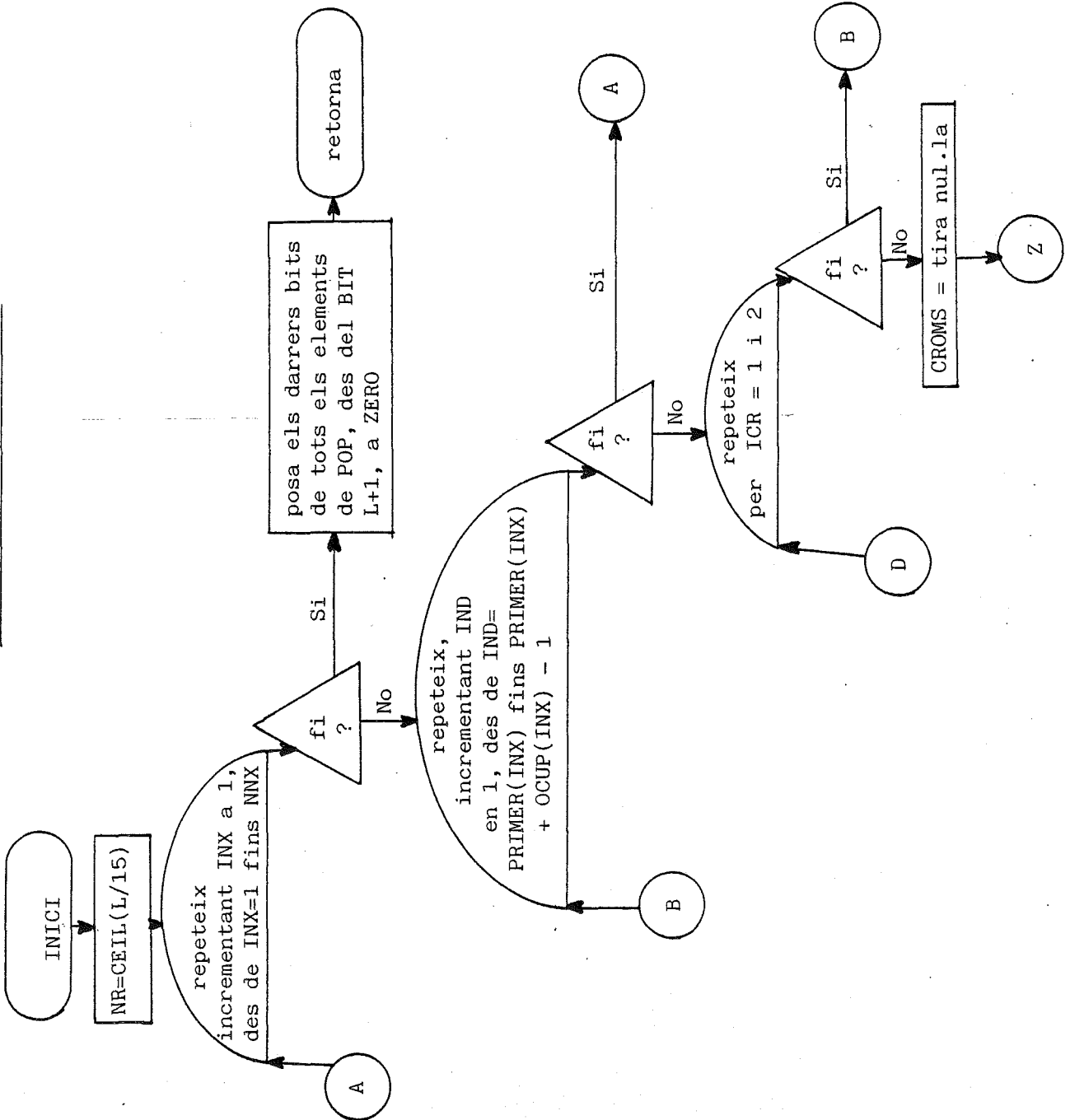
CROMS Tira de L bits, variable auxiliar per generar un cromosoma.

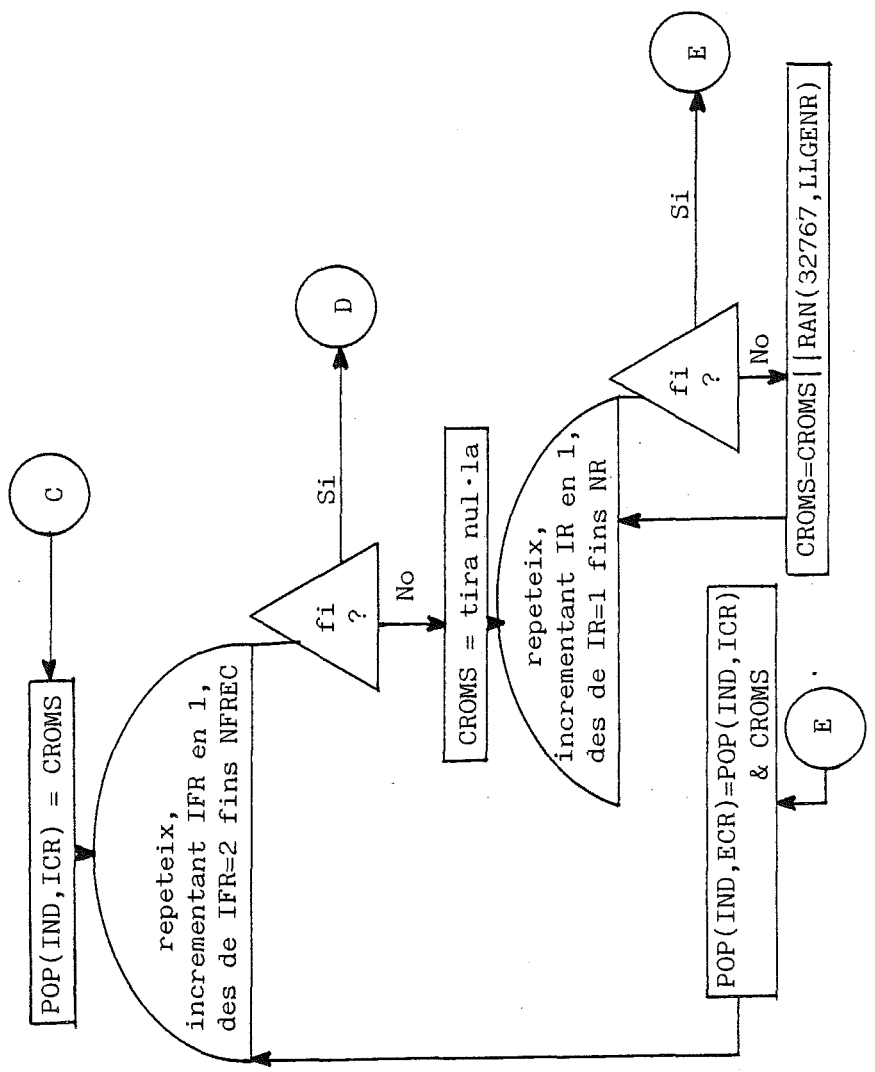
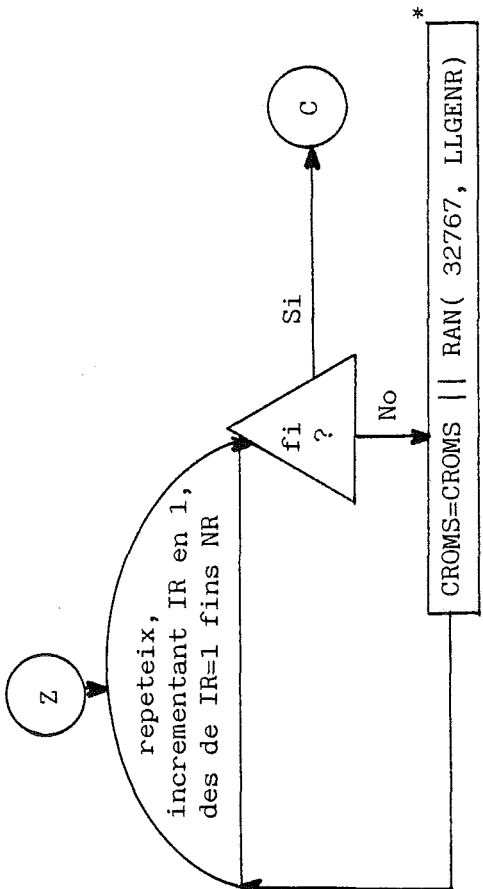
B Tira de NB-L bits posats a zero, s'assigna a l'estructura cromosòmica.

NR Variable entera definida com $NR=CEIL(L/15)$, essent $CEIL(x) = \min \{n \in \mathbb{Z} \mid n \geq x\}$. Indica el nombre de nombres aleatoris de 15 dígit binaris que cal generar per construir un cromosoma.

ICR, IFR, IND, INX, IR Indexos d'operacions repetitives.

La probabilitat que un locus presenti l'al·lel 1 es fa igual a $(0.5)^{NFREC}$ per $NFREC=1,2,\dots$ i 0.5 per $NFREC=0$.





A.5.2.3. Subrutina PARAM

En principi és lògic que la defineixi l'usuari ja que s'encarrega de les lectures de dades i inicialitzacions pròpies de les variables que ell hagi definit als seus subprogrames (p.e. els encarregats d'avaluar l'eficàcia segons alguna fórmula concreta). Totes aquestes variables han de tenir l'atribut EXTERNAL i estar definides als subprogrames concrets de l'usuari on es facin els càlculs que les utilitzin.

Ha de portar els paràmetres POP, N, L, NB, NST, NINX, PRIMER, OCUP, NNX, NMAX, BCONTROL ja descrits a A.5.2.1.

A la biblioteca hi ha el mòdul PARAMOO2 utilitzat als programes emprats en les simulacions descrites en aquesta memòria.

PARAMOO2 llegeix de fitxa perforada les següents variables:

EFER2	mitjana de la distribució de la variable aleatòria "nombre de zigots produïts en un aparellament".
SFER2	desviació típica de la mateixa variable. EFER2 i SFER2 també poden ser dos paràmetres qualsevol dels quals depengui aquesta distribució. Si no calen, per exemple en el cas de nombre constant de zigots, es poden deixar en blanc a la fitxa.
EQUI2	Mitjana (o un altre paràmetre) "nombre d'esdeveniments de recombinació".
SQUI2	desviació típica (o un altre paràmetre) de la mateixa variable. Si no calen es deixen en blanc.

NZIG2 nombre de zigots quan aquest és constant.

NINV2 nombre d'inversions a cada crida de NBRI quan aquest és constant.

NREP nombre total d'aparellaments produïts a la generació de la nova generació. Aquesta constant es fa servir a les funcions SIREP i FIREP per saber si cal finalitzar els esdeveniments de producció de nous individus.

Totes elles es llegeixen en format (COL(1), 4F(8,4),3F(4)).

Si les següents posicions de la variable BIT BCONTROL, llegida a CONTROL, valen:

1) Posició 6 val 1 (bit BAPR, hi ha selecció per diferències a l'habitat per aparellar-se) s'allotgen i es llegeixen els vectors LOCW de dimensió 2, i IOPW de dimensió NNX (nombre de nínxols). LOCW(1) indica el primer locus codificador del fenotip relacionat amb l'avaluació d'aquesta capacitat i LOCW(2) el darrer. IOPW(j) indica el fenotip òptim quant a obtenir màxima probabilitat d'aparellar-se al nínxol j, j=1,...,NNX.

2) Posició 8 val 0 (bit OVIN hi ha elecció aleatòria de la subdivisió ambiental on va a parar un nou individu) s'allotgen i llegeixen LOCO de dimensió 2 i IOPO de dimensió NNX, amb el mateix significat que abans.

3) Posició 9 val 1 (bit BFER, hi ha selecció per diferències a la fecunditat). Es llegeixen i allotgen LOCF i IOPF amb el significat habitual.

4) Posició 15 val 1 (bit BCLV, hi ha competició entre larves o nous individus). Es llegeixen i allotgen LOCL i IOPL. LOCL(1) indica el primer locus que codifica el fenotip relacionat amb la competició larvària i LOCL(2) el darrer. IOPL(j) l'òptim al nínxol j quant a la competició.

Tots aquests vectors es llegeixen en format

COL(1), 2F(4),
COL(1), (NNX) F(5)

A.5.2.4. Subrutina RECUP

S'encarrega de recuperar la informació necessària, guardada en disc, per recomençar un procés interromput.

A la biblioteca hi ha la versió catalogada amb el nom RECUP002. Els seus paràmetres són: POP, N, L, NB, NST, NINX, PRIMER, OCUP, NNX, NMAX, ABC, NCOD, tots descrits a A.5.2.1.

Altres variables són:

LLAVOR(15)	Ja descrita també a A.5.2.1.
ESTAT	Tira de 800 bits on es troba informació directament recuperada del disc, abans de passar-la a les diferents variables del programa.
INICI	Fitxer de lectura desde disc.
I,K	Variabls enteres utilitzades en algunes iteracions.
NSP	Nombre de bits ocupats per cada individu.

NPOS Darrera posició de ESTAT on hi pot cabre una
fila sencera de POP.

Si se li proporcionen les fitxes de control adequades, definint l'àrea de disc on ha de buscar, retorna POP, NINX, PRIMER OCUP, ABC i LLAVOR tal com estaven a l'enregistrament efectuat just abans de la interrupció. Cal que totes aquestes matrius i vector estiguin allotjats i L, NB, NST, NNX i NMAX inicialitzats.

A.5.2.5. Subrutina GRABA

Enregistra l'estat del procés en disc.

A la biblioteca hi ha el mòdul catalogat amb el nom GRABAOO2.

Els paràmetres i les variables del programa són les mateixes que a RECUP, excepte que el fitxer INICI és substituït pel fitxer de sortida GRAB. La lògica del programa també és la mateixa que a A.5.2.4.

Em magatzema en disc els valors actuals de les variables POP, NINX, PRIMER, OCUP, ABC i LLAVOR.

Escalars reals

SHI Quantitat d'informació de Shanon, segons les freqüències cariotípiques ($NSTC_1, \dots, NSTC_{NGT}$)

$$SHI = \log_2 N - \sum_{i=1}^{NGT} NSTC_i \log_2 NSTC_i$$
Escalars bit(1) (lògiques)

B200 Val 1 quan s'han arribat a comptar 200 cariotips diferents (*).

CR1, CR2 Tira de $L \cdot NST$ bits, variables auxiliars que contenen l'estructura d'un cromosoma.

Vectorials enteres

NSTC(200) Vector de freqüències cariotípiques.

Matricials bit

STCR(200,2) Matriu de 200 files i 2 columnes, on cada element és una tira de $L \cdot NST$ bits. Conté la codificació dels cariotips diferents trobats a la població.

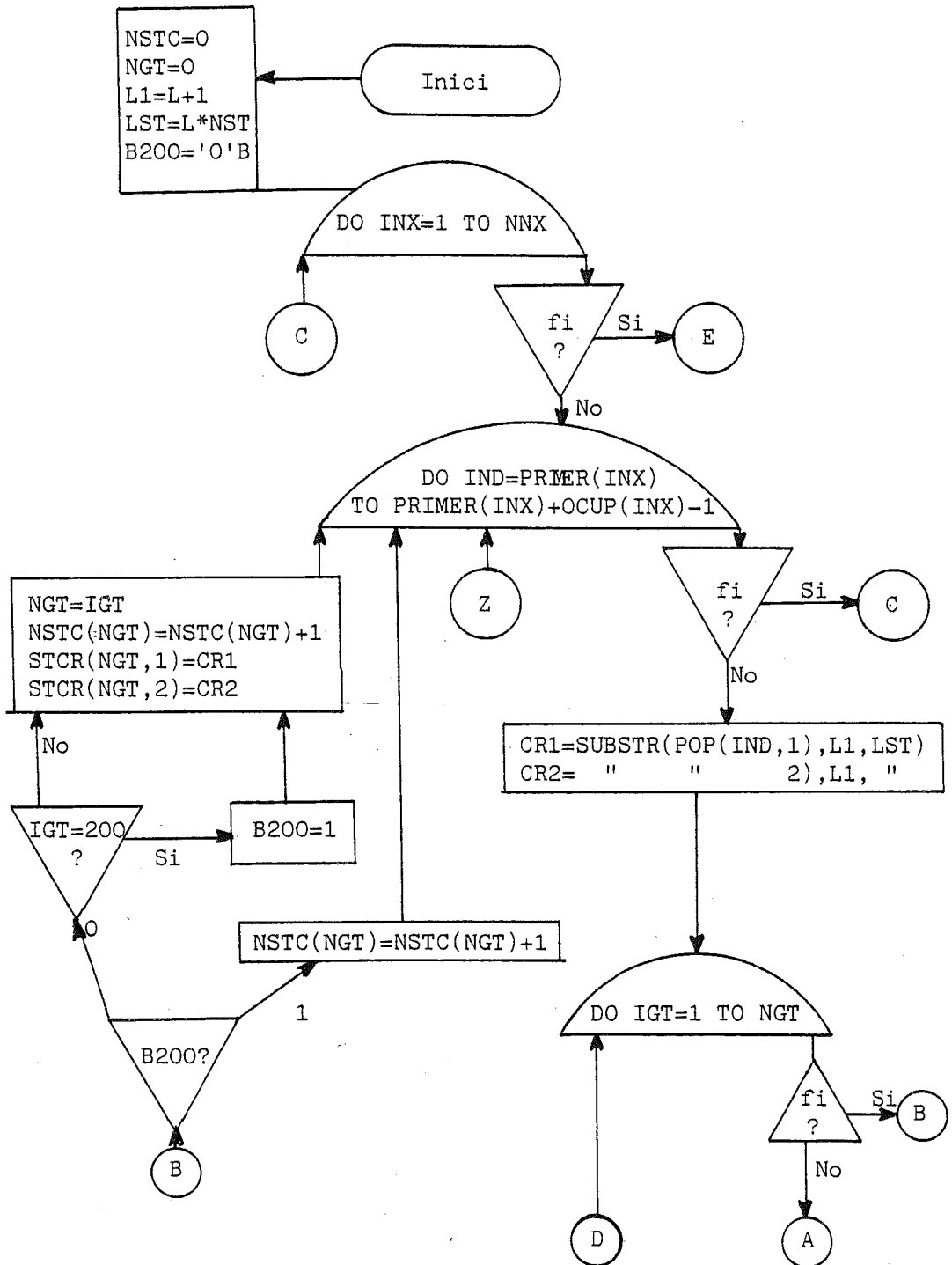
BGEN(1,2) BIT(1) Variable "fantasma" necessària per al funcionament correcte de DECOD.

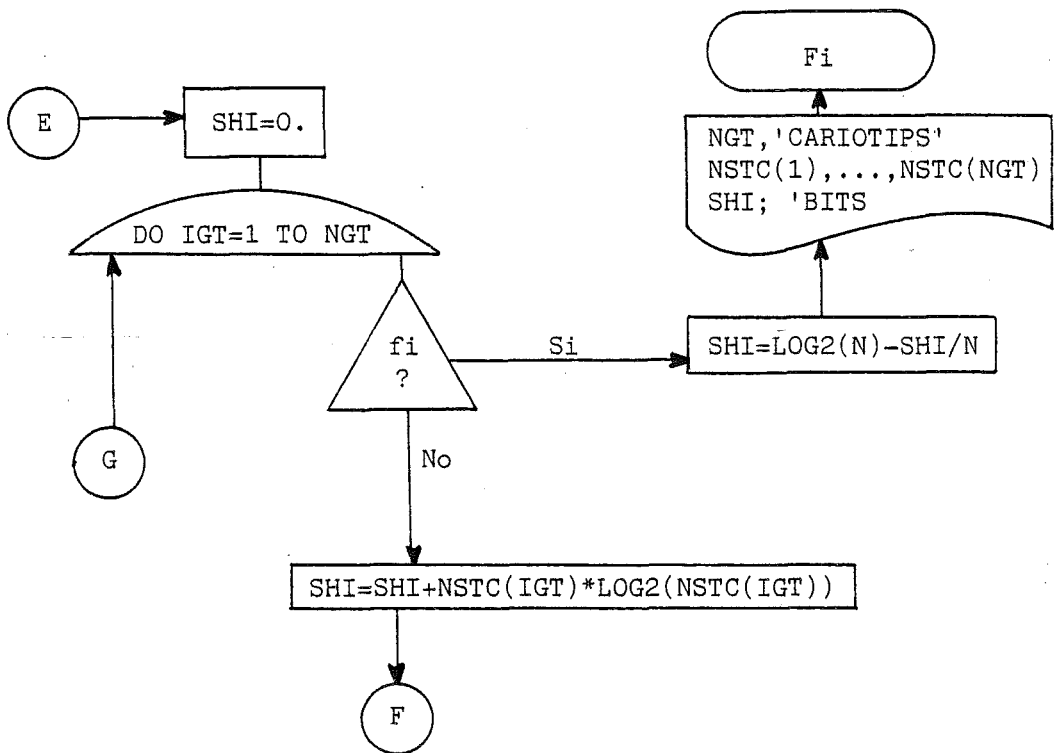
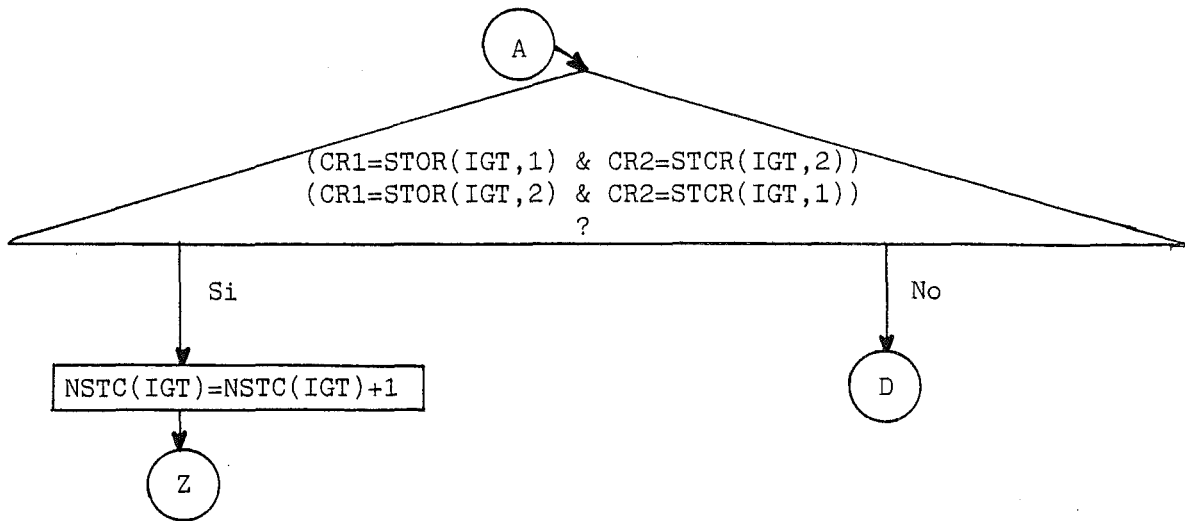
Tires de caràcters

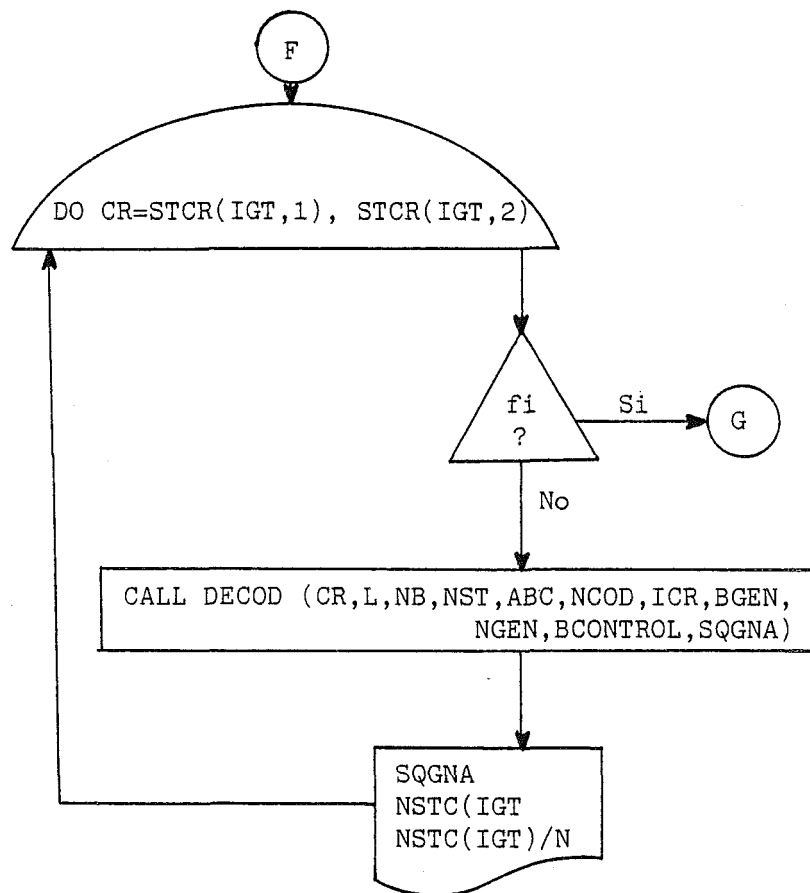
SQGNA CHAR(100) Conté la seqüència de loci d'un tipus cromosòmic i la genealogia per inversions. (Vegeu la descripció de DESCRTOT)

(*) Aleshores es compta la freqüència a una darrera classe (la 200) on s'inclouen junts tots els cariotips que no són els 199 primers.

DIAGRAMA DE FLUX DE DESCROO2







Mòdul DESCRTOT

Els paràmetres són els mateixos que a DESCROO2 (com a totes les versions d'una mateixa subrutina).

Dins de cada subdivisió compta les freqüències de cada cariotip, diferents genotips presents en aquell cariotip*, i les seves freqüències*, dóna la seqüència de loci present a cada tipus de cromosoma i les inversions que han portat de l'estructura estàndard a aquella seqüència. Finalment calcula la mitjana i la variància del nombre de loci heterozigòtics, també dins de cada nínxol i en la població global.

Per computar les freqüències es procedeix de la següent manera: es defineix un vector de "pointers", (POIN) amb un pointer per cada individu. Al principi el pointer de cada individu assenyala al següent, excepte el del darrer individu que val 9999 (senyal d'acabament). El cariotip d'un individu, p.e. el primer, s'emmagatzema en les variables E1 i E2 i, seguint la seqüència dictada pels pointers, es recorre tota la població comparant E1,E2 amb el cariotip dels individus. Cada cop que n'hi ha un d'igual s'incrementa en una la freqüència d'aquell cariotip,

(*) Això si la posició 12 de BCONTROL val 1.

A.5.2.6. Subrutina DESCR

Calcula i impremeix el valor d'alguns paràmetres i altres informacions útils per descriure l'estat de la població en un moment donat.

Actualment hi ha dues versions, catalogades amb el nombs de mòduls DESCROO2 i DESCRTOT.

Mòdul DESCROO2

Es una versió reduïda. Compta les freqüències cariotípiques i calcula l'índex de diversitat de Shanon sobre aquestes.

Els seus paràmetres son POP, N, L, NB, NST, NINX, PRIMER, OCUP, NNX, NMAX, BCONTROL, IGEN, NOM, ABC, NCOD tots ja descrits a A.5.2.1. Cal que la prosició 12 de BCONTROL valgui '0'B ja que no s'investiguen els genotips (vegeu la descripció de DESCRTOT). Altres variables del programa són:

Escalars enteres

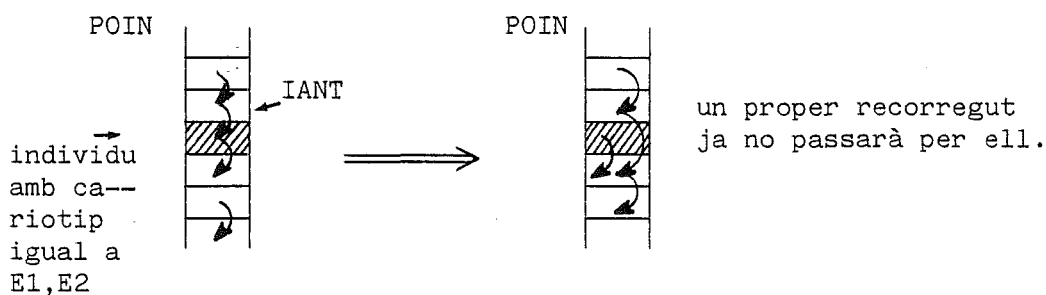
NGT En finalitzar l'execució conté el nombre de cariotips diferents presents ($NGT \leq 200$).

L1 L+1

LST L*NST

IGT, IND, INX Indexos.

s'enregistra el genotip de l'individu en una matriu de genotips associats a aquell cariotip, actualitzant també un vector de freqüències d'eixos genotips, i finalment al pointer de l'individu immediatament anterior d'entre els que no tenien la mateixa estructura que E1, E2 (posició IANT de POP) se li assigna el valor del pointer de l'individu investigat en aquell moment (de manera que un proper recorregut ja no passarà per aquest individu).



Quan s'acaba el recorregut de POP es recomença el mateix procés, assignant a E1, E2 el cariotip de l'individu IPRIM (variable on s'ha guardat l'adreça de la primera fila (individu) de POP amb cariotip diferent d'aquell assignat a E1, E2 a la busca anterior).

Això es repeteix fins que POIN(IPRIM) val 9999.

Les següents variables estan definides al programa:

Variables enteres de 15 dígitos binaris:

POIN(MAXOC) POIN(i) és l'individu que serà investigat després de (i).

MAXOC és igual al màxim de OCUP.
 INIC índex de l'individu el cariotip del qual s'assigna
 a E1, E2.
 IND índex de cada individu que es compara amb E1, E2.
 I1, I2 índex dels cromosomes de IND.
 IANT índex de l'individu investigat avans que IND.
 FST freqüència absoluta del cariotip E1, E2.
 NSTC nombre de cariotips diferents
 FGEN(100) freqüències del diferents genotips associats amb
 E1,E2.
 L1,LST ja descrites a DESCRO02.
 IPRIM és igual a PRIMER(j), j=1,...,NNX.
 ICR,ILOC,IGEN,INX,INX2 índexos d'operacions repetitives.

Variables lògiques

BLOCI BIT(1) posició 12 de BCONTROL. Si BLOCI='1'B("Si") es
 busquen i contenen les freqüències de tots els geno-
 tips associats amb cada cariotip.
 BHOM Val '1' B si IND és homocariotip.
 B100 Val '1' B si ja s'han trobat 100 genotips diferents
 associats amb un cariotip.
 B1 Val '1' B si encara no s'ha trobat un individu amb
 cariotip diferent de E1, E2.

Tires de bits

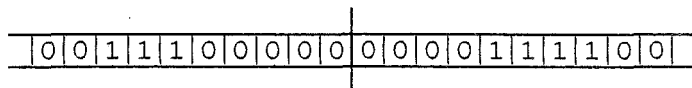
E1,E2 BIT(LST) codificació de l'estructura dels cromosomes 1
 i 2 de IND.
 EA BIT(LST) Variable auxiliar, que pot valer E1 o E2.
 G1,G2 BIT(L) genotip de IND.

BGEN(100,2) BIT(L) matriu, de 100 files i 2 columnes, cada fila representa un genotip, dels associats amb E1,E2. La freqüència del genotip BGEN(j,*) es troba a FGEN(j).

Tires de caràcters

SQGNA CHAR(100) conté la seqüència de loci d'un tipus cromosòmic i la genealogia.

Si un cromosoma amb $L=10$; $ABC=A,B,C,D,E,F,G,H,I,J$;
 $NST=2(i \text{ LST}=20)$, estés codificat com



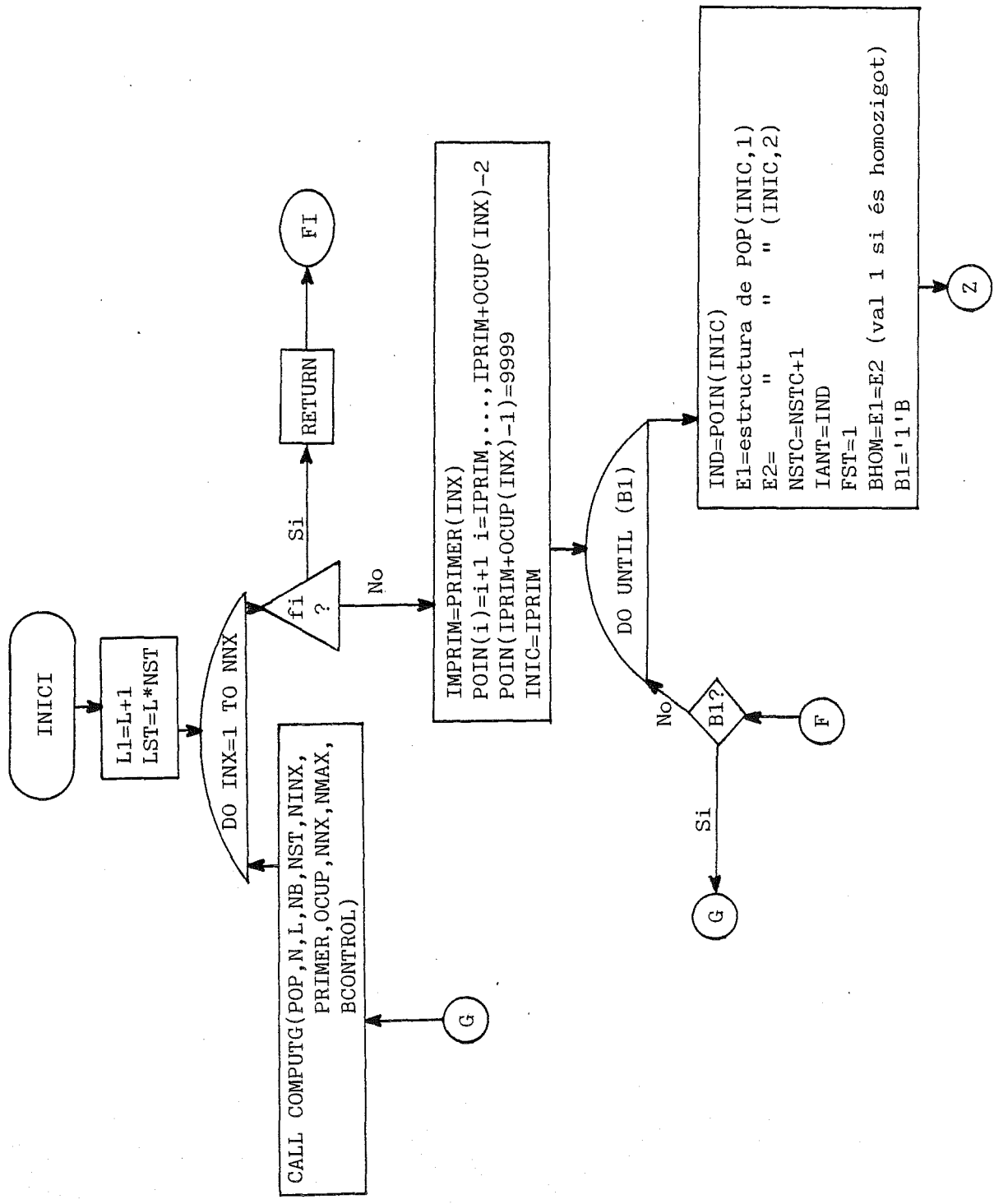
és-a-dir, que sobre estàndard (tot zeros) s'hagués produït una inversió que afectés del loci 2 al 5 i sobre l'estructura resultant una altra afectant del 4 al 8, SQGNA seria (tal com el retorna la subrutina DECOD):

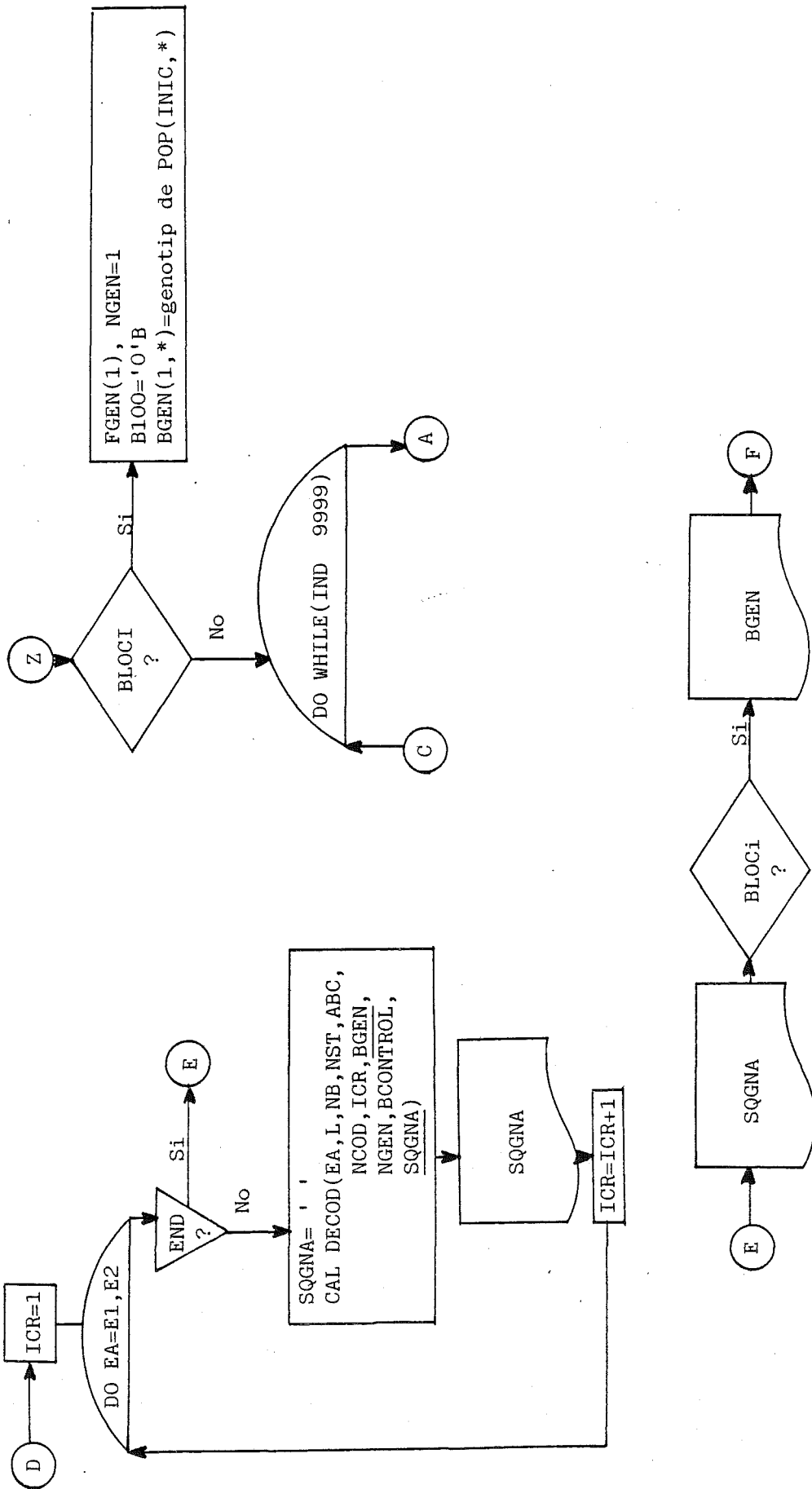
73 blancs

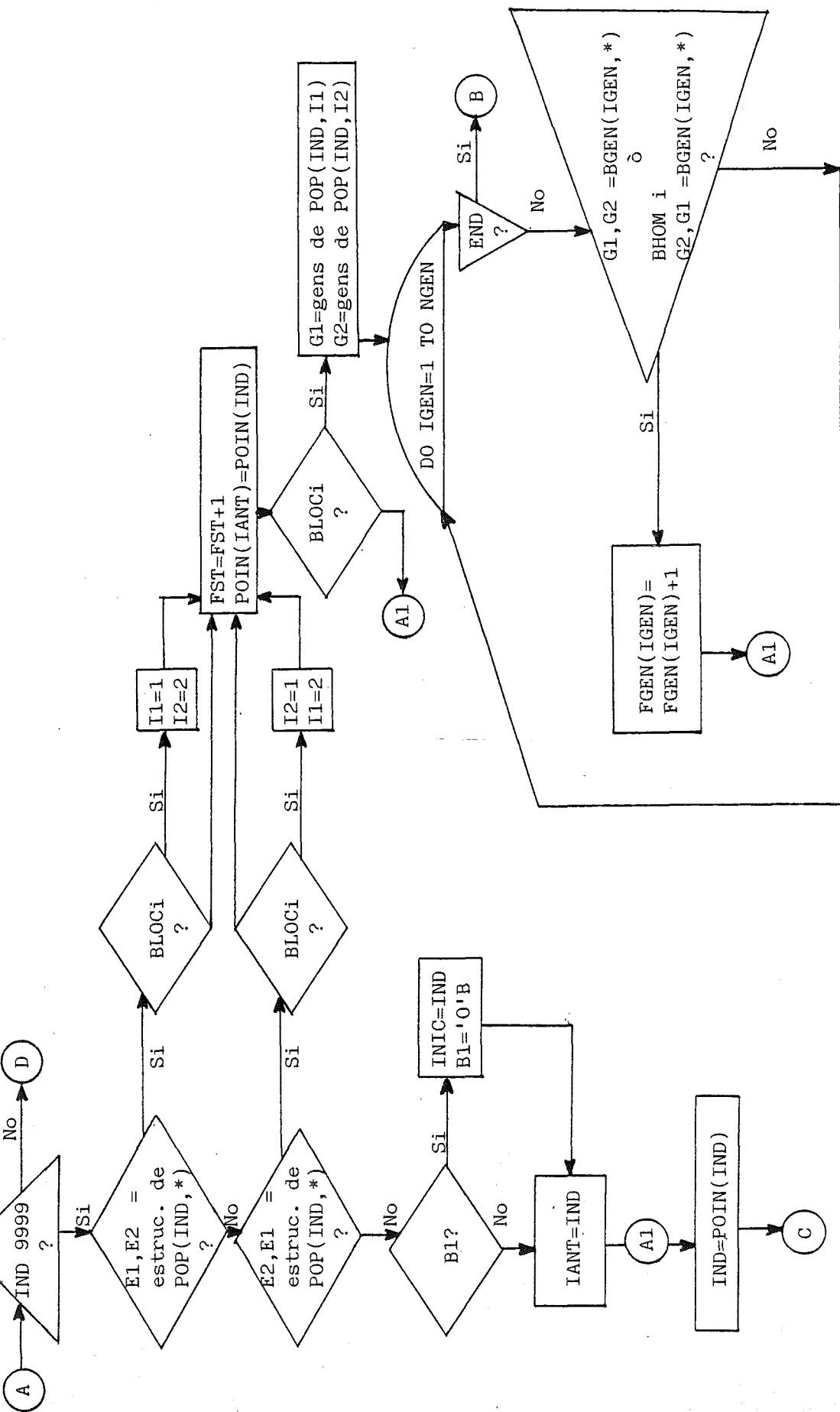
AEDHGFBCIJØ(Ø2,Ø5)ØØ(Ø4,Ø8)ØØ

indicant Ø el caràcter blanc (absència d'impressió).

DIAGRAMA DE FLUX DE DESCRTOT







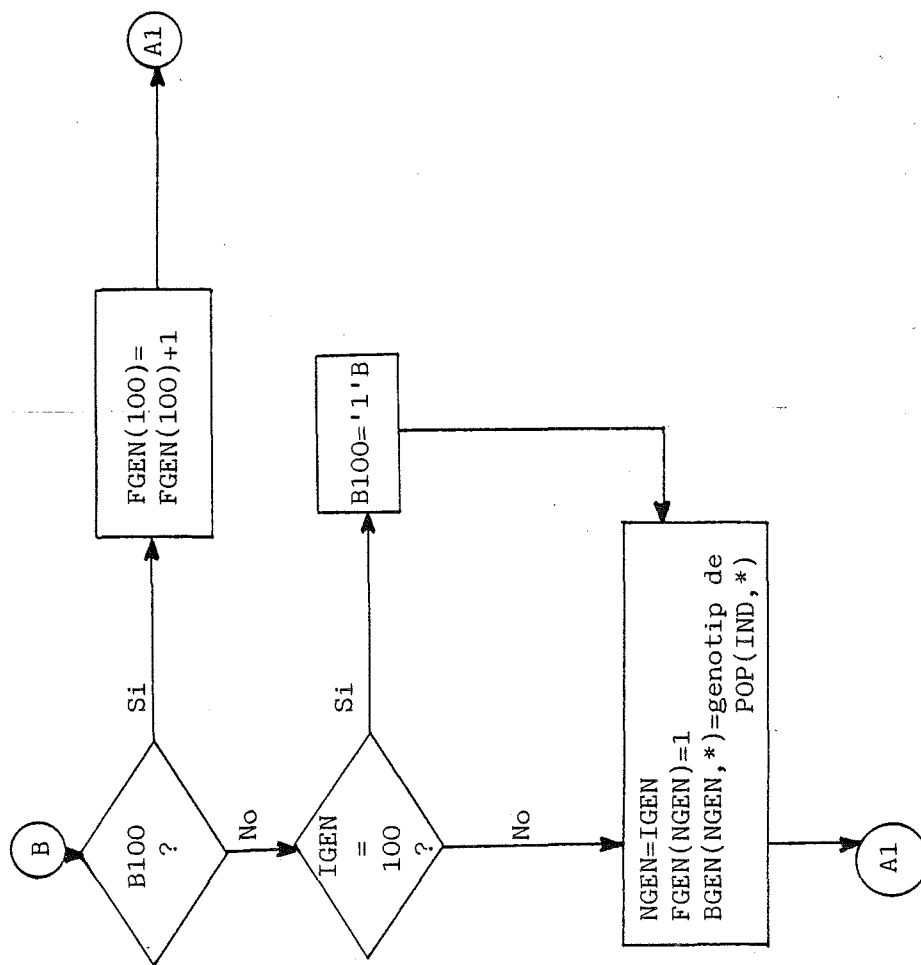
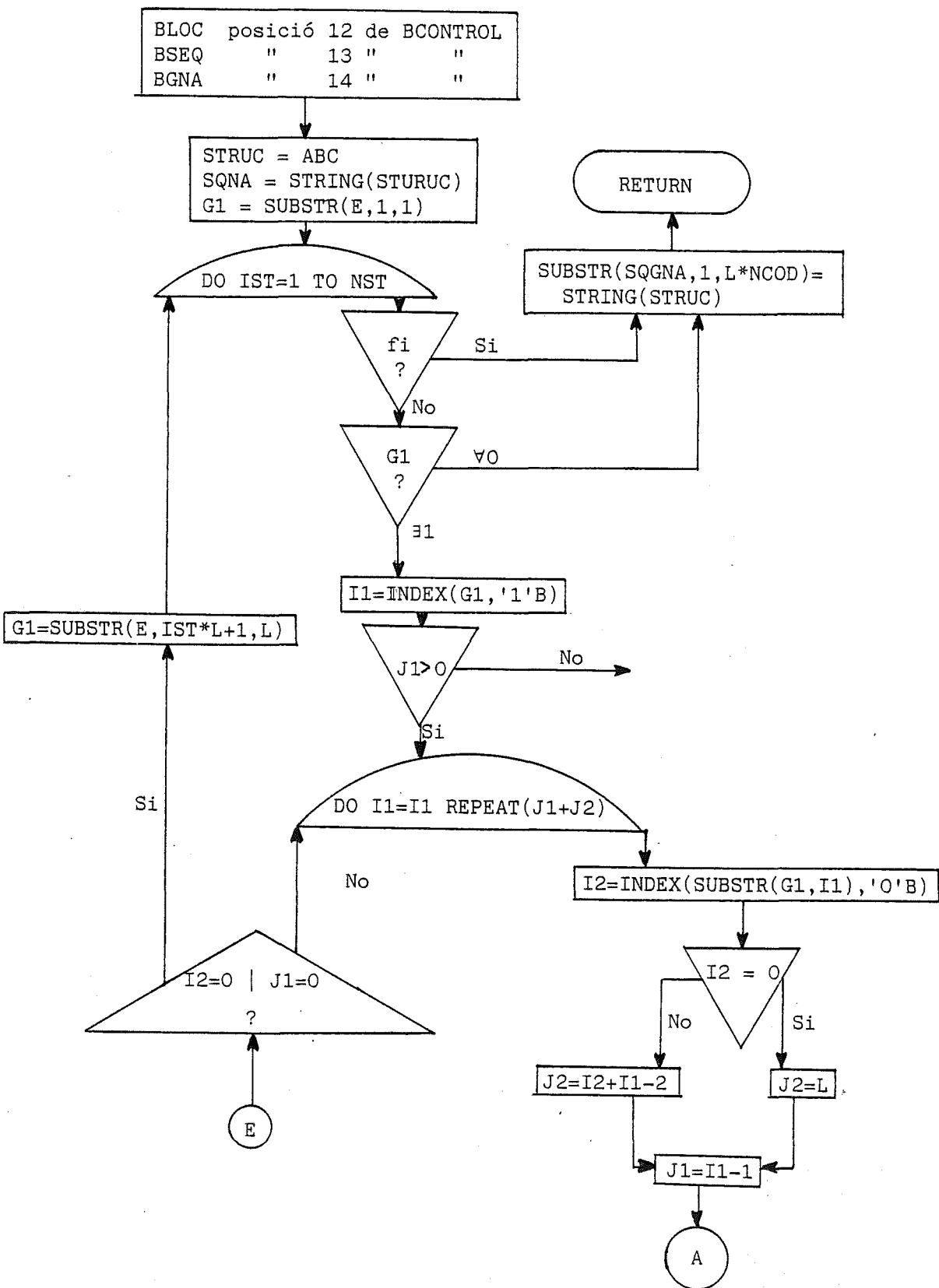
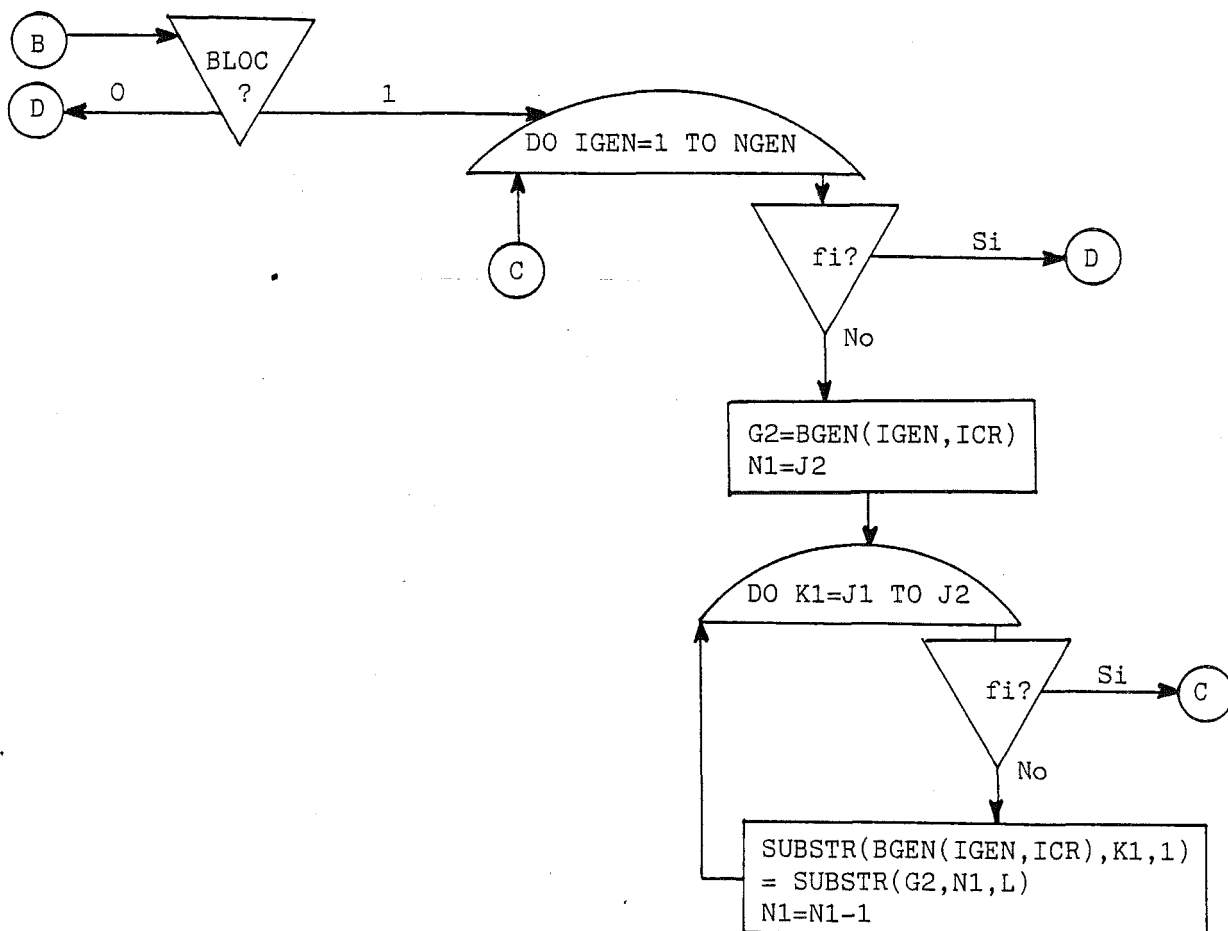
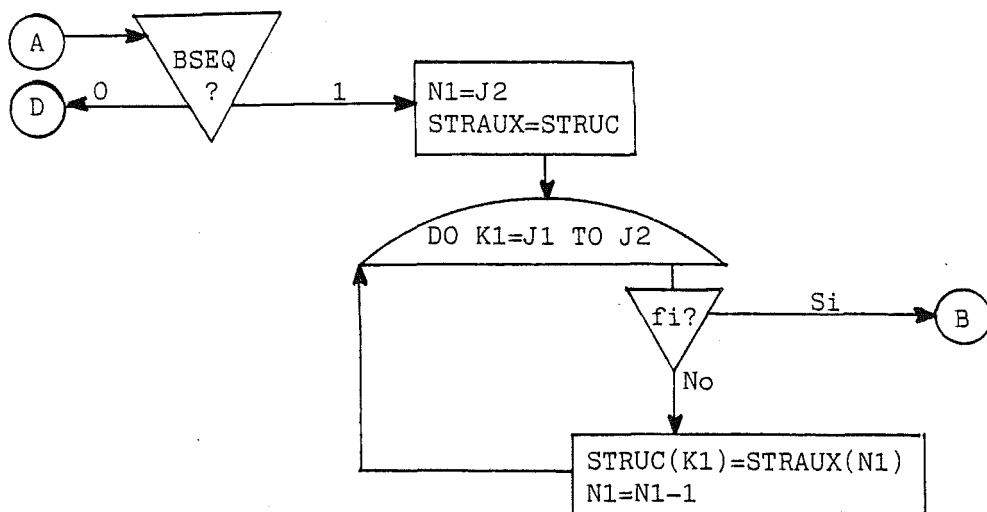
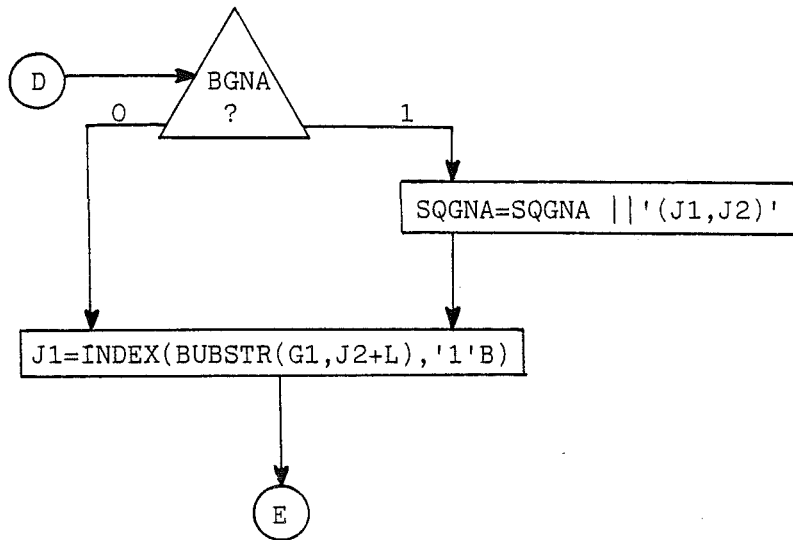


DIAGRAMA DE FLUX DE DECODTOT







A.5.2.7. Subrutina DECOD

Decodifica l'estructura d'un gamet.

Hi ha la versió DECODTOT de paràmetres E,L,NB,NST,ABC,NCOD,ICR,BGEN,NGEN,BCONTROL,SQNA. Tots aquests paràmetres han estat descrits a l'apartat anterior. Donada la codificació d'un gamet, continguda a la tira E de NST*L bits, retorna una tira de caràcters SQNA que conté la seqüència de loci, segons els noms d'aquests especificats a ABC, i la genealogia d'inversions des d'estàndard a l'estructura codificada a E. La forma concreta d'aquesta tira de caràcters s'indica a A.5.2.6.

A.5.2.8. Subrutina COMPUTG2

Calcula estadístics que caracteritzen l'estat de la població a una generació donada.

Hi ha la versió COMPUTG2, orientada a descriure els resultats segons el model de la secció 6, que calcula i imprimeix els següents valors:

- 1) Nombre mitjà de loci heterozigòtics per individu.
- 2) Variància del nombre de loci heterozigòtics.
- 3) Valor additiu mitjà
- 4) Variància del valor additiu
- 5) Llast genètic mitjà (llast calculat con allunyament a l'òptim)

- 6) Variància del llast genètic.
- 7) Correlació entre valors additius de dues zones diferents del genotip (corresponents normalment a la que codifica el caràcter mètric estudiat i a la que codifica les probabilitats d'oviposició a cada clapa).
- 8) Probabilitat d'oviposició a cada clapa (segons les fórmules del model de la secció 6).
- 9) Freqüències gèniques.
- 10) Probabilitat d'heterozigosi a cada locus.
- 11) Matriu de desequilibris el lligament i matriu de correlacions entre loci.

Els paràmetres de COMPUTG són POP,N,L,NB,NST,NINX,PRIMER, OCUP,NNX,NMAX,BCONTROL.

A.5.2.9. Subrutina INVER

Simula la producció d'inversions o d'altres mutacions cromosòmiques a POP. Només hi ha la versió catalogada al mòdul INVER002. Els paràmetres són: POP,N,L,NB,NST,NINX,PRIMER,OCUP, NNX,NMAX,BCONTROL, tots ja descrits.

Descripció d'INVER002

Simula la producció d'inversions a POP. Des d'INVER es crida la subrutina NBRI que retorna (NINV) el nombre d'inversions produïdes. Posteriorment, es repeteix NINV vegades el següent: es crida la subrutina POSI -això dona l'individu (IND) on s'ha produït cada inversió, el cromosoma (ICR), el locus on comença (IN) i la llargada (LON)- modificant a continuació l'estructura cromosòmica a aquesta posició específica.

Variables

LST	L*NST
NINV	nombre d'inversions produïdes.
IND	individu on s'ha produït una inversió.
ICR	cromosoma (1 o 2) de l'individu anterior.
IN	locus de la primera trencadura.
LON	nombre de loci implicats
IST,JINV	indexos d'operacions repetitives.

Crida les subrutines NBRI, amb arguments POP, ..., NMAX, NINV, que retorna el nombre d'inversions NINV, i POSI, amb arguments POP, ..., NMAX, IND, ICR, IN, LON, que retorna IND, ICR, IN i LON.

A.5.2.10. Subrutina NBRI

Retorna el nombre d'inversions produïdes. Paràmetres: POP, N, L, NB, NST, NINX, PRIMER, OCUP, NNX, NMAX, NINV el darrer és el nombre d'inversions produïdes, els altres ja han estat descrits.

Només hi ha la versió catalogada amb el nom NBRI0002 que retorna un valor constant, llegit a la subrutina PARAM i amagatzemat a la variable externa NINV2 que s'assigna a NINV.

A.5.2.11. Subrutina POSI

Retorna l'individu (IND) on s'ha produït una inversió, el cromosoma (ICR), el primer locus (INC) i el nombre de loci im-

- APSF BIT(1) Posició 7 de BCONTROL, si val '1'B la contribució en progenitors de cada compartiment de POP no és proporcional a la seva grandària, sinó segons les probabilitats PNX.
- OVIN Posició 8 de BCONTROL, si val '1'B hi ha "imprinting" absolut a l'oviposició: tota femella deposita els ous fecundats al seu compartiment d'origen.
- BFER Posició 9 de BCONTROL, si val '1'B la fecunditat (paràmetres EFER, SFER de la subrutina DSTF) és dependent del genotip.
- BZIG Posició 10 de BCONTROL, si val '1'B el nombre de zigots produïts (NZIG) es genera segons una distribució estadística de paràmetres EFER, SFER.
- BQUI Posició 11 de BCONTROL, si val '1'B els paràmetres EQUI i SQUI de la subrutina DSTQ (generació estocàstica del nombre de quiasmes) són funció del genotip.
- BDNQ Posició 12 de BCONTROL, si val '1'B el nombre de quiasmes NQ es genera segons una distribució estadística de paràmetres EQUI i SQUI.
- BOV1 Posició 16 de BCONTROL, si val '1'B només a la primera oviposició dels productes d'una fecundació s'avalua PRO (tendència a ovipositar a cada compartiment).
- ZIGOT(2) BIT(NB) Zigot produït en una fecundació. ZIGOT(1) és el cromosoma patern, ZIGOT(2) el matern.

Variables reals

- EFER Mitjana de la distribució de probabilitats de la variable aleatòria "nombre de zigots produïts en una fecundació" (o qualsevol altre paràmetre).
- SFER Desviació típica (o qualsevol altre paràmetre) de la mateixa distribució.
- EQUI Mitjana (o qualsevol altre paràmetre) de la variable aleatòria "nombre de quiasmes".
- SQUI Desviació típica (o qualsevol altre paràmetre) de la mateixa variable.

Variables enteres escalars:

INX (Compartiment de POP) on té lloc l'aparellament.

INX1 (idem) de procedència del mascle (o primer membre de la parella).

INX2 (idem) de procedència de la femella (o segon)

INXOV (idem) d'oviposició

I1 índex del mascle (lloc a POP)

I2 índex de la femella (lloc a POP)

NZIG nombre de zigots produïts a un aparellament.

NQ nombre de quiasmes a una gametogènesi.

IZIG,IPROG,K índexos a operacions repetitives.

Variables enteres vectorials

W(NMAX) W(i) és la probabilitat acumulada que té l'individu i de participar com a reproductor en la constitució de la generació següent.

PNX(NNX) PNX(j) es la probabilitat acumulada de què el proper aparellament tingui lloc al compartiment j (de POP) o entre individus de j.

PRO(NNX) PRO(j) és la probabilitat acumulada que el proper zigot sigui dipositat a la casella j.

LQUI(MAXQ+1,2) LQUI(K,1) és la posició del k-èssim quiasma respecte dels loci, LQUI(k,2) ho és respecte de l'estructura cromosòmica.

Variables lògiques i tires de bits

BAPR BIT(1) Posició 6 de BCONTROL, si val '1'B hi ha selecció per la capacitat per l'aparellament.

plicats (LON). Paràmetres: POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,IND,ICR,INC,LON.

Hi ha la versió POSI0002 on totes les eleccions es fan a l'atzar: $\text{Prob}(\text{IND}=i) = 1/N$, $\text{Prob}(\text{ICR}=j) = \frac{1}{2}$, $\text{Prob}(\text{INC}=k) = 1/L$, $\text{LON} = |\text{INC2}-\text{INC}|+1$ amb INC2 també escollit a l'atzar.

A.5.2.12. Subrutina REPRO

Dirigeix la seqüència d'operacions necessàries per substituir la generació actual per la següent, canviant adequadament el contingut de POP, N i OCUP. Els seus paràmetres són: POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,BCONTROL,MINZ,MAXZ,MINQ,MAXQ, tots ells descrits a l'apartat dedicat al bloc CONTROL (A.5.2.1)

Hi ha les versions REPRO002 i NOREPRO la darrera no fa res, es pot incloure quan es vulgui utilitzar el bloc CONTROL per a altres funcions que no siguin produir noves generacions, com descriure l'estat d'un procés interromput, guardat en un dispositiu de memòria auxiliar (incluint també DESCRTOT, DECODTOT i COMPUTG, per exemple).

Mòdul REPRO002

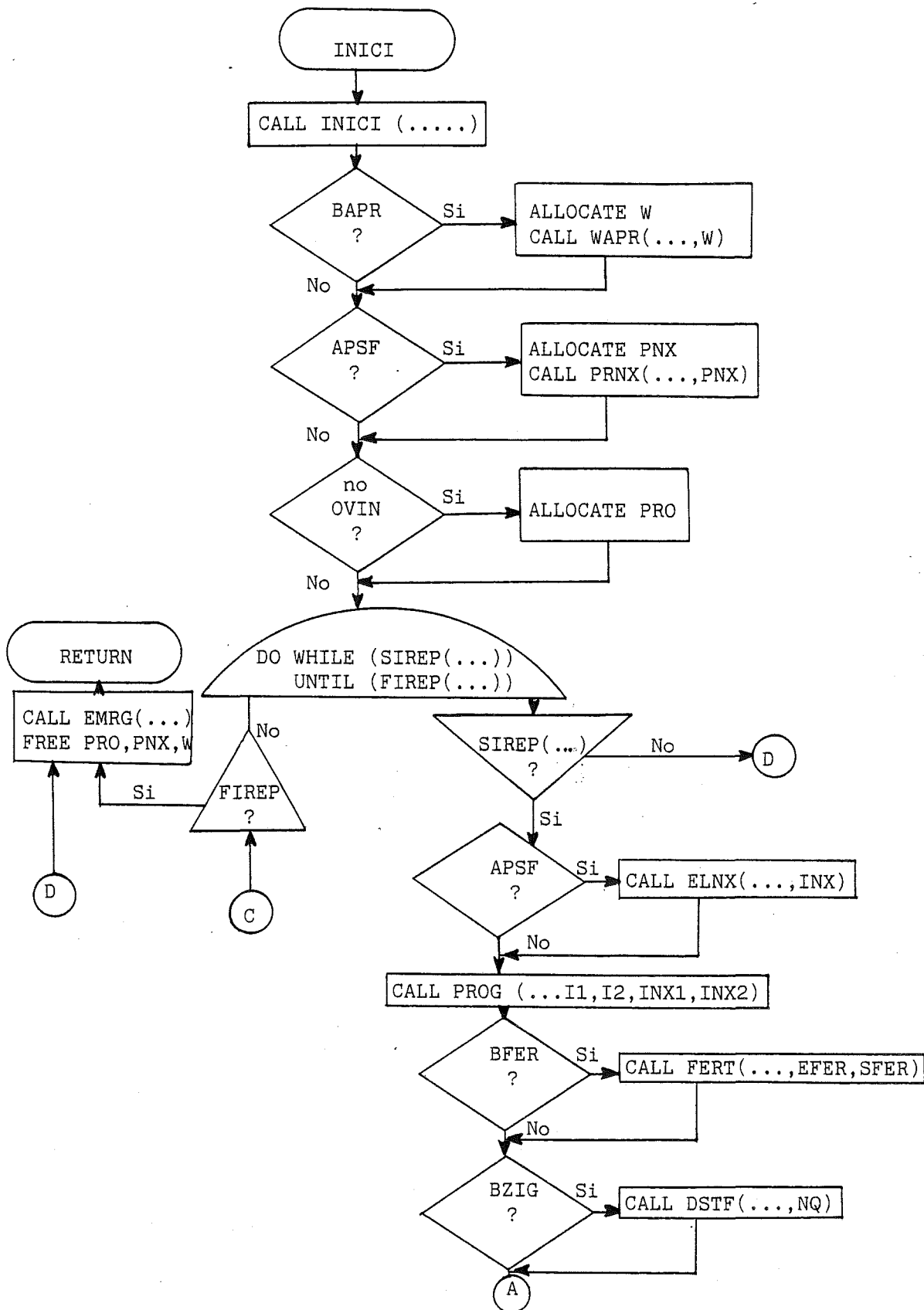
Variabls del programa:

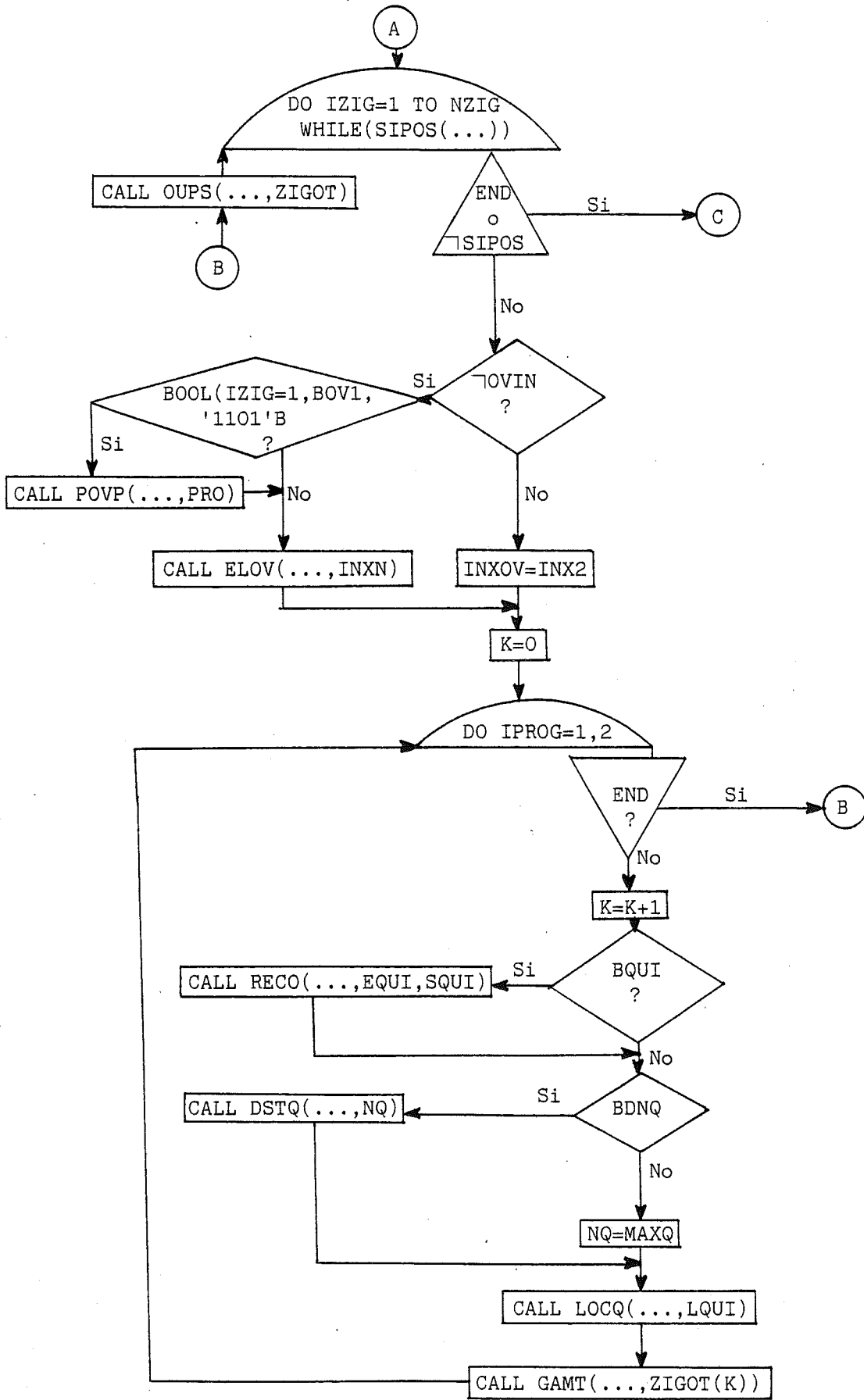
Subprogrames cridats des de REPRO

- INIC Subrutina cridada des de REPRO amb arguments POP, N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,BCONTROL,EFER,SFER,NZIG,EQUI,SQUI. Inicialitza les variables que calguin a REPRO i als subprogrames cridats des d'ella.
- WAPR Subrutina d'arguments POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,BCONTROL,W. Calcula el vector enter W.
- PRNX Subrutina d'arguments POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,BCONTROL,W,PNX. Calcula el vector PNX.
- SIREP Funció d'arguments POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,BCONTROL,PNX. Retorna un bit '1'B si les condicions són adequades per iniciar un esdeveniment d'aparellament i reproducció.
- FIREP Funció d'arguments POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,BCONTROL,INX,I1,I2,INX1,INX2,PRO,INXON. Retorna un '1'B si s'ha produït una condició per finalitzar els aparellaments - reproduccions.
- ELNX Subrutina d'arguments POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,BCONTROL,PNX,INX. Escolleix aleatòriament el compartiment on tindrà lloc la reproducció, segons el vector de probabilitats PNX.
- PROG Subrutina d'arguments POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,BCONTROL,W,INX,I1,I2,INX1,INX2. Escull dos elements de POP (I1 i I2, procedents dels compartiments INX1 i INX2 respectivament) com a reproductors, segons allò especificat a BCONTROL (p.e. a l'atzar, segons W, etc).
- FERT Subrutina d'arguments POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,BCONTROL,INX,I1,I2,INX1,INX2,EFER,SFER. Avalua la fecunditat (EFER,SFER) de la parella I1,I2.
- DSTF Subrutina d'arguments EFER,SFER,MINZ,MAXZ,NZIG. Genera aleatòriament el nombre de zigots (NZIG) produïts per una parella, segons una distribució de paràmetres EFER,SFER.

- SIPOS Funció d'arguments POP,N,L,NB,NST,NINX,PRIMER,OCUP
NNX,NMAX,INX,I1,I2,INX1,INX2,BCONTROL. Retorna
un bit '1'B si no es produeix cap condició per
finalitzar l'oviposició dels resultats d'un apare-
llament.
- POVP Subrutina d'arguments POP,N,L,NB,NST,NINX,PRIMER,
OCUP,NNX,NMAX,INX,I1,I2,INX1,INX2,BCONTROL,PRO.
Calcula el vector PRO per la femella I2 de INX2
que s'ha aparellat amb I1 de INX1.
- ELOV Subrutina d'arguments POP,N,L,NB,NST,NINX,PRIMER,
OCUP,NNX,NMAX,INX,I1,I2,INX1,INX2,BCONTROL,PRO,
INXOV. Escolleix el compartiment on s'ovipositarà,
possiblement aleatòriament segons el vector PRO.
- RECO Subrutina d'arguments POP,N,L,NB,NST,NINX,PRIMER,
OCUP,NNX,NMAX,BCONTROL,INX,I_{PROG},K,EQUI,SQUI.
Avalua EQUI i SQUI en funció del genotip (normal-
ment de POP(I_{PROG},*)).
- DSTQ Subrutina d'arguments EQUI,SQUI,MINQ,MAXQ,NQ. Ge-
nera aleatòriament, segons una distribució de parà-
metres EQUI i SQUI, el nombre de quiasmes NQ a
la producció d'un gamet.
- LOCQ Subrutina d'arguments POP,N,L,NB,NST,NINX,PRIMER,
OCUP,NNX,NMAX,INX,I_{PROG},K,LQUI,NQ,MINQ,MAXQ. Retor-
na la matriu LQUI amb les posicions dels quiasmes
i NQ eventualment modificat (per exemple si hi
ha inversions).
- GAMT Subrutina d'arguments POP,N,L,NB,NST,I_{PROG},K,LQUI,
NQ,ZIGOT(K). Genera un gamet (ZIGOT(k),k=1 o 2)
a partir dels cromosomes POP(I_{PROG},*) i segons
els quiasmes LQUI.
- OVPS Subrutina d'arguments POP,N,L,NB,NST,NINX,PRIMER,
OCUP,NNX,NMAX,BCONTROL,INXOV,ZIGOT. Assigna ZIGOT
en alguna posició específica de POP o d'alguna
altra matriu d'individus, simulant l'oviposició.
- EMRG Subrutina d'arguments POP,N,L,NB,NST,NINX,PRIMER,
OCUP,NNX,NMAX,BCONTROL. Simula el naixement o
l'arribada a l'estat d'adults dels zigots pro-
duïts. Normalment s'assignem a POP, substituint
els valors originals.

DIAGRAMA DE FLUX DE REPRO





A.5.2.13. Subrutina INIC

Inicialitza les variables necessàries a REPRO i els blocs cridats des d'ella, tant variables del sistema com les creades per l'usuari. Per aquest motiu és molt dependent de cada programa particular i és lògic que la defineixi cada usuari. Els seus paràmetres han d'ésser: POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX BCONTROL,EFER,SFER,NZIG,EQUI,SQUI, ja descrits en indicar les variables dels subprogrames CONTROL (A.5.2.1)' i REPRO (A.5.2.2).

Hi ha la versió INIC0002. Breument, es pot dir que INIC0002 dóna valors a EFER,SFER,NZIG,EQUI i SQUI si fa falta, a partir dels llegits a PARAM002, inicialitza a zero N2 i IREP, variables enteres que compten el nombre de nous individus i el nombre d'aparellaments realitzats a una execució de REPRO, respectivament, i allotja POP2 i LARVOC, matriu d'individus "preadults", amb les mateixes característiques que POP, i vector de dimensió NNX que compta el grau d'ocupació de POP2(LARVOC(j) és el nombre d'individus (preadults) al compartiment j de POP2), respectivament. N2, IREP, POP2 i LARVOC són variables externes.

A.5.2.14. Subrutines WAPR,PRNX,ELNX,FERT i RECO

Ja descrites en parlar dels blocs cridats (eventualment) des de REPRO (A.5.2.12). No hi ha cap versió al sistema. Els paràmetres de WAPR han d'ésser: POP,N,L,NB,NST,NINX,PRIMER,OCUP,

A.5.2.16. Funció FIREP

Retorna un bit, '1'B quan es produeix alguna condició que impliqui que cal acabar els esdeveniments de reproducció. Els seus paràmetres són: POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,BCONTROL,INX,I1,I2,INX1,INX2,PRO,INXOV.

Hi ha la versió FIREPOO2 que retorna un '1'B quan IREP=NREP a més incrementa IREP en 1 cada cop que es crida (vegeu A.5.2.15).

A.5.2.17. Subrutina PROG

Escull dos progenitors, els individus (files) I1 i I2 de POP, que procedeixen respectivament dels compartiments INX1 i INX2 de POP. Els seus paràmetres son: POP,N,L,NB,NST,NINX,PRIMER OCUP,NNX,NMAX,BCONTROL,W,INX,I1,I2,INX1,INX2.

Hi ha la versió PROG0002 la qual es descriu tot seguit.

Mòdul PRG0002

Pot escollir els progenitors a l'atzar o segons el vector de probabilitats acumulades W (segons que BAPR, posició 5 de BCONTROL, valgui '0'B o '1'B). Aquesta elecció es farà dins del compartiment INX si APSF val '1'B (posició 7 de BCONTROL) o a tot POP si APSF val '0'B.

Variables del subprograma:

NXPR indica el compartiment de POP on començarà la recerca de I1 i I2. Si APSF='1'B, NXPR=INX, en cas contrari NXPR=1.

NXDR indica el compartiment de POP on finalitzarà la recerca de I1 i I2. Si APSF='1'B, NXDR=INX, en cas contrari NXDR=NNX.

LLAVOR(15) vector extern de les llavors de nombres aleatoris, l'element LLAVOR(3) s'empra en aquest bloc.

N1 N1=PRIMER(NXPR)

NWO,NWN Les probabilitats acumulades que un individu sigui escollit estan definides a l'interval (NWO,NWN).

APSF,BAPR Variables lògiques ja descrites.

Subrutines internes cridades des de PROG:

ELECO(IJ,INXJ) realitza l'elecció a l'atzar dins de tot POP.

ELEC(IJ,INXJ) realitza l'elecció segons les probabilitats W. IJ és l'individu i INXJ el seu compartiment d'origen.

A.5.2.18. Subrutina DSTF

Genera el nombre de zigots NZIG produïts en un aparellament segons una distribució de paràmetres EX,VX. Paràmetres de la subrutina: EX,VX,NMIN,NMAX,NZIG. Hi ha la versió DSTF01 que és en realitat un generador de nombres amb distribució normal N(EX,VX) (VX representa la desviació típica). Es basa en la coneguda propietat que si U_1 i U_2 són variables aleatòries independents amb distribució uniforme a (0,1), aleshores $X=\sqrt{-2\log(U_1)} \cdot \cos(2 U_2)$ té una distribució N(0,1).

DIAGRAMA DE FLUX DE PROG002

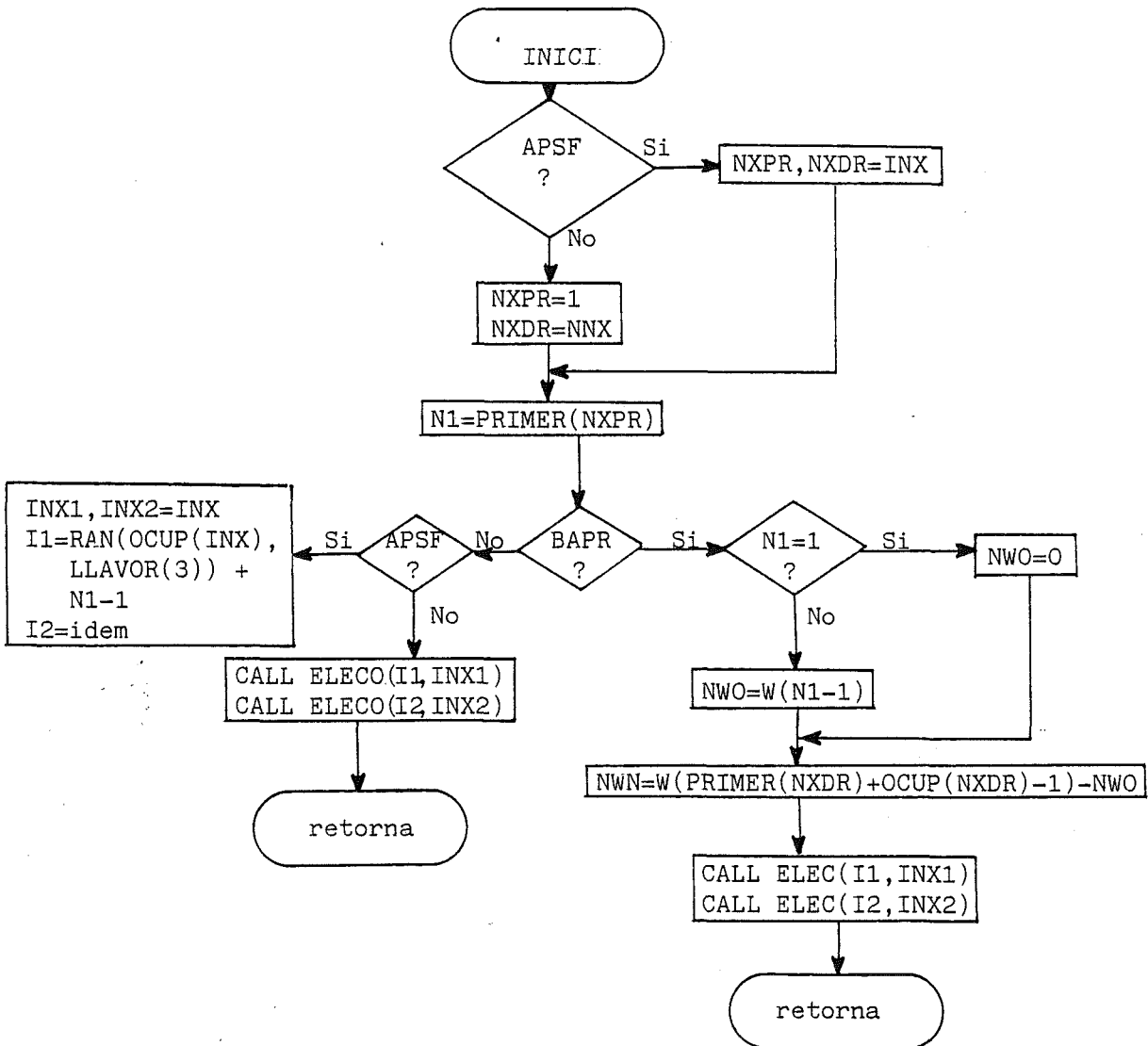


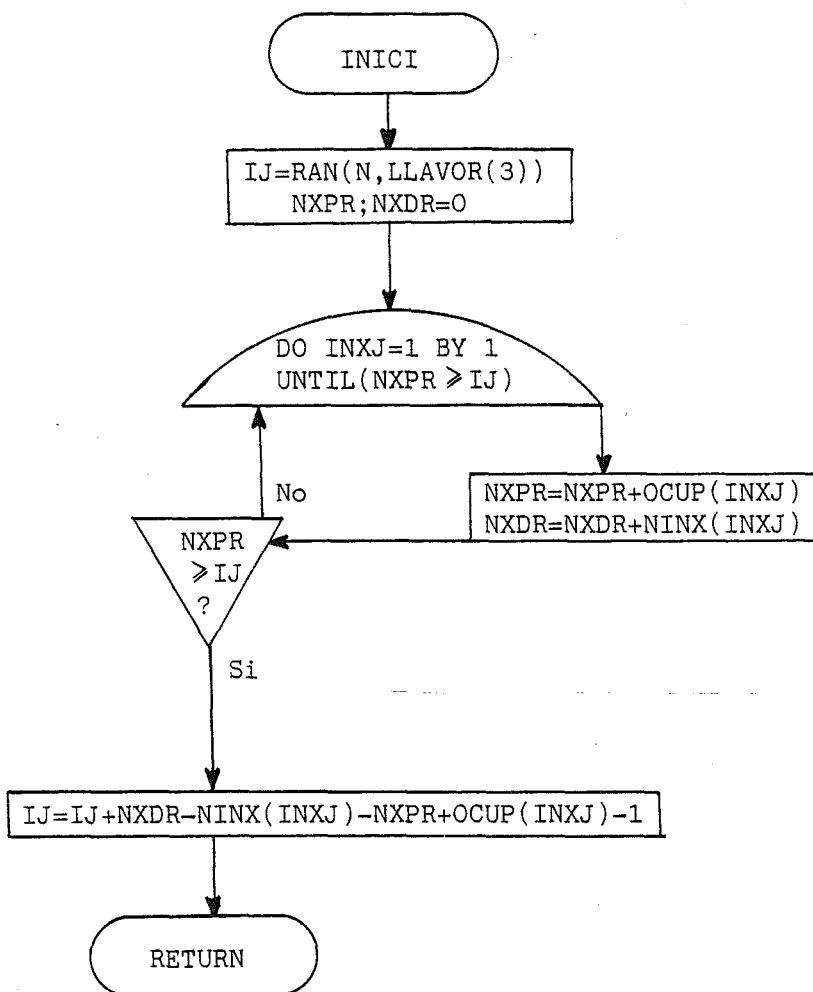
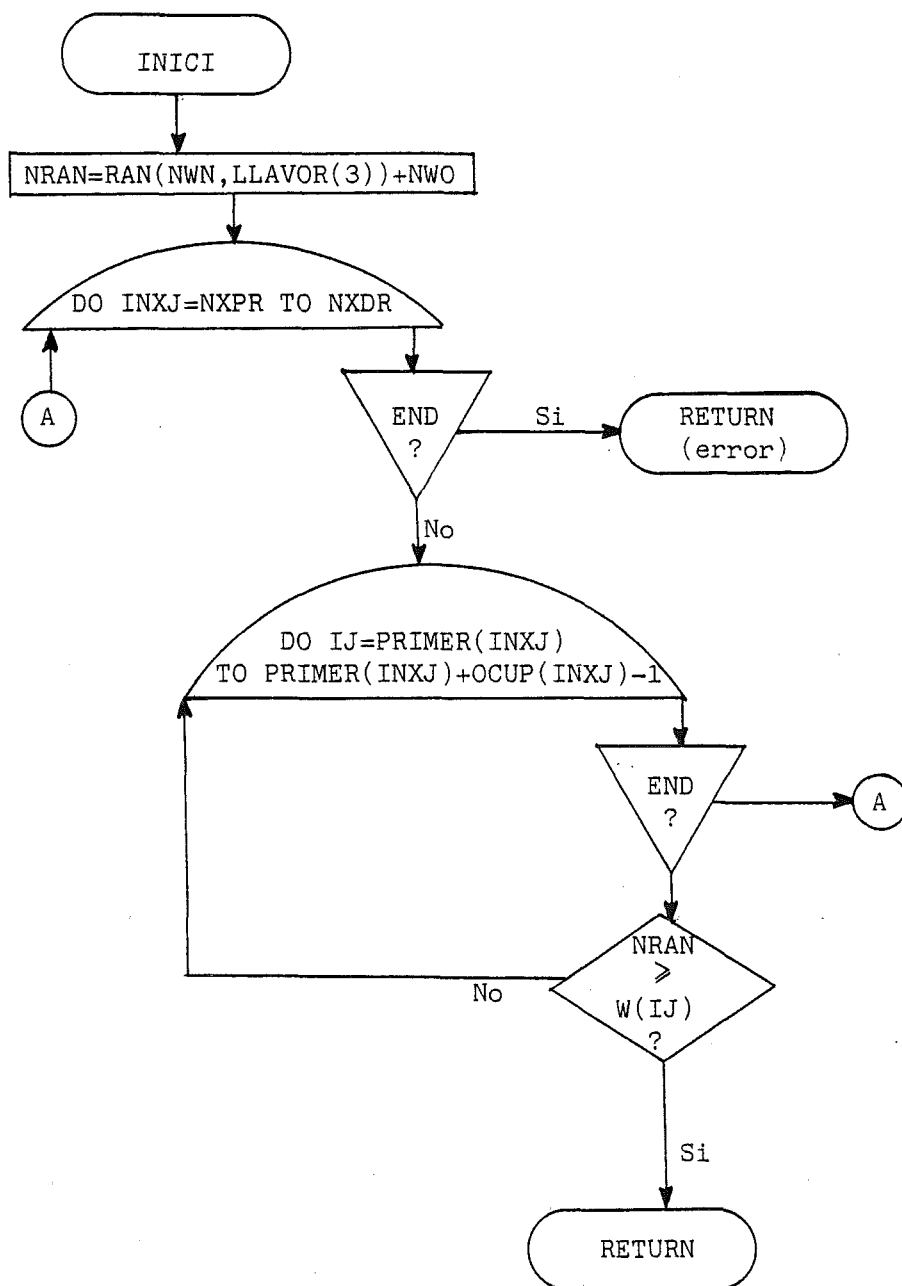
DIAGRAMA DE FLUX DE ELECO

DIAGRAMA DE FLUX DE ELEC

Variables del programa:

EX mitjana de la variable aleatòria normal.
 VX desviació típica
 X1,X2 nombres generats a partir d'una distribució uniforme a
 (0,1)
 X nombre generat segons una $N(0,1)$
 NMIN valor mínim de NZIG
 NMAX " màxim de NZIG
 NZIG Valor, arrodonit a un nombre enter, generat segons una
 $N(EX, VX)$

L'usuari pot definir les seves pròpies DSTF molt fàcilment fent únicament des de la seva versió de DSTF una crida a una subrutina generadora de nombres segons alguna distribució estadística.

A.5.2.19. Funció SIPOS

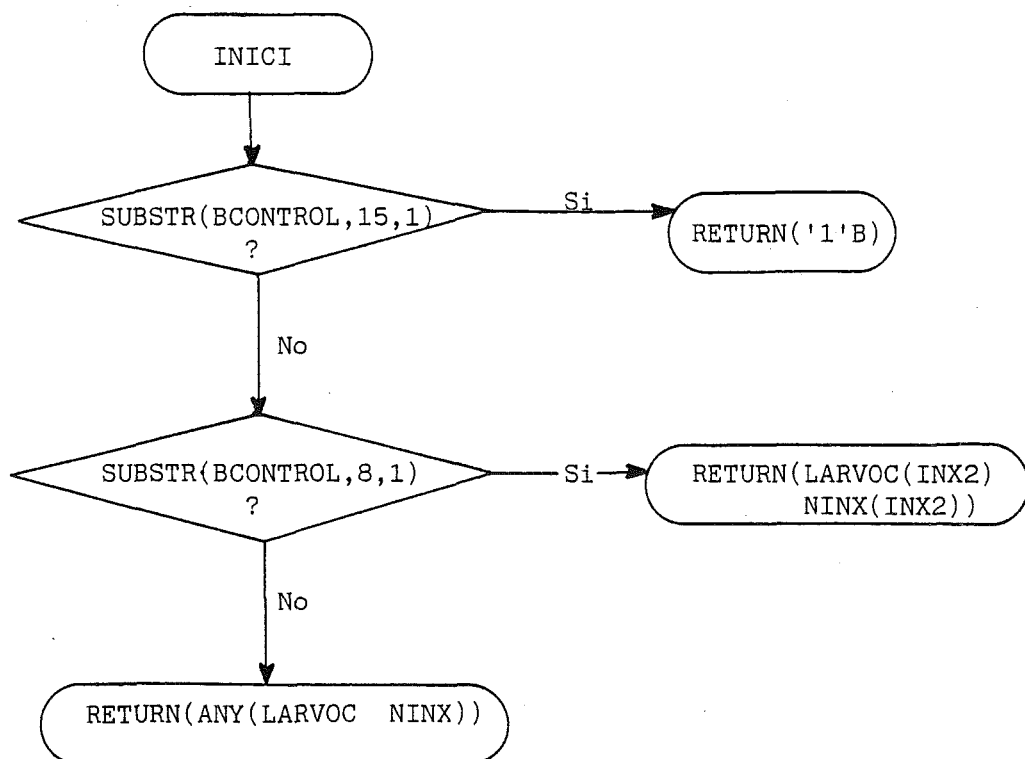
Retorna un bit '1'B quan no es produeix cap condició que impliqui finalitzar la producció de zigots i la posta.

Els seus paràmetres són POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,INX,I1,I2,INX1,INX2,BCONTROL.

Hi ha la versió SIPS002. Retorna un '1'B si la posició 15 de BCONTROL (BCLV) és '1'B. Això vol dir que hi ha competència larvària i que no hi haurà cap restricció a la oviposició. En cas contrari, si la posició 8 de BCONTROL(OVIN) és '1'B, és a dir, hi ha "imprinting" absolut i cada femella oviposita al seu

compartiment d'origen, es retorna un '1'B si aquell compartiment encara no és ple, la qual cosa es sap mitjançant un vector extern LARVOC(NNX), inicialitzat a zeros a INICI, i que s'incrementa $LARVOC(INX)=LARVOC(INX)+1$ cada cop que s'oviposita en una posició no ocupada del compartiment INX (vegeu la descripció de la subrutina VPS). Si OVIN val '0'B només es comprova que algun compartiment encara no sigui ple.

DIAGRAMA DE FLUX



A.5.2.20. Subrutina POVP

Retorna un vector (paràmetre PRO) amb les probabilitats acumulades d'oviposició als diferents compartiments de POP.

Els seus paràmetres són POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,INX,I1,I2,INX1,INX2,BCONTROL,PRO.

Hi ha les versions POVP1002 i POVP2002. La primera és nul·la, no calcula cap valor a PRO, es pot utilitzar quan es vol elegir el lloc d'oviposició totalment a l'atzar (vegeu ELOV1002) El mòdul POVP2002 retorna un vector de probabilitats acumulades PRO dependent additivament del genotip.

A.5.2.21. Subrutina ELOV

Escull aleatòriament el compartiment (paràmetre INXOV) on s'ovipositarà un zigot concret.

Els seus paràmetres son POP,N,L,NB,NST,NINX,PRIMER,OCUP,NNX,NMAX,INX,I1,I2,INX1,INX2,BCONTROL,PRO,INXOV.

Hi ha les versions ELOV1002 i ELOV2002. La primera fa una elecció totalment a l'atzar. La seva única instrucció és INXOV=
=RAN(NNX,LLAVOR(4)) (vegeu RAN). ELOV2002 excull INXOV segons les probabilitats acumulades PRO.

A.5.2.22. Subrutina DSTQ

Genera aleatòriament el nombre de quiasmes NQ a una gametogènesi concreta.

Hi ha la versió DSTQ01, de paràmetres EX,VX,MINQ,MAXQ,NQ, l'única instrucció de la qual es una crida a POISS (vegeu més endavant), és a dir que genera NQ segons una distribució de Poisson de paràmetre EX. D'una manera similar es podria definir qualsevol altra versió de DSTQ en què es volgués generar NQ segons alguna distribució concreta.

A.5.2.23. Subrutina LOCQ

Reparteix els NQ quiasmes al llarg del cromosoma, retornant LQUI que n'especifica la posició.

Hi ha la versió LOCQ0002, de paràmetres POP,N,L,NB,NST,NINX PRIMER,OCUP,NNX,NMAX,INX,IPOG,K,LQUI,NQ,MINQ,MAXQ.

Utilitza la llavor de nombres aleatoris LLAVOR(6) per generar-ne NQ, compresos entre 1 i L, assignant-los a les files 1,, NQ de LQUI. LQUI(1,j) especifica la posició del quiasma j respecte del loci, LQUI(2,j) respecte de l'estructura cromosòmica. En principi LQUI(1,j)=LQUI(2,j), però segons l'estructura cromosòmica, la subrutina STCR, cridada des de LOCQ, pot modificar LQUI(1,*) i LQUI(2,*) de maneres diferents. Finalment, amb probabilitat $\frac{1}{2}$, es produeix un quiasma addicional a la posició

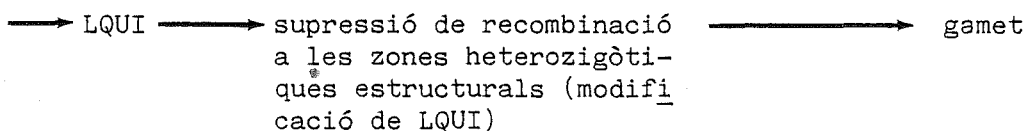
1 o no es produeix. Això té el sentit d'iniciar la formació del gamet amb un o l'altre cromosoma, amb igual probabilitat.

A.5.2.24. Subrutina STCR

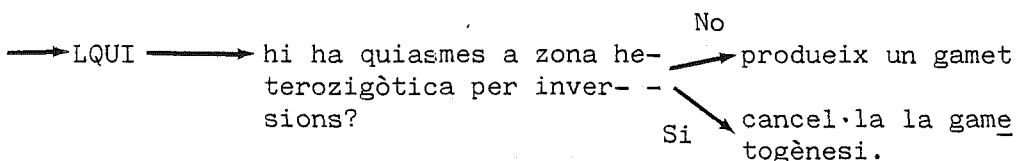
Modifica LQUI segons l'estructura cromosòmica.

Hi ha la versió STCR01 que suprimeix la recombinació als llocs on hi ha heterozigosi estructural, modificant també adequadament LQUI(2,j) si hi ha homozigosi per inversions. En realitat seria més correcte suprimir tots aquells gamets amb algun quiasma dins la zona heterozigòtica estructural; però suposant la producció d'un nombre molt gran de gamets i que una baixa proporció d'ells arriben a produir zigots, aquest procediment sembla aproximadament correcte i molt més ràpid.

Esquemàticament, STCR01 fa:



I seria més realista:



Aquest darrer procés, repetit moltes vegades, hauria de donar unes freqüències de recombinació com les del primer.

Els paràmetres de SATCRO1 són POP,N,L,NB,NST,IProg,K,LQUI,NQ,MINQ,MAXQ. Altres variables del programa són:

LOC LOC=LQUI(KIASM,2), KIASM=1,2,...,NQ. Lloc de cada un dels quiasmes.

I1 Variable auxiliar, indicador d'una posició dins E1,E2. S'utilitza per buscar l'inici d'una inversió.

I2 Posició terminal d'una inversió.

E1,E2 BIT(L) Tira de bits codificant un nivell d'estructura dels cromosomes 1 i 2 respectivament.

B1,B2 BIT(1) Posició LOC de E1 i E2 respectivament.

BL1,BL2 BIT(1) Posició I1 de E1 i E2 respectivament.

Des de STCR s'avalua la funció interna, sense arguments, BLIM, que retorna un bit. BLIM es crida si una quiasma cau en una zona invertida als dos cromosomes d'un individu. Si ambdues inversions són idèntiques (comencen i acaben al mateix lloc) retorna un bit '1'B, en cas contrari retorna '0'B.

A.5.2.25. Subrutina GAMT

Simula la producció d'un gamet (paràmetre GAMET, una tira de $L*(NST+1)$ bits) a partir dels cromosomes de POP(IProg,*) i dels punts de recombinació LQUI. El paràmetre GAMET normalment

DIAGRAMA DE FLUX DE STCR

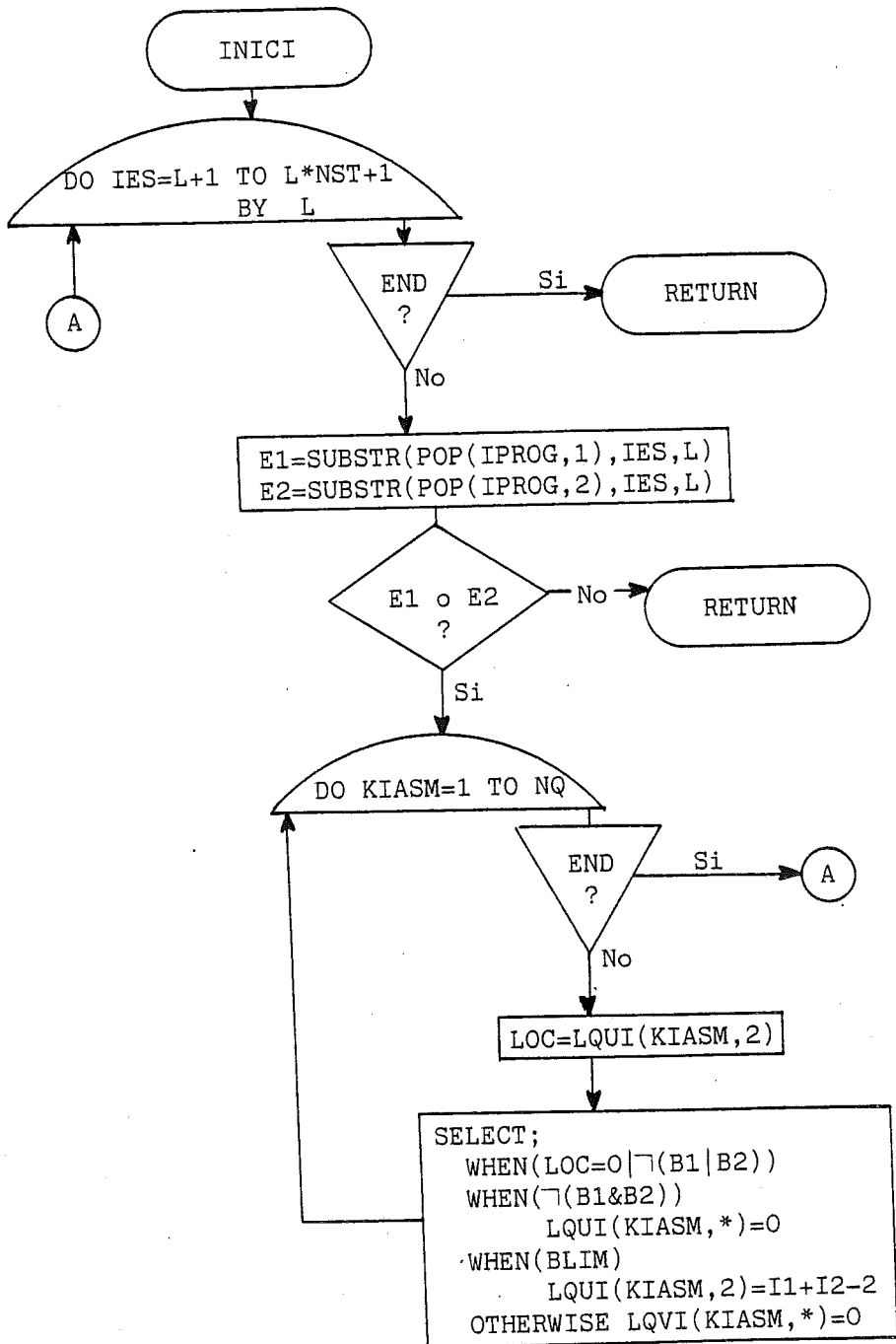
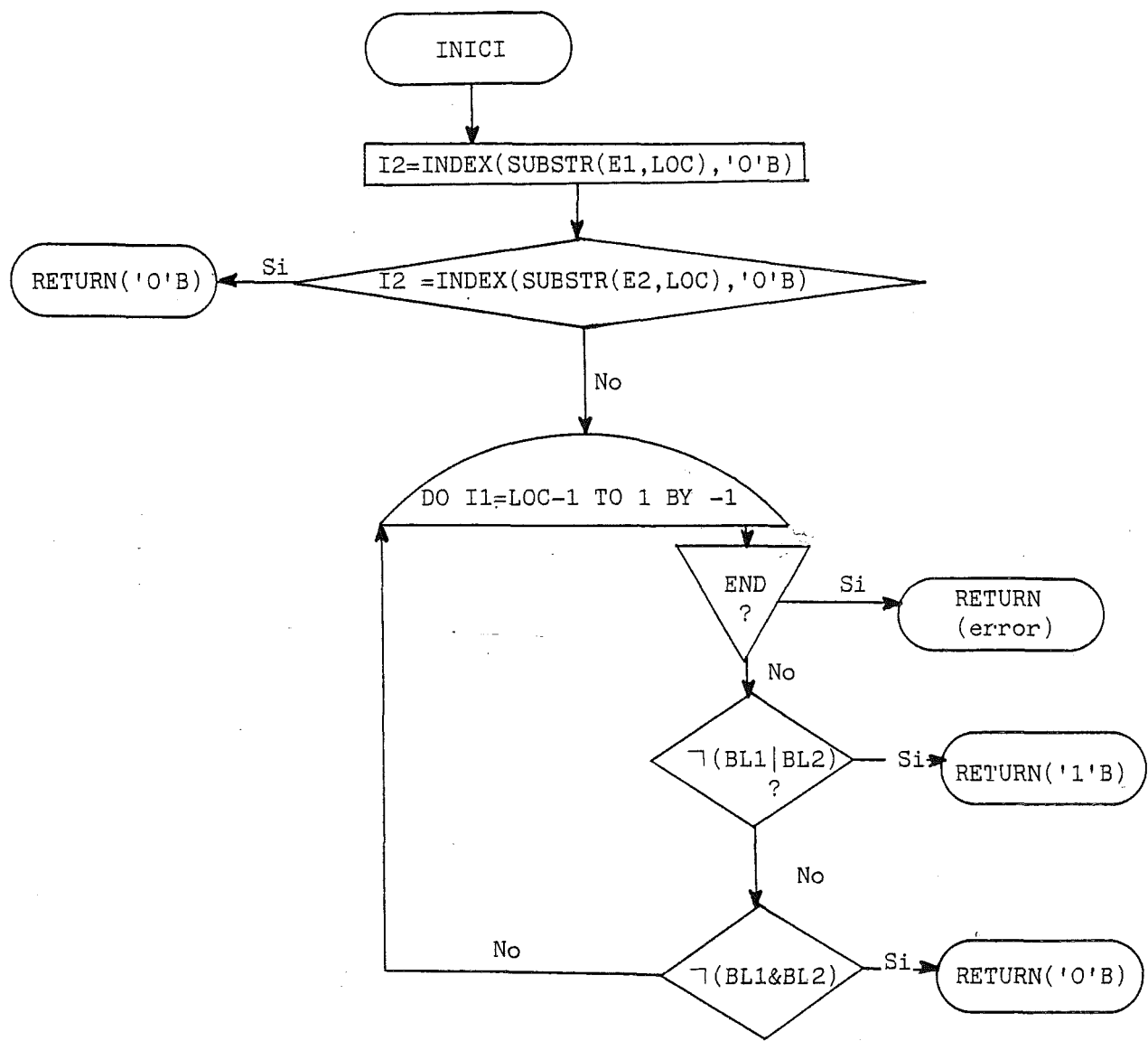


DIAGRAMA DE FLUX DE BLIM



correspon a l'argument ZIGOT(K), K=1,2, descrit a A.5.2.10 (RE-PRO).

Hi ha la versió GAMT0002 que és una versió millorada de l'algorisme proposat a Fraser i Burnell (1970) i a Crosby (1973) Introdueix també algunes idees de Rico (1975).

Els seus paràmetres són: POP,N,L,NB,NST,I PROG,K,LQUI,NQ, GAMET.

La manera possiblement més "semblant a la realitat" de construir GAMET és copiar un fragment d'un dels cromosomes, elegit a l'atzar, fins al punt de la primera recombinació, dels d'on es copiarà l'altra fins al següent punt de recombinació on es tornarà a canviar, etc. GAMT0002 fa això seguint el següent procediment:

a) Construeix dues tires de L bits (variable bit LB, tires LB(1) i LB(2)) tals que si considerem ordenats de més petit a més gran els valors de LQUI(j,*) (j=1 o 2): LQUI(j,1) LQUI(j,2) ...
 ... LQUI(j,NQ), la part de LB(j) compresa entre LQUI(j,i) i LQUI(j,i+1)-1 (i=1,2,...,NQ) està formada exclusivament de bits '1'B si i és imparell i de bits '0'B si i és parell. Per exemple si L=10, NQ=3, LQUI(j,1)=2, LQUI(j,2)=4, LQUI(j,3)=7, aleshores LB(j) serà 0110001111. És a dir, cada vegada que cal canviar

de cromosoma copiat, canvia el tipus de seqüència.

b) Es defineix una tira LOC de NB bits formada per la concatenació de LB(1) amb NST vegades LB(2). LB(1) expressa els al·lels que es copiaran i LB(2) les parts de la descripció de l'estructura cromosòmica.

c) Si es defineix l'operació \oplus com: $1\oplus 1=0, 0\oplus 0=0, 1\oplus 0=0, 0\oplus 1=1$ (suma lògica, equivalent a la suma dins dels enters mòdul 2), és fàcil comprovar que la fórmula

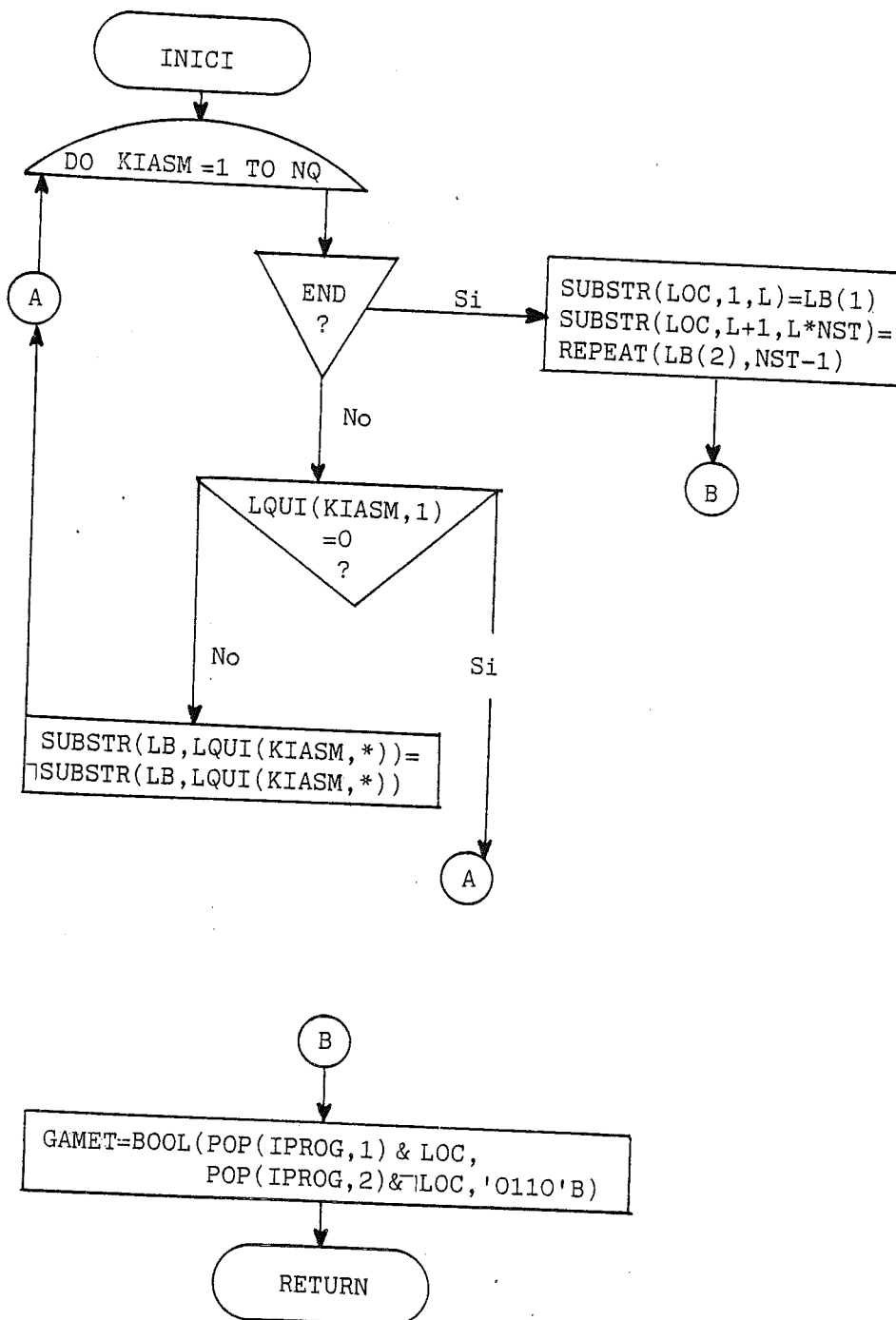
$$(\text{POP}(\text{IPROG},1)\&\text{LOC}) \oplus (\text{POP}(\text{IPROG},2)\&\neg\text{LOC})$$

és equivalent a GAMET tal com ha estat definit més amunt. L'operació $\&$ es defineix com $1\&1=1, 1\&0=0, 0\&1=0, 0\&0=0$ (copulació lògica) i l'operació \neg com $\neg 0=1, \neg 1=0$ (negació lògica), aplicant aquestes operacions bit a bit dins de les tires de bits que són els operants.

A.5.2.26. Subrutina OVPS

Simula l'arribada al "medi", on passa a "larva", d'un "zigot". Aquest "medi" pot ser la mateixa matriu POP (larves o estats preadults i adults competeixen el mateix nínxol), una part d'ella o una matriu similar apart. Allí pot arribar directament a adult, competir amb altres larves, etc.

DIAGRAMA DE FLUX DE GAMTOOO2



Hi ha la versió OVPS0002 de paràmetres POP,N,L,NB,NST,NINX, PRIMER,OCUP,NNX,NMAX,BCONTROL,INXOV,ZIGOT, tots descrits a A.5.

10. OVPS0002 està molt orientada a les simulacions concretes fetes en aquesta memòria. Assigna el contingut de ZIGOT a la primera posició lliure de la matriu POP2, variable externa amb les mateixes característiques que POP,i allotjada quan s'executa INIC0002. Aquesta posició és PRIMER(INXOV)+LARVOC(INXOV)-1, LARVOC és un vector de NNX elements, LARVOC(j) és el nombre de preadults a j. Quan el compartiment INXOV és ple (LARVOC(INXOV)=NINX(INXOV)), si la posició 15 de BCONTROL és '1'B es crida la subrutina CLRV d'arguments POP2,N,L,NB,NST,NINX,PRIMER,OCUP, NNX,NMAX,BCONTROL,INXOV,ZIGOT, que simula competició pel nínxol entre ZIGOT i un individu elegit a l'atzar del compartiment INXOV de POP2, en cas contrari es perd ZIGOT.

A.5.2.27. Subrutina CLRV

Hi ha la versió CLRVO002 de paràmetres POP,N,L,NB,NST,NINX, PRIMER,OCUP,NNX,NMAX,BCONTROL,INXOV,ZIGOT. Cal remarcar que ara el paràmetre POP correspondrà normalment a l'argument POP2 i no a la variable del sistema POP.

Avalua additivament el genotip de ZIGOT i de POP(IND,*) essent $IND=PRIMER(INXOV)+RAN(NINX(INXOV), LLAVOR(13))-1$. Si ZIGOT és més proper a l'òptim del compartiment INXOV, substitueix l'anterior contingut de POP(IND,*).

A.5.2.28. Subrutina EMRG

Simula l'arribada a adults dels preadults d'una generació, que substituiran els adults de l'anterior.

Hi ha la versió EMRG0002, de paràmetres POP,N,L,NB,NST,NINX PRIMER,OCUP,NNX,NMAX,BCONTROL i amb les variables externes definides POP2, LARVOC i N2. Assigna LARVOC a OCUP, POP2 a POP, N2 a N i zero a N2. Finalment allibera l'espai de memòria ocupat per POP2 i LARVOC.

A.5.2.29. Subrutina POISS

Genera un nombre enter NPOIS segons una distribució de Poisson de paràmetre $\lambda=EX$, NPOIS està comprès entre un mínim NMIN i un màxim NMAX.

Hi ha la versió POISS01 de paràmetres EX,NMIN,NMAX,NPOIS, ISEM. Com s'ha dit EX representa el paràmetre λ de la distribució de Poisson, NMIN i NMAX els límits entre els que es generaran els valors ($0 \leq NMIN \leq NMAX \leq 32767$), NPOIS el valor general i ISEM la llavor de nombres aleatoris. El procediment seguit és el següent:

a) Si la variable aleatòria X segueix una distribució de Poisson de paràmetre λ , $X \sim \mathcal{P}(\lambda)$, aleshores si $\lambda \rightarrow \infty$ la distribució de la variable $(X-\lambda)/\sqrt{\lambda}$ és asimptòticament $N(0,1)$. Quan $\lambda > 6$

es considera que l'aproximació és suficientment bona, de manera que el programa, si $EX(=\lambda) > 6$, genera un nombre Y amb distribució normal $N(0,1)$ i fa el canvi de variable $NPOIS = [Y \cdot \sqrt{EX} + EX]$ on $[\cdot]$ indica la part entera. El procediment seguit per generar Y s'explica a l'apartat A.5.2.16.

b) Quan $EX \leq 6$ es genera NPOIS utilitzant la relació existent entre la distribució de Poisson i l'exponencial: si els temps entre esdeveniments d'un determinat tipus es distribueix segons una exponencial de paràmetre α i si són variables aleatòries estocàsticament independents, aleshores per un interval arbitrari de temps Δt , el nombre d'esdeveniments produïts durant Δt segueix una distribució de Poisson de paràmetre $\lambda = \frac{\Delta t}{\alpha}$. Si arbitràriament es pren $\Delta t = 1$, suposem que es generen $T_0, T_1, T_2, \dots, T_{NPOIS}, T_{NPOIS+1}$ nombre segons una exponencial de paràmetre $\alpha = 1/\lambda$, tals que $\sum_{i=0}^{NPOIS} T_i \leq 1 < \sum_{i=0}^{NPOIS+1} T_i$, aleshores NPOIS ha d'estar distribuïda segons una Poisson $\lambda = 1/\alpha$. L'expressió anterior és equivalent a

$$-\sum_{i=0}^{NPOIS} \log(U_i) \leq \lambda < -\sum_{i=0}^{NPOIS+1} \log(U_i)$$

$$\prod_{i=1}^{NPOIS} U_i \leq e^{-\lambda} < \prod_{i=1}^{NPOIS+1} U_i \quad (*)$$

éssent U una variable amb distribució uniforme entre 0 i 1, ja que (secció 1) el mètode de la inversa de la funció de distribució permet generar exponencials a partir d'uniformes mitjançant l'expressió

$$T_i = -\lambda \log(U_i) = -\frac{1}{\lambda} \log(U_i).$$

El programa genera nombre U uniformement distribuïts entre 0 i 1, fins que es compleix (*), i retorna el valor NPOIS.

A.5.2.30. Subrutina GAUSS

Genera un nombre real X segons una distribució normal amb mitjana EX i desviació típica SX , $N(EX, SX)$.

Hi ha la versió GAUSS01 de paràmetres $EX, SX, XMIN, XMAX, X, ISEM$. Com s'ha dit al paràgraf anterior, X representa el nombre generat i EX, SX els paràmetres de la distribució. $XMIN$ i $XMAX$ no s'empren en aquesta versió $ISEM$ és la llavor de nombres aleatoris, paràmetre del generador RAN de nombres aleatoris.

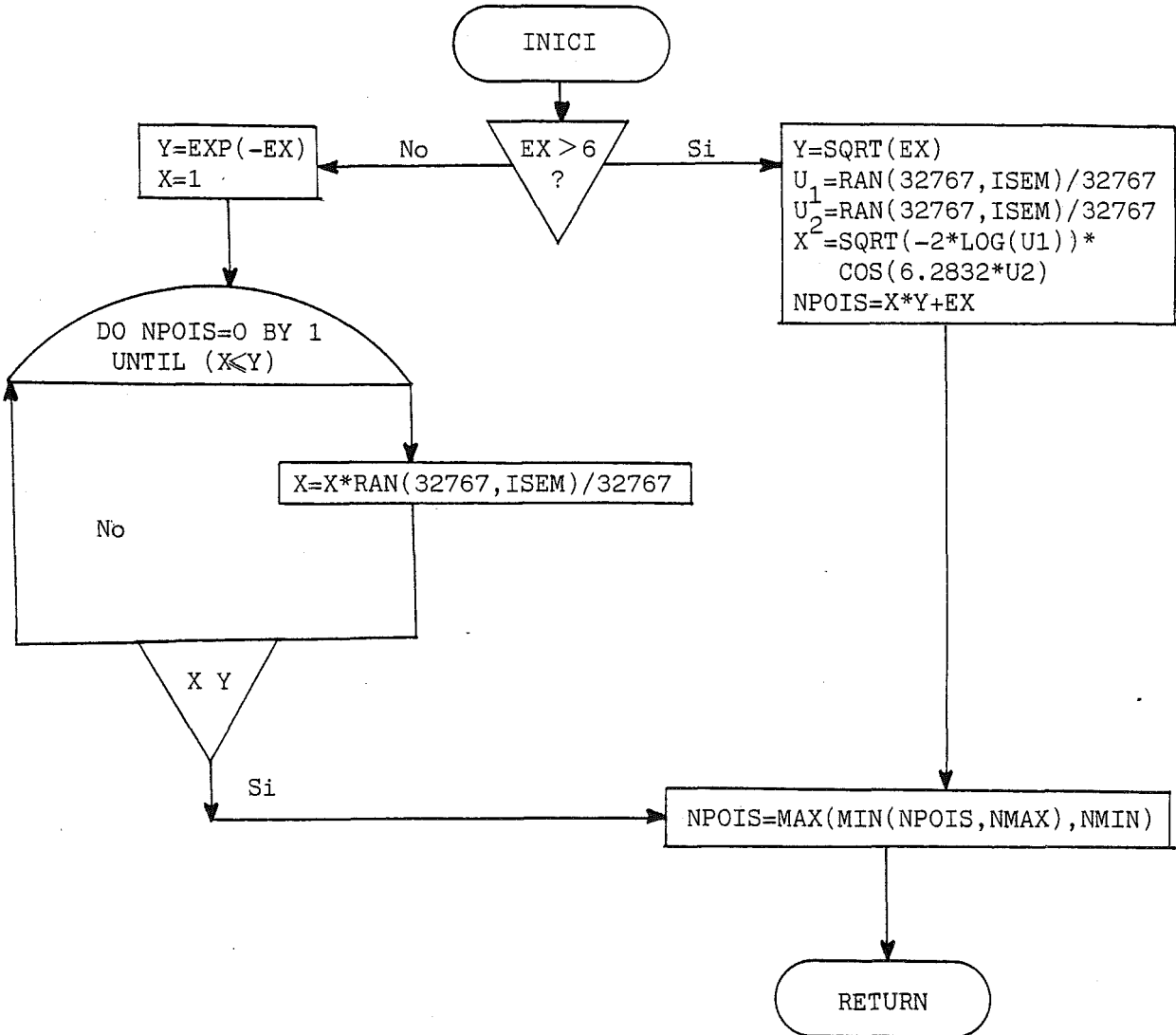
GAUSS01 utilitza el fet que si U_1 i U_2 són variables aleatòries independents amb distribució uniforme a $(0,1)$, la variable $X = \sqrt{-2 \log U_1} \cos(2\pi U_2)$ té distribució $N(0,1)$.

A.5.2.31. Subrutina RAN

Es un generador de nombres aleatoris enters.

Els seus paràmetres són $K, ISEM$. El segon és la llavor de

DIAGRAMA DE FLUX DE POISSO1



nombres aleatoris. Si K és 32767 la funció RAN retorna un nombre enter entre 0 i 32767, escollint-lo amb la mateixa probabilitat $1/32768$ per a tots. Si $K < 32767$ es retorna un nombre enter entre 1 i K , escollint-los amb probabilitat $\text{prob}(i)=1/K$, $i \in \{1,2,\dots,K\}$.

Hi ha la versió JORRANMOZ que és un generador congruencial (vegeu Fishman (1973), capítol 7), és a dir generant el nombre N_i a partir de l'anterior N_{i-1} mitjançant l'expressió

$$N_i \equiv aN_{i-1} + b \pmod{m}$$

on a, b, m són constants adequadament escollides.