# Simbiosis de las Técnicas Evolutivas y el Procesamiento Estadístico del Lenguaje Natural

Lourdes Araujo

lurdes@sip.ucm.es

Dpto. Sistemas Informáticos y Programación

Universidad Complutense de Madrid

- Esquema General de Aplicación de Técnicas Evolutivas al Procesamiento del Lenguaje Natural (PLN)
- Aplicación al Etiquetado Léxico de textos.
- Aplicación al Análisis Sintáctico.

# Motivación

- PLN incluye una gran cantidad de procesos complejos:
  - etiquetado léxico,
  - análisis sintáctico,
  - determinación de los antecedentes de pronombres y cláusulas de relativo, etc.
- Muchos de estos procesos pueden verse como una búsqueda de la estructura correcta.

- Los métodos estadísticos han conseguido avances importantes en muchos de estos problemas.
- Estos métodos permiten considerar el problema lingüístico a tratar como una optimización.

# Alg. Evolutivos y técnicas estadísticas

- Las medidas estadísticas usadas en los enfoques estadísticos a PLN proporcionan una función de evaluación natural.
- Los textos de entrenamiento permiten el ajuste automático de los parámetros del algoritmo evolutivo.
- Los algoritmos evolutivos aportan su robustez a la búsqueda y optimización involucrados en los problemas de PLN.

# Esquema de Aplicación

- Individuos: dependientes del problema.
- Función de Adaptación: modelos estadísticas.
- Operados genéticos: dependientes del problema.
- Parámetros del algoritmo: ajustados mediante corpus de entrenamiento.

# Dos Aplicaciones

- Etiquetado Léxico
- Análisis Sintáctico

# Etiquetado Léxico

- Muchas palabras son ambiguas (pertenecen a distintas categorías léxicas):

| | |
|---|---|
| Rice: | NOMBRE |
| flies: | NOMBRE, VERBO |
| like: | PREP, VERBO |
| sand: | NOMBRE |

# Etiquetado Léxico Evolutivo

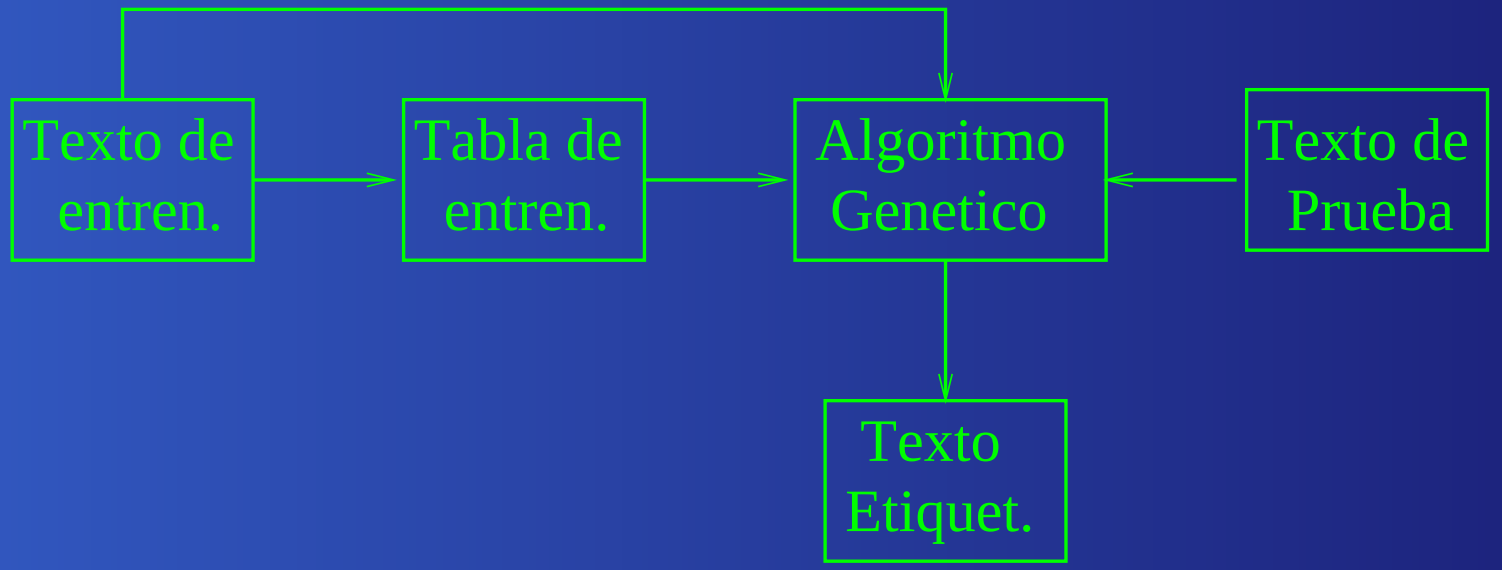
- La asignación depende del contexto de la palabra: etiquetas de las palabras circundantes.
- La evaluación de los individuos se basa en los datos extraídos de un corpus etiquetado manualmente.

# Etiquetado Léxico Evolutivo

- La tabla de entrenamiento se construye a partir del corpus de entrenamiento.
- Registra los distintos contextos para cada etiqueta y sus frecuencias.
- Los cromosomas se evalúan en función de la tabla de entrenamiento: maximización de la probabilidad total del etiquetado.

# Etiquetado Léxico Evolutivo

- Ajuste automático de parámetros:



# Representación de los Individuos

- Los cromosomas son una secuencia de etiquetas de cada palabra de la sentencia.

**Población Inicial**

- Selección aleatoria, proporcional a la frecuencia, en un diccionario de una de las etiquetas válidas de cada palabra.
- A las palabras ausentes se les asigna la etiqueta que aparece más frecuentemente en el contexto correspondiente.

# Aptitud de los Individuos

- Probabilidad total de la secuencia de etiquetas de la sentencia.
  $n$: número de palabras de la sentencia
  $w_i$ palabra en la posición $i$ (gen $g_i$).
  $f(g_i)$ aptitud del gen $g_i$.

$$Aptitud = \sum_{i=1}^{n} f(g_i)$$

# Aptitud de los Individuos

- Se consideran los contextos:

$$LC(T_{l_1}, \cdots, T_{l_{l_{LC}}}), T, RC(T_{r_1}, \cdots, T_{r_{l_{RC}}})$$

$T \in \mathcal{T}$: etiqueta asignada a $w_i$.
$\mathcal{T}$: conjunto de etiquetas posibles de $w_i$.
*LC*: Parte izq. del contexto (longitud $l_{LC}$)
*RC*: Parte derecha (longitud $l_{RC}$)

# Aptitud de los Individuos

Evaluación de cada gen:

$$f(g_i) = log(\frac{occ_i}{sum_i})$$

- $occ_i$: número de apariciones del contexto en la tabla.
- $sum_i$: suma $\forall T' \in \mathcal{T}$ de apariciones de contextos:

$$LC(T_{l_1}, \cdots, T_{l_{LC}}), T', RC(T_{r_1}, \cdots, T_{r_{RC}})$$

# Aptitud de los Individuos

- Si no hay entradas en la tabla para ese contexto se reduce su tamaño.
- Si incluso el contexto más corto no aparece en la tabla:

$$f(g_i) = log \frac{\#T \text{ en cualquier contexto}}{\sum_{T' \in \mathcal{T}} \#T' \text{ en cualquier contexto}}$$

# Operadores Genéticos: Cruce

- Se seleccionan dos individuos con probabilidad proporcional a su aptitud.
- Se elige aleatoriamente un punto de cruce.
- La primera parte de un padre se combina con la segunda del otro produciendo dos hijos.

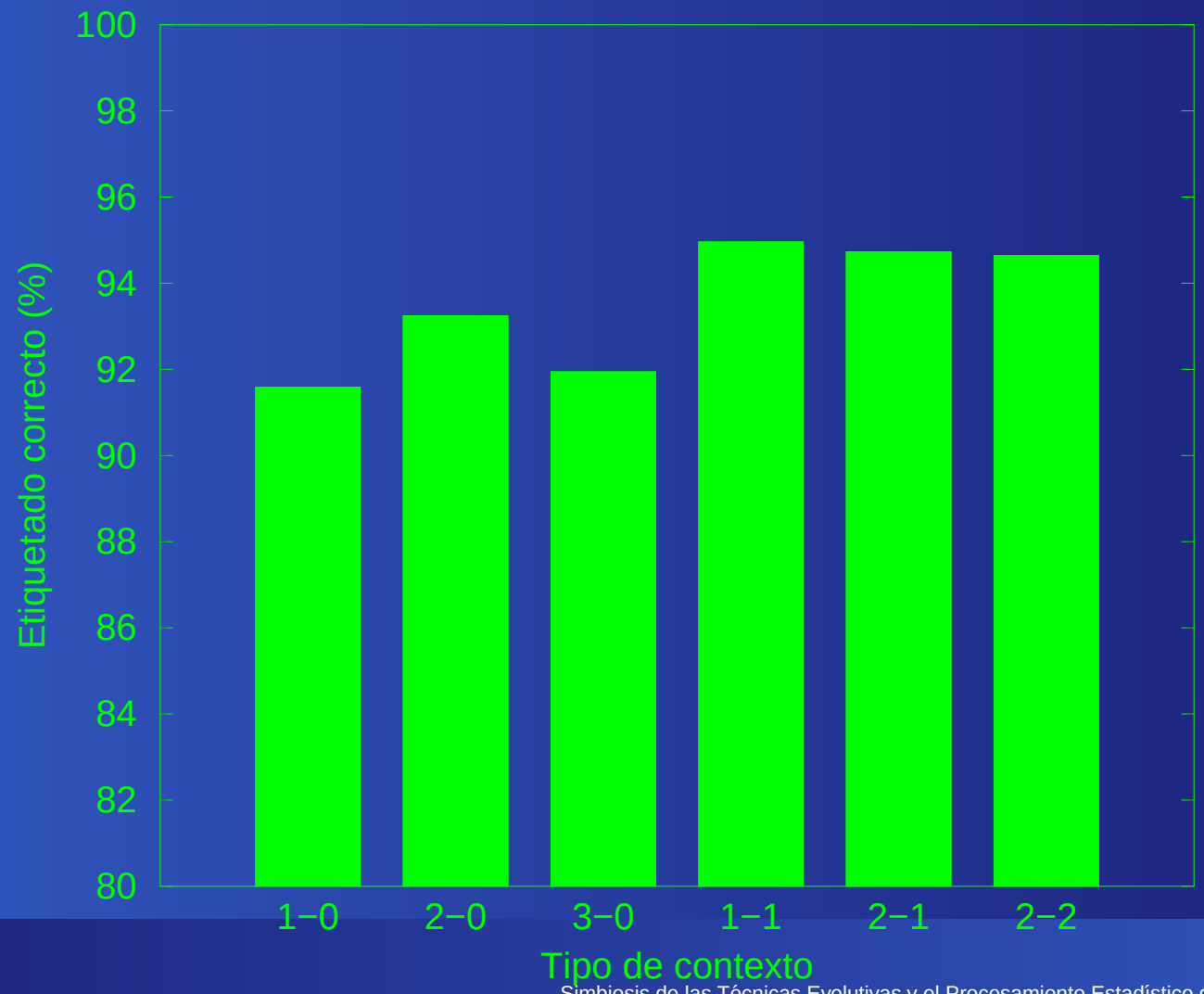
# Operadores Genéticos: Mutación

- Se aplica a los genes de los individuos procedentes del cruce, con probabilidad $P_m$.
- La etiqueta del punto de mutación se reemplaza por otra de las etiquetas válidas de esa palabra.
- La nueva etiqueta se selecciona con probabilidad proporcional a su frecuencia.

# Resultados Experimentales

- Tabla de entrenamiento obtenida a partir del corpus de Brown:
  - Tamaño apropiado del conjunto de etiquetas.
- Estudio de factores influyentes en la precisión del etiquetado:
  - Tamaño y forma de los contextos.
  - Tamaño del corpus de entrenamiento.
  - Parámetros Evolutivos.

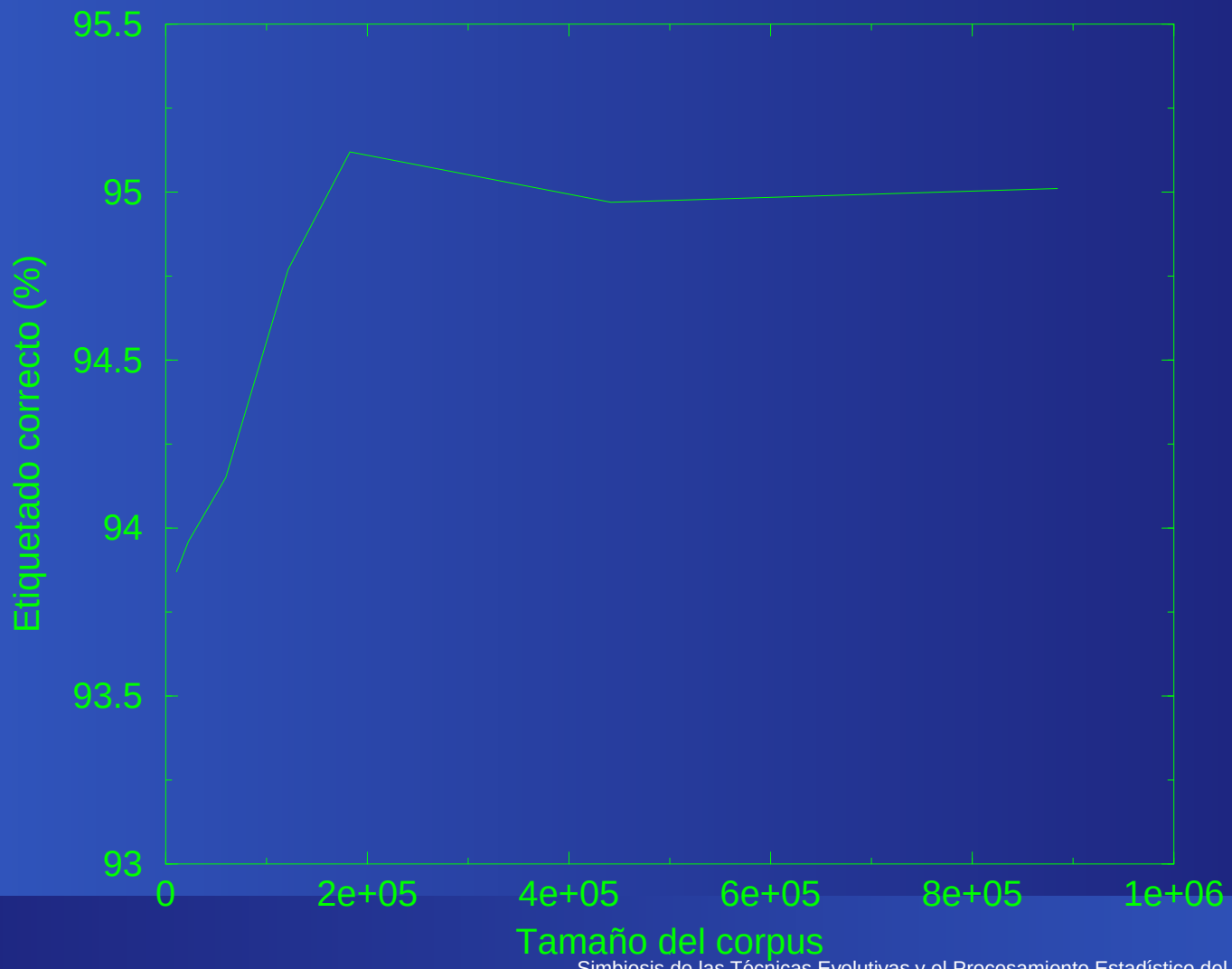
# Influencia de los Contextos



# Influencia de los Contextos

- Mejor rendimiento para contextos pequeños como 1-1.
- Contextos mayores producen entradas poco significativas en la tabla de entrenamiento.

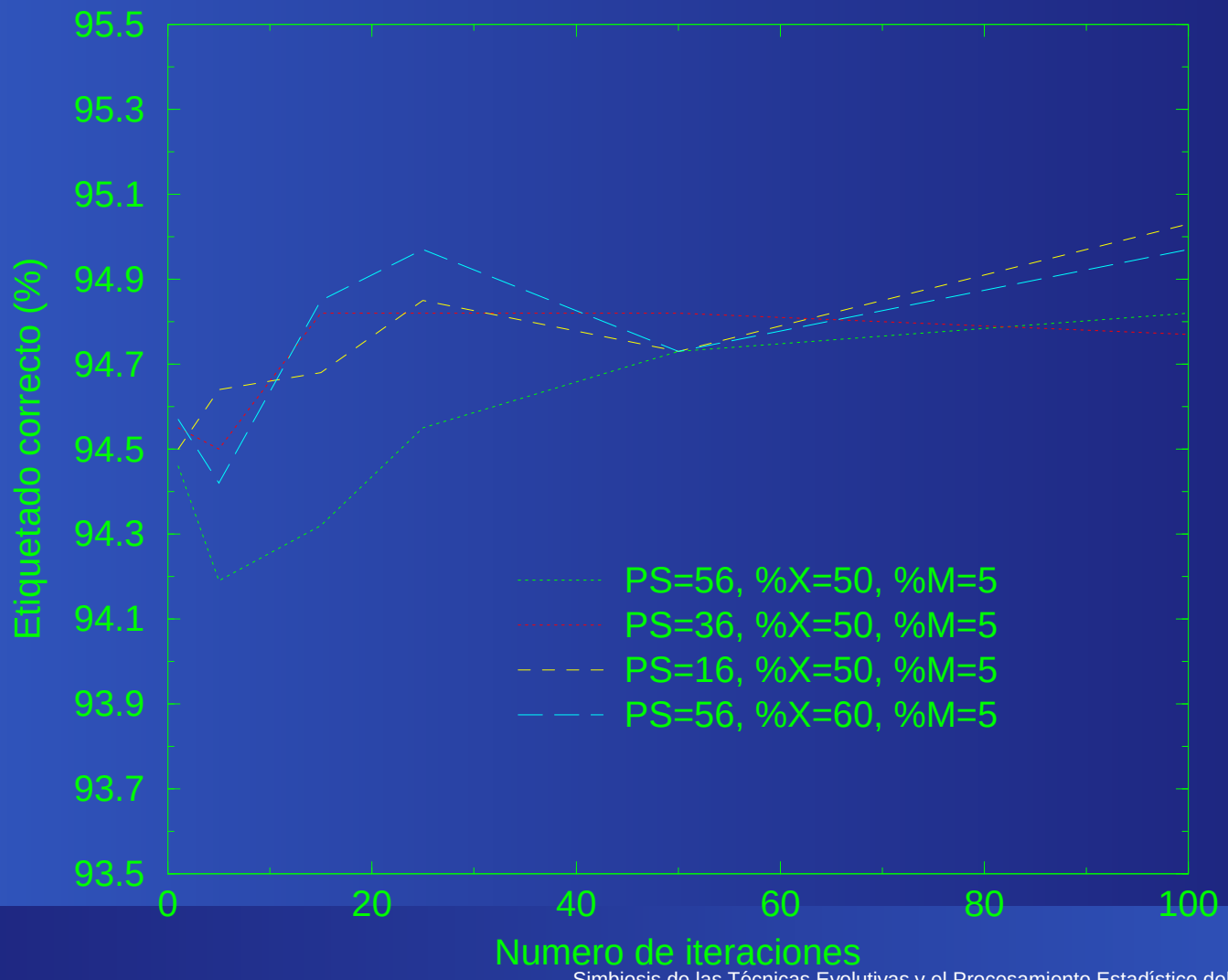
# Tamaño del texto de entrenamiento



# Tamaño del texto de entrenamiento

- El incremento de la precisión con el tamaño del corpus alcanza saturación (alrededor de 200,000 palabras).
- Resultados comparables a los obtenidos con otros enfoques probabilísticos.
- Los algoritmos evolutivos son más robustos.

# Parámetros del Algoritmo



# Parámetros del Algoritmo

- Pequeñas poblaciones son suficiente: algoritmo eficiente.
- Los porcentajes de cruce y mutación deben estar en correspondencia con el tamaño de la población.

# Conclusiones

- La programación evolutiva es suficientemente robusta para tratar el etiquetado léxico.
- Los experimentos indican la importancia de la longitud de los contextos.
- Los resultados muestran la importancia del tamaño de los textos de entrenamiento.
- Sin embargo, hay un límite en la mejora obtenida.

# Estudio de etiquetados erróneos

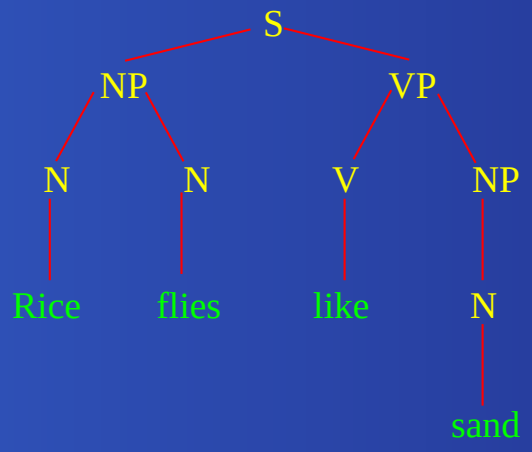
- Los resultados mejoran con la longitud de la sentencia.
- Palabras que requieren una etiqueta poco frecuente o que aparece en un contexto raro tienden a etiquetarse erróneamente.

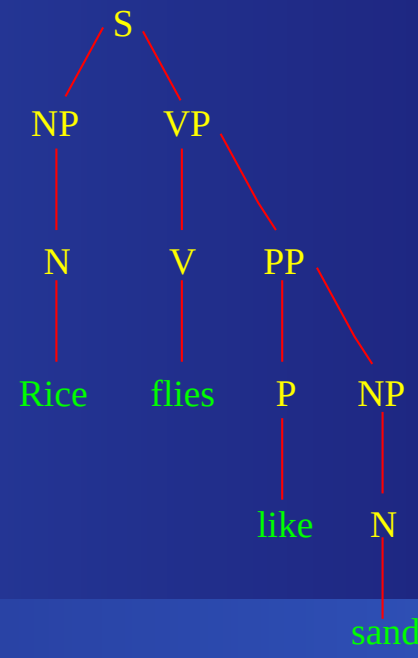
# Estudio de etiquetados erróneos

- El incremento del tamaño del texto de entrenamiento mejora los resultados de palabras que requieren una de sus etiquetas más comunes y que aparecen en contextos frecuentes.
- El etiquetado de palabras que requieren una etiqueta poco frecuente puede empeorar con el tamaño del corpus.

# Análisis gramatical

- La búsqueda del significado de una sentencia requiere extraer su estructura gramatical: análisis sintáctico.
- Ambigüedad gramatical:



# Análisis gramatical

- El análisis es un proceso de búsqueda de las estructuras correctas.
- Las gramáticas probabilísticas permiten establecer preferencias entre estas estructuras: optimización $\Longrightarrow$ programación evolutiva.
- El problema es aún muy complejo $\Longrightarrow$ Paralelización

# Gramáticas probabilísticas

- Asignan una probabilidad a cada regla de la gramática.
- La probabilidades de las reglas de una misma categoría sintáctica suman uno.
- La probabilidad de un análisis es el producto de las probabilidades de las reglas usadas en su construcción.

# Análisis Evolutivo: Individuos

- Los cromosomas son posibles análisis para la sentencia y gramáticas dadas.
- El conjunto de categorías léxicas de cada palabra de la sentencia de entrada se buscan en un diccionario (lexicón)
- Un cromosoma contiene una lista de genes que son análisis de secuencias de palabras de la sentencia.

# Análisis Evolutivo: Individuos

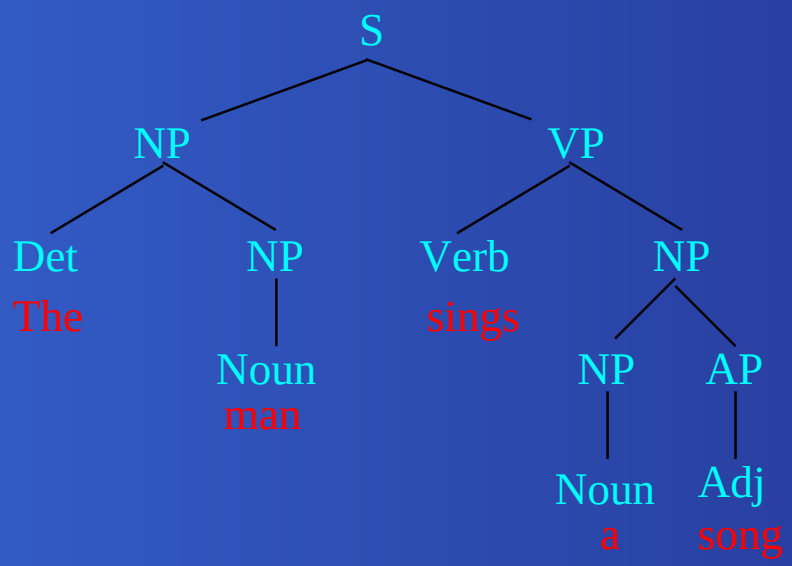
Cada gen contiene:

- Secuencia de palabras que le corresponde analizar.
- Regla gramatical usada.
- Si el lado derecho de la regla tiene símbolos no terminales, lista de referencias a los genes que realizan los análisis de esos símbolos.

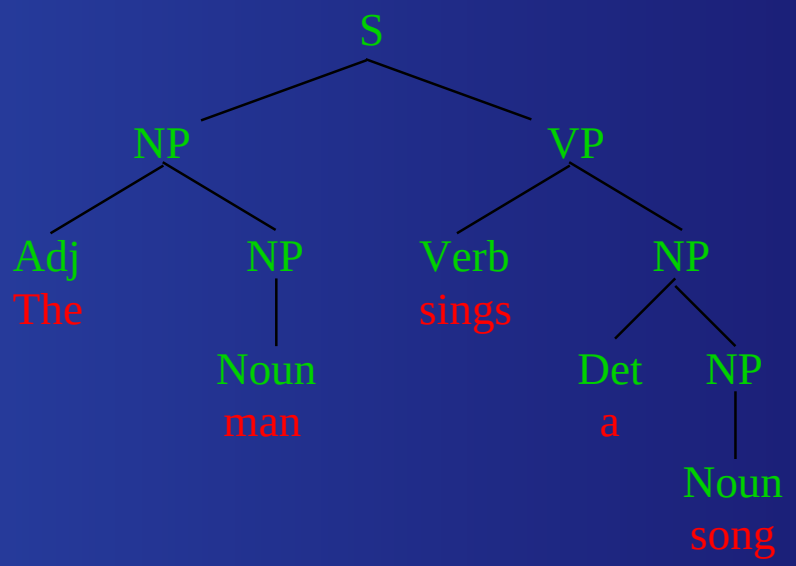
# Estructura de datos

| (n. of genes): 4 | | "the man sings a song" |
|---|---|---|
| (n. gen) | (regla) | (descomposición): (primera pal., n pal, gen): |
| (1) | $S \rightarrow NP, VP$ | NP:(1, 2, 2) |
| | | VP:(3, 3, 3) |
| (2) | $NP \rightarrow Det, Noun$ | Det: $The$ |
| | | Noun: $man$ |
| (3) | $VP \rightarrow Verb, NP$ | Verb: $sings$ |
| | | NP:(4, 2, 4) |
| (4) | $NP \rightarrow Det, Noun$ | Det: $a$ |
| | | Noun: $song$ |

# Cromosomas ejemplo




Cromosoma 1




Cromosoma 2

# Población Inicial

Generación aleatoria con condiciones:

- El conjunto de palabras de la sentencia se divide aleatoriamente, dejando al menos un verbo en la segunda parte (VP principal).
- Las palabras asignadas al $NP$ se analizan eligiendo aleatoriamente una regla $NP$. Las palabras del $VP$ se analizan con una regla $VP$ elegida aleatoriamente.
- Se da preferencia a las reglas capaces de analizar el número correcto de palabras del gen.

# Función de evaluación

$$Fitness = w_{\text{coher}} f_{\text{coher}} + w_{\text{prob}} f_{\text{prob}}$$

- $f_{coher}$ mide la capacidad del cromosoma para analizar la sentencia objetivo.
- $f_{prob}$ mide la probabilidad de las reglas empleadas en el análisis.

$$f_{prob} = \prod_{i=1}^{n} \text{Prob}(g_i)$$

donde $\text{Prob}(g_i)$ es la probabilidad de la regla del gen $g_i$.

# Función de evaluación

$f_{coher}$ se basa en el número relativo de genes coherentes. Un gen es coherente si

a) corresponde a una regla cuyo lado derecho sólo tiene terminales, y estos se corresponden con la categorías de las palabras que analizan.

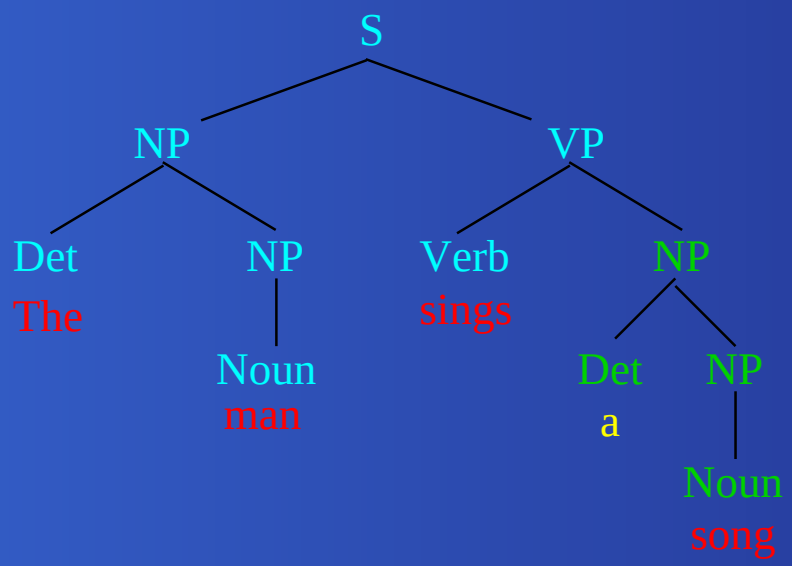
b) si corresponde a una regla con no-terminales y cada uno de ellos se analiza por un gen coherente.

# Operadores genéticos: Cruce

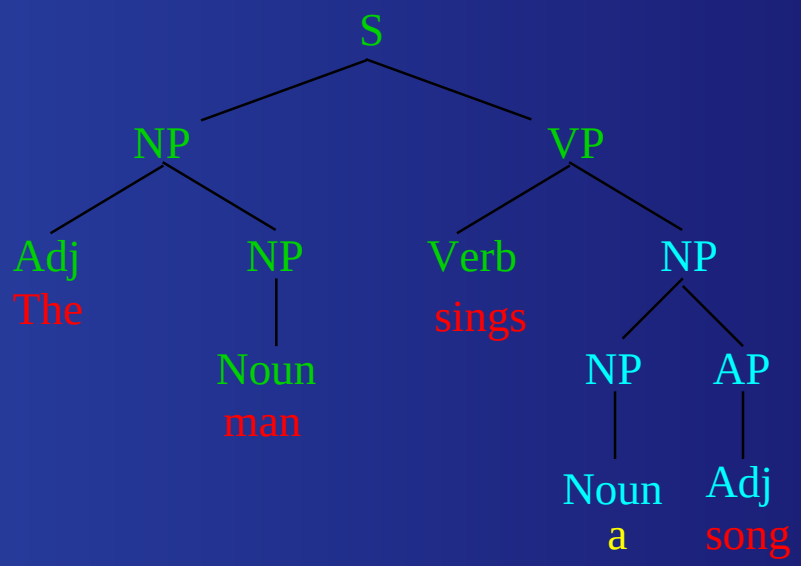
El cruce intercambia los subárboles más pequeños de dos padres que contienen una palabra seleccionada aleatoriamente y cumplen:

- Los subárboles (genes) intercambiados corresponden a la misma categoría sintáctica (NP, VP, etc).
- Los intercambios no producen inconsistencias en las secuencia de palabras de la sentencia.

# Cruce de los cromosomas ejemplo




Hijo 1

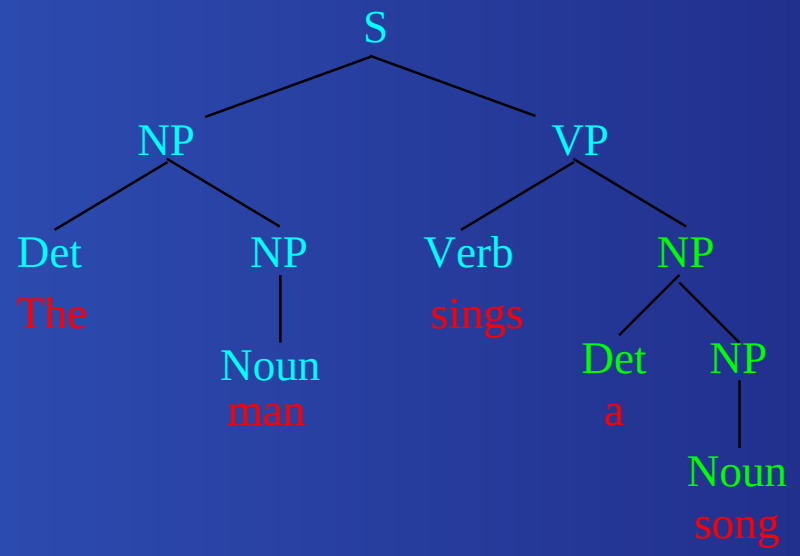



Hijo 2

# Operadores Genéticos: Mutación

Genera un nuevo análisis para un gen seleccionado aleatoriamente.

Mutación en el cromosoma 1

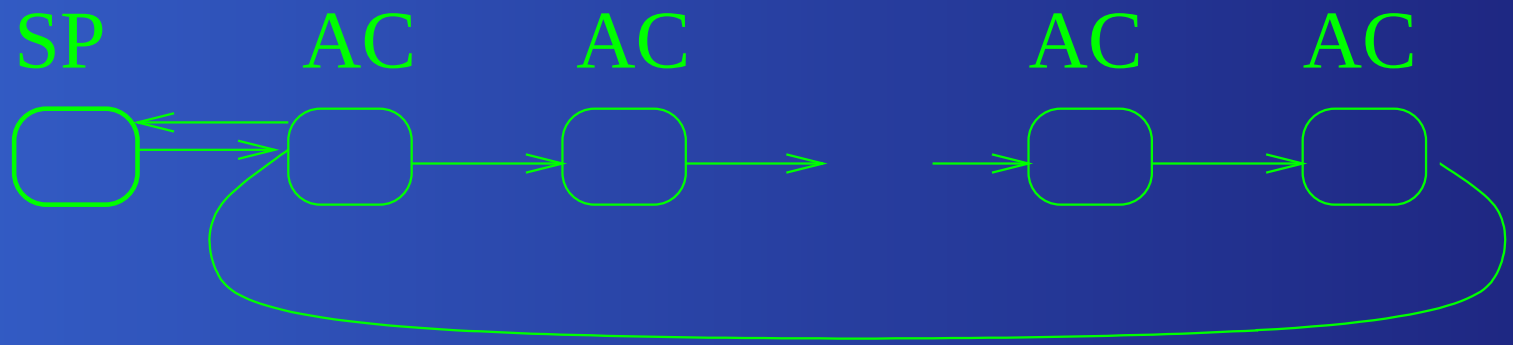


# Modelo Paralelo de **Islas**

Componentes del sistema:

- Analizadores Cooperativos (AC)
- Selector principal (SP)

Política de Migración:



# Modelo Paralelo de **Islas**

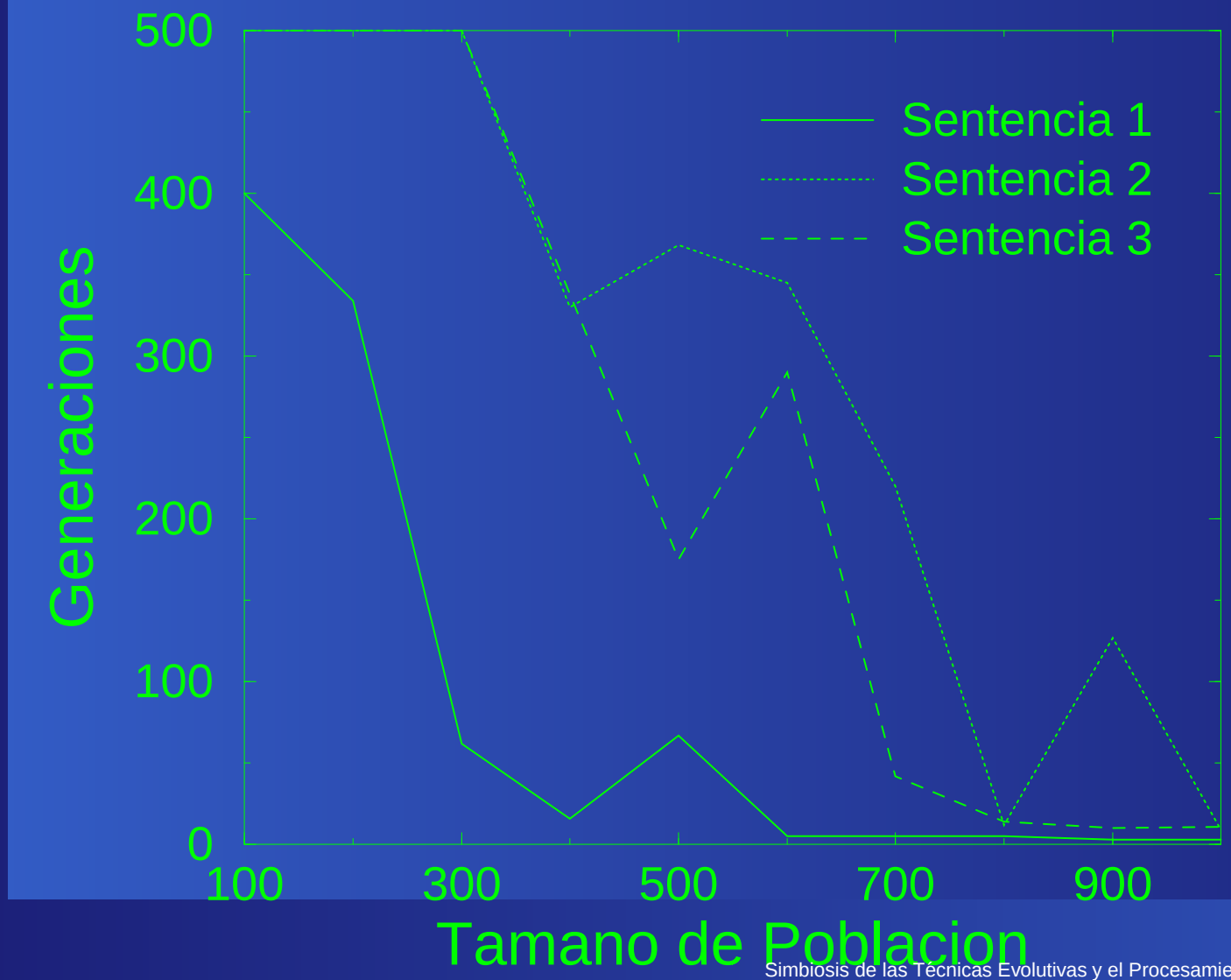
- Modelo Asíncrono
- Política de convergencia: Un analizador cooperativo que alcanza convergencia envía su mejor individuo al selector principal.
- Selección de los individuos a migrar: elegidos aleatoriamente con probabilidad proporcional a la aptitud.
- Selección de los individuos a reemplazar por los 'inmigrantes': aleatoriamente con igual probabilidad.

# Resultados Experimentales

Implementado en C++ con PVM sobre un SGI-Cray ORIGIN 2000.
Sentencias usadas en los experimentos:

| | |
|---|---|
| 1 | Jack(noun) regretted(verb) that(wh) he(pro) ate(verb) the(det) whole(adj) thing(noun) |
| 2 | The(det) man(noun) who(wh) gave(verb) Bill(noun) the(det) money(noun) drives(verb) a(det) big(adj) car(noun) |
| 3 | The(det) man(noun) who(wh) lives(verb) in(prep) the(det) red(adj) house(noun) saw(verb) the(det) thieves(noun) in(prep) the(det) bank(noun) |

# Ejecución secuencial



# Ejecución paralela

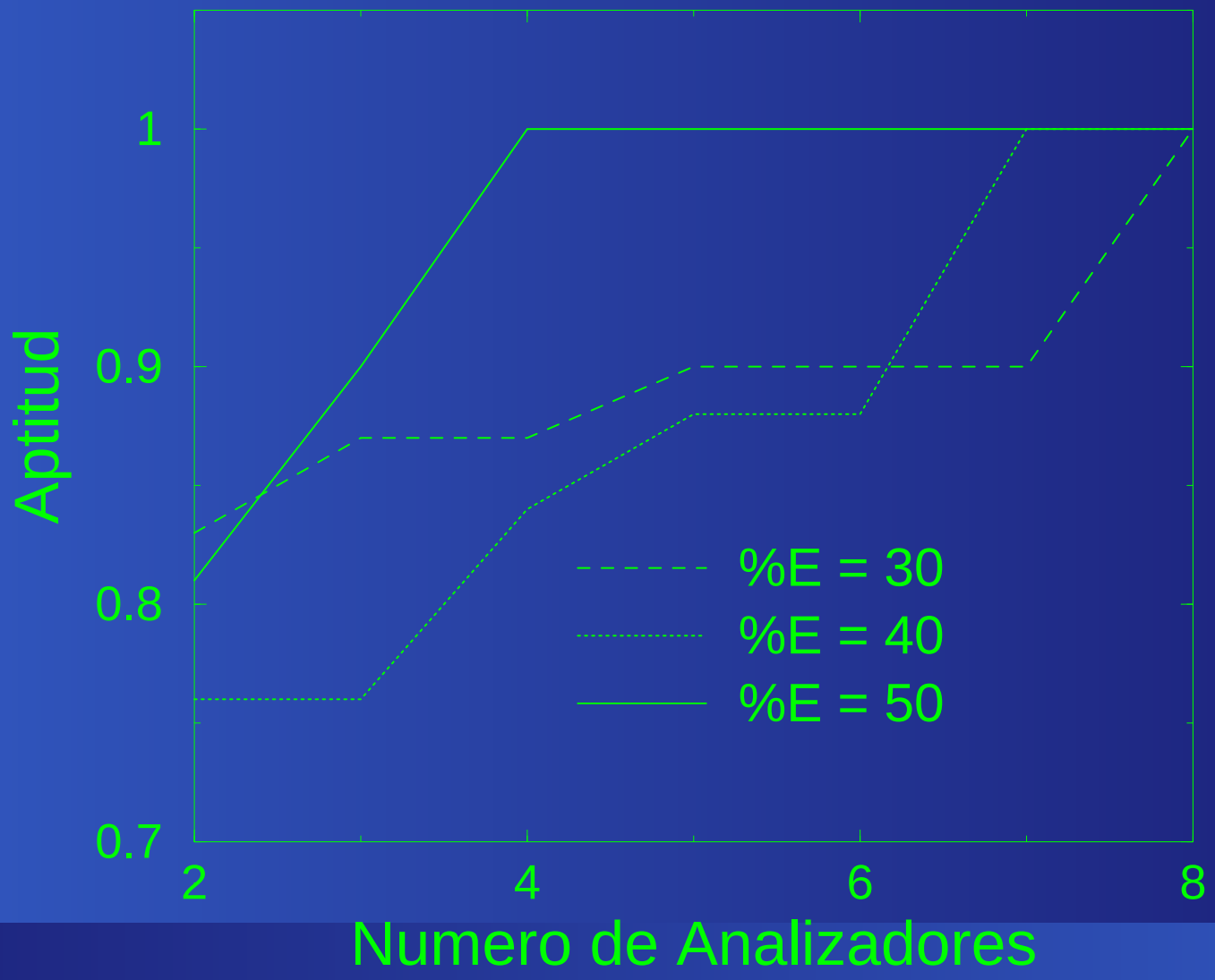
| Sent | Sec. | Paralela | | | | |
|---|---|---|---|---|---|---|
| | | 2 P. | 4 P. | 6 P. | 8 P. | 10 P. |
| sent1 | 16.55 | 10.48 | 3.08 | 3.09 | 2.09 | 2.09 |
| sent2 | 50.03 | 19.12 | 15.02 | 10.64 | 3.48 | 3.49 |
| sent3 | 52.70 | 25.40 | 22.71 | 19.34 | 14.93 | 14.79 |

Tiempo en segundos. Población de 200. %C = 50%. %M = 20%. Población emigrante de 40. Intervalo de 15 generaciones entre migraciones.

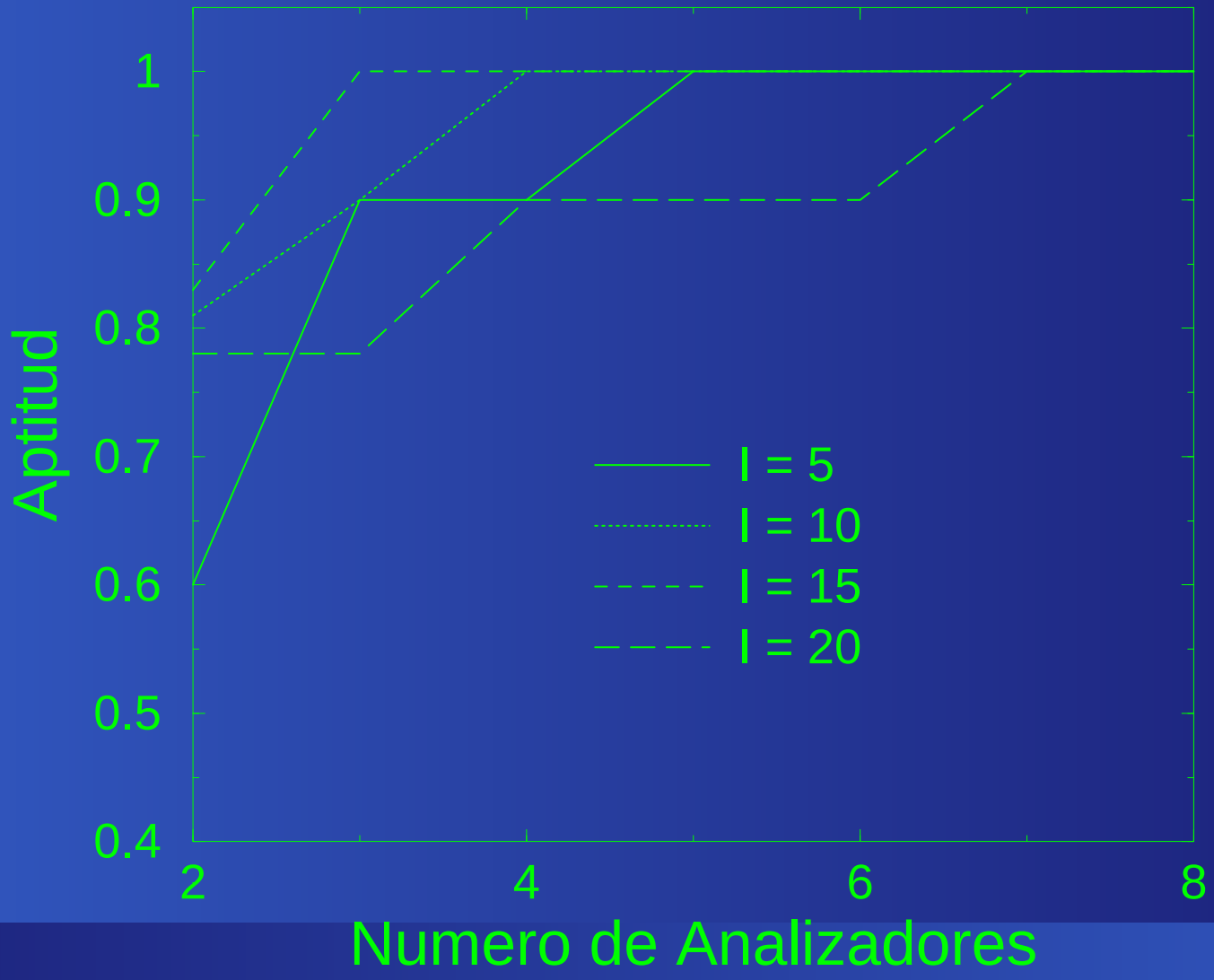
# Ejecución paralela

- La ejecución paralela consigue una mejora importante incluso con solo 2 procesadores.
- Se alcanza saturación para cierto número de procesadores.

# Tamaños de la población emigrante



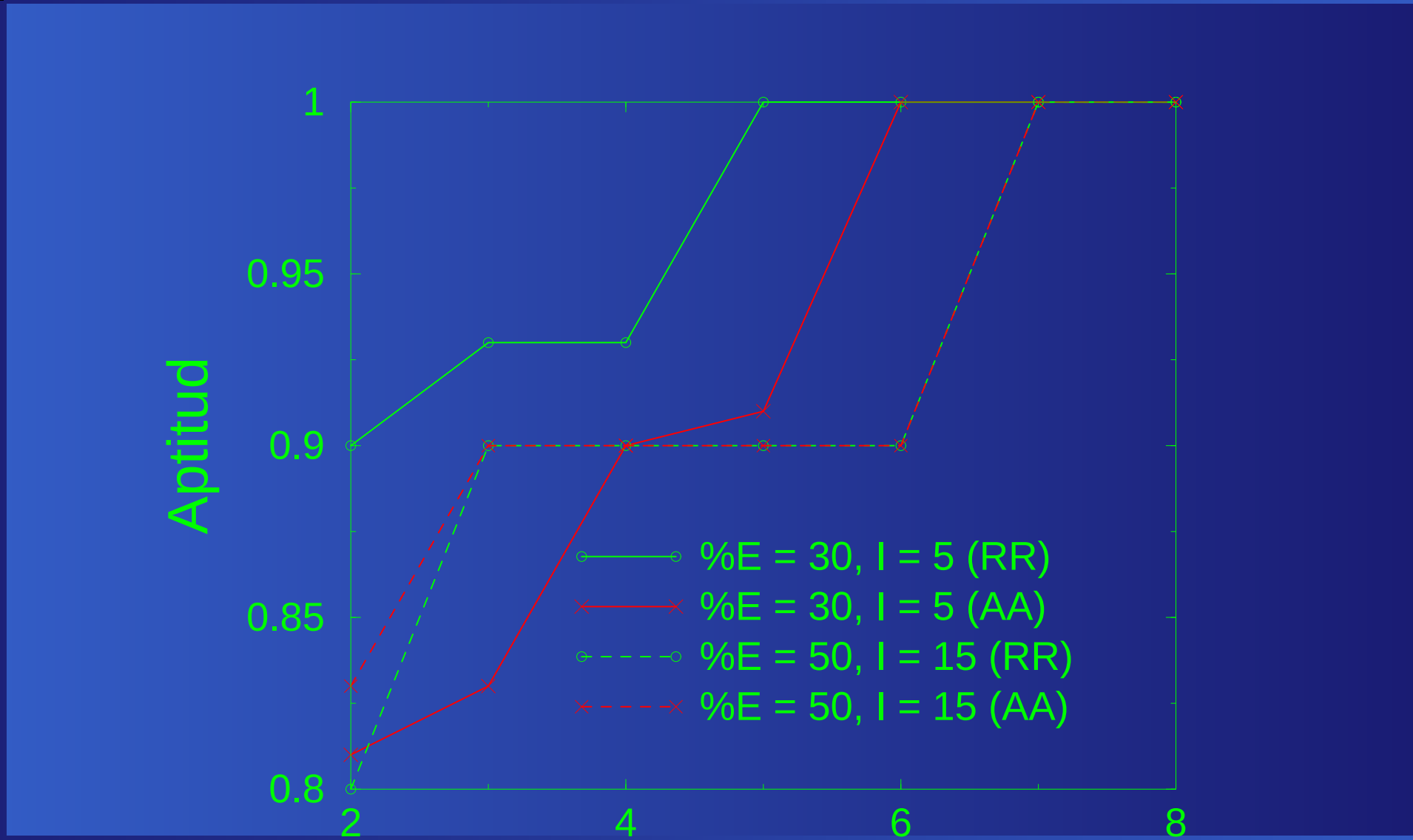
# Intervalos de migración



# Comparación de políticas de migración

- Round-Robin: cada analizador envía la población emigrante al siguiente en una secuencia anular.
- Todos-a-todos (All-to-all): Cada analizar envía la población emigrante a todos los demás.

# Comparación de políticas de migración



# Comparación de políticas de migración

- Los resultados de ambas políticas son similares, aunque la round-robin es ligeramente mejor.
- Se adopta esta política, que obteniendo resultados similares reduce las comunicaciones.

# Conclusiones

- La programación evolutiva es válida para tratar el problema del análisis sintáctico.
- El problema tiene suficiente granularidad para ser paralelizado en forma de modelo de islas.
- Los intercambios de población con una política round-robin son tan efectivos como con una política todos-a-todos.