Computacion Avanzada en la Prediccion de la Estructura de Proteinas

Natalio Krasnogor
www.cs.nott.ac.uk/~nxk

Proteinas a Vuelo de Pajaro

- Las proteinas son c estas por (algunos) de los 20 aminoacidos.

A partir de estos objetos!

Queremos predecir estos objetos

- Se conectan formando una cadena lineal

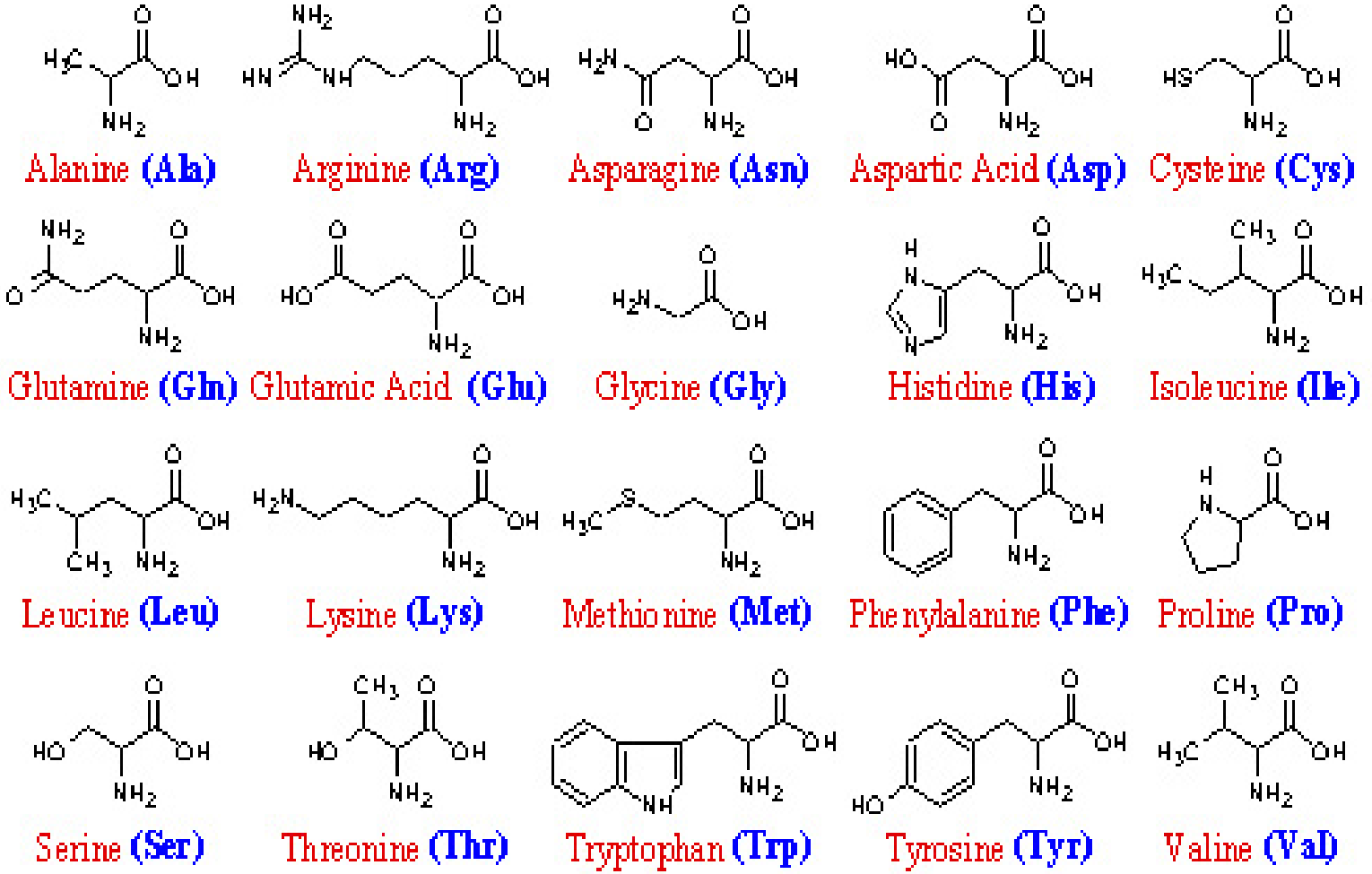
- Esta cadena lineal o secuencia se llama estructura primaria (compuesta por un string en el alphabeto de tamanio 20).

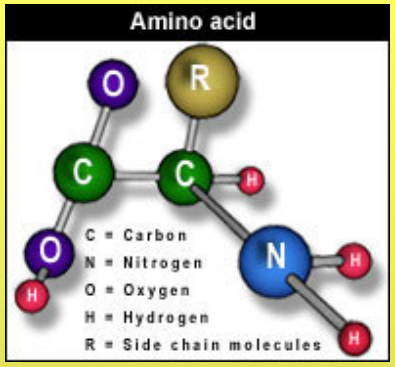
- La estructura primaria forma estructuras locales secundarias

- Las estructuras secundarias dan lugar a la estructura terciaria

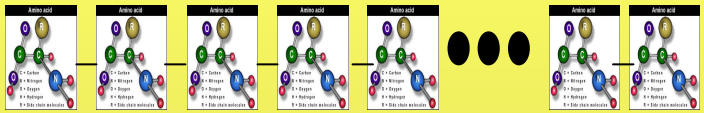
Schematic diagrams of the 20 amino acids

(picture taken from www.chemistry.pomona.edu)

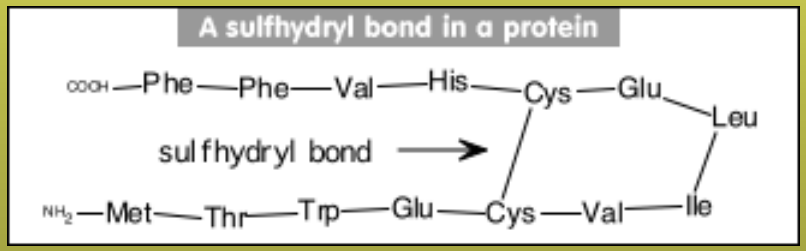




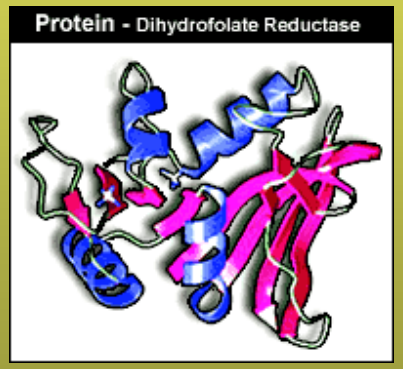
Estructura primaria



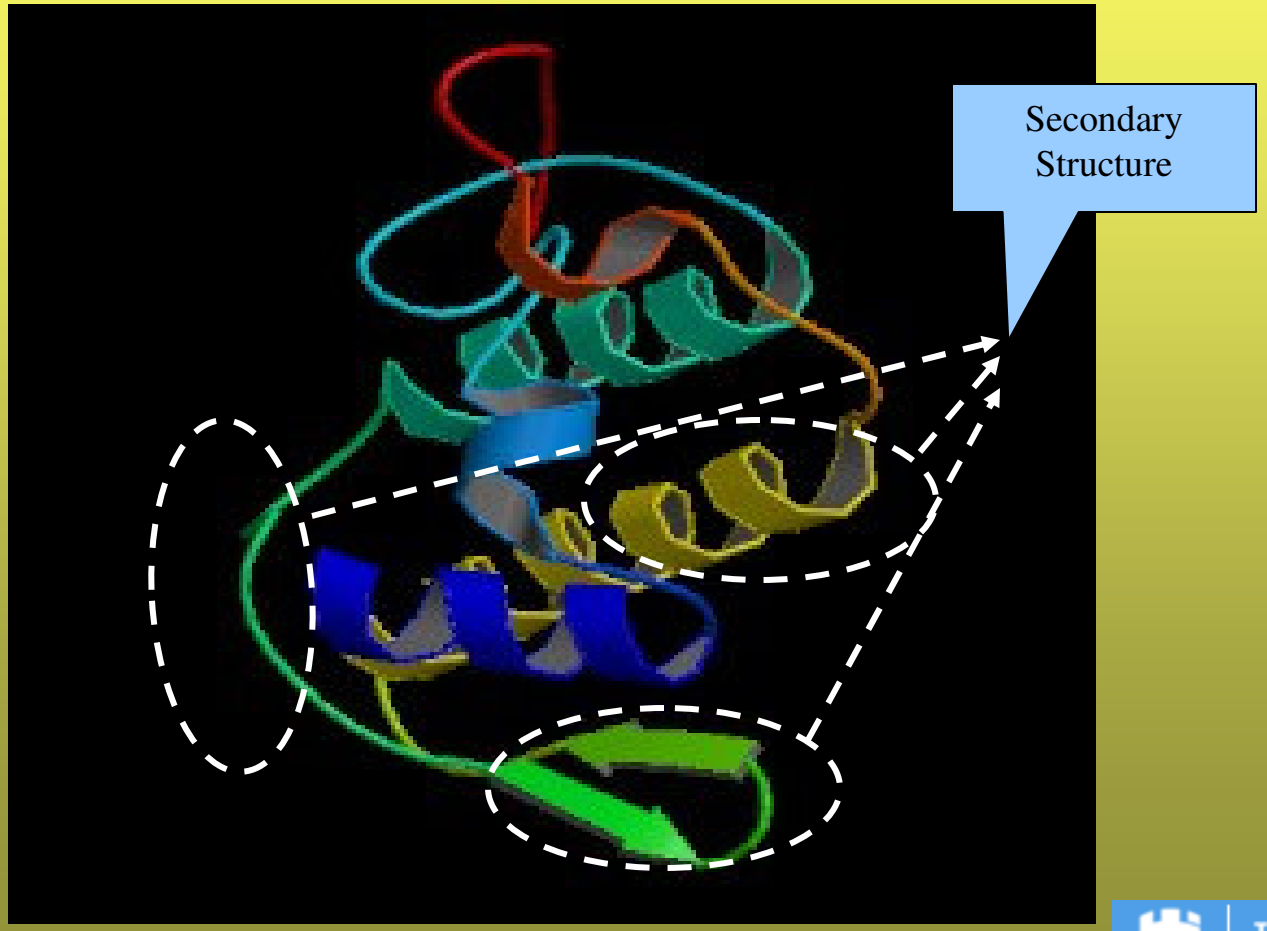
Estructura Secundaria



Estructura terciaria



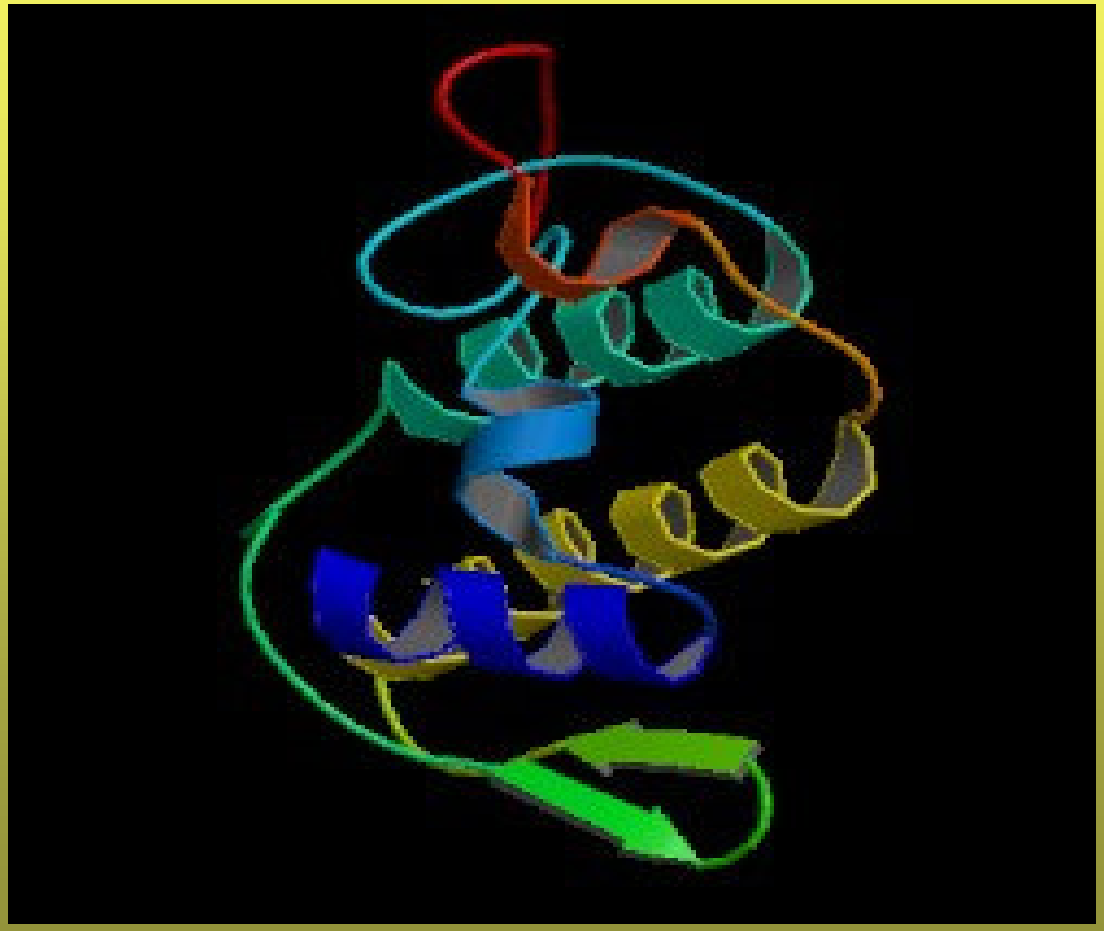
Example: 1CLP



Example: 1CLP

Tertiary Structure

Native State



El role de las proteinas en la vida

- **Proteinas estructurales:** los bloques constructivos del organismo, ej. colageno, unias, pelo, etc.

- **Enzimas:** maquinaria biological que regulan una multitud de funciones metabolicas. Generalmente las enzimas son muy especificas y catalizan un solo tipo de reaccion, pero pueden jugar algun papel en mas de una reaccion.

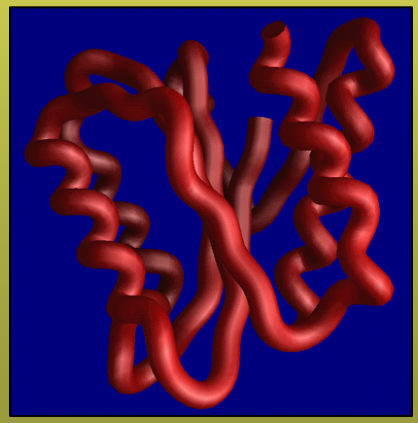
- **Proteinas de las membranas:** son las “amas de casa” de las celulas, ej. Mediante la regulacion del volumen celular, extraccion y concentracion de pequenias moleculas del ambiente extracelular y la generacion gradientes ionicos essentiales para el funcionamiento muscular y nervioso (la bomba de sodio/potasio es un ejemplo)

Problema de Prediccion de la Estructura Terciaria de Proteinas:

- INPUT: una secuencia de aminoacidos

- OUTPUT: la estructura terciaria que esta adopta

```
MKIVYWSGTGNTEK
MAELIAKGIIESGK
DVNTINVSDVNI
DELLNEDILILGCS
AMGDEVLEESEFEP
FIEEISTKISGK
KVALFGSYGWGDGK
WMRDFEERMNGYGC
VVVETPLIVQNE
PDEAEQDCIEFGKK
IANI
```



• No existe concenso entre los cientificos de cual es el modelo atomico adecuado para predecir el plegado de una proteina

• Aun cuando se conociese ese modelo, hacer una simulacion atomo-por-atomo del plegado seria muy costoso computacionalmente

• Modelos simplificados de proteinas han impactado en nuestro entendimiento:

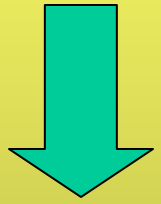
- del comportamiento de las mismas

- de los algoritmos necesarios para predecir el plegado.

Estos modelos minimalistas se han usado entre otras cosas para:

- Estudiar la naturaleza del energy landscape (NakSasSas2001)
- La unicidad del estado nativo y las secuencias degeneradas (SunBreChaDil96)
- La transicion de dos estados de estructuras de tipo globins (YueDil94)
- La existencia de plegado cooperativo, la relacion entre la estructura y la funcion (Hir99)

• Predecir la estructura de proteinas reales por medio de una combinacion de informacion experimental sobre estructura secundaria y terciaria con modelos optimisados obtenidos de simulaciones en latices, ej. (SamXiaHuaLev99, XiaHuaLevSam2000, FeiBro2000, KihLuKolSko2001b, KolBetKihRotSko2001)



Los modelos simplificados continuan jugando un rol importante en:

- la mejora de nuestro conocimiento de los fundamentos fisicos de proteinas reales

- facilitando el desarrollo de mejores algoritmos para la prediccion de estructura.

El problema de predecir el estado nativo de una proteina se descompone en:

• Modelado: especificacion y seleccion de la funcion de energia, geometria, uso de “side-chains”, etc.

• Optimisation: se concentra en las tecnicas algoritmicas que son necesarias para buscar la estructura optima de una secuencia especifica asumiendo un modelo particular.

En el resto de la charla asumiremos una familia de modelos minimalistas y nos concentraremos en los algoritmos necesarios para resolverlos.

Los Modelos Utilizados

- Modelo HP (Dil85)

- Modelo Functional Model Proteins, FMP,(Hir99,BlaHir01,BlaHir03)

- Embebidos en una geometria 2d o 3d, reticular, ej., cubicos, diamante, centrado en las caras, etc.

- HP y FMP abstraen las interacciones hidrofobicas y reducen el alfabeto de 20 aminoacidos a 2: no-polares o hidrofobicos (H) y polares (P) o hidrofilicos

- Una proteina de n aminoacidos se representa por una secuencia $s \nexists \{H,P\}^n$ que se mapea al reticulado donde cada residuo ocupa una celda diferente. El mapeo debe ser self-avoiding.

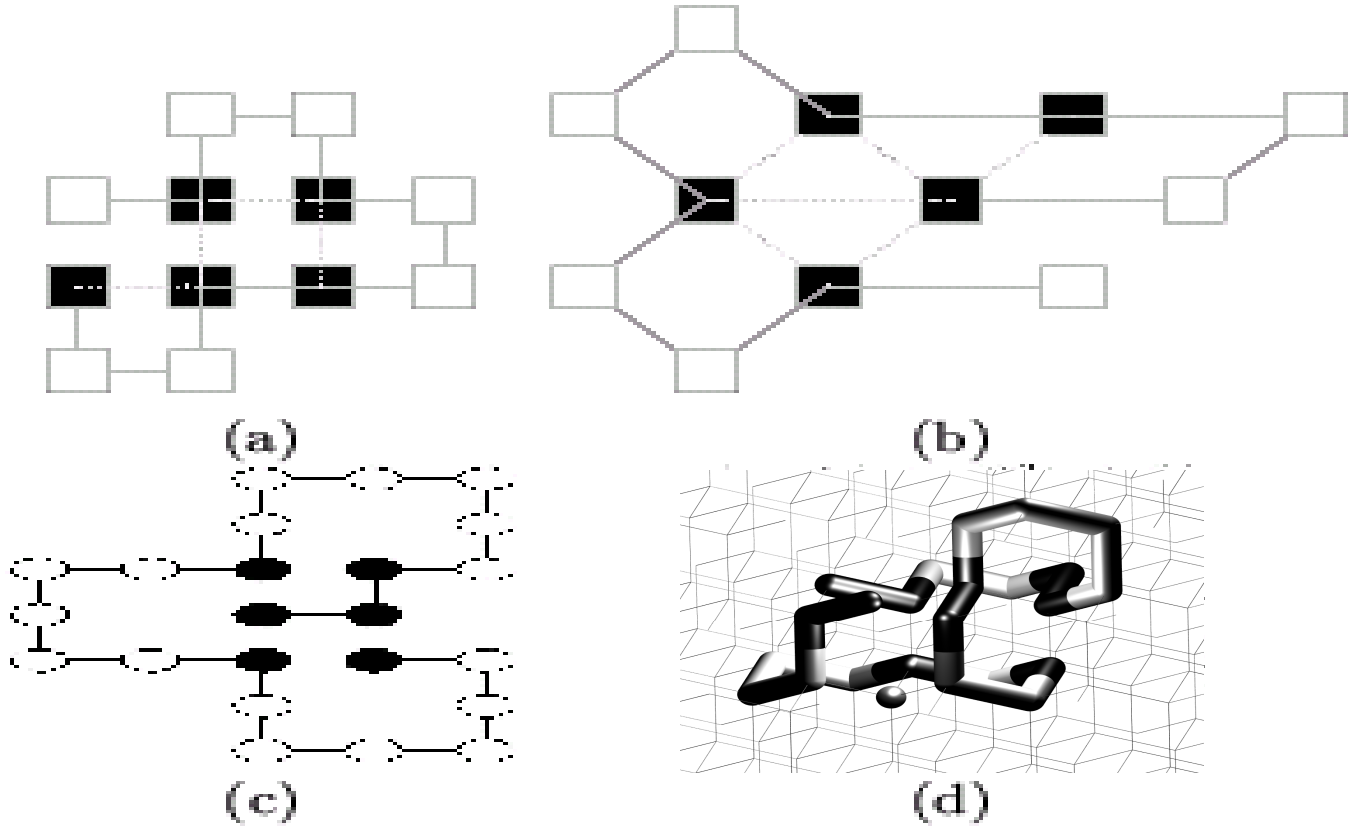
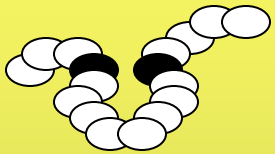


FIG. 1: *HP protein embedded in the square lattice (a) and triangular lattice(b). Functional Model protein embedded in the square lattice(c) and diamond (3D) lattice(d). In (c) and (d) native structures are not maximally compact as they must have a "binding pocket".*

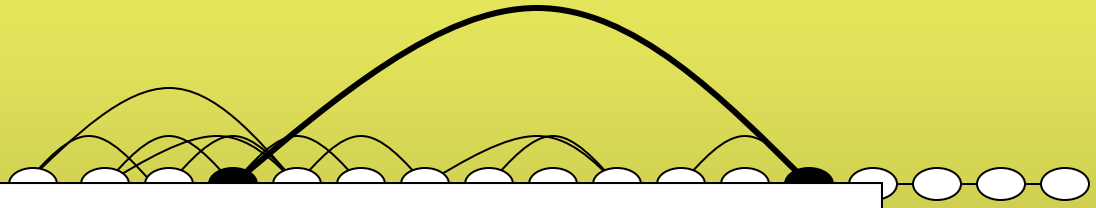
La Funcion de Energia (funcion objetivo)

Una proteina:

Su estructura:



El mapa de contacto



$$E(s) = \sum_{i<j \ ; \ 1 \leq i,j \leq n} (\Delta_{i,j} * \epsilon_{i,j})$$

where

$$\Delta_{i,j} = \begin{cases} 1 & \text{if } i,j \text{ are in contact and } |i-j| > 1 \\ 0 & \text{otherwise} \end{cases}$$

asap automated scheduling optimisation & planning research

The University of Nottingham

• Una proteina del FMP debe plegarse a un estado nativo unico, tener un “binding pocket” y un “energy gap”

• Estas restricciones adicionales al model HP lo hacen un modelo mas dificil de resolver.

• Es mas probable quedarse atrapado en un optimo local pobre

• No se puede asumir compacticidad de la estructura

• Existe solo un optimo global

• Varias versiones de los modelos HP son NP-Hard (BerLei98, Cre98).

En Resumen

- 20 tipos de residuos
  - No-polares o hydrophobicos (H)
  - Polares (P) o hydrophilicocs
- Una proteina de *n* residuos se representa por una sequencia *s*
- La secuencia se mapea al latice
- Cada residuo en s ocupa una celda del latice diferente
- El mapeo debe ser “self-avoiding”

$$E(s) = \sum_{i<j \ ; \ 1 \leq i,j \leq n} (\Delta_{i,j} \epsilon_{i,j})$$

- El potencial de energia reflaja la propensidad de los H a formar close contacts
- Standard HP model
  - HP and PP assigned energy 0
  - HH contact assigned energy -1

$$\Delta_{i,j} = \begin{cases} 1 \text{ if } i,j \text{ are in contact and } |i-j| > 1 \\ 0 \text{ otherwise} \end{cases}$$

Metaheuristicas para la Prediccion de Estructuras

(Glo86) define metaheuristicas como:

...a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality. The heuristics guided by such a meta-strategy may be high-level procedures or may embody nothing more than a description of available moves for transforming one solution into another, together with an associated evaluation rule.

Algunas de las metaheuristicas mas exitosas para estos modelos incluyen:

- Automatas Celulares (KraMarPelRiz98)
- Algoritmos Geneticos (Krasnogor99)
- Hibridos Evolutivos-Monte Carlo (LiaWon2001)
- Busqueda por Vecindarios Variables Fuzzy (KraPel2002)
- Algoritmos Memeticos (KraBlaBurHir2002)
- Optimisacion por Colonia de Hormigas (ShmAguHoo02)
- Hibridos Geneticos-Busqueda Tabu (JiaCuiShiMa03)

Principios de Disenio para Algoritmos Exitosos

Los algoritmos mas exitosos usan al menos 3 de los siguientes:

1. The use of global search mechanisms in contrast with purely local search.
2. The explicit use of diversity preserving mechanisms.
3. The use of a number of distinct move operators rather than a single move operator.
4. The explicit use of mechanisms to jump-out of poor local optima and/or traverse large neutral plateaus.
5. The use a "memory" of visited configurations.

El primero principio de disenio es generalmente implementado por medio de un (multi)set de soluciones.

Teniendo varias soluciones es mas dificil quedarse atrapado en optimos locales pobres.

Sin embargo, este principio se relaciona con el segundo, la preservacion de la diversidad. En el peor caso todo el (multi)set representaria la misma solucion lo cual implicaria una perdida de recursos computacionales.

Las soluciones exploradas deben ser diversas de tal manera que representen varias regiones distintas del espacio de busqueda.

En relacion a la diversidad y a los optimos locales se introduce el tercer principio de disenio: el uso de varios operadores de busqueda.

Si el algoritmo tiene a su disposicion una variedad de “move operators” entonces es mas dificil que pierda diversidad o que se quede atrapado en optimos locales pobres.

En general los optimos locales de va alentes

Sin embargo, este mecanismo por si solo le pu antizar no quedarse atrapado en optimos loca

El cuarto principio de diseni punta a ello. Hay que incluir mecanismos que detecten optimos locales (o mesetas) para poder escaparse de ellos.

Finalmente, el quinto principio apunta a mantener alguna forma de memoria de configuraciones visitadas.

- principios no ortogonales
- pueden ser conflictivos

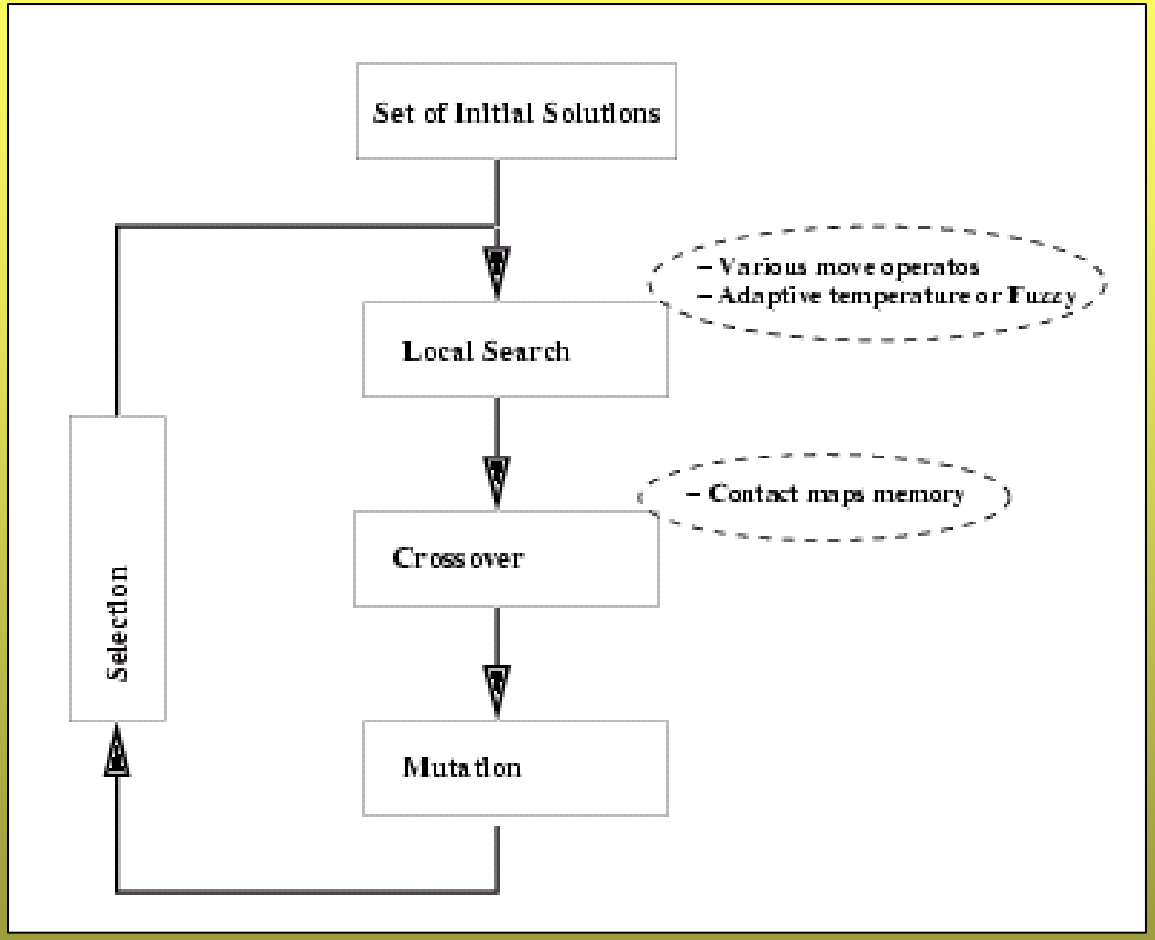
Ejemplos de Algoritmos Recientes que Emplean Algunos de Estos Principios

Hibridos Evolutivos-Monte Carlo

(KraSmi99b) utiliza 1 (poblacion) , 2 (diversificacion) & 4 (evitar optimos locales)

Usa un hibrido (algoritmo memetico) Evolutivo-Monte Carlo

El Evolutivo implementa el principio 1 y el MC modificado el 2&4


Set of Initial Solutions
– Various move operatos
– Adaptive temperature or Fuzzy
Local Search
– Contact maps memory
Crossover
Mutation
Selection

El MC modificado sirve de buscador local cuando la poblacion es diversa y de diversificador cuando la poblacion del Memetico es homogenea

La temperatura del MC se define antes de cada ejecucion como:

$$T = \frac{1}{|E(v_h) - E(v_l)|}$$

Y se acepta una nueva solucion de acuerdo a :

$$random(0,1) < \exp^{-k * \frac{\delta_{v,v_{random}}}{T}}$$

cuando la poblacion es diversa, $T \rightarrow 0$, acepta solo mejoras

cuando la poblacion es Homogenea, $T \rightarrow \infty$, acepta mejores/peores soluciones

(LiaWon2001) tambien usa un Memetico con MC e implementan los principles 1, 2 & 3:

• mantienen una poblacion de soluciones (1)

• usan varios operadores de busqueda que le permite al algoritmo inducir un paisaje de busqueda rico (3)

• incluyen un operador de intercambio para las temperaturas del MC lo que le permite al algorimo mantener la diversidad (2) ya que puede funcionar como muy explorativo (altas T) a muy explotativo (bajas T):

- parallel tempering (GeyTho95, YanPab99)

- single chain Metropolis-Hastings algorithm.

Colonia de Hormigas

En (ShmAguHoo02) se implementa una CH

A diferencia de los Memeticos con MC, la CH mantiene una poblacion de hormigas (no soluciones explicitas) (1).

Cada hormiga tiene asociado un vector de probabilidades de instanciar una solucion particular

El principio 2 (diversidad) se implementa con un operador de re-inicio que realoca probabilidades a estos vectores

Ademas la CH utiliza dos operadores de busqueda local (principio 3)

Hibridos Evolutivos- Busqueda de Operador Variable Fuzzy

En (KraPel2002) los autores usan busqueda de vecindarios variables fuzzy (FANS).

Los principios 1, 3 & 4 son empleados.

Un algoritmo evolutivo implementa el primer principio y se hibridiza con FANS.

FANS se usa para enfocar la busqueda en buenas regiones del espacio de busqueda inducido por varios operadores (3)

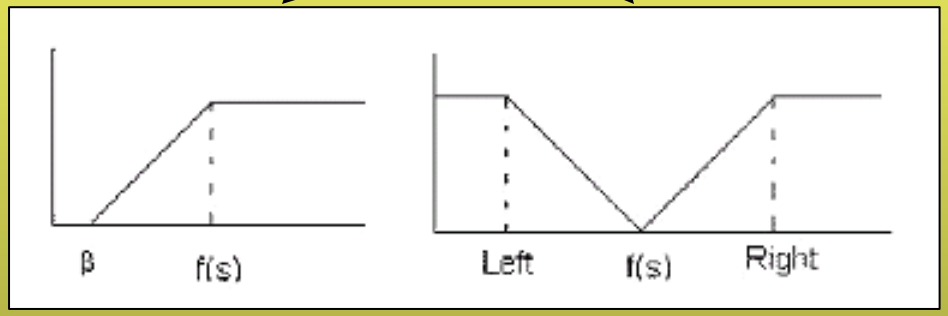
Un criterio fuzzy se usa de manera semejante al MC modificado de manera que de vez en cuando se aceptan soluciones con energias mas altas (4).

$$\mu(E(v), E(v_{new})) \geq \lambda$$

funcion de pertenencia al conjunto de soluciones "aceptables"

Dos conceptos distintos de aceptabilidad

Promueve mejoras



Promueve diversidad

$$\mu_1(v, v_{new}) = \begin{cases} 0.0 & \text{if } E(v_{new}) > \beta \\ \frac{(E(v_{new}) - \beta)}{(E(v) - \beta)} & \text{if } \beta \geq E(v_{new}) \geq E(v) \\ 1.0 & \text{if } E(v_{new}) < E(v) \end{cases}$$

$$\mu_2(v, v_{new}) = \begin{cases} 1.0 & \text{if } Right \geq E(v_{new}) \geq Left \\ \frac{(E(v) - E(v_{new}))}{(E(v) - Left)} & \text{if } Left > E(v_{new}) \geq E(v) \\ \frac{(E(v_{new}) - E(v))}{(Right - E(v))} & \text{if } E(v) > E(v_{new}) \geq Right \end{cases}$$

Algunos Resultados (1)

Static Move Operators	#O/#R	Mean FHT
GA (no move operators)	0/10	-
MA with Macro Mutation (r=4)	2/10	27.5
MA with Macro Mutation (r=8)	3/10	53.3
MA with Macro Mutation (r=16)	2/10	43.0
MA with Reflect (r=4)	3/10	20.6
MA with Reflect (r=8)	1/10	79.0
MA with Reflect (r=16)	1/10	45.0
MA with Stretch (r=4)	0/10	-
MA with Stretch (r=8)	0/10	-
MA with Stretch (r=16)	0/10	-
MA with Pivot	5/10	27.0
MultiMeme (all local searchers)	8/10	16.87

Principio 3: cero, uno o varios operadores
modelo hp 2d, rectangular

Algunos Resultados (2)

#	Sequence	Size	Opt.	MMA
1	$HPHPPHHPHPPHPHHPPHPH$	20	-9	-9
2	$PPPHHPPHHPPPPPPHHHHHHHPPHHPPPPPHHPPHPP$	36	-14	-14
3	$H^2(PH)^4H^3PHP^3HP^3HP^4HP^3HP^3HPH^4(PH)^4H$	50	-21	-21
4	$H^{12}(PH)^2(P^2H^2)^2(PPH)^2(HP^2H)^2(P^2H^2)^2P^2(HP)^2H^{12}$	**64**	**-42**	**-39**
5	$HPHPPHHPHPPHPHHPPHPH$	20	-9	-9
6	$PPHPPHHPPPPHHPPPPHHPPPPHH$	25	-8	-8
7	$(P^2H)^2HP^2H^2P^5H^{10}P^6(H^2P^2)^2HP^2H^5$	48	-22	-22(-23)
8	$PHPPHPHHHPHHPH^5$	18	-9	-9
9	$HPHPHHHPPPHHHHPPHH$	18	-8	-8
10	$HHPPPPPHHPPPHPPPHP$	18	-4	-4
11	$HHHPPHPHPHPPHPHPHPPH$	20	-10	-10
12	$PPH^3PH^8P^3H^{10}PHP^3H^{12}P^4H^6PHHPHP$	60	-34	-34(-35)

Principio 4: Fuzzy logic para salir de optimos locales.
Nuevos optimos en el modelo HP, 2d

Algunos Resultados (3)

#	Sequence	Size	Opt.	MMA
1	*HHPPHPHPHPHPHP*	14	-11	-11
2	*HHPPHPPHPHPHPH*	14	-11	-11
3	*HHPHPPHPPHPPHPPH*	16	-11	-11
4	*HHPPHPPHPHPHPPHP*	16	-11	-11
5	*HHPPHPPHPPHPPHPPH*	17	-11	-11
6	*HHPHPHPHPHPHPHPHH*	17	-17	-17
7	*HHPPHPPHPHPHPPHPHPHH*	20	-17	-17
8	*HHPHPHPHPHPPHPPHPPHH*	20	-17	-17
9	*HHPPHPPHPHPPHPHPPHPHH*	21	-17	-17
10	*HHPHPPHPPHPHPHPPHPPHH*	21	-17	-17
11	*HHPPHPHPHPPHPHPPHPPHH*	21	-17	-17
12	*HHPPHPPHPHPHPPHPPHPPHH*	22	-17	-17
13	*HHHPHPHPHPHPHPHPHPHPHHH*	23	-25	-25
14	*HHPPHPPHPPHPPHPPHPPHPPHH*	**24**	**-17**	**-16**
15	*HHHPHPHPPHPHPHPHPHPHPHHH*	24	-25	-25
16	*HHHPHPHPHPPHPHPHPHPHPHHH*	24	-25	-25
17	*HHHPPHPPHPPHPPHPHPPHPHPPHPPHHH*	**30**	**-25**	**-24**
18	*HHHPPHPPHPPHPHPPHPHPPHPPHPPHHH*	**30**	**-25**	**-24**
19	*HHHPPHPPHPPHPHPHPPHPPHPPHPPPPPHPHPHHH*	**37**	**-29**	**-26**

Principio 3: MC modificado en el reticulado triangular, modelo hp 2d.

Algunos Resultados (4)

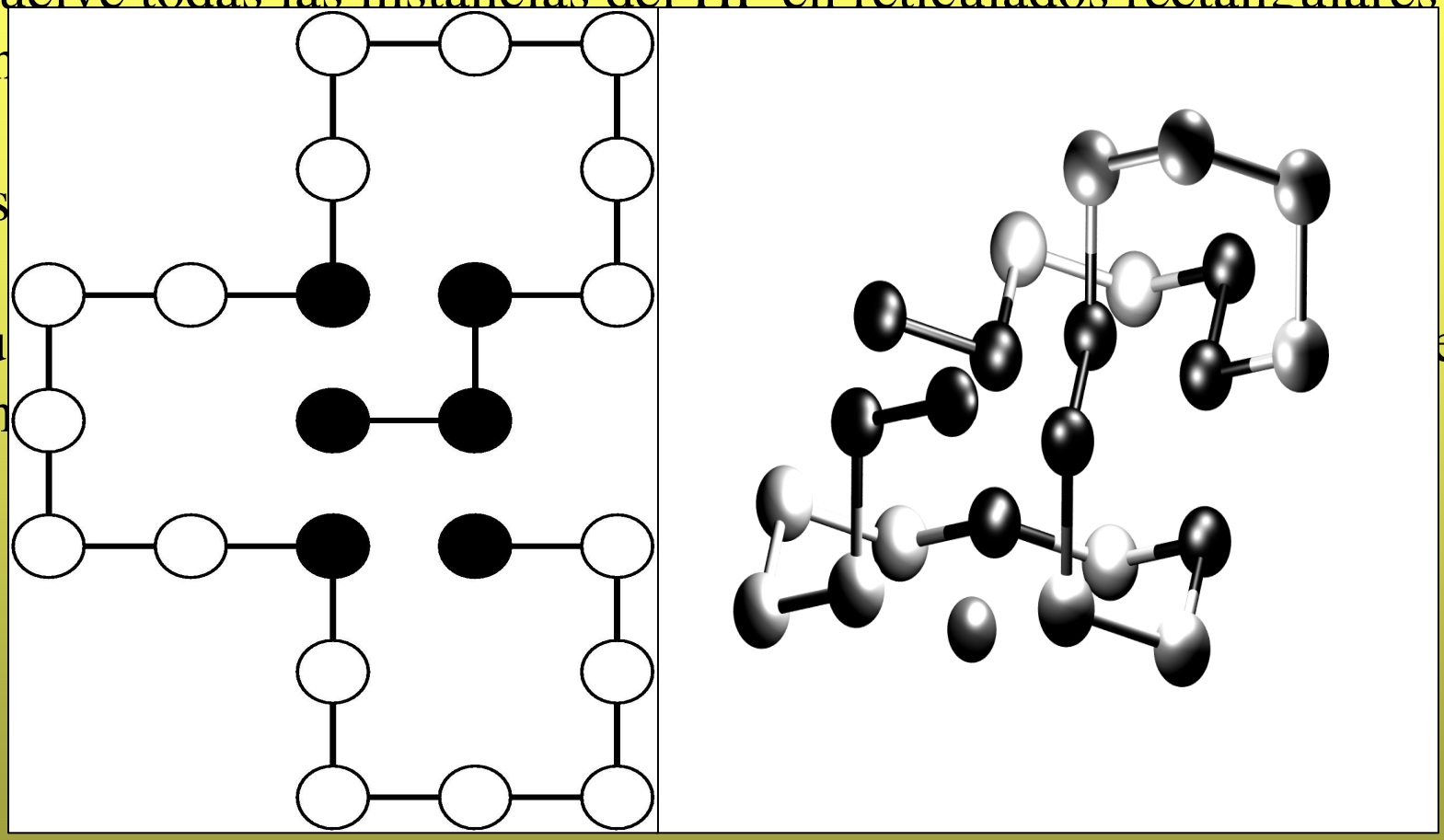
Un algoritmo memetico que usa multiples operadores, el MC modificado y memoria en forma de contact maps (los 5 principios):

```
Reproduction Stage(P):
Begin
  /*  The parent population is stored in P  */
  O = ∅;
  /* O, the offspring set, is initialised */
  Generate the contact map memory for P → CMM.;
  While ( Not enough offspring generated ) Do
    Select two parents to mate, p1,p2;
    crossover(p1,p2) → offspring;
    If (compatible(offspring,CMM)) Then
      O = O ⋃ offspring;
      Inherit move operator to offspring using SIM;
    Fi
  Od
  Return 0;
End.
```

Resuelve todas las instancias del HP en reticulados rectangulares y trian

Mas

resu el
mon



Conclusiones

El desarrollo de algoritmos robustos en modelos simplificados son un “steping-stone” para alcanzar algoritmos efectivos en modelos mas realistas que no pueden ser resueltos por “threading” o “homology”

Varios de los algoritmos mas recientes presentados en CASP utilizan un sampleo de estructuras en latices

En esta charla destile varios principios de disenio importantes para algoritmos en modelos simplificados. Es improbable que un algoritmo que no utilice esto principios pueda mejorar los resultados actuales

Nuestro algoritmo memetico (de 5 principios) es el mas robusto ya que resuelve instancias de los modelos HP y FMP en dos y tres dimensiones y en varias geometrias (rectangular, triangular, fcc)

Veremos luego que estos modelos,
extremadamente simple,
no nos alejan demasiado de los problemas
reales!