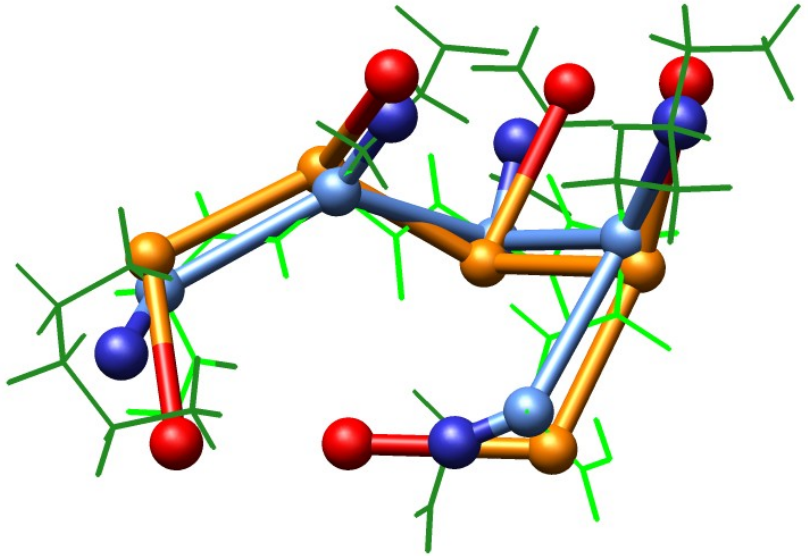
Constraint-based Lattice Model Refinement

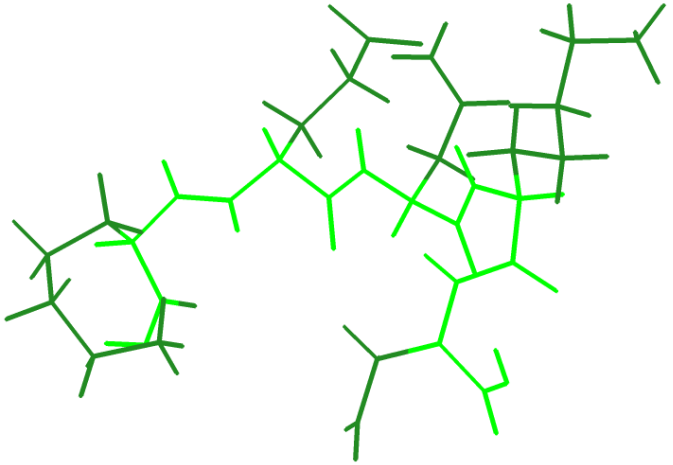


- Martin Mann and Alessandro Dal Palu.
  Lattice model refinement of protein structures. ?, page 9, April 2010. Submitted.
  [COLAfit-src.tar.gz] [pdf] [bib]
- Martin Mann, Rhodri Saunders, Cameron Smith, Rolf Backofen, and Charlotte Deane.
  Latfit - producing high accuracy lattice models from protein atomic co-ordinates including side chains. ?, page 9, April 2010. Submitted.
  [pdf] [bib]

The Problem

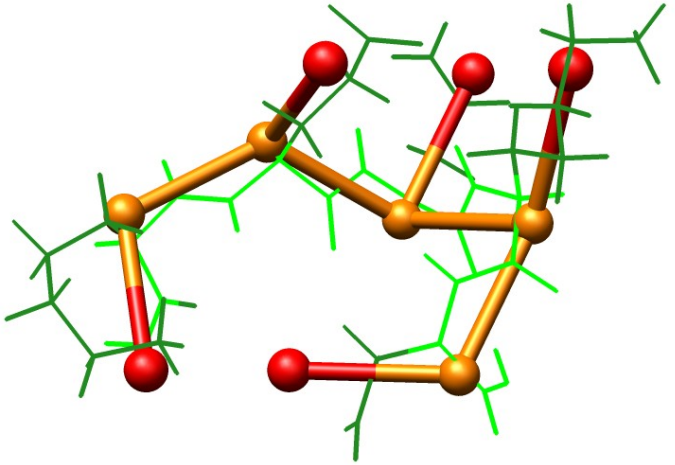
Given:

- Protein in full data (PDB)



Goal:

- Lattice protein model with low deviation



(NP-complete in 3D-cubic for cRMSD optimization [Manuch&Gaur,2008])

Approaches

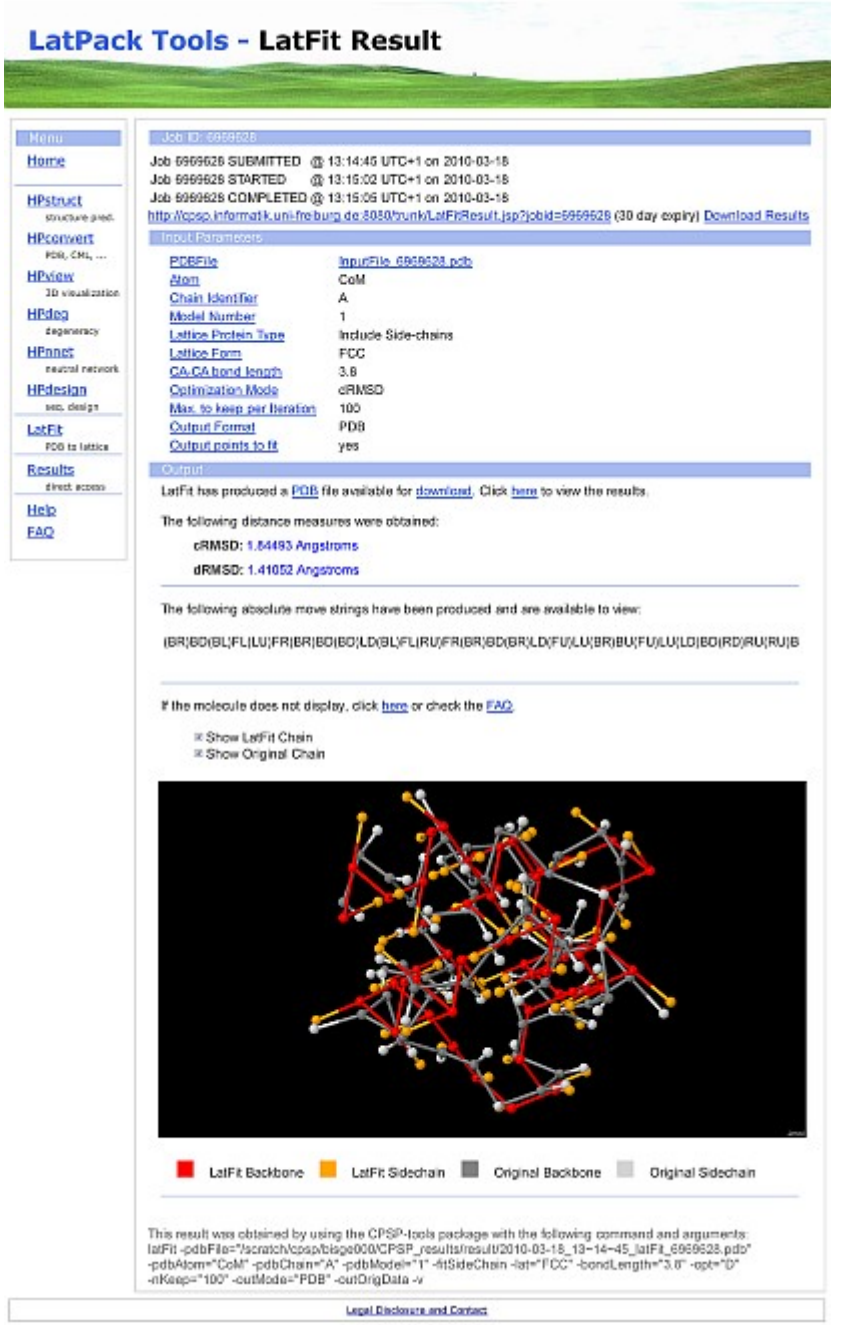
- Full enumeration [Covell&Jernigan,1990]
- Dynamic programming [Hinds&Levitt,1992]
- Chain growth [Park&Levitt,1995]
- Force field optimization [Reva *et al.*,1998]
- Simulated annealing [Ponty *et al.*,2008]

► Mainly heuristics

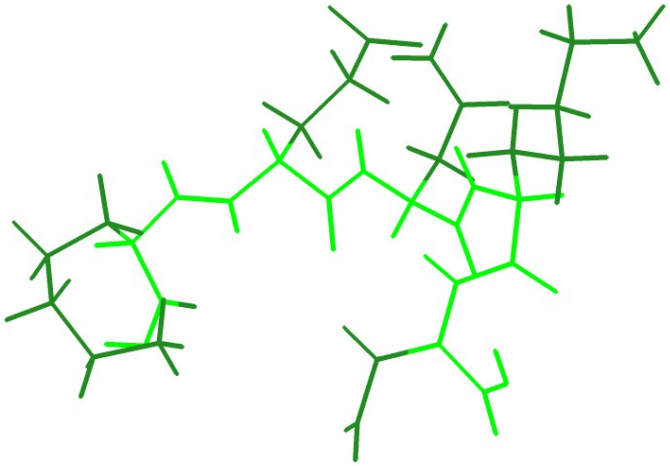
... biased results ?

LatFit – our heuristics

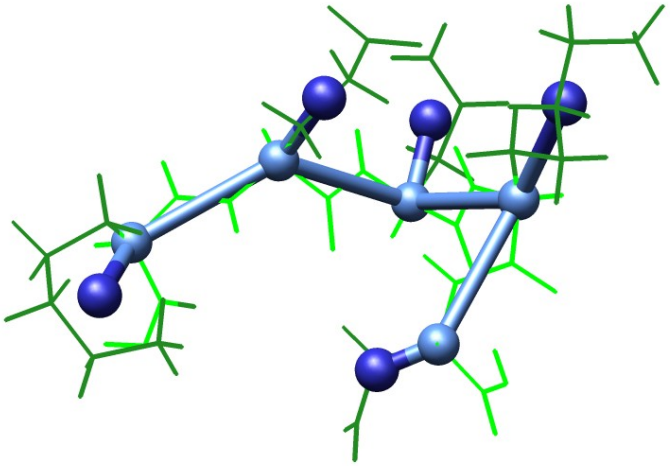
- Chain growth algorithm
- Optimizes either
  - distance (d)RMSD or
  - coordinate (c)RMSD
- Open to backbone-only and side chain models
- Web interface available



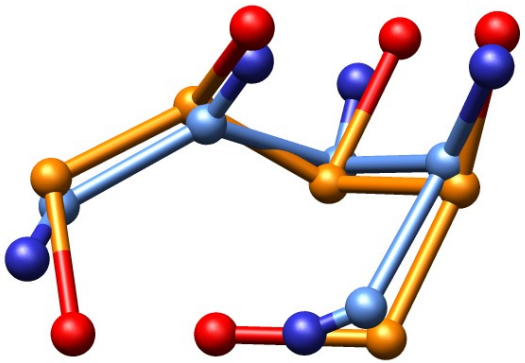
LatFit - Workflow



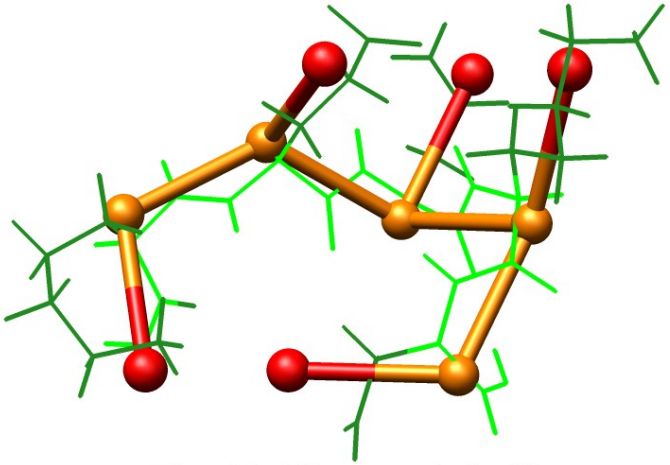
Initial full atom model (PDB)



Coordinate extraction



Best lattice fit calculation



Final lattice protein fit

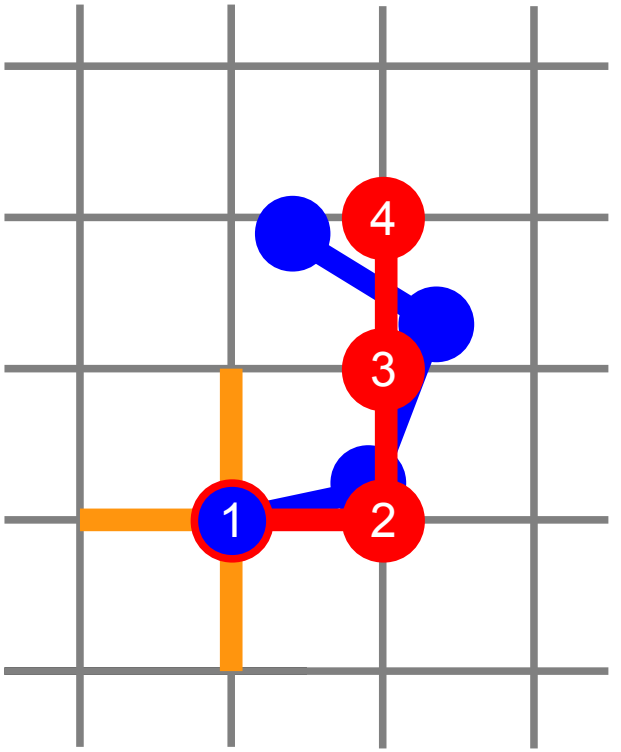
LatFit – cRMSD fitting procedure

For each Monomer:

- Generate **all possible** placements
- Select **best**
- Iterate until all placed

Problem:

- Once placed position is fixed



LatFit - Results

Backbone-only models

	Results taken from Park and Levitt [17]		Results taken from Ponty *et al.* [21]	LatFit cRMSD optimisation	
	dRMSD	cRMSD	cRMSD	dRMSD	cRMSD
	μ	μ	μ (rescaled to $\mathring{A}$)	μ / σ	μ / σ
cub	2.34	2.84	3.5 $(0.923 \cdot 3.8)$	2.042 / 0.228	2.539 / 0.234
fcc	1.46	1.78	-	1.319 / 0.086	1.641 / 0.090
210	1.02	1.24	-	0.931 / 0.060	1.154 / 0.060

Side chain models

	LatFit - dRMSD optimisation		LatFit - cRMSD optimisation	
	dRMSD	cRMSD	dRMSD	cRMSD
	μ / σ	μ / σ	μ / σ	μ / σ
cub	2.779 / 0.754	4.157 / 1.331	2.609 / 0.481	3.286 / 0.624
fcc	1.496 / 0.153	2.104 / 0.246	1.495 / 0.061	1.839 / 0.068
210	1.126 / 0.068	1.601 / 0.100	1.185 / 0.042	1.450 / 0.047

Can we do better ?

Yes We Can !


I admit that 'Bob the Builder' was the main inspiration behind my 'Yes, We Can!' campaign.
Bob the Builder
Yes We Can!
Thanks!

Lattice Model Refinement

Given:

- Original protein
- Lattice model (LatFit)

Idea:

- Allow some relaxation
- Find best among these

Goal:

- Refined lattice model based on first model

Method:

- Constraint Programming
- Limited discrepancy search

Constraint Satisfaction Problem

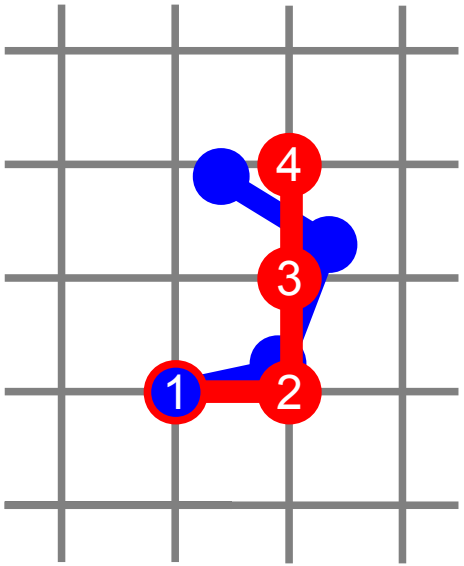
Definition

A Constraint Satisfaction Problem (CSP) consists of

- *variables* $\mathcal{X} = \{X_1, \ldots, X_n\}$,
- the domain D that associates finite domains $D_1 = D(X_1), \ldots, D_n = D(X_n)$ to $\mathcal{X}$.
- a set of constraints C.

A solution is an assignment of variables to values of their domains that satisfies the constraints.

Refinement CSP



$X = \{X_1, \ldots, X_n\}$

one for each monomer

$D(X_i) = \{ p : |p\text{-}M_i| \leq \mathbf{d_{max}}\}$

surrounding sphere

$C_1 = SAW\ (X_1, \ldots, X_n)$

self avoiding walk

$C_2 = |\{ i : X_i \neq M_i \}| \leq \mathbf{K}$

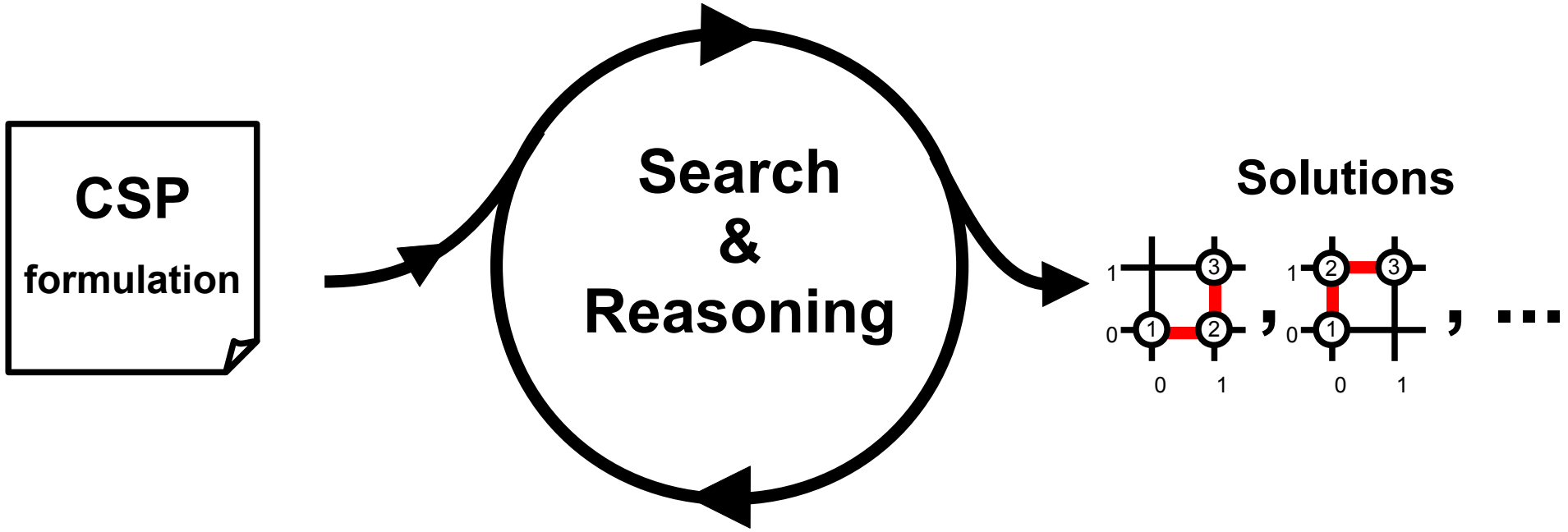
limited discrepancy

Relaxation Parameters:

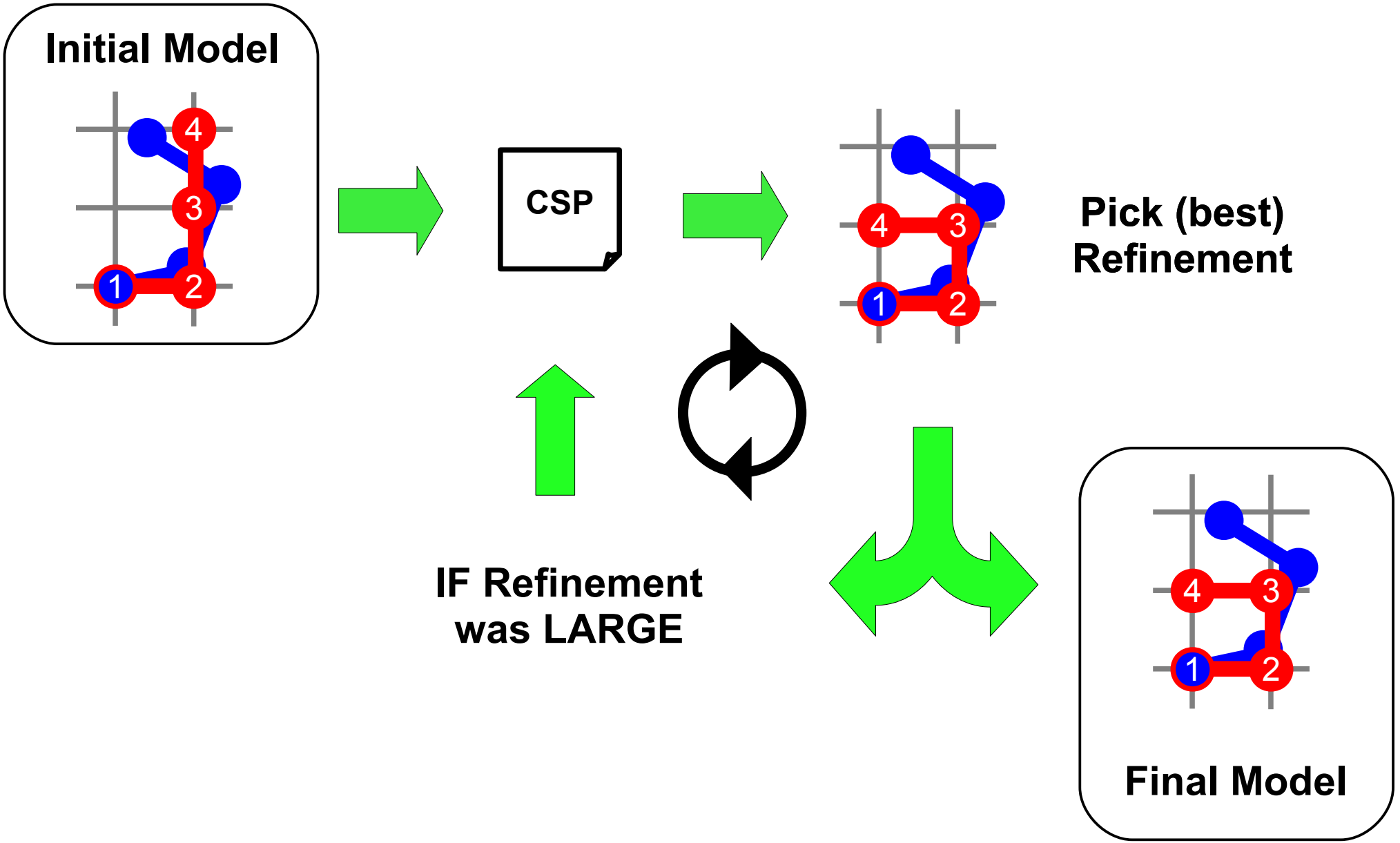
D_{max} : maximal displacement

K : number of replaced monomers

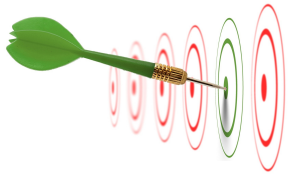
CP Workflow


CSP
formulation
Search
&
Reasoning
Solutions
1
0
3
1
2
0
1
0
1
1
0
2
3
1
0
1
0
1
, , ...

Local Search Cycle



Refinement - Results



dRMSD

8RXN		K = 1	2	3	4
$d_{\max}$	0	1.2469	1.2469	1.2469	1.2469
	1	1.2319	1.2172	1.1639	1.1189
	2	1.2319	1.1674	1.1596	1.0884
	3	1.2319	1.1674	1.1596	1.0884

1CKA		K = 1	2	3	4
$d_{\max}$	0	1.2370	1.2370	1.2370	1.2370
	1	1.2226	1.2226	1.2226	1.2226
	2	1.2026	1.1887	1.1887	1.1887
	3	1.2026	1.1887	1.1887	1.1887

2FCW		K = 1	2	3	4
$d_{\max}$	0	1.1353	1.1353	1.1353	1.1353
	1	1.1353	1.1324	1.1317	1.1309
	2	1.1321	1.1300	1.1254	1.1200
	3	1.1321	1.1300	1.1254	1.1200

time in seconds

8RXN		K = 1	2	3	4
$d_{\max}$	0	0.048	0.081	0.040	0.039
	1	0.112	0.790	2.365	20.70
	2	0.068	0.983	6.500	106.6
	3	0.106	0.499	7.399	124.0

1CKA		K = 1	2	3	4
$d_{\max}$	0	0.031	0.030	0.027	0.037
	1	0.402	0.615	3.442	39.27
	2	0.225	0.456	7.595	120.6
	3	0.421	0.616	8.573	140.2

2FCW		K = 1	2	3	4
$d_{\max}$	0	0.043	0.050	0.058	0.078
	1	0.118	1.997	49.99	1128
	2	0.294	7.192	341.8	14235
	3	0.332	8.129	394.5	16140

Refinement – Future Work

- Extension to side chain models
- Combination with local search
  - "local neighboring search" – Gelato
    - Gradient walk
    - Simulated Annealing
- Local move application

Any comments ?