Chapter 2

Dramatic Performance Enhancements for the FASTER Optimization Algorithm

The text of this chapter was adapted from a manuscript coauthored with Stephen L. Mayo.

Allen, B. D.; Mayo, S. L., Dramatic performance enhancements for the FASTER optimization algorithm. *Journal of Computational Chemistry* **2006,** *27* (10), 1071–1075.

Abstract

FASTER is a combinatorial optimization algorithm useful for finding low-energy side-chain configurations in side-chain placement and protein design calculations. We present two simple enhancements to FASTER that together improve the computational efficiency of these calculations by as much as two orders of magnitude with no loss of accuracy. Our results highlight the importance of choosing appropriate initial configurations, and show that efficiency can be improved by stringently limiting the number of positions that are allowed to relax in response to a perturbation. The changes we describe improve the quality of solutions found for large-scale designs and allow them to be found in hours rather than days. The improved FASTER algorithm finds low-energy solutions more efficiently than common optimization schemes based on the dead-end elimination theorem and Monte Carlo. These advances have prompted investigations into new methods for force field parameterization and multiple state design.

Introduction

Computer programs for protein design and structure prediction typically include a module used to optimize side-chain coordinates in the context of fixed backbone coordinates. To perform this type of calculation, side-chain conformations (rotamers) of one or more amino acid types are oriented onto each residue position, and all possible pairwise rotamer-backbone and rotamer-rotamer interaction energies are calculated using a molecular mechanics force field. This system of interactions is then optimized to find a rotamer configuration of low molecular mechanics energy. The difficulty of finding the lowest-energy configuration increases dramatically with the number of positions designed and the number of rotamers allowed at each position.[1] Useful optimization strategies include Monte Carlo with simulated annealing (MC),[1–4] methods based on dead-end elimination (DEE),[5, 6] methods based on self-consistent mean field theory,[1, 7] genetic algorithms,[1, 8, 9] and the FASTER method.[10] The DEE-based methods have proven especially useful because they ensure that the global minimum energy configuration (GMEC) is identified when they converge.[5] This feature allows researchers to conclude with certainty that any deviations between simulation and experiment are due to problems with the energy functions or simulation model, and are not the result of incomplete optimization. However, current DEE-based algorithms often fail to converge to a single solution when challenged with difficult optimization problems.[6] For this reason, we have begun to favor the FASTER algorithm described by Desmet, Spriet, and Lasters[10] for difficult designs.

Like Monte Carlo, FASTER is a stochastic optimization algorithm that makes perturbations to intermediate solutions and keeps the improvements that it finds.

However, FASTER discovers low-energy solutions far more efficiently, and frequently finds the GMEC as determined by DEE-based algorithms. In cases for which DEE does not converge, it cannot be determined whether or not the solution produced by FASTER is optimal. We typically treat these cases by running many FASTER trajectories in parallel with different random number seeds until the lowest-energy solution has been found multiple times. At this point the solution is considered satisfactory; we refer to such a solution as a FASTER-determined minimum energy configuration (FMEC). This procedure can be time-consuming for problems with many positions and many rotamers at each position. In this paper we present two simple modifications to the published FASTER algorithm that improve the efficiency with which it finds FMEC solutions by as much as two orders of magnitude. In our laboratory, this improvement has reduced the turnaround time for very large designs from days to hours, and has allowed us to begin developing new methods for force field parameterization and multiple state design.

Improvements to FASTER

Original FASTER

As originally described,[10] a FASTER optimization trajectory is computed by executing the following five steps in order: backbone-derived minimum energy configuration (BMEC), iterative batch relaxation (iBR), conditional iBR (ciBR), single perturbation and relaxation (sPR), and double perturbation and relaxation (dPR). The output rotamer configuration of each step is used as input for the next, as follows. BMEC: Generate a starting rotamer configuration by choosing the rotamer at each position with the most favorable interactions with the backbone; rotamer-rotamer

interactions are ignored. iBR: At each position, find the best rotamer in the context of the input configuration at all other positions. Simultaneously update the rotamers at every position after all positions have been considered. Repeat until convergence or cyclic behavior is detected. ciBR: Proceed as in iBR, but randomly accept the new rotamer found at each position with 0.8 probability. sPR: One position at a time, perturb the structure by fixing a rotamer at that position, and allow all other positions to relax as in one round of iBR. The resulting configuration is accepted only if it has the lowest energy found so far. Pick positions for perturbation in random order. Repeat until convergence. dPR: Proceed as in sPR, but perturb pairs of rotamers at different positions together.

Improvement to starting configurations

Regarding the choice of initial rotamer configuration to use as input to FASTER, Desmet et al. noted that the positions of many side-chains can be accurately placed on the protein backbone without considering interactions with other side-chains.[10] Although they showed that this BMEC can serve as an adequate input to FASTER for side-chain placement calculations, our results indicate that the BMEC is suboptimal when FASTER is applied to more difficult protein design problems. Because rotamer-rotamer interactions are ignored, the BMEC is usually a poor solution in terms of amino acid sequence and energy compared to the optimized solutions found by FASTER and other algorithms. Furthermore, the optimization scheme we employ involves computing many separate FASTER trajectories with different random number seeds; because neither the BMEC nor iBR are stochastic, all trajectories are identical until the ciBR step. We hypothesized that FASTER would be able to find the FMEC more effectively if a pool of

partially optimized solutions were generated and initial configurations drawn from that pool. Therefore, we replace the BMEC step at the beginning of each trajectory with a short Monte Carlo run starting from a random configuration. This procedure gives diverse starting solutions with energies significantly better than the BMEC at negligible computational cost.

Improvement to sPR via selective relaxation

As described above, a step of sPR or dPR involves perturbation of the rotamer configuration at one or two positions, followed by relaxation of all the remaining positions in response to the perturbation. In general, however, only a subset of the other positions actually interact significantly with a perturbed position. Thus, the time spent selecting a new rotamer at each of the potentially numerous uncoupled or weakly coupled positions is essentially wasted. This problem can be addressed by limiting the set of positions that are relaxed after every perturbation to those that interact most strongly with the perturbed position. The interaction between a perturbed position and a potential relaxing position may be assessed according to the absolute value of the pairwise interaction energy between the positions before the perturbation. Before a position is perturbed, all the other positions are sorted into a list based on their interactions with the position to be perturbed. The positions to be relaxed are then chosen either by using a number cutoff (the *n* most strongly interacting positions), or an energy cutoff. The optimal value for an energy cutoff depends on the magnitudes of the energies produced by the force field, whereas a number cutoff does not. Therefore, we report calculations

performed with number cutoffs, so that our results might be more useful to researchers using different energy functions.

Methods

The performance of FASTER was tested on four full sequence designs using each method for generating initial configurations (BMEC and MC), and with the number of relaxing positions limited to various values of *n*. We calculated designs for a 28-residue DNA-binding domain of mouse zinc finger Zif268 (PDB code 1AAY, residues 133–160),[11] the 34-residue WW domain from human rotamase Pin1 (1PIN, residues 6–39),[12] the 56-residue B1 domain of streptococcal protein G (1PGA),[13] and the 66-residue cold-shock protein *Bc*-Csp from *Bacillus caldolyticus* (1C9O, chain A).[14] These small, stable, monomeric domains have been the targets of several protein design and stability studies.[15–18]

For each of the four designs, all nonprotein atoms and residues outside the ranges given above were removed; hydrogens were added using REDUCE.[19] All positions were designated core, boundary, or surface as described previously.[15] The amino acids Ala, Val, Leu, Ile, Met, Phe, Tyr, and Trp were allowed at core positions; Ala, Ser, Thr, Asp, Asn, His, Glu, Gln, Lys, and Arg were allowed at surface positions; amino acids from the combination of both sets were allowed at boundary positions. All positions were designed except those with proline or glycine in the wild-type sequence. We used the Dunbrack backbone-dependent rotamer library[20] with expansions of +/- one standard deviation around $\chi 1$ and $\chi 2$ for aromatic amino acids and around $\chi 1$ for hydrophobic amino acids. The average number of rotamers per position over all four designs was 212. Pairwise

energies were computed using energy functions as previously described,[6, 21] except the polar hydrogen burial term was omitted. The design choices reported here reflect the procedures typically used in our laboratory for full-sequence designs.

Optimizations with FASTER were performed as follows. First, rotamers with rotamer-backbone interaction energies greater than 20 kcal/mol or pairs with pairwise interaction energies greater than 50 kcal/mol were eliminated from consideration.[6, 22] Then, simple Goldstein DEE singles elimination was applied until no further rotamers could be eliminated.[6, 23] The input configuration for each trajectory was either the BMEC or the result of a short MC run. The MC was performed by starting with a random configuration and optimizing for 1 cycle of $1x10^6$ steps using a linear temperature gradient from 4500 K to 150 K, followed by quenching[1] of the best-energy sequence that was found. iBR was applied to the input configuration until convergence, followed by 20 cycles of ciBR. Finally, sPR was run with a user-defined value of *n* until convergence. dPR was deemed too computationally expensive to use on all trajectories, and was only applied to the 10 best solutions from each calculation in order to assess whether the FMEC was optimal.

For comparison with FASTER, we also optimized the designs using Monte Carlo. The Monte Carlo optimization was performed according to the procedure described above for FASTER, except that the iBR, ciBR, and sPR passes were skipped, the number of Monte Carlo steps was increased to $2x10^7$, and the low temperature decreased to 0 K. For each design, we computed the same number of trajectories using this Monte Carlo procedure as we had when using FASTER. We also attempted to optimize the designs

using our DEE-based hybrid exact rotamer optimization algorithm (HERO), according to the published procedure.[6]

Results and discussion

The four designs described above were each optimized using 10 different combinations of parameters. We tested values of *n* (the number of positions to relax) from the set (5, 10, 15, 20, N), where N is the total number of positions in the protein. For each *n* tested, we tried FASTER starting from the BMEC solution, and also starting from solutions generated by MC. Starting from the BMEC and setting *n* = N corresponds to FASTER as originally reported by Desmet et al.[10] For each of the four designs, and for each of the 10 parameter combinations tested, we computed 2000 separate FASTER trajectories (8000 for 1AAY). The results of these calculations are presented in Table 1.

Whereas a typical FASTER run might comprise 100 trajectories, here we examined at least 2000 in each case to more accurately assess how easily the FMEC could be found. In particular, we note that when using the original FASTER procedure (BMEC and *n* = N) for 1AAY, as few as 0.01% of the trajectories actually found the FMEC. In this case, the probability of finding the FMEC during a standard run of 100 trajectories approaches zero.

Table 1: **Test calculations illustrating performance enhancements for FASTER**

Design	n [a]	# FMEC[b]		% FMEC[c]		t (minutes)[d]		S[e]		x [f]	
		BMEC	MC	BMEC	MC	BMEC	MC	BMEC	MC	BMEC	MC
1AAY	5	4	29	0.05	0.36	0.24	0.25	485	69	14	98
	10	5	42	0.06	0.53	0.38	0.41	604	79	11	86
	15	5	41	0.06	0.51	0.53	0.59	848	114	8	59
	20	4	23	0.05	0.29	0.69	0.74	1370	257	5	26
	N=28	1	25	0.01	0.31	0.85	0.85	6780	273	1	25
1PIN	5	112	53	5.60	2.65	0.26	0.22	5	8	15	9
	10	113	71	5.65	3.55	0.37	0.36	7	10	11	7
	15	105	77	5.25	3.85	0.50	0.47	10	12	7	6
	20	98	87	4.90	4.35	0.60	0.56	12	13	6	6
	N=34	23	65	1.15	3.25	0.82	0.74	71	23	1	3
1PGA	5	0	9	0.00	0.45	1.9	1.7	—	378	—	16
	10	10	73	0.50	3.65	3.1	2.8	620	77	10	78
	15	10	110	0.50	5.50	4.6	4.0	920	73	7	83
	20	21	110	1.05	5.50	6.2	5.2	590	95	10	63
	N =56	4	116	0.20	5.80	12.0	14.0	6000	241	1	25
1C9O	5	0	12	0.00	0.60	1.3	1.4	—	233	—	99
	10	1	26	0.05	1.30	2.0	1.8	4000	138	6	166
	15	2	35	0.10	1.75	3.0	2.6	3000	149	8	155
	20	1	36	0.05	1.80	3.9	3.2	7800	178	3	129
	N=66	1	54	0.05	2.70	11.5	8.8	23000	326	1	71

[a] The number of positions relaxed after every perturbation during sPR

[b] The number of trajectories that found the FMEC

[c] The percent of trajectories that found the FMEC. The total number of trajectories attempted was 8000 for 1AAY and 2000 for all others.

[d] The time in processor-minutes required to compute a single trajectory, averaged over all trajectories in the run

[e] The score S, representing the number of processor-minutes required, on average, to find the FMEC once. Calculated as $S = t / f$, where f is the fraction of trajectories that found the FMEC. Smaller values are better. "—" indicates that S is undefined because $f = 0$.

[f] The multiplicative factor of improvement compared to the original FASTER protocol

Each combination of parameters may be compared via the score $S = t / f$, where t is the average number of processor-minutes required to compute a single trajectory, and f is the probability that a trajectory would find the FMEC, estimated using the data in Table 1. Thus, S represents the number of processor-minutes it would take, on average, to find the FMEC once; smaller values are better. Using this score as our metric, an improvement in efficiency may occur due to an increase in the fraction of trajectories that find the FMEC, or a decrease in the average convergence time per trajectory, or both.

Table 1 clearly illustrates the utility of starting with an MC solution rather than with the BMEC; when n = N, the improvements in efficiency x observed on switching to MC range from a factor of 3 (1PIN) to a factor of 71 (1C9O). Improvements in this range are also observed for most other values of n we tested; notable exceptions are the 1PIN designs with smaller values of n, for which the BMEC was more effective. In each case, the observed improvements in efficiency when using MC were predominantly due to the greater fraction of trajectories that found the FMEC. For each trajectory, the running time was dominated by the sPR step, and the additional cost of MC was negligible.

With the choice of BMEC/MC held constant, observed changes in f due to the reduction of n from N to (20,15,10) have different magnitudes and signs in the four designs. However, the average time t required to complete a single trajectory was always reduced, typically by a factor of 3–5 when n = N is compared with n = 10. Thus, significant improvements in the computational efficiency S were always observed when reducing n to the range of 10–20. For 1PGA and 1C9O when n = 5, the FMEC was never found when the BMEC was used as an input structure; we therefore avoid the use of n

smaller than 10. Although we have not systematically evaluated parameter combinations for designs larger than 66 positions, we do not anticipate problems using values of n in the range of 10–20 for larger designs.

The overall performance of FASTER is dramatically improved when both enhancements are used together. When using MC instead of the BMEC and with $n = 10$, the computational efficiency S of the 1AAY calculation was improved compared to the original FASTER by a factor of 86. Optimizations for the other designs 1PIN, 1PGA, and 1C9O were improved by factors of 7, 78, and 166, respectively. We note that this improvement in efficiency is not only a convenience. Because users have limited time and computer resources, they will rarely be able to compute as many trajectories for a given design as we describe in this paper. Thus, the improvements allow protein designers to find solutions that are better than those they would have found with the original FASTER protocol, and not merely to find the same solutions more rapidly.

In an attempt to show that the FMEC solutions found by FASTER were optimal, we performed DEE-based optimizations using HERO. HERO converged for the 1PIN design, yielding a sequence and energy identical to the FMEC found by the FASTER trajectories; the other three HERO calculations failed to converge, and so the optimality of the FMEC solutions for the 1AAY, 1PGA, and 1C9O designs is not known. We also tested the optimality of the FMEC solutions by applying dPR until convergence to the top ten solutions found in every FASTER calculation. In no case did this dPR optimization yield a better solution than the FMEC, giving us further confidence that the FMECs used to generate the values in Table 1 are the best solutions that FASTER can provide.

To determine whether the improved FASTER procedure we describe performs better than when Monte Carlo is used alone, we repeated the optimizations with a more extensive MC section and with the FASTER-specific passes omitted, as described above. Table 2 shows that the improved FASTER algorithm is able to find the FMEC solution for each design much more frequently than MC alone, even though the MC trajectories used somewhat more processor time than the FASTER trajectories. Notably, the pure Monte Carlo procedure was never able to find the FMEC for the 1PGA design. For the 1AAY, 1PIN, and 1C9O designs, the improved FASTER algorithm was more efficient than Monte Carlo alone by factors of 10, 7, and 8, respectively. Interestingly, the improvement factors reported in Table 2 also indicate that Monte Carlo is actually more powerful for these three designs than the original FASTER algorithm. Nevertheless, the improved FASTER procedure we report is clearly preferable for all four designs.

Table 2: Comparison of the improved FASTER to Monte Carlo

Design	Opt[a]	# FMEC[b]		% FMEC[c]		*t* (minutes)[d]		*S*[e]		*x*[f]	
		w/[g]	w/o[g]	w/	w/o	w/	w/o	w/	w/o	w/	w/o
1AAY	Monte	12	10	0.15	0.1	1.13	1.25	753	1000	9	7
	Faster	42	25	0.53	0.3	0.41	0.90	78	288	86	24
1PIN	Monte	53	26	2.65	1.3	1.28	1.43	48	110	1	1
	Faster	71	66	3.55	3.3	0.36	0.80	10	24	7	3
1PGA	Monte	0	0	0.00	0.0	3.63	3.73	—	—	—	—
	Faster	73	80	3.65	4.0	2.80	5.05	77	126	78	48
1C9O	Monte	11	6	0.55	0.3	5.68	5.70	1033	1900	22	12
	Faster	26	17	1.30	0.8	1.80	3.92	138	461	166	50

[a] The optimization strategy that was used. Monte: pure MC trajectories as described in Methods. Faster: FASTER trajectories as described in Methods; the number of interacting residues in sPR was limited to 10, and the BMEC step was replaced with MC. The total number of trajectories attempted for both Monte and Faster was 8000 for 1AAY and 2000 for all other designs.

[b–e] See Table 1.

[f] The multiplicative factor of improvement compared to data for the original FASTER protocol reported in Table 1

[g] Indicates whether or not Goldstein singles elimination was performed before the other optimization steps.

The improved FASTER algorithm and Monte Carlo were also assessed without the pre-elimination of singles by Goldstein DEE. Table 2 shows that the DEE step significantly improved the convergence times of FASTER trajectories, and slightly improved the convergence times for the MC trajectories. Furthermore, the use of DEE typically increased the fraction of trajectories that found the FMEC for both FASTER and MC, improving overall efficiency by a factor of 2–4 for FASTER and by close to 2 in

one case for MC. We conclude that the pre-elimination of singles by Goldstein DEE is a worthwhile enhancement to these optimization strategies.

Conclusions

FASTER is a stochastic optimization algorithm that can efficiently find low-energy solutions to difficult protein design problems. We report two simple enhancements to FASTER that together result in up to two orders of magnitude better computational performance with no loss of accuracy. The first improvement replaces the backbone-derived initial configuration with a short Monte Carlo run. The second improvement limits the number of relaxing positions in the perturbation and relaxation steps to a fixed value. The dramatic performance enhancements provided by these changes make FASTER significantly more powerful than alternative methods, and allow better solutions to be found more quickly for larger, more complex designs. We expect the improved algorithm to facilitate the development of next-generation protein design tools that treat multiple states and explicit backbone flexibility.

References

1. Voigt, C. A.; Gordon, D. B.; Mayo, S. L., Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. *Journal of Molecular Biology* **2000,** *299* (3), 789–803.

2. Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P., Optimization by Simulated Annealing. *Science* **1983,** *220* (4598), 671–680.

3. Kuhlman, B.; Dantas, G.; Ireton, G. C.; Varani, G.; Stoddard, B. L.; Baker, D., Design of a novel globular protein fold with atomic-level accuracy. *Science* **2003,** *302* (5649), 1364–1368.

4. Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E., Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics* **1953,** *21* (6), 1087–1092.

5. Desmet, J.; Demaeyer, M.; Hazes, B.; Lasters, I., The Dead-End Elimination Theorem and Its Use in Protein Side-Chain Positioning. *Nature* **1992,** *356* (6369), 539–542.

6. Gordon, D. B.; Hom, G. K.; Mayo, S. L.; Pierce, N. A., Exact rotamer optimization for protein design. *Journal of Computational Chemistry* **2003,** *24* (2), 232–243.

7. Koehl, P.; Delarue, M., Application of a Self-Consistent Mean-Field Theory to Predict Protein Side-Chains Conformation and Estimate Their Conformational Entropy. *Journal of Molecular Biology* **1994,** *239* (2), 249–275.

8. Desjarlais, J. R.; Handel, T. M., De-Novo Design of the Hydrophobic Cores of Proteins. *Protein Science* **1995,** *4* (10), 2006–2018.

9. Jones, D. T., De-Novo Protein Design Using Pairwise Potentials and a Genetic Algorithm. *Protein Science* **1994,** *3* (4), 567–574.

10. Desmet, J.; Spriet, J.; Lasters, I., Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins* **2002,** *48* (1), 31–43.

11. Elrod-Erickson, M.; Rould, M. A.; Nekludova, L.; Pabo, C. O., Zif268 protein-DNA complex refined at 1.6 angstrom: A model system for understanding zinc finger-DNA interactions. *Structure* **1996,** *4* (10), 1171–1180.

12. Ranganathan, R.; Lu, K. P.; Hunter, T.; Noel, J. P., Structural and functional analysis of the mitotic rotamase Pin1 suggests substrate recognition is phosphorylation dependent. *Cell* **1997,** *89* (6), 875–886.

13. Gallagher, T.; Alexander, P.; Bryan, P.; Gilliland, G. L., 2 Crystal-Structures of the B1 Immunoglobulin-Binding Domain of Streptococcal Protein-G and Comparison with NMR. *Biochemistry* **1994,** *33* (15), 4721–4729.

14. Mueller, U.; Perl, D.; Schmid, F. X.; Heinemann, U., Thermal stability and atomic-resolution crystal structure of the Bacillus caldolyticus cold shock protein. *Journal of Molecular Biology* **2000,** *297* (4), 975–988.

15. Dahiyat, B. I.; Mayo, S. L., De novo protein design: Fully automated sequence selection. *Science* **1997,** *278* (5335), 82–87.

16. Kraemer-Pecore, C. M.; Lecomte, J. T. J.; Desjarlais, J. R., A de novo redesign of the WW domain. *Protein Science* **2003,** *12* (10), 2194–2205.

17. Malakauskas, S. M.; Mayo, S. L., Design, structure and stability of a hyperthermophilic protein variant. *Nature Structural Biology* **1998,** *5* (6), 470–475.

18. Perl, D.; Schmid, F. X., Electrostatic stabilization of a thermophilic cold shock protein. *Journal of Molecular Biology* **2001,** *313* (2), 343–357.

19. Word, J. M.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C., Asparagine and glutamine: Using hydrogen atom contacts in the choice of side-chain amide orientation. *Journal of Molecular Biology* **1999,** *285* (4), 1735–1747.

20. Dunbrack, R. L.; Cohen, F. E., Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science* **1997,** *6* (8), 1661–1681.

21. Gordon, D. B.; Marshall, S. A.; Mayo, S. L., Energy functions for protein design. *Current Opinion in Structural Biology* **1999,** *9* (4), 509–513.

22. DeMaeyer, M.; Desmet, J.; Lasters, I., All in one: A highly detailed rotamer library improves both accuracy and speed in the modelling of sidechains by dead-end elimination. *Folding & Design* **1997,** *2* (1), 53–66.

23. Goldstein, R. F., Efficient Rotamer Elimination Applied to Protein Side-Chains and Related Spin-Glasses. *Biophysical Journal* **1994,** *66* (5), 1335–1340.