International Journal of
Molecular Sciences

Article

Protein Structure Refinement Using Multi-Objective Particle Swarm Optimization with Decomposition Strategy

Cheng-Peng Zhou [1], Di Wang [1], Xiaoyong Pan [1] and Hong-Bin Shen [1,2,*]

[1] Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China; anakinchou@sjtu.edu.cn (C.-P.Z.); wangdi19941224@sjtu.edu.cn (D.W.); 2008xypan@sjtu.edu.cn (X.P.)

[2] Department of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China

[*] Correspondence: hbshen@sjtu.edu.cn; Tel.: +86-21-34205320; Fax: +86-21-34204022

Abstract: Protein structure refinement is a crucial step for more accurate protein structure predictions. Most existing approaches treat it as an energy minimization problem to intuitively improve the quality of initial models by searching for structures with lower energy. Considering that a single energy function could not reflect the accurate energy landscape of all the proteins, our previous AIR 1.0 pipeline uses multiple energy functions to realize a multi-objectives particle swarm optimization-based model refinement. It is expected to provide a general balanced conformation search protocol guided from different energy evaluations. However, AIR 1.0 solves the multi-objective optimization problem as a whole, which could not result in good solution diversity and convergence on some targets. In this study, we report a decomposition-based method AIR 2.0, which is an updated version of AIR, for protein structure refinement. AIR 2.0 decomposes a multi-objective optimization problem into a number of subproblems and optimizes them simultaneously using particle swarm optimization algorithm. The solutions yielded by AIR 2.0 show better convergence and diversity compared to its previous version, which increases the possibilities of digging out better structure conformations. The experimental results on CASP13 refinement benchmark targets and blind tests in CASP 14 demonstrate the efficacy of AIR 2.0.

Keywords: protein structure prediction; structure refinement; multi-objective particle swarm optimization; decomposition strategy; AIR

check for updates

Citation: Zhou, C.-P.; Wang, D.; Pan, X.; Shen, H.-B. Protein Structure Refinement Using Multi-Objective Particle Swarm Optimization with Decomposition Strategy. *Int. J. Mol. Sci.* **2021**, *22*, 4408. https://doi.org/10.3390/ijms22094408

Academic Editor: Daisuke Kihara

Received: 30 March 2021
Accepted: 20 April 2021
Published: 23 April 2021

1. Introduction

The functions of a protein are closely related to its 3D structure, and high-resolution protein structure can increase the understanding of what it does and how it works. In the past decades, dramatic progress has been made in structure determination using wet-lab experimental methods, such as X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, and recent electron microscopy techniques [1]. However, these experiments are still expensive and time-consuming [2]. Many popular automated protein structure prediction methods play important complementary roles [3–5], such as AlphaFold [6], trRosetta [7], I-TASSER [8], and MULTICOM [9,10]. Especially in recent years, protein structure prediction performance has been largely improved due to the advances in both theoretical and computational studies as demonstrated in recent CASP (Critical Assessment of protein Structure Prediction) assessment, e.g., coevolution analysis-based investigation [11–14], powerful deep learning computational techniques [15–17], etc.

Although remarkable results have been achieved in protein structure prediction, the predicted models still contain inaccurate regions deviating from the native structures [18]. Thus, there have been increasing efforts on improving predicted models via refinement as a following step. Since the 8th competition of Critical Assessment of protein Structure Prediction, the protein structure refinement task has been introduced to evaluate the performance of computational methods for structure refinement by given an initial predicted model [19–21]. However, it is a challenging task until now, as it is a blind refinement and on some hard targets, refinement methods degrade their initial models rather than improve them.

One of the common strategies for protein structure refinement is to implement the work pipeline through the combination of energy functions and optimization algorithms [22–25]. The energy function is designed to describe a protein's state that is near-native or non-native from its view, which will guide the refinement search to its lower energy state. Considering its importance, a number of molecular mechanics force fields and knowledge-based energy functions have been proposed, i.e., AMBER [26], CHARMM [27], OPLS [28], RWplus [29], DFIRE [30], GOAP [31], and Rosetta [32]. However, it is still difficult to apply a single energy function to exactly describe the states of all proteins due to the large diversities of the protein structures. Each energy function would have its advantages and disadvantages on specific targets, which is a potential reason the performance of the refinement algorithms often varies with the targets in the CASP experiments.

In addition to the energy functions, the optimization algorithms are also crucial in protein structure refinement, which are designed to search for the lowest-energy structure conformation. Popular optimization algorithms include Molecular Dynamics (MD) simulation [33] and Monte Carlo (MC) simulation [34]. It is still very challenging to achieve consistent refinement over initial models, and one potential reason is that most existing approaches are conducted based on a single energy function.

Motivated by those observations, we have developed one multi-objective-based refinement method called AIR [35] to alleviate the potential bias caused by minimizing only one energy function. The AIR is a multi-objective particle swarm optimization (PSO)-based protocol [36], where each structure is treated as a particle. The quality of the particles in each iteration is evaluated by three energy functions based on dominance relations [37], and the non-dominated particles are put into a set called Pareto set (*PS*) [38], which is used to select the final refined structures.

However, the dominance-based AIR has no direct control over the movement of each particle in the swarm and no suitable mechanism to maintain the diversity of Pareto front (*PF*) [39]. The loss of diversity may deteriorate the advantage of multi-objective optimization. Moreover, the crucial parameter *Gbest* in PSO is difficult to choose, since there are many non-dominated candidates in the *PS*. Using Pareto dominance alone would deteriorate the selection pressure toward the *PF* and slow down the searching process [40], since the update of another important parameter *Pbest* only needs to reduce one energy function.

To solve the above problems, we present a decomposition-based approach AIR 2.0, which is an updated version of AIR 1.0 to further increase the conformation optimization capability. In AIR 2.0, each particle is associated with a unique subproblem defined by a weight vector, which is different from the protocol of AIR 1.0 that solves a multi-objective optimization problem as a whole. Thus, the diversity is accordingly improved, since each particle is moving toward *PF* in its own direction. In addition, the *Pbest* and *Gbest* of each particle have a determined choice according to its own subproblem, helping avoid oscillation in the searching process. The benchmark tests of CASP13 refinement targets and blind tests in CASP14 demonstrate the efficacy of the new updated version of AIR refinement pipeline.

2. Experiments and Results

We have evaluated AIR 2.0 pipeline on the refinement targets of CASP13 and CASP14. To demonstrate the advantage of AIR 2.0, we compare it with state-of-the-art methods, including its previous version AIR 1.0 and other state-of-the-art refinement methods in CASP13 such as FEIGLAB [18], BAKER [24], and Zhang-Refinement [41]. Global distance test total score (GDT-TS) [42], template modeling score (TM-score) [43], and root mean square deviation (RMSD) are the metrics for evaluating the effectiveness of AIR 2.0.

For each target, the number of divisions H in (10) (see Methods) is set to 10 according to our local tests, resulting in $N = \begin{pmatrix} H+M-1 \\ M-1 \end{pmatrix} = \begin{pmatrix} 10+3-1 \\ 3-1 \end{pmatrix} = 66$ weight vectors or subproblems and the same number of particles. $M = 3$ is the number of objectives. The single initial model provided by CASP is taken as input, and another 65 models are generated by applying random perturbations to the initial model. The neighborhood size T is set to 8 according to [44], and the maximum number of iterations is set to 3000 as AIR 1.0. We set S = 20,000 in (9) (see Methods) to get a stable result and output the top five ranked solutions.

2.1. Effectiveness of AIR 2.0 on CASP13

We test AIR 2.0 on the 29 CASP13 refinement targets (two cancelled targets were excluded), and the results are summarized in Figure 1. We compare the best model and Model 1 with the initial model. The best model achieves consistent improvements over the initial model and almost all targets are to a certain degree refined. The average gains in the quality of the best model are +1.98 in GDT-TS, +0.014 in TM-score, and −0.18 Å in RMSD. Compared to Model 1, the average improvement in GDT-TS is 1.22, and 82% of the targets (24 out of 29 targets) are refined. In terms of TM-score and RMSD, the average improvements are +0.0076 and −0.0752 Å with 72% (21 out of 29 targets) and 69% (20 out of 29 targets) being refined, respectively.

It is observed that the targets with a medium quality are more likely to be refined (Figure 2). Specifically, AIR 2.0 improves those targets with the following quality: (1) initial GDT-TS is between 60 and 80, (2) initial RMSD is between 2 and 5 Å, and (3) initial TM-score is between 0.65 and 0.8. The potential reason is that high quality models leave a few spaces to refine, while the relatively bad models might be trapped in a deep local minimum caused by a rough energy landscape.

2.2. AIR 2.0 Is Superior to AIR 1.0

We compare the updated method AIR 2.0 with our previous AIR 1.0. The CASP13 results of AIR 2.0 and AIR 1.0 are summarized in Table 1. Figure 3 illustrates the GDT-TS of all targets refined by AIR 2.0 and 1.0 based on the best model and Model 1. The results indicate that AIR 2.0 achieves better or comparable performance over AIR 1.0. Compared to AIR 1.0, AIR 2.0 obtains a better quality in 21 out of 29 targets for the best model and 24 of 29 targets for Model 1.

For AIR 1.0, a refinement of hard targets in CASP13 often obtained model degradation rather than improvement, such as R0949, R0977D2, R0996D4, and R1016. However, AIR 2.0 achieves promising refinement results on these hard targets. The potential reason is due to the diversity of PS introduced by the decomposition strategy. In the case of R0981D5, the non-dominance solutions in the final PS of AIR 2.0 and 1.0 are plotted in Figure 4b. We can see that AIR 2.0 finds more non-dominated solutions than AIR 1.0, and these solutions are distributed with a high diversity. Moreover, as shown in Figure 4a,c,d, the solutions obtained by AIR 2.0 completely dominate those obtained by AIR 1.0, indicating a more convergence toward the true PF. Thus, the overall quality of the solutions obtained by AIR 2.0 is higher than those of AIR 1.0, which is beneficial for the selection of high quality Model 1.

Table 1. Overall performance of AIR 1.0 and AIR 2.0 in terms of GDT-TS on 29 refinement targets from CASP13.

Method	Best Model (GDT-TS)	Model 1 (GDT-TS)
AIR 1.0	+1.07	+0.16
AIR 2.0	+1.98	+1.22

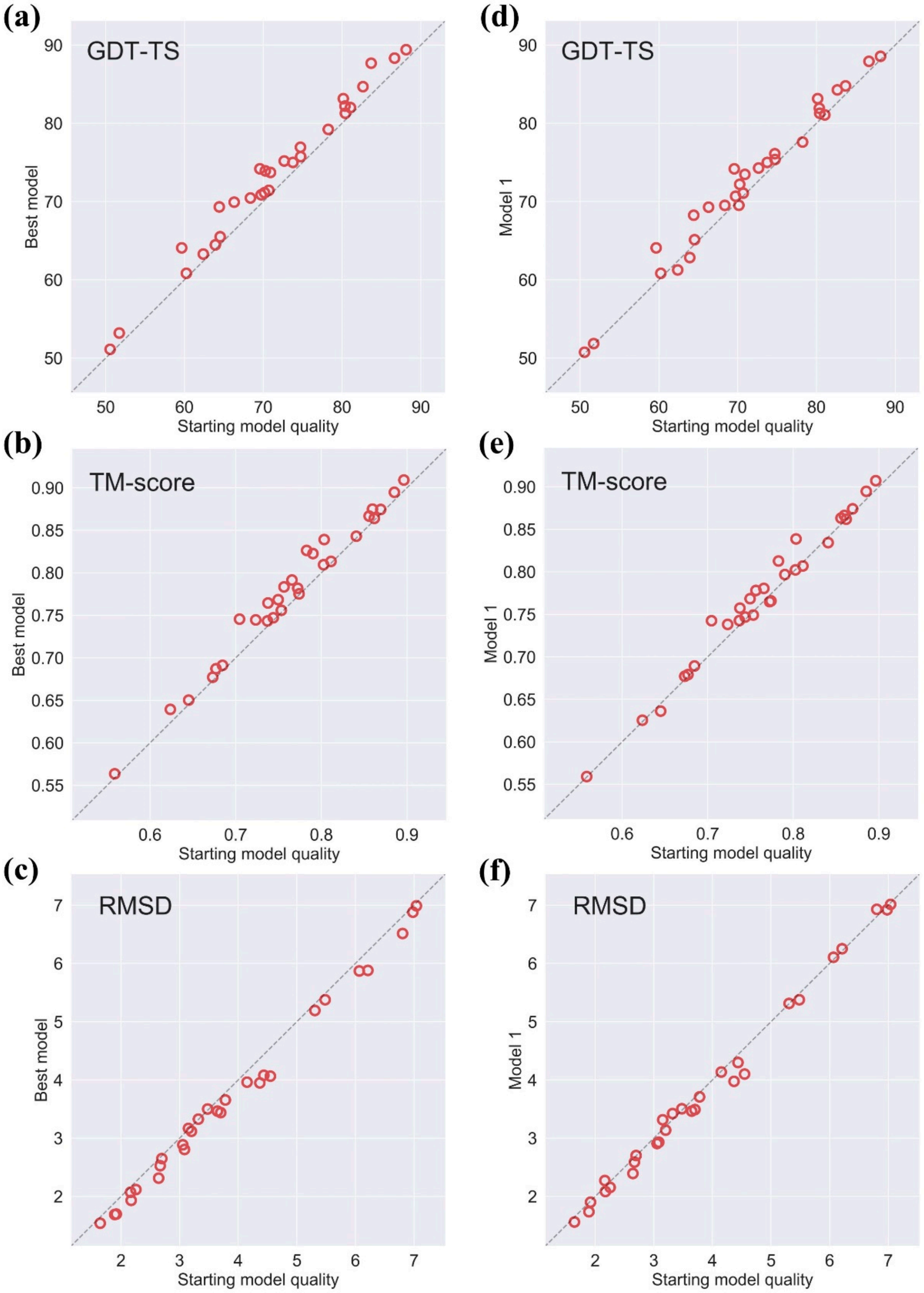


Figure 1. Effectiveness of AIR 2.0 on CASP13 measured by GDT-TS, TM-score, and RMSD. The comparison of the best model refined by AIR 2.0 and the initial model in terms of GDT-TS, TM-score, and RMSD are shown in (**a**–**c**), respectively. The comparison of AIR 2.0 refined Model 1 and the initial model in terms of GDT-TS, TM-score, and RMSD are given in (**d**–**f**) respectively.

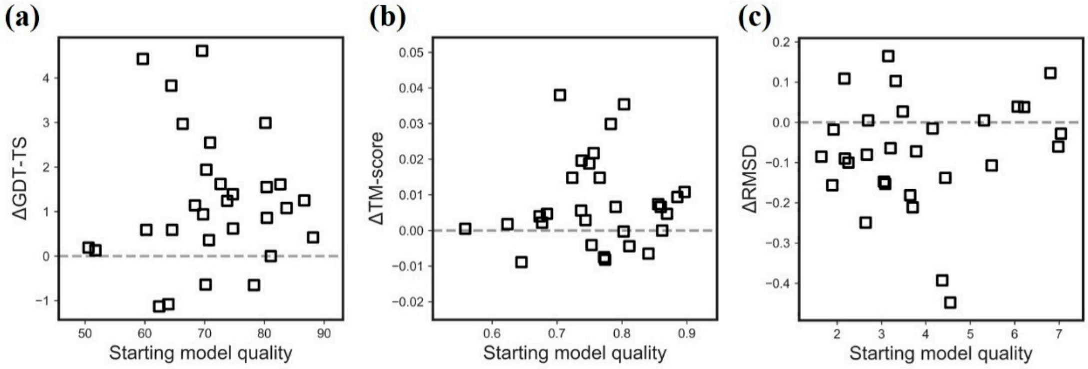
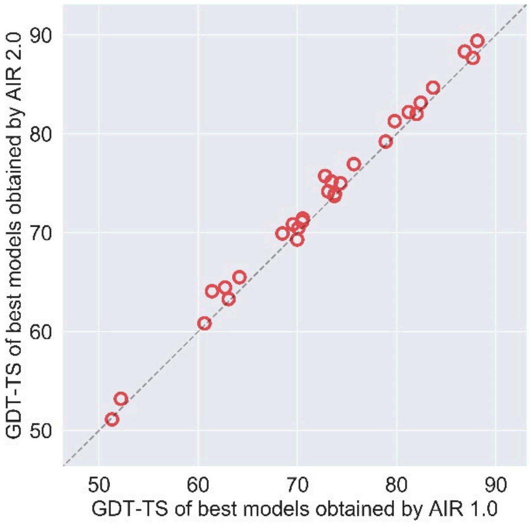
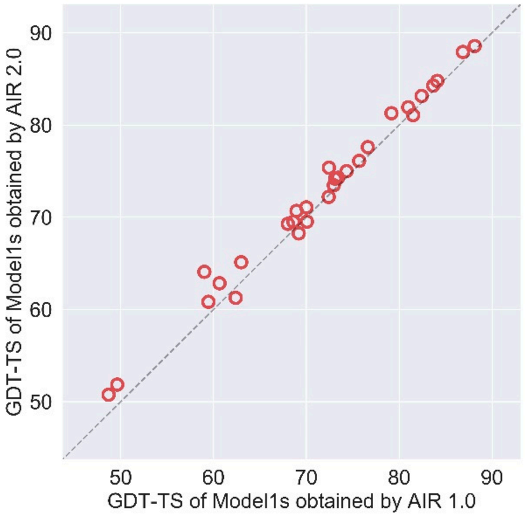


Figure 2. Refinement improvement over the initial model in terms of (**a**) GDT-TS (**b**) TM-score (**c**) RMSD.



(**a**)



(**b**)

Figure 3. The GDT-TS comparison of (**a**) best models and (**b**) Model1s between AIR 1.0 and AIR 2.0 on the targets of CASP13.

The dominance-based method AIR 1.0 drives the whole population toward the *PF* without direct control over the movement of each individual in the population. Thus, AIR 1.0 prefers the regions that are easy to access and does not sufficiently account for the diversity. As a result, the solutions obtained by AIR 1.0 are only distributed in a small area. Moreover, due to the lack of stable guidance on *Pbest* and *Gbest* (see Methods), the searching process of each particle would be difficult. However, the decomposition strategy in AIR 2.0 assigns a single objective optimization subproblem for each particle. In this way, each particle has an exact update direction or a clear target position in *PF*, which results in better diversity and convergence features. This is the potential reason why AIR 2.0 outperforms AIR 1.0 in most targets.

2.3. Comparison with Other State-of-the-Art Refinement Methods

The test data consist of those targets in which each group performs the best on CASP13 in order to highlight the characteristics of each method. The refined models of BAKER, FEIGLAB, and Zhang are available at the CASP official website. As shown in Table 2, AIR 2.0 yields promising improvement over initial models. For the target R0949, R0979, and R0989D1, AIR 2.0 is the only method that achieves improvement rather than degradation over initial models and the GDT-TS gains is 0.59, 3.53, and 0.37, respectively. However, for the targets such as R0968s1, R0974s1, and R0986s1, the GDT-TS gains obtained by BAKER or FEIGLAB are larger than AIR 2.0. The potential reason is probably that AIR 2.0 uses multiple energy functions that constrain each other, resulting in a limited deviation from initial models. These results also highlight that the protein structure map is huge, and it is very hard to achieve a general better refinement algorithm on all proteins. For hard targets, AIR 2.0 would be reliable, since we extend the one-dimension optimization to a new three-dimension space optimization, partially alleviating the bias caused by using only one energy function.

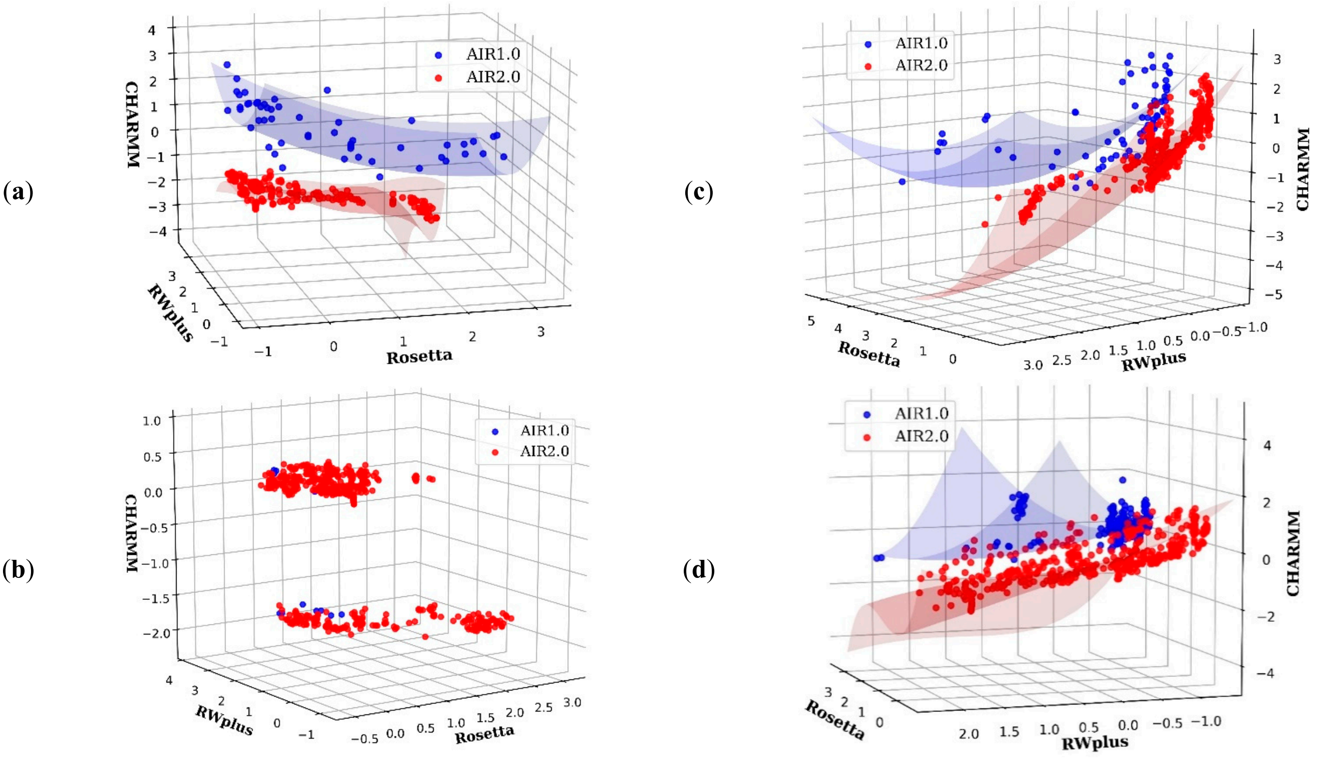


Figure 4. Pareto fronts of some targets obtained by AIR 2.0 (red) and AIR 1.0 (blue). (**a**) R0976D2, (**b**) R0981D5, (**c**) R0999D3, and (**d**) R1001. AIR 2.0 finds more non-dominated solutions than AIR 1.0, and these solutions are distributed with a high diversity on the target (**b**). Moreover, for targets such as (**a**,**c**,**d**), the solutions obtained by AIR 2.0 completely dominate those obtained by AIR 1.0, indicating a more convergence toward the true Pareto front.

Table 2. GDT_TS comparison of AIR 2.0 and other refinement methods on CASP13 refinement targets. The results of BAKER, FEIGLAB, and Zhang come from the CASP official website. The best model among the four methods is bolded on each target.

Target	Initial Model	AIR 2.0	BAKER	FEIGLAB	Zhang
R0949	64.53	**65.12**	56.01	62.98	64.53
R0957	60.97	**64.08**	60.32	61.45	61.61
R0968s1	66.74	69.71	**78.81**	72.25	69.07
R0974s1	84.78	85.96	**99.64**	97.10	84.06
R0976D2	83.06	84.27	**89.11**	80.64	83.87
R0979	70.65	**74.18**	60.60	70.38	70.38
R0986s1	80.16	83.15	90.76	**93.21**	77.99
R0989D1	50.75	**51.12**	44.22	50.75	N/A
R0999D3	75.14	**76.94**	76.11	**76.94**	74.31
R1002D2	88.14	88.56	**89.41**	79.24	88.14
R1004D2	78.57	77.60	81.49	**93.51**	79.22
R1016	81.06	**82.11**	78.22	81.68	80.45

2.4. Blind Test in CASP14

We also test our method in the recent CASP14 blind test. Overall, our AIR ranks the ninth among 37 groups in the competition according to SUM Zscore > −2.0 (including the reference group named STARTING MODEL). There are 51 refinement targets in total, where two targets were cancelled during the competition and five targets do not have a native structure for reference. The results of AIR on the remaining 44 targets are summarized in Figure 5 (for more details, please refer to the CASP14 website). The average gains in the quality of the best model among Model 1–5 are +0.36 in GDT-TS, which is slightly lower than CASP13. The main reason is that our solution ranking method performed relatively poorly on these targets, and we found locally that a number of better structures were not selected as the top five submission models. This is also one of our future efforts to further improve the AIR program. However, there are also some successful cases. For example, AIR ranks the first on the target R1042v2 among the models submitted by 31 groups (https://predictioncenter.org/casp14/results.cgi?view=tables&target=R1042v2&model=1&groups_id=, accessed on 21 April 2021). Considering the Model 1 submission models, the AIR approach is the only one that successfully achieves improvement rather than degradation over the initial model. Considering the best submission models, our predictions for the target R1029 are among the most accurate in all submissions (https://predictioncenter.org/casp14/results.cgi?view=tables&target=R1029&model=all&groups_id=, accessed on 21 April 2021).

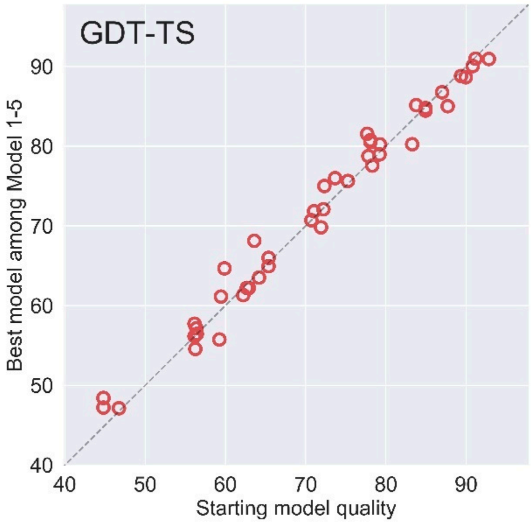


Figure 5. The overall performance on 44 refinement targets of AIR in CASP14 blind test measured by GDT-TS.

The success of these two cases indicates the potential advantages of multi-objective optimization and the PSO algorithm that can efficiently explore the high-dimensional energy landscape to get a reliable refined model. On the other hand, the performance of AIR in CASP14 also indicates that there is still room for improvement of our algorithm. For instance, on the target R1042v2, the improvement is still limited to a moderate level. For target R1029, our Model 2 submission is better than our Model 1, implying that we still need to investigate how to rank the final solution. In our AIR's future development, we will go on to find a better mechanism that could guide the search process to achieve significant improvement and a new method to accurately score all the candidate solutions.

3. Discussion

3.1. The Importance of the Diversity on AIR 2.0

In AIR 1.0, we have shown that multi-objective optimization is a promising way to improve protein structure refinement. The two goals of the multi-objective optimization are: (1) a set of solutions as close as possible to the PF; (2) a broadly distributed set of solutions that cover the entire PF [45]. The two goals are also referred to convergence and diversity. In the field of protein structure refinement, the diversity of the solutions is important. When given a model to be refined, we have no idea which direction or what combination of multiple energy functions is feasible for improvement due to the diversity of protein structures. In order to improve the initial model, AIR 2.0 tries different directions to obtain a well-distributed PF that covers all potential solutions. As mentioned before, dominance-based method AIR 1.0 prefers the regions that are easy to access, resulting in insufficient diversity, which may lose the solution diversity.

To give a more intuitive understanding of the conformation solution diversity, Figure 6 shows a comparison between AIR 1.0 and AIR 2.0 on the target R0949 from CASP13 whose PF is irregular, consisting of at least two parts. The solutions of AIR 1.0 cover only one part, and the GDT-TS of the Model 1 is 62.98, which is a degradation of the initial model with a GDT-TS of 64.53. However, Model 1 of AIR 2.0 yields a GDT-TS of 65.12 on the other part of the PF, demonstrating improvement over the initial model. Therefore, if we do not take the diversity carefully, the possibility of improvement would decrease.

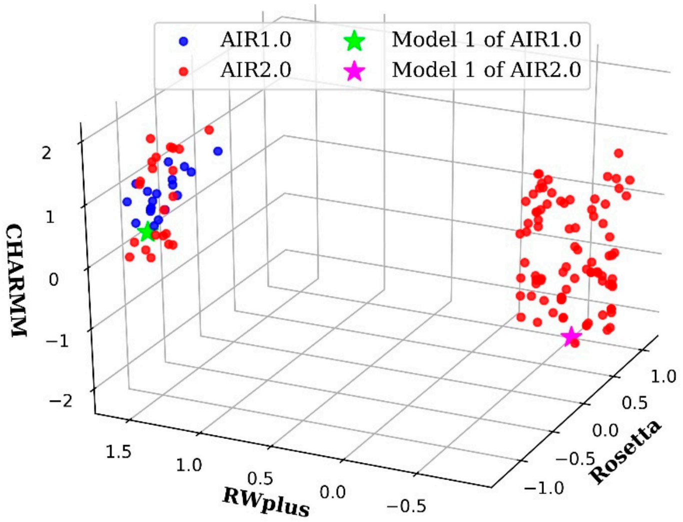


Figure 6. Candidate solutions obtained by AIR 2.0 (red) and AIR 1.0 (blue) on the target R0949. The Pareto front of R0949 is irregular and consists of at least two parts. The solutions of AIR 1.0 covers only one part, and the GDT-TS of its Model 1 marked with a green star is 62.98, which is a degradation to the initial model with a GDT-TS of 64.53. However, Model 1 of AIR 2.0 marked with a magenta star yields a GDT-TS of 65.12 on the other part of the Pareto front, demonstrating improvement over the initial model.

3.2. The Influence of Hyperparameters on AIR 2.0

The neighborhood size T (see Methods) is a major control parameter in AIR 2.0 since the solutions in the neighborhood of a subproblem can be used to guide the searching process. In a sense, each subproblem with its neighborhood is regarded as a swarm. To test the influence of T, we perform some experiments on those targets with different size. The results are summarized in Table 3. When $T = 3$, the neighborhood is too small, and the particles in a swarm are similar, resulting in the inability to explore the new area and achieve a good result. It should also be noted that AIR 2.0 performs similarly with T larger than 8. That is because the $gbest$ for each subproblem depends on only a certain number of neighbors, while others play a small role. Moreover, a large T will increase the computational burden and might undermine the diversity of solutions. Thus, we set $T = 8$ to make a tradeoff between the performance and running time.

The penalty value θ (see Methods) in the PBI decomposition approach is another important parameter [46]. In this study, we adopt an adaptive penalty scheme (APS) [47] that linearly increases θ with the number of generations from 5 to 20. At the early stage, a small θ is beneficial for convergence toward PF, and the value of θ is gradually increased to promote the diversity of solutions. For the number of iterations $MaxIt$ and the number of particles N, generally, the larger the two numbers are, the better the performance is.

However, large values of these two parameters will increase the time cost. To make a tradeoff, we finally set $MaxIt = 3000$ and $N = 66$.

Table 3. Comparison of GDT-TS gains on different value of T.

Target	Length	$T = 3$	$T = 8$	$T = 15$	$T = 30$
R0974s1	69	−0.27	1.18	0.81	1.24
R1004D2	77	−1.05	−0.97	−0.98	−0.95
R0968s1	118	2.58	2.97	3.18	2.94
R0981D5	127	−0.79	0.59	0.44	0.20
R0959	189	3.30	3.83	3.97	3.74
R0981D3	203	0.49	0.13	0	0.13

4. Methods

4.1. Overview of Refinement Pipeline AIR 2.0

As shown in Figure 7, AIR 2.0 consists of three main steps. The first step is swarm initialization, which generates multiple particles. If a single initial model is used as the input, other particle models can be generated by applying perturbations to the initial one. In total, an initial swarm of N particles is obtained. In the second step, each particle is associated with a unique weight vector generated by the simplex-lattice design method [48]. Then, the main loop of optimization is performed, where Rosetta, RWplus, and CHARMM are selected as three fitness functions. Each particle updates the position according to its own subproblem formulated by the weight vector. In each iteration, the non-dominated solutions in the whole population are added to the Pareto set. In the third step, all solutions in the Pareto set are ranked using the expected utility rule [49] and the top five of them are selected as the final refined structures.

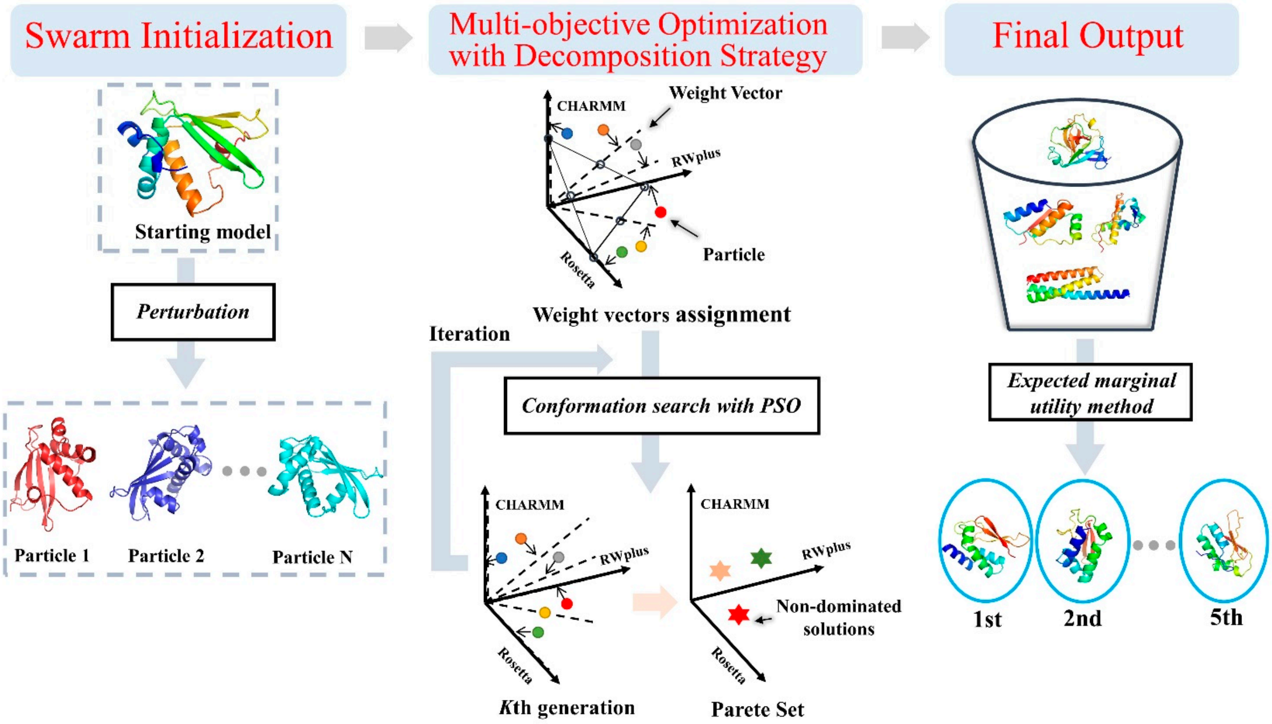


Figure 7. Overall refinement protocol of AIR 2.0. The protocol consists of swarm initialization, multi-objective optimization with decomposition strategy, and solution ranking. In the second step, the different colored circles and stars denote particles and non-dominated solutions respectively.

4.2. Representations of Protein Conformations

Mathematically, protein conformation could be represented by the Cartesian coordinates of the atoms or internal coordinates (bond lengths and angles) [50]. The former is suitable for describing physical force fields, and the latter is a better representation to describe bonded interactions as well as certain kinds of experimental information [51]. In AIR 2.0, we use both coordinate systems.

During the sampling stage, the protein backbone is represented by a list of main-chain torsion angles using internal coordinates:

$$C = [\phi_1, \psi_1, \omega_1, \ldots, \phi_L, \psi_L] \tag{1}$$

where L stands for the protein length. We further use the Denavit and Hartenberg (DH) method [52] to convert internal coordinates to corresponding Cartesian coordinates, since certain energy functions, such as the Rosetta, explicitly encode Cartesian energy terms. This conversion goes back and forth until the end of the pipeline.

4.3. Multi-Objective Optimization

Similar to its previous version, the AIR 2.0 uses three energy functions to perform conformation search in a 3D energy space composed by Rosetta, RWplus, and CHARMM. It is crucial to select accurate force fields for protein structure refinement. There are roughly two types of force fields in the community. One is physics-based force fields that are designed on the basis of all kinds of interactions at the atomic and molecular levels. The other is knowledge-based energy function deduced from diverse sets of known protein structures. Each type of force field has it its merits and drawbacks. To take advantage of both types of force fields, we choose one popular physics-based force field CHARMM and one typical knowledge-based energy function RWplus. Rosetta energy function could be classified into both types and is widely used in protein structure prediction and refinement for its good performance. Therefore, we use it as a complementary part for the other two force fields. This will formulate the protein structure refinement as a multi-objective optimization (MOP) problem as follows:

$$\begin{gathered} minimize\ F(C) = \left(f_{Rosetta}(C), f_{RWplus}(C), f_{CHARMM}(C)\right)^T \\ subject\ to\ C \in \Omega \end{gathered} \tag{2}$$

where C is the conformation of a protein and Ω is the overall conformational space. $f_{Rosetta}(C)$, $f_{RWplus}(C)$, and $f_{CHARMM}(C)$ are three energy values in terms of C.

Due to potential conflicts among multiple objectives, usually, one single solution (conformation) cannot optimize all objectives simultaneously. Instead, a set of optimal solutions representing the trade-offs among different objectives could be obtained. A dominance relation between different solutions is often used to suggest the acceptance of current conformations.

Let $C_i, C_j \in \Omega$; we say that C_i *dominates* C_j (denoted as $C_i < C_j$) if and only if $\forall k = 1,2,3, f_k(C_i) \le f_k(C_j)$ and $F(C_i) \ne F(C_j)$, where f_1, f_2, and f_3 correspond to $f_{Rosetta}(C)$, $f_{RWplus}(C)$, and $f_{CHARMM}(C)$ respectively. If $C^* \in \Omega$ and there is no other solution in Ω that dominates C^*, then C^* is considered as a Pareto optimal solution. The Pareto set (PS) is defined as:

$$PS = \{C \in \Omega | C\ is\ a\ Pareto\ optimal\ solution\}. \tag{3}$$

The energy map of all Pareto optimal solutions in PS is called a Pareto front (PF) [53] and can be described as:

$$PF = \left\{ F(C) = \left(f_{Rosetta}(C), f_{RWplus}(C), f_{CHARMM}(C)\right)^T \middle| C \in PS \right\}. \tag{4}$$

The goal of multi-objective optimization is to obtain widely distributed Pareto optimal solutions that are as close to true PF as possible.

4.4. Decomposition Approach in Multi-Objective Optimization

Our original protocol AIR 1.0 solves the multi-objective optimization based on Pareto dominance [37]. It mainly evaluates each solution by its Pareto dominance relations to other solutions and aims to drive the population toward the PF as a whole. However, the movement of each particle in the population and the distribution of computational effort over different ranges of the PF can be further investigated. Otherwise, the whole population would prefer to the regions that are easily accessible and cannot maintain the diversity of the solutions.

Generally speaking, a Pareto optimal solution for the multi-objective optimization problem can be seen as the optimal solution of a scalar optimization subproblem whose objective is an aggregation function of all the individual objectives ($f_{Rosetta}$, f_{RWplus}, f_{CHARMM}) [44]. Thus, a multi-objective optimization problem can be decomposed into a number of optimization subproblems, and each subproblem is distinguished by one unique weight vector. Then, Pareto solutions could be achieved by minimizing such subproblems. There exists several methods to construct the aggregation function [54] for each subproblem with a weight vector, such as weight sum approach [44], Tchebycheff approach [55], penalty-based boundary intersection (PBI) [44], etc. Here, AIR 2.0 uses the PBI approach to construct the aggregation function for each subproblem. Formally, an optimization subproblem in AIR 2.0 can be stated as:

$$\begin{gathered} minimize\ g^{pbi}(C|\lambda_i, z^*) = d_1 + \theta d_2 \\ subject\ to\ C \in \Omega \end{gathered} \tag{5}$$

where

$$d_1 = \frac{\| (F(C) - z^*)^T \lambda_i \|}{\| \lambda \|} \tag{6}$$

$$d_2 = \| F(C) - \left(z^* + d_1 \frac{\lambda_i}{\| \lambda_i \|} \right) \| \tag{7}$$

where C is a candidate solution (conformation) that belongs to overall conformational space. $F(C) = \left(f_{Rosetta}(C), f_{RWplus}(C), f_{CHARMM}(C) \right)^T$ consists of three components from Rosetta, RWplus, and CHARMM. $\lambda_i = (\lambda_i^1, \lambda_i^2, \lambda_i^3)^T$ is the weight vector of the ith subproblem satisfying $\lambda_i^j \geq 0$ and $\sum_{j=1}^{3} \lambda_i^j = 1$. $z^* = \left(z^*_{Rosetta}, z_{RWplus}, z^*_{CHARMM} \right)^T$ is the ideal objective vector with $z_k^* \leq \min_{C \in \Omega} f_k(C)$, $k \in \{Rosetta, RWplus, CHARMM\}$, θ is a user-defined penalty parameter. d_1 is the distance between the ideal objective vector z^* and the solution $F(C)$, d_2 is the direction error between λ_i and *F(C)*. The PBI approach tries to minimize both d_1 and d_2, and their relative importance is control by θ.

Figure 8 presents a simple example to illustrate the PBI approach given the weight vector $\lambda_i = (0.33,\ 0.33, 0.33)^T$. *F(C)* and z^* are denoted as two points in the energy map. The orange plane is PF, and d_1, d_2 are marked in the figure. It is clear that the intersection of the weight vector and PF, which is marked by a black point, is the optimal solution of the subproblem defined by PBI with λ_i .

Thus, the optimal solution to (5) is a Pareto optimal solution to (2). We use λ_i to emphasize that (5) is the ith subproblem defined by a weight vector. In order to obtain a set of different Pareto optimal solutions, we can use different weight vectors. A natural idea comes that if we have a large number of uniformly distributed weight vectors, we could get a set of Pareto optimal solutions that approximates PF very well.

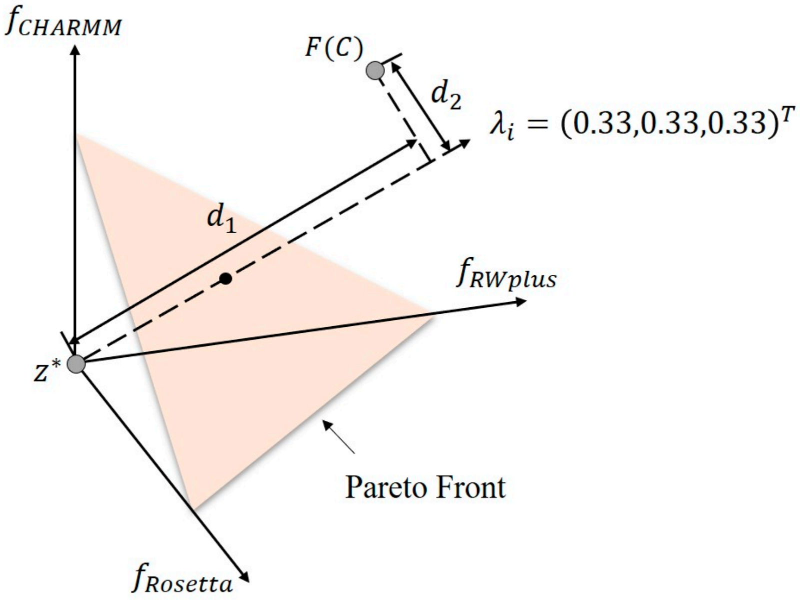


Figure 8. Illustration of PBI approach with a weight vector $\lambda_i = (0.33,\ 0.33, 0.33)^T$. *F(C)* is the corresponding point of the candidate solution in the energy space. z^* is the ideal objective vector. d_1 is the distance between the ideal objective vector z^* and the solution $F(C)$, d_2 is the direction error between λ_i and *F(C)*. The Pareto front is represented by an orange plane. The PBI approach tries to minimize both d_1 and d_2. Obviously, the intersection of the weight vector and the Pareto front marked by a black point is the optimal solution of the subproblem defined by PBI with λ_i. It should be noted that the Pareto front is always irregular and discontinuous in practice. Here, we use a simple plane to represent it just for clarity.

4.5. Particle Swarm Optimization

Particle swarm optimization (PSO) [56] is a meta-heuristic algorithm simulating the behaviors of groups of birds and fishes. It solves a problem by iteratively improving candidate solutions with the information coming from their population. Each candidate solution not only has its own exploration behavior, but its trajectory is also affected by other solutions in the population. In PSO, every individual in the population is called a particle, and swarm is another name for population. In AIR 2.0, each particle in the swarm represents a candidate conformation in the overall conformational space. A particle is characterized by its position and velocity, where the position is the conformation of a protein represented by (1) and the velocity represents the change of torsion angles. The particle uses the position of the selected global leader and its own personal movement trajectory to update the velocity and position values using (8) and (9).

$$v_i^{t+1} = w * v_i^t + c_1 * r_1 * \left(Pbest_i^t - C_i^t\right) + c_2 * r_2 * \left(Gbest_i^t - C_i^t\right) \tag{8}$$

$$C_i^{t+1} = C_i^t + v_i^{t+1} \tag{9}$$

where v_i^t is the velocity of the ith particle in the tth generation, C_i^t is the new conformation of the ith particle in the tth generation, and w is the inertia weight.

According to our previous study [35], we set w to 1.3 at the beginning; it linearly decreases to 0.7 as the number of iterations increases. c_1 and c_2 are two learning coefficients that are both set to be 2 [57]. r_1,r_2 $\in$ $[0, 1]$ are uniformly distributed random variables. $Pbest_i^t$ is the best conformation that the ith particle has ever been until the tth generation. Similarly, $Gbest_i^t$ is the best conformation that the whole swarm has ever met until the tth generation. Each time the conformation updates, the non-dominated ones are added into PS.

For only one objective, each solution can be ranked according to the objective. Thus, both $Pbest_i^t$ and $Gbest_i^t$ have a determined choice. However, for multiple objectives, there are always many non-dominated solutions that are equally good under the concept of the dominance. Thus, it is difficult to choose $Pbest_i^t$ and $Gbest_i^t$ to lead the searching process. In our previous protocol of AIR 1.0, $Pbest_i^t$ is updated when any one of the three energy functions decreases, and $Gbest_i^t$ is randomly selected from current PS. This will severely deteriorate the selection pressure toward the PF and considerably slows down the searching process due to ambiguous search direction.

4.6. Obtaining Pareto Optimal Set with Multi-Objective Particle Swarm Optimization Based on Decomposition Strategy

Due to the diversity of protein structures, we use multiple energy functions as multi-objectives to alleviate the bias problem caused by minimizing one single energy function. Given a particular protein, which energy function or what combination of these energy functions is appropriate for a particular protein is still unknown. Each candidate solution on PF represents a potential optimization direction. Thus, the diversity of solutions on the PF is of importance for multi-objective optimizations in protein structure refinement. In order to make good use of three energy functions, we need to find as many Pareto optimal solutions as possible and maximize the distribution of solutions in the PF. However, using Pareto dominance alone could discourage the diversity of solutions, since it has no direct control over the movement of each individual in its population and no good mechanism to control the distribution of the computational effort over different ranges of the PF. As a result, the whole population is updated in a random direction and prefers those regions that are easily accessible. Finally, the solutions will end up in a small range of PF, resulting in the loss of the diversity.

In order to overcome the above shortcomings, AIR 2.0 uses a decomposition strategy to define a single objective optimization subproblem for each particle. A Pareto optimal solution to an MOP could be an optimal solution of a scalar optimization subproblem, in which the objective is an aggregation of all the objectives in AIR 2.0. In this way, each particle has an exact updating direction and increasing evolutionary pressure, which is beneficial to the convergence. In addition, the diversity is inherently guaranteed since each particle is moving toward PF in its own direction. The general framework is as follows.

At the beginning, a set of weight vectors $\{\lambda_1, \ldots, \lambda_N\}$ (N is the number of particles) are generated using the canonical simplex-lattice design method [48], whose weight vectors are sampled from a unit simplex.

$$\begin{cases} \lambda_i = (\lambda_i^1, \lambda_i^2, \lambda_i^3) \\ \lambda_i^j \in \left\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\right\}, \sum\limits_{j=1}^{3} \lambda_i^j = 1 \end{cases} \tag{10}$$

where $i = 1, \ldots, N$ is the index of uniformly distributed weight vector. λ_i has three components corresponding to three energy functions, Rosetta, RWplus, and CAHRMM. $H > 0$ is the number of divisions along each objective coordinate. In total, there are $N = \begin{pmatrix} H+M-1 \\ M-1 \end{pmatrix} = \begin{pmatrix} H+3-1 \\ 3-1 \end{pmatrix}$ different weight vectors for $M = 3$ objectives. Then, each particle is associated with a different weight vector, which defines a unique subproblem. Solving these subproblems is equivalent to solving the original multi-objective optimization problem.

In AIR 1.0, the velocity and position of a particle are updated using the information from its individual best $Pbest_i^t$ and the global best $Gbest_i^t$. However, it is difficult to select a suitable one, since multiple objectives result in a large number of equally good non-dominated solutions. Now with the decomposition strategy, $Pbest_i^t$ is obvious for a particle with a weight vector λ_i using the aggregation function $g^{PBI}(C|\lambda^i, z^*)$. For $Gbest_i^t$, there is a small difference, since each particle corresponds to a different subproblem. However, if two weight vectors λ_i and λ_j are close enough, the optimal solutions to both two subproblems,

$g^{PBI}(C|\lambda_i, z^*)$ and $g^{PBI}\left(C\middle|\lambda_j, z^*\right)$, will also be similar. Therefore, the information from the searching process of λ_i is useful to λ_j and vice versa. According to this observation, a neighborhood of the weight vector λ_i is defined as a set of weight vectors $\{\lambda_{i_1}, \lambda_{i_2} \ldots, \lambda_{i_T}\}$ that are closest to λ_i, where T is the size of the neighborhood. Correspondingly, the neighborhood of the ith particle is composed of those particles whose weight vectors are in the neighborhood of λ_i. With the notion of neighborhood, $Gbest_i^t$ is defined as the best position in the neighborhood of the ith particle during t generations. Then, we could use the particle swarm optimization algorithm to optimize those subproblems simultaneously and finally obtain a Pareto optimal set. The pseudocode of the main framework for AIR 2.0 is summarized in Algorithm 1.

Algorithm 1 Main Framework of AIR 2.0

```
Input: Initial model $C_0$, the maximum number of iterations $MaxIT$, the number of particles $N$.
Output: Pareto set $PS$.
/*Initialization*/
Generate weight vectors $\{\lambda_1, \ldots, \lambda_N\}$ using the simplex lattice method.
Create initial population $\{C_1^0, \ldots, C_N^t\}$ by perturbing $C_0$ and assign weight vectors to each particle individually.
Compute the Euclidean distances between any two weight vectors. For each particle $C_i^0$, $i = 1, \ldots, N$, set $B(i) = \{i_1, ..i_T\}$, where $\lambda_{i_1}, \ldots, \lambda_{i_T}$ are the $T$ closest weight vectors to $\lambda_i$.
Initialize ideal objective vector $z^*$, set the initial velocity randomly, $pbest_i^0 = gbest_i^0 = C_i^0$, add initial non-dominated particles into $PS$. /*Main Loop*/
while $t < MaxIt$ do:
    for $i = 1, \ldots, N$ do:
        Update the position of the $C_i^t$ using PSO Formulas (8) and (9).
        Update $z^*$
        if $g(C_i^t|\lambda_i) < g(pbest_i^t|\lambda_i)$ then
            $pbest_i^t = C_i^t$
        end if
        for each $j \in B(i)$ do:
            if $g\left(C_i^t\middle|\lambda_j\right) < g\left(gbest_j^t\middle|\lambda_j\right)$ then
                $gbest_j^t = C_i^t$
            end if
        end for
        Remove all the vectors dominated by $C_i^t$ from $PS$.
        Add $C_i^t$ into $PS$ if no vectors in $PS$ dominate it.
    end for
end while
```

4.7. Model Selection

After enough iterations, there are plenty of non-dominated solutions or candidate models in the PS. To obtain the final refined structures, we need to assess and rank the generated models. Many methods for estimation of model accuracy have been described such as MULTICOM_CLUSTER [10], Pcons [58], PRESCO [59], DeepAccNet [60], and ProQ3D [61]. Here, for a quick ranking purpose, we use a widely used knee-based ranking method [49] in a multi-objective optimization problem to rank those models. Since the three energy functions are treated equally, AIR 2.0 has no preference to any regions of the PF. However, there will be some special solutions called 'knee' in the PF. In such 'knees' solutions, a small improvement in one objective will cause large depravation in other objectives. Thus, three objectives reach a balance that all objectives are relatively optimal and no objective can decrease further without seriously increasing other objectives.

To obtain these 'knee' solutions, we adopt a utility-based method similar to the AIR 1.0, which uses the expected marginal utility to measure the importance of the solutions in the PS:

$$\begin{aligned} &U_{C,w} = w_1 f_1(C) + w_2 f_2(C) + w_3 f_3(C)) \\ &s.t. \quad w_1 + w_2 + w_3 = 1 \; and \; w_1, w_2, w_3 \geq 0 \end{aligned} \tag{11}$$

where C is the non-dominated solution in the PS and w_1, w_2, w_3 are the weight coefficients. The expected margin utility is approximated by random sampling of w_i. For each conformation, we obtain a large number of utility values $\left\{U^i_{C,w}, i = 1, \ldots S\right\}$ by using different combinations of weight coefficients. The expected margin utility could be approximated by the average of these sample values:

$$E(U_{C,w}) = \frac{1}{S} \sum_{i=1}^{S} U^i_{C,w}. \tag{12}$$

Then, we can rank the solutions in PS according to individual expected marginal utility and output the top-ranking solutions.

5. Conclusions and Future Direction

In this study, we developed a decomposition-based method AIR 2.0 for protein structure refinement. AIR 2.0 employs a decomposition strategy that divides the multi-objective optimization into a set of subproblems and optimizes them in a collaborative manner. The performance on CASP 13 refinement targets and a blind test on CASP 14 shows that AIR 2.0 is capable of achieving promising results. In the future, we will further improve the AIR refinement protocol to use deep learning methods to design new energy functions that could guide the search process and identify those local structure regions need to be refined. With this information, we could reduce the searching space largely and make the sampling process more efficient. Moreover, the new energy function could be used as the final model selection criterion to rank the models in Pareto set, which may bridge the gap between Model 1 and the best model.

Author Contributions: Conceptualization, C.-P.Z. and H.-B.S.; Data curation, D.W.; Formal analysis, C.-P.Z., D.W. and H.-B.S.; Funding acquisition, H.-B.S.; Investigation, C.-P.Z.; Methodology, C.-P.Z., D.W., X.P. and H.-B.S.; Project administration, H.-B.S.; Resources, H.-B.S.; Software, C.-P.Z.; Supervision, H.-B.S.; Validation, C.-P.Z., D.W. and H.-B.S.; Visualization, C.-P.Z.; Writing—Original draft, C.-P.Z.; Writing—Review & editing, X.P. and H.-B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 61725302, 62073219, 61903248, 92059206), and the Science and Technology Commission of Shanghai Municipality (No. 20S11902100).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data about our method could be obtained through the implementation above. The data about other methods is available on CASP official website.

Conflicts of Interest: The authors declare no conflict of interest.

Availability and Implementation: AIR 2.0 is available online: www.csbio.sjtu.edu.cn/bioinf/AIR/ (accessed on 21 April 2021).

References

1. Schwede, T.; Peitsch, M.C. *Computational Structural Biology: Methods and Applications*; World Scientific: Singapore, 2008.
2. Dill, K.A.; MacCallum, J.L. The protein-folding problem, 50 years on. *Science* **2012**, *338*, 1042–1046. [CrossRef]
3. Kihara, D.; Lu, H.; Kolinski, A.; Skolnick, J. TOUCHSTONE: An ab initio protein structure prediction method that uses threading-based tertiary restraints. *Proc. Natl. Acad. Sci. USA* **2001**, *98*, 10125–10130. [CrossRef]

4. Leaver-Fay, A.; Tyka, M.; Lewis, S.M.; Lange, O.F.; Thompson, J.; Jacak, R.; Kaufman, K.; Renfrew, P.D.; Smith, C.A.; Sheffler, W.; et al. ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol.* **2011**, *487*, 545–574.
5. Xu, D.; Zhang, Y. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins Struct. Funct. Bioinform.* **2012**, *80*, 1715–1735. [CrossRef]
6. Senior, A.W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Zidek, A.; Nelson, A.W.R.; Bridgland, A.; et al. Improved protein structure prediction using potentials from deep learning. *Nature* **2020**, *577*, 706–710. [CrossRef]
7. Yang, J.; Anishchenko, I.; Park, H.; Peng, Z.; Ovchinnikov, S.; Baker, D. Improved protein structure prediction using predicted interresidue orientations. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 1496–1503. [CrossRef]
8. Yang, J.; Yan, R.; Roy, A.; Xu, D.; Zhang, Y. The I-TASSER suite: Protein structure and function prediction. *Nat. Methods* **2014**, *12*, 7–8. [CrossRef] [PubMed]
9. Renzhi, C.; Debswapna, B.; Badri, A.; Jilong, L.; Jianlin, C. Massive integration of diverse protein quality assessment methods to improve template based modeling in CASP11. *Proteins Struct. Funct. Bioinform.* **2015**, *84*, 247–259.
10. Hou, J.; Wu, T.; Cao, R.; Cheng, J. Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13. *Proteins Struct. Funct. Bioinform.* **2019**, *87*, 1165–1178. [CrossRef] [PubMed]
11. Kamisetty, H.; Ovchinnikov, S.; Baker, D. Assessing the utility of coevolution-based residue-residue contact predictions in a sequence- and structure-rich era. *Proc. Natl. Acad. Sci. USA* **2013**, *110*, 15674–15679. [CrossRef]
12. Wu, T.; Guo, Z.; Hou, J.; Cheng, J. DeepDist: Real-value inter-residue distance prediction with deep residual convolutional network. *BMC Bioinform.* **2021**, *22*, 1–17. [CrossRef]
13. Adhikari, B.; Cheng, J. CONFOLD2: Improved contact-driven ab initio protein structure modeling. *BMC Bioinform.* **2018**, *19*, 22. [CrossRef]
14. Adhikari, B.; Hou, J.; Cheng, J. Protein contact prediction by integrating deep multiple sequence alignments, coevolution and machine learning. *Proteins Struct. Funct. Bioinform.* **2018**, *86*, 84–96. [CrossRef] [PubMed]
15. Kandathil, S.M.; Greener, J.G.; Jones, D.T. Recent developments in deep learning applied to protein structure prediction. *Proteins* **2019**, *87*, 1179–1189. [CrossRef] [PubMed]
16. Lee, G.R.; Won, J.; Heo, L.; Seok, C. GalaxyRefine2: Simultaneous refinement of inaccurate local regions and overall protein structure. *Nucleic Acids Res.* **2019**, *47*, W451–W455. [CrossRef] [PubMed]
17. Hou, J.; Adhikari, B.; Cheng, J. DeepSF: Deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics* **2018**, *34*, 1295–1303. [CrossRef] [PubMed]
18. Heo, L.; Arbour, C.F.; Feig, M. Driven to near-experimental accuracy by refinement via molecular dynamics simulations. *Proteins* **2019**, *87*, 1263–1275. [CrossRef] [PubMed]
19. Hovan, L.; Oleinikovas, V.; Yalinca, H.; Kryshtafovych, A.; Saladino, G.; Gervasio, F.L. Assessment of the model refinement category in CASP12. *Proteins* **2018**, *86* (Suppl. 1), 152–167. [CrossRef]
20. Modi, V.; Dunbrack, R.L., Jr. Assessment of refinement of template-based models in CASP11. *Proteins* **2016**, *84* (Suppl. 1), 260–281. [CrossRef]
21. Read, R.J.; Sammito, M.D.; Kryshtafovych, A.; Croll, T.I. Evaluation of model refinement in CASP13. *Proteins* **2019**, *87*, 1249–1262. [CrossRef]
22. Heo, L.; Feig, M. Experimental accuracy in protein structure refinement via molecular dynamics simulations. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 13276–13281. [CrossRef] [PubMed]
23. Heo, L.; Park, H.; Seok, C. GalaxyRefine: Protein structure refinement driven by side-chain repacking. *Nucleic Acids Res.* **2013**, *41*, W384–W388. [CrossRef] [PubMed]
24. Park, H.; Lee, G.R.; Kim, D.E.; Anishchenko, I.; Cong, Q.; Baker, D. High-accuracy refinement using Rosetta in CASP13. *Proteins* **2019**, *87*, 1276–1282. [CrossRef] [PubMed]
25. Terashi, G.; Kihara, D. Protein structure model refinement in CASP12 using short and long molecular dynamics simulations in implicit solvent. *Proteins Struct. Funct. Bioinform.* **2018**, *86*, 189–201. [CrossRef] [PubMed]
26. Tian, C.; Kasavajhala, K.; Belfon, K.A.A.; Raguette, L.; Huang, H.; Migues, A.N.; Bickel, J.; Wang, Y.; Pincay, J.; Wu, Q.; et al. ff19SB: Amino-Acid-Specific Protein Backbone Parameters Trained against Quantum Mechanics Energy Surfaces in Solution. *J. Chem. Theory Comput.* **2020**, *16*, 528–552. [CrossRef] [PubMed]
27. Huang, J.; Rauscher, S.; Nawrocki, G.; Ran, T.; Feig, M.; de Groot, B.L.; Grubmuller, H.; MacKerell, A.D., Jr. CHARMM36m: An improved force field for folded and intrinsically disordered proteins. *Nat. Methods* **2017**, *14*, 71–73. [CrossRef]
28. Jorgensen, W.L.; Maxwell, D.S. Development and testing of the OPLS all-atom force field on conformational energetics. *J. Am. Chem. Soc.* **1996**, *118*, 11225–11236. [CrossRef]
29. Zhang, J.; Zhang, Y. A Novel Side-Chain Orientation Dependent Potential Derived from Random-Walk Reference State for Protein Fold Selection and Structure Prediction. *PLoS ONE* **2010**, *5*, e15386. [CrossRef]
30. Zhou, H.; Zhou, Y. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein Sci.* **2002**, *11*, 2714–2726. [CrossRef]
31. Zhou, H.; Skolnick, J. GOAP: A Generalized Orientation-Dependent, All-Atom Statistical Potential for Protein Structure Prediction. *Biophys. J.* **2011**, *101*, 2043–2052. [CrossRef] [PubMed]

32. Alford, R.F.; Leaver-Fay, A.; Jeliazkov, J.R.; O'Meara, M.J.; DiMaio, F.P.; Park, H.; Shapovalov, M.V.; Renfrew, P.D.; Mulligan, V.K.; Kappel, K.; et al. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *J. Chem. Theory Comput.* **2017**, *13*, 3031–3048. [CrossRef] [PubMed]
33. Heo, L.; Feig, M. PREFMD: A web server for protein structure refinement via molecular dynamics simulations. *Bioinformatics* **2018**, *34*, 1063–1065. [CrossRef]
34. Rohl, C.A.; Strauss, C.E.M.; Misura, K.M.S.; Baker, D. Protein structure prediction using Rosetta. *Methods Enzymol.* **2003**, *383*, 66.
35. Wang, D.; Geng, L.; Zhao, Y.J.; Yang, Y.; Huang, Y.; Zhang, Y.; Shen, H.B. Artificial intelligence-based multi-objective optimization protocol for protein structure refinement. *Bioinformatics* **2020**, *36*, 437–448. [CrossRef] [PubMed]
36. Moore, J.; Chapman, R.; Dozier, G. ACM Press the 38th annual. In Proceedings of the 38th Annual on Southeast Regional Conference, ACM-SE 38, Multiobjective Particle Swarm Optimization, Clemson, SC, USA, 7–8 April 2000; p. 56.
37. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
38. Coello, C.A.C.; Pulido, G.T.; Lechuga, M.S. Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 256–279. [CrossRef]
39. Hui, L.; Qingfu, Z. Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.* **2009**, *13*, 284–302. [CrossRef]
40. Trivedi, A.; Srinivasan, D.; Sanyal, K.; Ghosh, A. A Survey of Multiobjective Evolutionary Algorithms based on Decomposition. *IEEE Trans. Evol. Comput.* **2016**, *21*, 440–462. [CrossRef]
41. Zhang, J.; Liang, Y.; Zhang, Y. Atomic-level protein structure refinement using fragment-guided molecular dynamics conformation sampling. *Structure* **2011**, *19*, 1784–1795. [CrossRef] [PubMed]
42. Cozzetto, D.; Kryshtafovych, A.; Fidelis, K.; Moult, J.; Tramontano, A. Evaluation of template-based models in CASP8 with standard measures. *Proteins Struct. Funct. Bioinform.* **2009**, *77* (Suppl. 9), 18–28. [CrossRef]
43. Zhang, Y.; Skolnick, J. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* **2005**, *33*, 2302–2309. [CrossRef]
44. Qingfu, Z.; Hui, L. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
45. Cheng, R.; Jin, Y.; Olhofer, M.; Sendhoff, B. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. *IEEE Trans. Evol. Comput.* **2016**, *20*, 773–791. [CrossRef]
46. Mohammadi, A.; Omidvar, M.N.; Li, X.; Deb, K. Sensitivity analysis of Penalty-based Boundary Intersection on aggregation-based EMO algorithms. In Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 25–28 May 2015.
47. Yang, S.; Jiang, S.; Jiang, Y. Improving the Multiobjective Evolutionary Algorithm Based on Decomposition with New Penalty Schemes. *Soft Comput.* **2016**, *21*, 4677–4691. [CrossRef]
48. Das, I.; Dennis, J.E. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *Siam J. Opt.* **1996**, *8*, 631–657. [CrossRef]
49. Branke, J.; Deb, K.; Dierolf, H.; Osswald, M. Finding knees in multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*; Springer: Berlin/Heidelberg, Germany, 2004.
50. Parsons, J.; Holmes, J.B.; Rojas, J.M.; Tsai, J.; Strauss, C.E. Practical conversion from torsion space to Cartesian space for in silico protein synthesis. *J. Comput. Chem.* **2005**, *26*, 1063–1068. [CrossRef]
51. AlQuraishi, M. Parallelized Natural Extension Reference Frame: Parallelized Conversion from Internal to Cartesian Coordinates. *J. Comput. Chem.* **2019**, *40*, 885–892. [CrossRef] [PubMed]
52. Zhang, M.; Kavraki, L.E. A New Method for Fast and Accurate Derivation of Molecular Conformations. *J. Chem. Inform. Model.* **2002**, *42*, 64–70. [CrossRef]
53. Tripathi, P.K.; Bandyopadhyay, S.; Pal, S.K. Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients. *Inform. Sci.* **2007**, *177*, 5033–5049. [CrossRef]
54. Zapotecas Martínez, S.; Coello Coello, C.A. A multi-objective particle swarm optimizer based on decomposition. In Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, Dublin, Ireland, 12–16 July 2011; p. 69.
55. Miettinen, K. *Nonlinear Multiobjective Optimization*; Springer: Berlin/Heidelberg, Germany, 1998.
56. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In *Book Particle Swarm Optimization*; BoD—Books on Demand GmbH: Norderstedt, Germany, 2002.
57. Parsopoulos, K.E.; Vrahatis, M.N. Particle swarm optimization method in multiobjective problems. In Proceedings of the 2002 ACM Symposium on Applied Computing, Madrid, Spain, 10–14 March 2002.
58. Wallner, B.; Fang, H.; Elofsson, A. Automatic consensus-based fold recognition using Pcons, ProQ, and Pmodeller. *Proteins Struct. Funct. Bioinform.* **2003**, *53*, 534–541. [CrossRef]
59. Kim, H.; Kihara, D. Detecting local residue environment similarity for recognizing near-native structure models. *Proteins* **2014**, *82*, 3255–3272. [CrossRef] [PubMed]
60. Hiranuma, N.; Park, H.; Baek, M.; Anishchenko, I.; Dauparas, J.; Baker, D. Improved protein structure refinement guided by deep learning based accuracy estimation. *Nat. Commun.* **2021**, *12*, 1340. [CrossRef] [PubMed]
61. Uziela, K.; Menendez Hurtado, D.; Shu, N.; Wallner, B.; Elofsson, A. ProQ3D: Improved model quality assessments using deep learning. *Bioinformatics* **2017**, *33*, 1578–1580. [CrossRef] [PubMed]