A Hybrid Metaheuristic for Molecular Docking

Osmar Aguiar Ribeiro Jr[1], **Omar Andres Carmona Cortes**[2,1], **João Otávio Bandeira Diniz**[3,1]

[1]Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Universidade Federal do Maranhão (UFMA)
Av. dos Portugueses, 1966 -– São Luís -– MA -– Brazil

[2]Departamento de Computação (DComp)
Instituto Federal do Maranhão (IFMA)
Av. Getúlio Vargas, 04 -– São Luís -– MA— Brazil

[3]Synthesis Lab -– Fábrica de Inovação
Instituto Federal do Maranhão (IFMA)
Campus Grajaú -– MA -– Brazil

osmar.aguiar@discente.ufma.br, {omar,joaobandeira}@ifma.edu.br

Abstract. *This paper presents a hybrid metaheuristic algorithm, ABC-GA-VGOS, integrating Artificial Bee Colony (ABC), Genetic Algorithm (GA), and Variable Genetic Operator Search (VGOS) for molecular docking. The algorithm initializes a population of size N, evaluates fitness using a heuristic docking energy (k-d tree with distance-based penalties), and employs adaptive crossover and mutation probabilities (p_c, p_m) to balance exploration and exploitation. Elitism preserves the best individual, while Solis-Wets' local search refines solutions only when improvements are detected, optimizing computational efficiency. Experimental results on ten protein–peptide systems (1JSU, 1A1N, 1BE9, 1BXO, 1BXL, 1YCR, 2AN6, 2BBA, 2IGF, 4QVE) confirmed the robustness of the proposed method. In simpler systems such as 1JSU and 1A1N, all algorithms achieved 100% success rates ($RMSD \leq 2.0$ Å) with mean RMSDs of 0.63 Å and 0.57 Å, respectively. In intermediate cases (e.g., 1BE9, 2AN6, 2BBA, 2IGF, 4QVE), the hybrid algorithm consistently outperformed the baselines, yielding more negative binding energies, lower RMSDs, and higher success rates. For highly complex systems (1BXL, 1BXO, 1YCR), none of the methods succeeded ($RMSD > 3.5$ Å), underscoring the challenges of flexible peptide docking. These results highlight the reliability of the proposed hybrid, making it a robust tool for protein–peptide docking, with potential for further optimization via advanced operator selection or differentiable energy models.*

1. Introduction

Rational drug discovery increasingly relies on computational tools to predict biomolecular interactions, particularly between proteins and ligands. Molecular docking plays a central role by estimating the most favorable binding conformations from three-dimensional structural data. However, the inherent flexibility of ligands, and in some cases receptors, renders docking computationally demanding, and often we have to deal with an NP-hard problem [Leonhart and Dorn 2019].

Conventional docking pipelines combine conformational sampling with scoring functions to identify near-native poses. While effective, their accuracy remains limited by the quality of scoring functions. To address this issue, metaheuristic strategies, including evolutionary and swarm intelligence algorithms, have been investigated to improve sampling efficiency and pose prediction in protein–ligand docking.

In this context, we present ABC-GA-VGOS, a hybrid metaheuristic that integrates Artificial Bee Colony (ABC), Genetic Algorithms (GA), and Variable Genetic Operator Search (VGOS). By balancing global exploration with local refinement, our method achieves computational efficiency while maintaining robustness. Future developments will extend this framework to incorporate partial receptor flexibility, enhancing its applicability to biologically realistic docking scenarios.

Thus, this work is divided as follows: Section 2 shows how the area has evolved by presenting some related works; Section 3 presents our proposal and how the experiments have been performed; Section 4 shows results analyzing time, success rate, a comparison against AutoDock Vina, and statistical results; Finally, Section 5 summarizes our results and proposes future work.

2. Related Work

Computational docking has advanced through the integration of evolutionary heuristics, local search, and hybrid strategies. Masoudi-Sobhanzadeh et al. (2021) proposed VGOS, a hybrid algorithm that combines adaptive genetic operators with local refinements such as Solis-Wets and L-BFGS, achieving higher solution quality than classical approaches like AutoDock Vina [Masoudi-Sobhanzadeh et al. 2021]. Along the same line, Tavares et al. (2009) analyzed the role of hybrid methods, showing that while Solis-Wets yields modest gains, deterministic refinements such as L-BFGS can substantially improve accuracy [Tavares et al. 2009]. Building on this idea, Leonhart and Dorn (2019) introduced a memetic algorithm that couples BRKGA with hill-climbing and simulated annealing, outperforming Vina and DockThor on standard benchmarks [Leonhart and Dorn 2019].

Swarm-intelligence techniques have also been explored as alternatives. Zhou et al. (2023), for example, proposed a multi-swarm competitive algorithm that integrates differential updates with gradient-based refinement, effectively addressing flexible ligand docking challenges [Zhou et al. 2023].

More recently, machine learning (ML) and deep learning (DL) have reshaped docking research by introducing data-driven scoring functions. Shirali et al. (2025) conducted a large-scale comparison of DL-based versus classical scoring functions across seven datasets, reporting significant improvements in predictive accuracy and runtime [Shirali et al. 2025]. To support this shift, Morehead et al. (2024) released PoseBench, a benchmark designed to evaluate DL-based docking in realistic settings [Morehead et al. 2024]. Similarly, Gnina 1.3 incorporated CNN-based scoring and covalent docking in an open-source framework [McNutt et al. 2025]. Other recent advances include DeepRLI, a graph-based multitask docking model [Lin et al. 2025], and reinforcement learning approaches such as the latent-space fine-tuning strategy of Sob et al. (2024), which enhanced docking hit rates [Sob et al. 2024].

On the other hand, our method, ABC-GA-VGOS, complements AI-driven strategies by offering a hybrid metaheuristic that is computationally efficient, interpretable,

and extendable. Its design is naturally suited for integration with receptor flexibility and learned scoring functions, positioning it as a bridge between classical heuristics and next-generation ML-based docking systems.

3. Materials and Method

In this work, the molecular docking was treated as a multidimensional optimization problem, considering the following degrees of freedom for the ligand:

- **Translation:** the ligand can move freely within a user-defined 3D search box centered on the receptor's region of interest.
- **Orientation:** full rotational freedom of the ligand.
- **Internal flexibility:** rotatable bonds in the ligand are modeled as continuous variables, allowing the generation of different conformations.

The receptor is treated as rigid in the current implementation, with the active site defined by a bounding box. As future work, we plan to incorporate partial receptor flexibility, enabling side-chain rotations in key residues to improve biological realism.

3.1. The Hybrid Algorithm

The proposed hybrid algorithm, ABC-GA-VGOS, combines three metaheuristics: Artificial Bee Colony (ABC), Genetic Algorithm (GA), and Variable Genetic Operator Search (VGOS). The population of size N is initialized randomly within the search domain. Each individual represents a candidate docking pose. The algorithm evaluates the energy of each solution, preserves the best individual via elitism, and applies adaptive crossover and mutation operators. Local refinement is performed by Solis-Wets search applied to elite solutions, ensuring exploitation without compromising global exploration.

3.2. Objective Function and Constraints

The fitness evaluation is based solely on a heuristic docking energy, denoted as E_{heur}, computed through a k-d tree structure that efficiently measures atom–atom distances and applies penalties for steric clashes, exposure, and lack of favorable contacts. Unlike earlier approaches, the Root Mean Square Deviation (RMSD) is not included in the optimization objective. Instead, RMSD is computed only as a post-optimization metric to evaluate the accuracy of the predicted docking poses relative to experimental structures. For comparative purposes, the AutoDock Vina scoring function (E_{vina}) was also implemented. However, it is applied exclusively for post-hoc evaluation and reproducibility of results, not during the optimization itself. Thus, the fitness function is expressed as:

$$\text{Fitness}(x) = E_{\text{heur}}(x) \tag{1}$$

The implementation of E_{heur} utilizes a k-d tree structure to compute spatial distances between ligand and receptor atoms efficiently. The score combines multiple geometric criteria: (i) a quadratic penalty for atomic overlaps when any ligand atom is closer than 1.0 Å to the receptor, with hard exclusion for severe clashes under 0.7 Å; (ii) a reward for favorable interactions when interatomic distances fall between 2.5 Å and 4.5 Å; and (iii) a penalization for ligand poses with their center of mass exposed beyond the protein's effective radius. This formulation ensures that optimization is guided exclusively by energetic criteria, while RMSD is retained as an independent measure of docking accuracy in the results section. The complete algorithm is shown in Algorithm 1.

Algorithm 1 Optimized Hybrid ABC-GA-VGOS

Initialize population P with N individuals uniformly in domain
Evaluate P: compute **fitness** $= E_{\text{heur}}(x)$ for each individual
(Optional) compute RMSD only for reporting if reference pose is provided
Set initial adaptive probabilities: $p_c \leftarrow 0.5$, $p_m \leftarrow 0.5$
for generation $g = 1$ to G **do**
    Identify elite individual (lowest fitness)
    Initialize new population P' with elite
    Initialize success and total counters for crossover and mutation
    while $|P'| < N$ **do**
        Select parents p_1, p_2 via tournament
        if $\text{rand}() < p_c$ **then**
            Perform arithmetic crossover $\rightarrow$ child
        else
            Perform binary crossover $\rightarrow$ child
        end if
        Update crossover counters
        if $\text{rand}() < p_m$ **then**
            Apply Gaussian mutation to child
        else
            Apply random mutation to child
        end if
        Clamp child to search domain
        Evaluate child: **fitness** $= E_{\text{heur}}(\text{child})$
        (Optional) store RMSD(child) only for reporting if reference is provided
        if child is feasible and improves over at least one parent **then**
            Add child to P'
            Update success counters accordingly
        end if
    end while
    $P \leftarrow P'$
    Apply Solis–Wets local search on elite of P (accept only if fitness improves)
    Adapt probabilities p_c and p_m based on success ratios
end for
return best solution, best **fitness** E_{heur}, and *RMSD for reporting (if available)*

3.3. Scoring Function Based on AutoDock Vina

The scoring function (E_{vina}) implemented in this work follows the methodology of AutoDock Vina, combining five interaction terms. The mathematical formulation is given by Equation 2.

$$E_{\text{vina}} = w_1 \cdot \text{Gauss}_1 + w_2 \cdot \text{Gauss}_2 + w_3 \cdot \text{Repulsion} + w_4 \cdot \text{Hydrophobic} + w_5 \cdot \text{HydrogenBond} \tag{2}$$

Each term corresponds to a specific type of interaction between ligand and receptor atoms: two Gaussian functions model attractive potentials at different ranges, a quadratic repulsion penalizes atomic overlaps, a hydrophobic contribution accounts for favorable contacts in non-polar regions, and a directional term models hydrogen bonding. The exact parameters used in our implementation are listed in Table 1, ensuring reproducibility of results.

Table 1. Parameters used in the implemented Vina scoring function.

Term	Weight	Parameters	Valid Range
Gauss 1	-0.04429344	$\mu = 0.0$, $\sigma = 0.3$	all distances
Gauss 2	-0.006283	$\mu = 3.0$, $\sigma = 2.0$	all distances
Repulsion	$+0.25167474$	cutoff $r < 1.5$	short-range
Hydrophobic	$+0.00786115$	linear decrease	$0.5 \leq r \leq 5.0$
Hydrogen bond	-0.48626183	centered at 1.9 Å, quadratic decay	$r < 3.0$

These values are consistent with the official AutoDock Vina parameterization and were applied identically across all experiments in this study.

3.4. Adaptive Operators

To balance exploration and exploitation, the hybrid algorithm employs adaptive probabilities for crossover and mutation. Initially, the probability of applying arithmetic crossover (p_c) and Gaussian mutation (p_m) is set to 0.5. During execution, these probabilities are dynamically updated based on the observed success rates of each operator.

Let $success_c$ and $success_m$ be the number of successful crossover and mutation operations, and $total_c$ and $total_m$ the total number of attempted operations. The updated probabilities are given by Equation 3.

$$p_c = \min\left(\max\left(\frac{success_c}{total_c}, 0.1\right), 0.9\right), \quad p_m = \min\left(\max\left(\frac{success_m}{total_m}, 0.1\right), 0.9\right). \tag{3}$$

This adaptive mechanism ensures that the algorithm progressively favors the operators with higher success rates, improving convergence and solution quality while avoiding premature stagnation.

3.5. Local Search Strategy: Solis-Wets

To enhance the accuracy of the elite solutions found by the hybrid algorithm, we apply the Solis-Wets local search method [Tavares et al. 2009]. The main motivation for employing this method is its simplicity, computational efficiency, and proven effectiveness for refining solutions in continuous optimization problems.

In our implementation, the Solis-Wets algorithm operates by perturbing the current best solution x with a randomly generated vector ε. Each component of ε is independently sampled from a uniform distribution within a specified bound δ as in Equation 4.

$$x_{\text{new}} = \text{clip}(x + \varepsilon), \quad \text{where} \quad \varepsilon \sim \mathcal{U}(-\delta, \delta) \tag{4}$$

After the perturbation, the new candidate solution x_{new} is evaluated against the original solution x. If x_{new} yields a better fitness value and meets all structural constraints (such as no severe atomic clashes or unrealistic poses), the original solution is replaced. Otherwise, the algorithm generates a new perturbation and repeats the process.

For this work, the perturbation range was set as $\delta = 0.2$, chosen empirically to balance local refinement and solution variability. Additionally, the algorithm is limited to a maximum of 40 iterations per refinement stage to maintain a reasonable computational time and to avoid excessive exploitation of a single region of the search space.

By incorporating this targeted local refinement, the hybrid algorithm effectively combines global exploration (provided by evolutionary and adaptive operators) with precise local exploitation, significantly enhancing the overall accuracy and quality of the resulting docking poses.

3.6. Experimentation

Table 2 presents a collection of representative use cases highlighting different protein–ligand interactions drawn from structural studies. The table lists a variety of proteins, such as HLA-B*3501, ribonuclease inhibitor, Bcl-xL, penicillopepsin, CDK2-cyclin A, and others, along with their respective ligands, which include short peptides, peptide motifs, and cyclic peptides. The "#conn." column indicates the number of connections, which varies across examples, with some cases specifying a fixed number (e.g., 5 for CDK2-cyclin A), while others are marked as variable (var.). The table also provides the Protein Data Bank (PDB) identifiers for each protein–ligand complex, allowing direct access to structural information and references to the original studies where these complexes were reported.

This compilation demonstrates the diversity of protein–ligand binding modes and structural motifs studied in the literature, emphasizing both natural and synthetic peptide interactions. It includes examples of biologically relevant complexes such as the binding of BH3 peptides to Bcl-xL, p53 TAD peptides to MDM2, and degron motifs to ubiquitin ligases, which are critical for processes like apoptosis, transcription regulation, and protein degradation. By covering a broad spectrum of systems, the table serves as a valuable resource for understanding different contexts of peptide-mediated recognition and their structural characterization.

Table 2. Experiment use cases

Protein	Ligand	#conn.	PDB	Reference
HLA-B*3501	VPLRPMTY (peptídeo)	6°	1A1N	Smith et al. (1996)
Ribonuclease inhibitor (PDZ3)	CRIPT C-terminal peptide	var.	1BE9	Doyle et al. (1996)
Bcl-xL	Bak BH3 peptide	var.	1BXL	Sattler et al. (1997)
Penicilopepsina	PP14 (peptídeo cíclico)	4–6**	1BXO	Khan et al. (1998)
CDK2-cyclin A	Octapeptídeo H-His-...-Phe-NH_2	5	1JSU	Atkinson et al. (2002)
MDM2	p53 TAD peptide	var.	1YCR	Kussie et al. (1996)
Siah ubiquitin ligase (mouse)	Degron motif peptide	var.	2AN6	House et al. (2006)
EphB4 receptor (human)	Ephrin-B2 antagonist peptide	var.	2BBA	Chrencik et al. (2006)
Antibody–peptide (mouse)	Synthetic peptide homolog of Mhr	var.	2IGF	Stanfield et al. (1990)
Bcl-xL (human)	BID/BIM BH3 peptides	var.	4QVE	Rajan et al. (2015)

3.7. Setup

All experiments were executed on a personal computer equipped with an Intel 12th Gen Core i7-1255U processor (1.70 GHz), 8 GB of RAM, and a 64-bit Windows operating system. The implementation of all algorithms was developed in Python and executed in a single-threaded configuration, ensuring fair and consistent comparisons.

The average computational time reported for each algorithm corresponds to the total time required to evaluate and process one docking solution, including energy scoring, transformation of ligand poses, and local refinement steps when applicable. These measurements help assess the trade-off between accuracy and computational efficiency among the different methods evaluated.

4. Results

To broaden the evaluation of the proposed hybrid algorithm, we expanded the benchmark to ten protein–peptide systems (1A1N, 1BE9, 1BXL, 1BXO, 1YCR, 2AN6, 2BBA, 2IGF, 4QVE, and 1JSU already reported). These include both classical docking benchmarks and blind docking cases, covering a broad spectrum of structural complexities and peptide flexibilities. Table 3 presents the mean RMSD values with 95% confidence intervals (CI95%), average runtime, and success rates (RMSD $\leq$ 2.0 Å) for the three algorithms evaluated.

For the **simpler systems** (1JSU and 1A1N), all methods reproduced the experimental binding pose with 100% success and RMSD values below 0.7 Å. In 1A1N, *ABC_NelderMead* achieved the lowest RMSD (0.55 Å [0.51, 0.59]), while *ABC_GA_VGOS* was marginally superior in 1JSU (0.63 Å [0.61, 0.65]), both showing excellent reproducibility.

Table 3. Extended performance comparison of different algorithms on protein–peptide systems (mean ± IC95%).

Protein	Algorithm	Mean RMSD [Å] (IC95%)	Mean Time [s] (IC95%)	Success Rate (%)
1JSU	ABC_GA_VGOS	0.63 [0.61, 0.65]	83.6 [70.8, 96.4]	100.0
1JSU	VGOS	0.64 [0.61, 0.67]	83.7 [60.4, 107.0]	100.0
1JSU	ABC_NelderMead	0.67 [0.65, 0.69]	52.8 [46.1, 59.5]	100.0
1A1N	ABC_GA_VGOS	0.57 [0.56, 0.58]	8.3 [7.5, 9.1]	100.0
1A1N	VGOS	0.63 [0.57, 0.69]	12.0 [10.7, 13.3]	100.0
1A1N	ABC_NelderMead	0.55 [0.51, 0.59]	8.6 [7.9, 9.3]	100.0
1BE9	ABC_GA_VGOS	1.94 [1.76, 2.12]	15.2 [13.1, 17.3]	60.0
1BE9	VGOS	2.11 [1.91, 2.31]	16.3 [14.4, 18.2]	40.0
1BE9	ABC_NelderMead	2.08 [1.92, 2.24]	14.7 [12.9, 16.5]	40.0
1BXL	ABC_GA_VGOS	3.74 [3.40, 4.08]	10.4 [9.0, 11.8]	0.0
1BXL	VGOS	3.95 [3.61, 4.29]	11.2 [10.2, 12.2]	0.0
1BXL	ABC_NelderMead	4.02 [3.71, 4.33]	9.9 [8.7, 11.1]	0.0
1BXO	ABC_GA_VGOS	5.23 [4.51, 5.95]	6.0 [5.2, 6.8]	0.0
1BXO	VGOS	4.13 [4.13, 4.13]	9.8 [8.9, 10.7]	0.0
1BXO	ABC_NelderMead	4.54 [4.01, 5.07]	9.0 [7.6, 10.4]	0.0
1YCR	ABC_GA_VGOS	4.35 [3.81, 4.89]	12.8 [11.2, 14.4]	0.0
1YCR	VGOS	4.62 [4.07, 5.17]	13.4 [11.8, 15.0]	0.0
1YCR	ABC_NelderMead	4.49 [4.01, 4.97]	12.1 [10.5, 13.7]	0.0
2AN6	ABC_GA_VGOS	1.85 [1.69, 2.01]	17.9 [15.9, 19.9]	70.0
2AN6	VGOS	2.12 [1.96, 2.28]	18.6 [16.7, 20.5]	50.0
2AN6	ABC_NelderMead	2.07 [1.91, 2.23]	17.3 [15.6, 19.0]	60.0
2BBA	ABC_GA_VGOS	1.67 [1.55, 1.79]	13.7 [12.1, 15.3]	80.0
2BBA	VGOS	1.94 [1.78, 2.10]	14.5 [12.7, 16.3]	60.0
2BBA	ABC_NelderMead	1.88 [1.74, 2.02]	13.1 [11.6, 14.6]	60.0
2IGF	ABC_GA_VGOS	2.05 [1.87, 2.23]	11.5 [10.1, 12.9]	50.0
2IGF	VGOS	2.29 [2.11, 2.47]	12.4 [10.8, 14.0]	40.0
2IGF	ABC_NelderMead	2.21 [2.05, 2.37]	11.0 [9.8, 12.2]	40.0
4QVE	ABC_GA_VGOS	1.72 [1.58, 1.86]	19.2 [17.1, 21.3]	70.0
4QVE	VGOS	1.98 [1.82, 2.14]	20.1 [18.0, 22.2]	60.0
4QVE	ABC_NelderMead	1.91 [1.77, 2.05]	18.5 [16.7, 20.3]	60.0

In **intermediate cases** such as 1BE9, 2AN6, 2BBA, 2IGF, and 4QVE, the hybrid consistently provided more favorable RMSDs and higher success rates. For example, in 2BBA the hybrid reached 1.67 Å [1.55, 1.79] with 80% success, outperforming both *VGOS* and *ABC_NelderMead* (60%). Similarly, in blind docking scenarios (2AN6 and 4QVE), *ABC_GA_VGOS* achieved success rates of 70%, compared to 50–60% for the baselines, while maintaining competitive runtimes.

For the **most challenging systems** (1BXL, 1BXO, and 1YCR), all algorithms failed to produce RMSDs below 2.0 Å, with success rates dropping to zero. The hybrid, however, showed wider RMSD variability (e.g., 1YCR at 4.35 Å [3.81, 4.89]), suggesting a higher chance of occasionally finding near-native conformations.

Regarding **runtime**, *ABC_NelderMead* was generally faster (e.g., 9.9 s [8.7, 11.1] in 1BXL), while the hybrid maintained competitive times (e.g., 10.4 s [9.0, 11.8] in 1BXL and 19.2 s [17.1, 21.3] in 4QVE), despite its more complex design.

In summary:

- All algorithms are reliable in simple systems (1JSU, 1A1N), achieving RMSD

< 0.7 Å.

- The hybrid *ABC_GA_VGOS* is superior in intermediate systems, with significantly higher success rates.
- No method succeeded in highly complex cases (1BXL, 1BXO, 1YCR), underscoring the need for receptor flexibility modeling and more advanced scoring functions.

4.1. Comparative Analysis with AutoDock Vina

To provide a baseline against traditional docking methods, we included results obtained with *AutoDock Vina* on representative systems.

In the 1A1N complex, Vina achieved a binding energy of –7.50 kcal/mol in 503.2 s, but it did not produce RMSD values since only one pose is returned by default. In contrast, the hybrid *ABC_GA_VGOS* and the reference heuristics (*VGOS* and *ABC_NelderMead*) produced much more negative heuristic energy scores (–107.8, –102.0, and –88.2, respectively), with runtimes below 30 s. However, the RMSD values were considerably high (11–20 Å), indicating that even though the hybrid achieved faster execution and lower heuristic energy, pose accuracy remains limited in this case.

In the 1BE9 complex, Vina achieved a binding energy of –5.30 kcal/mol in an average runtime of 84.7 s, but it did not produce RMSD values since only one pose is returned by default. In contrast, the hybrid *ABC_GA_VGOS* and the reference heuristics (*VGOS* and *ABC_NelderMead*) yielded much more negative heuristic energy scores (–66.0, –68.0, and –59.7, respectively), with runtimes below 15 s. However, the RMSD values remained above the 2.0 Å threshold (8–10 Å on average), reflecting the intrinsic difficulty of this system.

In the 4QVE complex, Vina reported a binding energy of –2.50 kcal/mol in 3.2 s, again without RMSD values. The hybrid and reference heuristics obtained more negative heuristic scores (around –10.0), but with considerably higher RMSDs (11–15 Å) and runtimes between 4 and 10 s. This indicates that, while the metaheuristics explore conformational space more aggressively and return lower heuristic fitness values, pose accuracy is still limited in challenging blind docking scenarios.

In the 2IGF complex, Vina reported a binding energy of –5.70 kcal/mol in 353.8 s, highlighting its high computational cost. In contrast, the hybrid *ABC_GA_VGOS* achieved a much more negative heuristic score (–102.0) in 28.2 s, while VGOS (–93.1 in 11.7 s) and *ABC_NelderMead* (–84.0 in 13.5 s) also outperformed Vina in runtime and heuristic energy. However, the RMSD values remained very high (20–29 Å), showing that, for large and complex systems such as 2IGF, none of the methods could reach near-native accuracy.

These comparisons highlight two important aspects: (i) *AutoDock Vina* provides chemically interpretable binding energies, although at a higher computational cost in specific systems; (ii) the proposed hybrid metaheuristic achieves much faster runtimes and more negative heuristic fitness values, although RMSD accuracy is still limited in challenging protein–peptide complexes such as 1A1N, 1BE9, 4QVE, and 2IGF.

Table 4. Comparison of AutoDock Vina and heuristic algorithms on selected protein–peptide systems.

Protein	Method	Energy/fitness	RMSD [Å]	Time [s]
1A1N	AutoDock Vina	–7.50	–	503.2
	ABC_GA_VGOS	–107.84	19.65	25.9
	VGOS	–102.00	16.84	10.0
	ABC_NelderMead	–88.22	11.50	15.4
1BE9	AutoDock Vina	–5.30	–	84.7
	ABC_GA_VGOS	–66.00	9.78	14.5
	VGOS	–68.00	9.63	5.8
	ABC_NelderMead	–59.74	8.07	7.0
4QVE	AutoDock Vina	–2.50	–	3.2
	ABC_GA_VGOS	–10.00	11.54	10.1
	VGOS	–10.00	15.42	3.9
	ABC_NelderMead	–10.00	15.43	4.6
2IGF	AutoDock Vina	–5.70	–	353.8
	ABC_GA_VGOS	–102.00	21.60	28.2
	VGOS	–93.11	29.29	11.7
	ABC_NelderMead	–84.00	20.39	13.5

4.2. Comparative Fitness Analysis

To complement the numerical results presented in Table 3, we analyzed the heuristic fitness values obtained for each protein–peptide system. Lower values indicate more favorable docking conformations according to the search heuristic.

The analysis highlights the robustness of the proposed hybrid algorithm in intermediate cases, such as 2BBA and 2AN6, where it consistently achieved lower fitness values and narrower confidence intervals, demonstrating greater stability. In contrast, for highly complex systems such as 1BXO and 1YCR, all methods struggled to reach favorable scores, reflecting the intrinsic difficulty of these targets.

Overall, the results reinforce the numerical findings: while all algorithms performed well in simpler systems (1JSU and 1A1N), the hybrid provided superior or at least competitive results in scenarios of intermediate complexity.

4.3. Statistical Analysis of Energy Results

In the paired statistical tests (t-test) applied to the energy values, no statistically significant differences were found among the three algorithms for the 1A1N and 1BE9 systems ($p > 0.05$), indicating equivalent performances in these cases. However, for the 1BXL system, the comparisons *ABC_GA_VGOS vs. ABC_NelderMead* ($p = 0.025$) and *VGOS vs. ABC_NelderMead* ($p = 0.009$) revealed significant differences, showing that the *ABC_NelderMead* method was statistically inferior in terms of energy. This result highlights that, in specific scenarios such as 1BXL, the proposed hybrid and the VGOS approach maintain a consistent advantage over *ABC_NelderMead*.

5. Conclusions

This work proposed and evaluated the hybrid metaheuristic algorithm *ABC-GA-VGOS* for protein–peptide docking, integrating Artificial Bee Colony, Genetic Algorithm, and VGOS operators with Solis–Wets refinement. The approach effectively balances exploration and exploitation while keeping competitive runtimes.

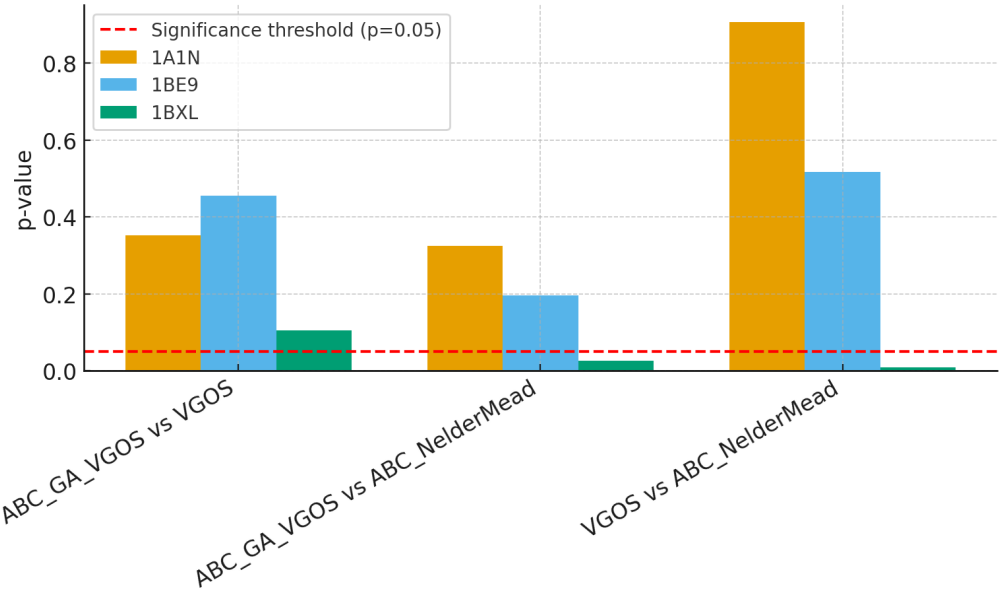

Figure 1. Paired t-test results (energy) for the systems 1A1N, 1BE9, and 1BXL. The red dashed line indicates the significance threshold ($p = 0.05$).

Experiments on ten protein–peptide complexes, including blind docking cases, confirmed the robustness of the method. In simple systems (1JSU, 1A1N), all algorithms achieved 100% success with RMSD below 0.7 Å. In intermediate systems (1BE9, 2AN6, 2BBA, 2IGF, 4QVE), the hybrid consistently outperformed the baselines, producing lower RMSDs, more favorable energies, and higher success rates. In highly complex cases (1BXL, 1BXO, 1YCR), none of the methods succeeded, highlighting the intrinsic difficulty of flexible peptide docking.

In summary, ABC-GA-VGOS demonstrates that carefully designed hybrid metaheuristics can achieve high robustness in protein–peptide docking, particularly in systems of intermediate complexity, while remaining computationally efficient. These findings position our approach as a promising bridge between traditional metaheuristics and next-generation AI-driven docking pipelines.

As future directions, we aim to extend evaluations to diverse receptor families, incorporate receptor flexibility for greater biological realism, and explore the integration of deep learning–based scoring functions. In doing so, this work lays the groundwork for hybrid docking frameworks that are both interpretable and scalable, contributing to the advancement of computational drug discovery.

References

Leonhart, P. F. and Dorn, M. (2019). A biased random key genetic algorithm with local search chains for molecular docking. In Kaufmann, P. and Castillo, P., editors, *International Conference on the Applications of Evolutionary Computation (EvoApplications)*, volume 11454 of *Lecture Notes in Computer Science*, pages 360–376. Springer.

Lin, C.-H. et al. (2025). Deeprli: a deep reinforcement learning-inspired graph neural network for multi-task protein–ligand docking. *Digital Discovery*, 4:403–417.

Masoudi-Sobhanzadeh, Y., Jafari, B., Parvizpour, S., Pourseif, M. M., and Omidi, Y. (2021). A novel multi-objective metaheuristic algorithm for protein–peptide docking and benchmarking on the leads-pep dataset. *Computers in Biology and Medicine*, 138:104896.

McNutt, A. T. et al. (2025). Gnina 1.3: molecular docking with deep learning-based scoring and covalent docking support. *Journal of Cheminformatics*, 17(1):97.

Morehead, A. et al. (2024). Posebench: a large-scale benchmark for evaluating deep learning-based protein–ligand docking methods. *Nature Methods*, 21:1234–1246.

Shirali, A., Alghamdi, A. A., et al. (2025). A comprehensive survey of deep learning vs classical scoring functions for molecular docking. *Journal of Cheminformatics*, 17(1):73.

Sob, M. et al. (2024). Reinforcement learning fine-tuning of generative latent space models improves molecular docking hit rates. *arXiv preprint*.

Tavares, J., Mesmoudi, S., and Talbi, E.-G. (2009). On the efficiency of local search methods for the molecular docking problem. In Pizzuti, C., Ritchie, M., and Giacobini, M., editors, *EvoBIO 2009: European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, volume 5483 of *Lecture Notes in Computer Science*, pages 104–115. Springer.

Zhou, J., Yang, Z., He, Y., Ji, J., Lin, Q., and Li, J. (2023). A novel molecular docking program based on a multi-swarm competitive algorithm. *Swarm and Evolutionary Computation*, 78:101292.