Examination of Annealing Schedules for RNA Design

Ryan McBride and Herbert H. Tsang
Applied Research Lab
Trinity Western University, Langley, British Columbia, Canada
Email: herbert.tsang@twu.ca

***Abstract*—Computational Intelligence is frequently applied to solve RNA design problems, to construct RNA sequences that fold into biochemically useful structures and alignments. RNA Design's NP-Hardness means that heuristic solutions, such as Evolutionary algorithms' Simulated Annealing, are commonly used to more effectively search for RNA sequences that fold into the target structure. Examples of Simulated Annealing in RNA Design include SIMARD, the ERD approach, and RNAPredict, all which aim to return RNA Sequences as close as possible to the target structure. However, such methods only use a single simulated annealing cooling schedule even though literature covers many schedules with varied convergence and performances guarantees. Since existing RNA Design cooling schedule surveys only cover at most four RNA design problems over two simulated annealing variants, we investigate the performance of four major simulated annealing schedules with ten variants on twenty-nine RNA design sequences. Relevant findings include a) the insensitivity of geometric schedule parameters, b) that logarithmic cooling schedules can solve RNA Design problems not solved by other schedules, c) suggestions for adjusting geometric schedule stopping conditions, and d) identifying common issues in popular adaptive and non-adaptive schedules for RNA Design.**

***Index Terms*—Bioinformatics, RNA Design, Simulated Annealing, Computational Intelligence**

I. Introduction

Bioinformatics relates to inferring properties of biological structures (e.g., RNA, DNA, and proteins) using Computational Intelligence [1]–[3], methods of combining human insight with the power of computation. Due to the complexity and importance of such biological processes, improved analysis is an active and ongoing frontier [4] [5]. Computational Intelligence is thus frequently applied in bioinformatics whether in aligning similar sequences of genes in species [6], designing databases and catalogues of existing RNA sequences [7], or inferring from sampled genetic subsequences what bacteria are present in a water supply [8].

Of particular interest is the RNA design problem: determining a sequence of RNA, a linear string of Adenine (A), Uracil (U), Guanine (G), and Cytosine (C), that folds into certain structures that guide biological function. Unlike DNA, RNA molecules are "single"-stranded and can therefore loop in on themselves to form complex shapes. The bonding of base pairs (e.g., A bonds to U, G bonds to C, and the less stable wobble pair of G bonded with U) results in diverse secondary structures. For example, the sequence in Fig. 1 folds and loops into itself to allow for unique functionality, such as sending "orders" or messages for specific enzymes to aid in digestion. Creating RNA sequences that can fold into a target structure of interest is therefore a significant bioinformatics problem [2].

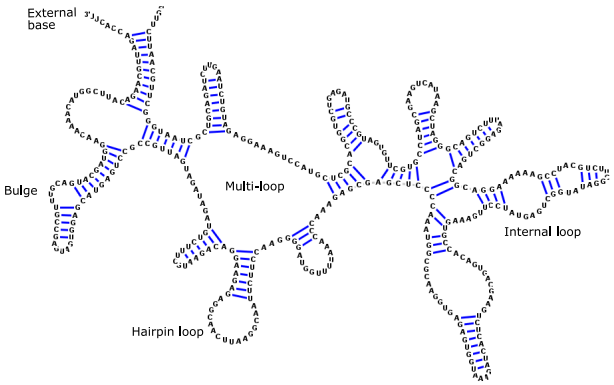

Fig. 1. The secondary structure of *Bacillus subtilis* (M13175) RNase P RNA. RNA Design problems involve finding RNA sequences that fold in certain target structures via Evolutionary Algorithms such as Simulated Annealing.

However, designing RNA sequences that fold into a specific structure is challenging. First, physically testing candidate RNA sequences is prohibitively expensive due to the cost of maintaining and using specialized equipment as well as the number of potential candidate RNA sequences. Furthermore, the problem is NP-Hard [2] due to the complexity of RNA folding in practice and how even a single change in one RNA molecule often results in a fundamentally different structure, even when using RNA folding models such as Vienna [9].

Due to this computational complexity, Computational Intelligence guided heuristics are thus used, particularly evolutionary algorithms such as our lab's **SIMARD** [2]: **Sim**ulated **A**nnealing to solve **R**NA **D**esign. Evolutionary algorithms are methods to iteratively improve solution as inspired by biology and survival of the fittest.

While most RNA Design research in evolutionary algorithms is on improving measures of *evolutionary fitness* or defining *useful mutations*, we instead propose surveying the affect of the chosen cooling schedule. The cooling schedule

The authors would like to acknowledge the research grants from the following agencies supported the work presented here: the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Trinity Western University. This research was enabled in part by support provided by WestGrid (www.westgrid.ca) and Compute Canada (www.computecanada.ca).

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

determines how the search's temperature, the tendency to explore the problem space rather than converge to a solution, is modified over algorithm runtime. This choice is essential to algorithm convergence and efficiency [10]. Commonly, Simulated Annealing for RNA Design uses one default schedule, a geometric schedule with default parameters [2] [7], while ignoring other available cooling schedule variants and settings [11]. Key questions we consider are the sensitivity of schedule parameters settings, if adaptive schedules that dynamically perturb temperature are effective [11], and how important a more "gradual" cooling schedule is to convergence and performance. This will further evaluate the effectiveness of RNA Design via Simulated Annealing.

II. Literature review

General RNA Design research in Simulated Annealing relates to using a single cooling schedule with set parameters, usually the geometric schedule, then tailoring evolutionary algorithm properties to RNA Design specifics. Some example approaches include using a database of biologically observed sequences to suggest more realistic candidate sequences [7] or choosing the best candidate of multiple neighbours to improve convergence [2]. However, all these approaches fail to examine cooling schedule variants and their relative effectiveness.

To our knowledge, the only RNA Design publication related to multiple cooling schedules was Erhan et. al's on applying three different cooling schedules for SIMARD on a handful of RNA sequences: the adaptive AARTs schedule and two variants of Geometric schedules [12]. Our new examination is valuable for many reasons. First, [12] was developed before many key improvements of SIMARD (e.g., using the ERD database to return more biologically consistent candidates, query pool selection, and the relative proportional temperature mutation); these improvements may change the suitability of particular schedules. Additionally, our new survey covers a larger scope of 29 RNA Design problems using four main scheduling variants and 10 total schedule parameterizations, a more robust analysis.

While there are other papers examine varied cooling schedules, their focus is on other problem domains such as the Travelling Salesmen Problem [11], Quadratic Assignment Problem (QAP) [11], or image smoothing [10]. Since evolutionary algorithms' performance varies for different domains [10] [13], we cannot conclude that these cooling schedules surveys would extend to RNA Design. Even echoing previous results is of interest to the wider computational intelligence community in the adaptability of cooling schedule algorithms in general.

III. Method

We will first discuss Simulated Annealing [14] solutions to RNA Design, particularly our lab's *SIMARD* [12] [15] [16] [17]. SIMARD and Simulated Annealing involves three major components:

1) A cost function that measures how "distant" a solution is to folding into the target structure;
2) The Neighbour or Mutator, a function that takes then perturbs the current solution;
3) A cooling schedule with parameters to update the temperature at iteration n, $T(n)$, after each chain of iterations. This temperature decides how likely the system is to update the current solution with the neighbor considering its cost.

One advantage of Simulated Annealing for RNA Design is that different biologists have different estimations of how an RNA sequence will fold, how close a candidate is to the intended structure, and estimates for how stable a configuration is based on the free or available energy. This plug-and-play property is important, especially due to the continued evolution of improved RNA folding models [9].

For more details, see the complete description and pseudocode Hampson et. al [2]. We will nonetheless summarize each aspect of SIMARD to give sufficient context for our cooling schedule experiment design in the next section.

A. Cost Setting

Given a candidate RNA sequence, the Vienna package is used [9] to predict how an RNA sequence would fold, which can then be compared to the target structure.

We represent RNA structures via dot-bracket notation [2], where the candidate sequence's "structure" is represented as a sequence of dots and brackets: a dot or "`.`", if the base is not connected, and brackets where an open bracket "`(`" indicates that the base connects to the next closing bracket in the sequence. For example the sequence `...(...)..` means that the 4th and 8th base are paired together while the rest are unpaired. More complex representations can be encoded as well. For example, a sequence such as `(....)` loops around to form a circle, a multi-loop.

Given that the candidate sequence folds into structure A represented in dot-bracket notation and a target structure B, in dot-bracket notation where A_i is the character at position i and B_i is the character at position i we can then compare the distance. We use the Hamming Distance metric where each position is compared for whether it correctly represents a dot or bracket:

$$Hamming\ Distance = \sum_{i=1}^{N-1} |A_i - B_i|$$

For example, the target structure $A =$`...(..)..` and the candidate structure $B =$`(.....).` differ in four locations (1st, 4th, 7th, and 8th), resulting in a hamming distance of four. As discussed, SIMARD's goal is to find a candidate sequence that folds as close to the intended structure as possible.

B. Mutation Setting

The neighbour operation represents how nearby RNA sequences are generated given the current solution. We used the ERD database method, a repository of common RNA sequences used to ensure that candidate solutions are "biologically consistent" [7] with observed sequences from nature.

Before the simulated annealing is run, an initialization step determines what candidate structures are involved in the target sequence. These structures are, for example, loops, bulges, or hairpin loops in Figure 1 of a certain length. The initial random solution is a random setting of each structure that is encoded as a RNA subsequence, a part of the whole RNA sequence.

The neighbour operator then uses a mutate operation that swaps candidate subsequences for other matching subsequences, such as swapping two sequences that are both expected to become a loop of the same length as in the target structure. In addition, we use the Dynamic Exploration Strategy (DES) [2] where multiple mutation steps are taken instead of one. Given a mutation factor, 0.33 by default, the number of mutations is based on the relative temperature, a constant that ranges from 0 to 1 that is highest at the start of the search:

$$Steps = \lceil mutation\_factor \cdot \#subsequences \cdot rel\_temp \rceil$$

For example, with a mutation factor of 0.33 at the initial temperature, a relative temperature of 1.00, results in the most swaps where 33% of these subsequences/structures are swapped out for alternatives from the ERD database. The main advantage of this approach is that many mutation steps can be taken at once, speeding up exploration [17].

C. Cooling Schedule

After every step of the algorithm, a decision is made whether to update the current solution with the new mutated neighbor. If the solution improves the current solution it is always accepted otherwise a uniform probability from 0 to 1 is drawn where a solution is replaced if this random probability exceeds $e^{\frac{-(cost_j - cost_i)}{T(n)}}$ where $T(n)$ is the temperature at the current "chain" of steps and $cost_i$ is the cost of the current solution and $cost_j$ is the cost of the candidate solution, the neighbor after the mutation step. The cooling schedule [14] determine how and when the temperature decreases, usually by updating after every "chain" (e.g., 1000 steps). The temperature therefore represents a tendency to explore more varied, poorer solutions to avoid converging to local optimums.

SIMARD and other Simulated Annealing variants (e.g., [7]) use the popular geometric schedule of:

$$T(n) = \beta \cdot T(n-1).$$

The temperature follows the geometric sequence in that $T(n)$ equals $\beta^n \cdot T(0)$: with a β of 0.8, temperature goes from $1.0 \cdot T(0)$ to $0.8 \cdot T(0)$ then $0.64 \cdot T(0)$. As motivated previously, studies of varying this β or leveraging more diverse cooling schedules have not been sufficiently examined for RNA Design. For example, there are logarithmic schedules [18] with stronger convergence guarantees and adaptive schedules [11] [19] that are less sensitive to exact parameter settings.

IV. Experimental Design and Data

Our goal is to compare and analyze multiple cooling schedules on SIMARD. We will discuss our choices of the RFAM data set, four cooling schedules, and our methodology.

A. Data

We consider the RFAM data set's 30 sequences [20], a popular data set for RNA design problems [2]. This data covers a mixture of RNA Design problems of varied lengths and difficulties. We, however, do not consider sequence 23 due to it requiring a pseudoknot when our Vienna folding model cannot support designing such a sequence [2], so we only consider the other 29.

B. Scheduling Variants

We considered four total schedule algorithms with a combined 10 parameterizations. One schedule is the adaptive AARTs methods while the remaining three are the non-adaptive geometric, linear, and logarithmic cooling schedules. The non-adaptive methods differ in how severely temperature is decreased after each chain of computation.

1) AARTs Adaptive Schedule: We first consider the adaptive scheduling algorithm of AARTs [11]. This method is adaptive because the cooling schedule parameters do not need to be tuned and adapted to specific problems, a valuable property because evolutionary algorithm parameterization is often more of an art than a science [11].

For example, setting the initial temperature usually involves considering factors such as range of the cost function between the best and worst possible solutions [11]. However, AARTs instead sets an initial temperature, $T(0)$, using the following equation:

$$T(0) = \overline{\Delta C^{(+)}}(\frac{m_2}{m_2\chi_0 - (1-\chi_0)m_1})^{-1}$$

by first generating a random candidate solution then running the neighbour operation a sufficient number of times based on the number of possible neighbours of this candidate; χ_0 is the initial acceptance ratio, $\overline{\Delta C^{(+)}}$ is the average distance or cost of neighbours that are worse than the initial random solution, m_1 is the number of lower distance neighbours and m_2 is the number of higher cost neighbors. This estimates the initial temperature that results in sufficient exploration given the cooling schedule update rule:

$$T(n) = T(n-1) \cdot (1 + \frac{\ln(1+\delta) \cdot T(n-1)}{3\sigma \cdot T(n-1)})^{-1} \quad (1)$$

δ is an adjustment factor, generally suggested to be 0.1 [14]. The σ is a variance term during the last chain of steps. A high variance thus results in a lower temperature drop.

While there are alternative adaptive schedule algorithms, many require far more tuning than AARTs; for example, setting a tolerance for when a temperature update is not beneficial that is essential to good convergence properties [19]. For this reason, we only consider AARTs though further adaptive schedules may be considered in a future work.

2) Geometric, Linear, and Logarithmic Schedules: We also choose three non-adaptive schedules where the temperature is decreased after every chain. First, we consider the ubiquitous *Geometric Schedule* of:

$$T(n) = \beta \cdot T(n-1);$$

where β ranges between 0.8 and 0.99, generally [18]. For SIMARD, a lower β of 0.7 was successful, possibly because of the mutation/neighbour policy that more quickly converges to high quality solutions; see [16] for more details. The geometric schedule is used in SIMARD [2] [17] [16] and other RNA related work, such as ERD's evolutionary RNA Database approach [7].

Another family of methods fix a starting temperature and ending temperature then fit a simple function to these constraints [21]. This ensures that the program ends after a certain number of iterations or are comparable to some other schedule's temperature curve. One approach is the *Linear schedule* [21], where a constant decrement d is used:

$$Linear(d) : T(n) = max(T(0) - n \cdot d, 0)$$

For example, setting d to $\frac{T(0)}{10}$ we ensure that the initial temperature is $T(0)$ and at $n = 10$ it equals 0.

While there are other "fitted" variants (e.g., fit a quadratic function, fit a cubic function, etc.), we focus on the linear function because of its simplicity and how it contrasts with the geometric schedules in that there is more time spent at higher temperatures while geometric sequences spend more time at lower temperatures. See Table I for a comparison of temperatures of similar schedules.

Our final choice is the *Logarithmic Schedule* [18] that ensures convergence of Simulated Annealing if:

$$T(n) \geq T(0) \cdot \frac{1}{log_2(1 + (n+1))}$$

However, this method often takes very long to converge [18]. We therefore represent a more general cooling schedule with a parameter alpha (α) to speed up the logarithmic schedule:

$$Log(\alpha) : T(n) = T(0) \cdot min(1, \frac{1}{log_2(1 + \alpha \cdot (n+1))})$$

Where $T(0)$ is always set to the initial temperature. A higher $\alpha \geq 1$ implies a faster asymptotic convergence to zero and therefore fewer iterations to reach a sufficiently low temperature to stop, though it does not fulfill the intended convergence property.

C. Methodology

We consider both the adaptive AARTs method and the three non-adaptive schedules, we consider three schedule categories: 1) low exploration settings, 2) medium exploration settings, and 3) high exploration settings. Recall that Simulated Annealing problems start in an exploration phase, high temperatures where worse solutions are more likely to replace the current solution to avoid local minimums, and exploitation, the lower temperatures where worse solutions are rarely considered [14]. Our goal is to consider different parameterizations where there is varying emphasis on the "exploration" temperatures.

For geometric, we consider βs of 0.7, 0.8, and 0.9. 0.7 therefore corresponds to low exploration, 0.8 to medium exploration, and 0.9 to high exploration. We then set the parameters of the other two schedules such that it equals the corresponding geometric schedule after ten iterations. For example, both Geo(0.7) and Linear(9.718) equal $0.03 \cdot T(0)$ at the 10th iteration. For the high exploration setting, we use Log($\alpha = 1$) instead of the true solution Log($\alpha = 0.527$) because both methods perform analogously in experiments and because the overall temperatures are similar, $0.35 \cdot T(0)$ versus $0.28 \cdot T(0)$ at the $n = 10$th iteration. These full results for this setting can be attained by contacting the paper's authors; the main difference is that Log($\alpha = 0.527$) takes longer to converge to its solutions.

Five runs for each schedule is used and we averaged the hamming distance and energy of the final solution. Based on prior experiment results, we use SIMARD with the QPS variant with a mutation factor of 0.33 with a pool size of 3, which achieved the lowest hamming distance in recent SIMARD experiments [2]. The initial temperature is set to 100 based on past SIMARD methods and the maximum runtime was set at 4 hours per each sequence, a time where all the geometric schedules converged successfully. All experiments were run on Compute Canada's WestGrid's Cedar.

The parameters of schedules, such as chain length, are set to the default settings in the ParSA software [22]. For example, the δ used for AARTs is set to 0.01 per the original AARTs schedule designers' recommendations.

V. Results and Discussion

We next apply this methodology to compare varied cooling schedules' temperatures and effectiveness.

A. Temperature Comparisons

A summary of the resulting temperatures for the non-adaptive schedules are listed in Table I for low exploration schedules, Table II for medium exploration, and Table III for high exploration. For example, the Geo(0.7) schedule only stays above 50% of the initial temperature, $T(0)$, for iterations $n = 0$ to $n = 2$ while Geo(0.9) stays above 0.5 for four iterations. Logarithmic tends to stay above 50% for at most one or two iterations then slowly decreases, it is more focused on a long exploitation period at low temperatures. Linear is more evenly spread between low and high temperatures.

Example AARTs schedule temperatures are shown in Table IV. These schedules tend to start at a middle temperature (e.g., 50.8 or a relative temperature of 0.508 compared to the initial temperature of the other schedules). These temperatures then decrease more slowly and stay in this high temperature range for a longer time than the other schedules.

TABLE I
TEMPERATURES FOR THE LOW EXPLORATION SCHEDULES.

	Relative Temperature at Iteration n						
Iteration	**n=0**	**n=1**	**n=2**	**n=3**	**n=5**	**n=7**	**n=10**
Geo(0.7)	1.00	0.70	0.49	0.34	0.17	0.08	0.03
Linear(9.718)	1.00	0.90	0.81	0.71	0.51	0.22	0.03
Log($4.3 \cdot 10^9$)	1.00	0.03	0.03	0.03	0.03	0.03	0.03

TABLE II
TEMPERATURES FOR THE MEDIUM EXPLORATION SCHEDULES.

	Relative Temperature at Iteration n						
Iteration	**n=0**	**n=1**	**n=2**	**n=3**	**n=5**	**n=7**	**n=10**
Geo(0.8)	1.0	0.80	0.64	0.51	0.33	0.21	0.11
Linear(8.92)	1.0	0.91	0.82	0.73	0.55	0.37	0.11
Log(57.7)	1.0	0.17	0.146	0.134	0.12	0.113	0.11

TABLE III
TEMPERATURES FOR THE HIGH EXPLORATION SCHEDULES.

	Relative Temperature at Iteration n						
Iteration	**n=0**	**n=1**	**n=2**	**n=3**	**n=5**	**n=7**	**n=10**
Geo(0.9)	1.00	0.90	0.81	0.73	0.59	0.48	0.35
Linear(6.51)	1.00	0.93	0.87	0.74	0.67	0.54	0.35
Log(1)	1.00	0.63	0.50	0.43	0.36	0.32	0.28

B. Hamming Distance Comparisons

Table V summarizes the hamming distances averaged over five runs for each schedule while Table VI ranks the schedules by the lowest distance. Table VIII, IX, X, and XI list the complete results over all twenty nine RFAM sequences for each method in each category of schedule.

Overall, the best performers are the geometric methods and the low exploration log schedule with $\alpha = 4.3 \cdot 10^9$ with an average hamming distance from 0.6 to 0.7. All these methods solve 21 to 22 of the 29 sequences. There is a statistically significant difference (p-value less than 5%) between these schedules and each of the lower quality schedules in comparing the average distances in the $5 \cdot 29$ runs, implying a meaningful improvement of RNA Design quality.

For general trends, the low exploration log option achieves the lowest average distance, 0.6, via solving an additional sequence not solved by any non-log approaches while solving other sequences equally well or slightly better. The fact that all geometric schedules performed similarly is a useful result that exact parameter setting is not essential. For the geometric methods, one trend is that none of the experiments timed out at our set four hour mark despite not achieving zero distance solutions. Perhaps the default convergence conditions should be tweaked to allow for more exploration time.

TABLE IV
AARTS SCHEDULE TEMPERATURES FOR EVERY 5TH SEQUENCE AND THE TOP-3 LONGEST SEQUENCES. NOTE THAT ITERATION 0 IS USED FOR CALIBRATION WITH A TEMPERATURE OF ZERO AND A NOT APPLICABLE (NA) ENTRY MEANS THAT THE SCHEDULED ALREADY CONVERGED TO A ZERO DISTANCE SOLUTION.

	Relative Temperature at Iteration n						
Name	**n=0**	**n=1**	**n=2**	**n=3**	**n=5**	**n=7**	**n=10**
RF00005	0.0	NA	NA	NA	NA	NA	NA
RF00010	0.0	0.659	0.579	0.519	0.428	0.366	0.297
RF00015	0.0	0.389	0.350	0.319	0.267	NA	NA
RF00020	0.0	0.139	0.125	0.113	0.093	0.077	0.061
RF00025	0.0	0.453	0.402	0.360	0.296	0.250	0.203
RF00030	0.0	0.690	0.616	0.555	0.462	0.394	0.320
RF00018	0.0	0.619	0.547	0.495	0.409	0.352	0.292
RF00011	0.0	0.869	0.763	0.681	0.559	0.479	0.394
RF00024	0.0	0.508	0.453	0.408	0.341	0.268	0.219

TABLE V
OVERALL RESULTS

Schedule	Avg. Distance	Avg Energy	# Solved
AARTs	1.53	-51.61	21/29
Geo(0.7)	0.68	-50.76	22/29
Linear(9.718)	1.32	-51.74	17/29
Log($4.3 \cdot 10^9$)	0.60	-51.11	23/29
Geo(0.8)	0.64	-50.73	21/29
Linear(8.92)	1.23	-51.07	18/29
Log(57.7)	1.07	-51.61	21/29
Geo(0.9)	0.66	-51.06	22/29
Linear(6.51)	1.57	-50.99	16/29
Log(1)	2.13	-52.41	19/29

TABLE VI
RANKING OF METHODS BY LOWEST AVERAGE HAMMING DISTANCE

Type	Schedule	Distance	Energy	# Solved
Low Exploration	Log($4.3 \cdot 10^9$)	0.60	-51.11	23/29
Med Exploration	Geo(0.8)	0.64	-50.73	21/29
High Exploration	Geo(0.9)	0.66	-51.06	22/29
Low Exploration	Geo(0.7)	0.68	-50.76	22/29
Med Exploration	Log(57.7)	1.07	-51.61	21/29
Med Exploration	Linear(8.92)	1.23	-51.07	18/29
Low Exploration	Linear(9.718)	1.32	-51.74	17/29
Adaptive	AARTs	1.53	-51.61	21/29
High Exploration	Linear(6.51)	1.57	-50.99	16/29
High Exploration	Log(1)	2.13	-52.41	19/29

The linear methods are three of the five worst performers because they prematurely end via freezing, achieving zero temperature, for eight to twelve of the 29 experiments. This suddenly ends the search without sufficient exploitation at low temperatures as in the other methods. Recall that linear formula differs from the other schedules in that it has a final iteration where the temperature is zero and no further state transitions are possible. This suggests that lower decrements would be preferable or more asymptotic cooling schedules should be used. Considering the difficulty and diversity of these problems, these fixed end schedules may be a poor choice and require too much calibration to be useful.

For the Logarithmic methods, they tended to timeout unless a very large alpha is used as in the low exploration setting. For example, the Log(1) method times out in 10 of 29 problems while Log(57.7) times out for 8 of 29. The Log(0.52) method an even higher exploration schedule which we did not include due to space limitations, timed out in 16 of 29. Nonetheless, the log methods were able solve sequence 3 when no other schedule could. Note that a zero distance implies that all five tests achieved the best possible solution. This shows that the log schedule is another useful tool in RNA design that we believe has not been considered in previous research.

For AARTs, we found that the temperature initialization and adaptive update schedule resulted in far more iterations to achieve similar results as the other methods: as an example, for RF00009.115.seq one AARTs run took 55386 steps while a Geo(0.7) run only took 11714 steps to achieve the same quality of solution. As shown in Table IV, AARTs' temperatures tends to start at a middle temperature then decrease more slowly and stays in this high range for a longer time. This plateauing ends up resulting in more timeouts with insufficient exploitation to achieve a low distance. We therefore do not recommend AARTs for RNA Design with SIMARD.

C. Energy Comparisons

Other than the distance objective, we also consider the lowest energy in Table VII. Compared to the distance results, this table is shuffled significantly where the method that timeouts the most actually achieves the best energy and worst distance. While non-intuitive, energy did not correlate well with distance in past RNA Design research [2]: a high energy may have low hamming distance or high hamming distance; for example, both AARTS and $Log(57.7)$ achieve an equivalent average energy, -51.61, but the average distance between them differs by roughly 0.5, implying a large difference in utility.

TABLE VII
RANKING OF METHODS BY LOWEST AVERAGE ENERGY.

Type	Schedule	Distance	Energy	# Solved
High Exploration	Log(1)	2.13	-52.41	19/29
Low Exploration	Linear(9.718)	1.32	-51.74	17/29
Adaptive	AARTS	1.53	-51.61	21/29
Med Exploration	Log(57.7)	1.07	-51.61	21/29
Low Exploration	Log($4.3 \cdot 10^9$)	0.60	-51.11	23/29
Med Exploration	Linear(8.92)	1.23	-51.07	18/29
High Exploration	Geo(0.9)	0.66	-51.06	22/29
High Exploration	Linear(6.51)	1.57	-50.99	16/29
Low Exploration	Geo(0.7)	0.68	-50.76	22/29
Med Exploration	Geo(0.8)	0.64	-50.73	21/29

In the RNA design field, any method requires models for both how a sequence would fold and an estimate of energy from this folding. These models affect reliability since it is approximating a complex biological system and can miss important factors, such as pseudoknots. In fact, the biologically preferred folded sequence often has energy within 20% of the lowest possible energy, which is unexpected under nature's basic principle of energy minimization [23]. An ongoing issue in RNA design is discovering a clearer understanding and modeling of distance and energy.

VI. CONCLUSION AND FUTURE RESEARCH

In this work, we highlight cooling schedules' impact on RNA Design. Example observations are the logarithmic schedule solving a new sequence, the need for adapting the convergence procedure for the geometric schedule, and the insensitivity of geometric schedule parameters. These results reconfirm the value and robustness of RNA Design via evolutionary Computational Intelligence. Future work may consider more schedule variants, more data sets, or use metaheuristics to suggest a preferred scheduling algorithm.

REFERENCES

[1] A. Deschênes and K. C. Wiese, "Using stacking-energies (inn and inn-hb) for improving the accuracy of rna secondary structure prediction with an evolutionary algorithm-a comparison to known structures," in *IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, 2004, pp. 598–606.

[2] D. J. D. Hampson and H. H. Tsang, "Incorporating dynamic exploration strategy for RNA design," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov 2018, pp. 1–7.

[3] X. Chen, S. He, D. Bu, F. Zhang, Z. Wang, R. Chen, and W. Gao, "FlexStem: improving predictions of RNA secondary structures with pseudoknots by reducing the search space," *Bioinformatics*, vol. 24, no. 18, pp. 1994–2001, 2008.

[4] G. Meister and T. Tuschl, "Mechanisms of gene silencing by double-stranded RNA ," *Nature*, vol. 431, pp. 343–349, 2004.

[5] B. Panning and R. Jaenisch, "Rna and the epigenetic regulation of X chromosome inactivation," *Cell*, vol. 93, no. 3, pp. 305–308, 1998.

[6] R. D. M. Page, "Comparative analysis of secondary structure of insect mitochondrial small subunit ribosomal RNA using maximum weighted matching," *Nucleic Acids Research*, vol. 28, no. 20, pp. 3839–3845, 2000.

[7] A. Esmaili-Taheri and M. Ganjtabesh, "Erd: a fast and reliable tool for rna design including constraints," *BMC bioinformatics*, vol. 16, no. 1, p. 20, 2015.

[8] M. A. Peabody, T. Van Rossum, R. Lo, and F. S. Brinkman, "Evaluation of shotgun metagenomics sequence classification methods using in silico and in vitro simulated communities," *BMC bioinformatics*, vol. 16, no. 1, p. 362, 2015.

[9] I. L. Hofacker, "Vienna RNA secondary structure server," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3429–3431, Jul 2003.

[10] W. Mahdi, S. A. Medjahed, and M. Ouali, "Performance analysis of simulated annealing cooling schedules in the context of dense image matching," *Computación y Sistemas*, vol. 21, no. 3, pp. 493–501, 2017.

[11] E. H. Aarts, F. M. de Bont, E. H. Habers, and P. J. van Laarhoven, "Parallel implementations of the statistical cooling algorithm," *INTEGRATION, the VLSI journal*, vol. 4, no. 3, pp. 209–238, 1986.

[12] H. E. Erhan, S. Sav, S. Kalashnikov, and H. H. Tsang, "Examining the Annealing Schedules for RNA Design Algorithm," in *IEEE Congress on Evolutionary Computation*, 2016.

[13] M. Locatelli, "Simulated annealing algorithms for continuous global optimization: convergence conditions," *Journal of Optimization Theory and applications*, vol. 104, no. 1, pp. 121–133, 2000.

[14] E. H. Aarts and P. J. Van Laarhoven, "Statistical cooling: A general approach to combinatorial optimization problems." *Philips J. Res.*, vol. 40, no. 4, pp. 193–226, 1985.

[15] S. Sav, D. J. D. Hampson, and H. H. Tsang, "SIMARD: A simulated annealing based RNA design algorithm with quality pre-selection strategies," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2016, pp. 1–8.

[16] D. J. D. Hampson, S. Sav, and H. H. Tsang, "Investigation of Multi-Objective Optimization criteria for RNA design," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2016, pp. 1–8.

[17] D. J. D. Hampson and H. H. Tsang, "Using matching substructures as an optimization objective for RNA design," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov 2017, pp. 1–7.

[18] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[19] M. Ortner, X. Descombes, and J. Zerubia, "An adaptive simulated annealing cooling schedule for object detection in images," INRIA, Research Report RR-6336, 2007. [Online]. Available: https://hal.inria.fr/inria-00181764

[20] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. R. Eddy, "Rfam: an RNA family database," *Nucleic Acids Research*, vol. 31, pp. 439—441, 2003.

[21] Y. Nourani and B. Andresen, "A comparison of simulated annealing cooling strategies," *Journal of Physics A: Mathematical and General*, vol. 31, no. 41, p. 8373, 1998.

[22] G. Kliewer, K. Klohs, and S. Oke, "Parallel simulated annealing library (parsa): User manual," 1999.

[23] A. Churkin and D. Barash, "Rna dot plots: an image representation for rna secondary structure analysis and manipulations," *Wiley Interdisciplinary Reviews: RNA*, vol. 4, no. 2, pp. 205–216, 2013.

TABLE VIII
AARTs RESULTS, A METHOD WITH ADAPTIVE SCHEDULING. TO MEANS A TIMEOUT ENDED THE SEARCH.

Sequence	Length	Avg. Distance (AARTs)	Avg. Energy (AARTs)
RF00008.11.seq	54	0	-14.6
RF00029.107.seq	73	0	-23
RF00005.1.seq	74	0	-24.32
RF00027.7.seq	79	0	-51.54
RF00019.115.seq	83	0	-27.3
RF00014.2.seq	87	0	-34.36
RF00006.1.seq	89	0	-18.28
RF00026.1.seq	102	0	-4.2
RF00001.121.seq	117	0	-38.36
RF00021.10.seq	118	0	-49.32
RF00020.107.seq	119	2 (TO)	-36.88
RF00016.15.seq	129	2 (TO)	-17.7
RF00015.101.seq	140	0	-34.82
RF00022.1.seq	148	0	-41.96
RF00002.2.seq	151	0	-23.24
RF00007.20.seq	154	0	-51.56
RF00003.94.seq	161	1.6 (TO)	-48.22
RF00013.139.seq	185	0	-61.98
RF00004.126.seq	193	0	-55.48
RF00025.12.seq	210	0	-41.68
RF00012.15.seq	215	0	-51.18
RF00017.90.seq	301	0	-121.4
RF00030.30.seq	340	0	-72.96
RF00028.1.seq	344	0.4 (TO)	-63.31
RF00009.115.seq	348	0	-58.74
RF00010.253.seq	357	9.2 (TO)	-117.92
RF00018.2.seq	360	7.6 (TO)	-66.22
RF00011.18.seq	382	16 (TO)	-120.02
RF00024.16.seq	451	5.6 (TO)	-126.06
Averages	191.86	1.53	-51.61

TABLE IX
FULL RESULTS, LOW EXPLORATION SCHEDULES. FOR SEQUENCES NOT SOLVED WITH ZERO DISTANCE, A (C) MEANS THE SIMULATION CONVERGED VIA A STOPPING CONDITION, (F) MEANS THE TEMPERATURE REACHED ZERO (I.E., BECAME FROZEN), AND (TO) MEANS A TIMEOUT.

Sequence	Length	Geo(0.7) Dist.	Geo(0.7) Energy	Linear Distance	Linear Energy	Log Distance	Log Energy
RF00008.11.seq	54	0	-14.24	0	-17.76	0	-17.02
RF00029.107.seq	73	0	-18.62	0	-20.26	0	-20.44
RF00005.1.seq	74	0	-23.78	0	-21.26	0	-21.56
RF00027.7.seq	79	0	-46.02	0	-52.2	0	-47.54
RF00019.115.seq	83	0	-20	0	-25.1	0	-25.94
RF00014.2.seq	87	0	-37.52	0	-36.7	0	-38.68
RF00006.1.seq	89	0	-22.5	0	-19	0	-20.32
RF00026.1.seq	102	0	-3.7	0	-4.24	0	-5
RF00001.121.seq	117	0	-31.02	0	-30.36	0	-30.82
RF00021.10.seq	118	0	-48.88	0	-46.68	0	-48.74
RF00020.107.seq	119	2 (C)	-35.38	2 (F)	-34.1	2 (TO)	-33.3
RF00016.15.seq	129	2 (C)	-23.84	2 (F)	-21.12	2 (TO)	-21.26
RF00015.101.seq	140	0	-31.74	0	-32.96	0	-34.66
RF00022.1.seq	148	0	-41.12	0	-42.4	0	-39.5
RF00002.2.seq	151	0	-22.48	1.2 (F)	-23.3	0	-23.86
RF00007.20.seq	154	0	-49.92	0	-53.3	0	-51.58
RF00003.94.seq	161	1.2 (C)	-47	2 (F)	-49.06	0	-43.06
RF00013.139.seq	185	0	-58.8	0	-59.86	0	-57.82
RF00004.126.seq	193	0	-58.18	0	-53.82	0	-54.96
RF00025.12.seq	210	0	-42.16	0.4 (F)	-40.34	0	-44.02
RF00012.15.seq	215	0	-53.68	0	-55.98	0	-45.24
RF00017.90.seq	301	0	-126.2	0	-124.26	0	-127.76
RF00030.30.seq	340	0	-70.18	1.6 (F)	-76.06	0	-74.24
RF00028.1.seq	344	0	-63.99	2.4 (F)	-61.45	0	-61.15
RF00009.115.seq	348	0	-58.68	0.4 (F)	-57.3	0	-61.24
RF00010.253.seq	357	3.8 (C)	-122.52	7.8 (F)	-125.78	3.8 (TO)	-122.84
RF00018.2.seq	360	4 (C)	-66.74	7.6 (F)	-74.56	4 (TO)	-62.42
RF00011.18.seq	382	4.8 (C)	-110.34	7.6 (F)	-116.54	3.6 (TO)	-114.94
RF00024.16.seq	451	2 (C)	-122.82	3.2 (F)	-124.7	2 (TO)	-132.4
Averages	191.86	0.68	-50.76	1.32 (F)	-51.74	0.6	-51.11

TABLE X

FULL RESULTS, MEDIUM EXPLORATION SCHEDULES. FOR SEQUENCES NOT SOLVED WITH ZERO DISTANCE, A (C) MEANS THE SIMULATION CONVERGED VIA A STOPPING CONDITION, (F) MEANS THE TEMPERATURE REACHED ZERO (I.E., BECAME FROZEN), AND (TO) MEANS A TIMEOUT.

Sequence	Length	Geo(0.8) Dist	Geo(0.8) Energy	Linear Distance	Linear Energy	Log Distance	Log Energy
RF00008.11.seq	54	0	-15.24	0	-13.9	0	-13.42
RF00029.107.seq	73	0	-16.92	0	-20.5	0	-20.88
RF00005.1.seq	74	0	-21.34	0	-27.24	0	-26.6
RF00027.7.seq	79.8	0	-46.2	0	-46.22	0	-52.54
RF00019.115.seq	83.8	0	-28.24	0	-21.28	0	-22
RF00014.2.seq	87.4	0	-34.06	0	-37.5	0	-38.36
RF00006.1.seq	91.6	0	-17.6	0	-20.54	0	-22.56
RF00026.1.seq	105	0	-7.48	0	-4.12	0	-3.84
RF00001.121.seq	117.2	0	-34.52	0	-34.76	0	-35.1
RF00021.10.seq	118.2	0.4 (C)	-45.38	0	-47.16	0	-49.16
RF00020.107.seq	121	2 (C)	-32.62	2 (F)	-35.58	2 (TO)	-39.5
RF00016.15.seq	131.2	1.6 (C)	-21.88	2 (F)	-17.22	2 (TO)	-18.5
RF00015.101.seq	141.6	0	-35.22	0	-33.52	0	-30.72
RF00022.1.seq	148.6	0	-38.14	0	-37.58	0	-39.98
RF00002.2.seq	151.6	0	-26.3	0.8 (F)	-22.64	0	-23.6
RF00007.20.seq	155.4	0	-53.28	0	-55.14	0	-47.76
RF00003.94.seq	165.8	0.8 (C)	-50.38	2 (F)	-47.4	0	-43.08
RF00013.139.seq	186.6	0	-59.58	0	-59.88	0	-68.42
RF00004.126.seq	196.4	0	-55.6	0	-55.62	0	-49.26
RF00025.12.seq	211	0	-46.36	0	-38.44	0	-44.68
RF00012.15.seq	232.2	0	-59.3	0	-51.06	0	-48.78
RF00017.90.seq	308.8	0	-123.06	0	-122.62	0	-121.92
RF00030.30.seq	340.8	0	-68.394	1.6 (F)	-71.54	1.6 (TO)	-83.58
RF00028.1.seq	344	0	-58.59	1.6 (F)	-61.71	1.6 (TO)	-61.21
RF00009.115.seq	348	0	-58.1	0.8 (F)	-62.3	0	-55.42
RF00010.253.seq	357	4.4 (C)	-121.28	6.6 (F)	-120.4	7.4 (TO)	-124.64
RF00018.2.seq	360	3.2 (C)	-63.64	7.6 (F)	-70.72	4.8 (TO)	-67.24
RF00011.18.seq	382	4.2 (C)	-113.36	7.8 (F)	-115.3	9.2 (TO)	-114.58
RF00024.16.seq	451	2 (C)	-119.02	2.8 (F)	-129.02	2.4 (TO)	-129.38
Averages	191.86	0.64	-50.73	1.23	-51.07	1.07	-51.61

TABLE XI

FULL RESULTS, HIGH EXPLORATION SCHEDULES. FOR SEQUENCES NOT SOLVED WITH ZERO DISTANCE, A (C) MEANS THE SIMULATION CONVERGED VIA A STOPPING CONDITION, (F) MEANS THE TEMPERATURE REACHED ZERO (I.E., BECAME FROZEN), AND (TO) MEANS A TIMEOUT. WE DID NOT INCLUDE THE LOG(0.52) SCHEDULE DUE TO SPACE, BUT IT IS SIMILAR TO THE LOG(1) SCHEDULE EXCEPT WITH MORE TIMEOUTS (16/29) AND WORSE, AS IN HIGHER, DISTANCES.

Sequence	Length	Geo(0.9) Dist	Geo(0.9) Energy	Linear Distance	Linear Energy	Log Distance	Log Energy
RF00008.11.seq	54	0	-15.18	0	-15.28	0	-13.32
RF00029.107.seq	73	0	-20.84	0	-19.7	0	-20.76
RF00005.1.seq	74	0	-23.42	0	-25.52	0	-23.84
RF00027.7.seq	79	0	-47.22	0	-54.36	0	-46.6
RF00019.115.seq	83	0	-23.62	0	-22.64	0	-25.06
RF00014.2.seq	87	0	-36.02	0	-37.42	0	-35.52
RF00006.1.seq	89	0	-21.02	0	-20.16	0	-19.06
RF00026.1.seq	102	0	-5.04	0	-3.9	0	-4.76
RF00001.121.seq	117	0	-33.74	0.8 (F)	-34.96	0	-38.92
RF00021.10.seq	118	0	-45.24	0	-47.22	0	-51.78
RF00020.107.seq	119	2 (C)	-31.2	2 (F)	-37.08	2 (TO)	-34.56
RF00016.15.seq	129	2 (C)	-18.66	2 (F)	-20.58	2 (TO)	-21.66
RF00015.101.seq	140	0	-30.98	0	-31.64	0	-33.66
RF00022.1.seq	148	0	-39.5	0	-35.6	0	-46.52
RF00002.2.seq	151	0	-21.3	2 (F)	-22.76	0	-22.78
RF00007.20.seq	154	0	-52.04	0	-52.88	0	-48.24
RF00003.94.seq	161	1.6 (C)	-46.04	2 (F)	-45.14	0.8 (TO)	-40.56
RF00013.139.seq	185	0	-60.98	0	-59.56	0	-64.18
RF00004.126.seq	193	0	-55.66	0	-54.8	0	-53.78
RF00025.12.seq	210	0	-41.38	0.4 (F)	-44.24	0	-44.24
RF00012.15.seq	215	0	-54.82	0	-49.5	0	-48.02
RF00017.90.seq	301	0	-126.14	0	-123.8	0	-118.14
RF00030.30.seq	340	0	-80.16	1.2 (F)	-69.16	5.6 (TO)	-79.06
RF00028.1.seq	344	0	-59.63	2.4 (F)	-62.71	3.2 (TO)	-64.45
RF00009.115.seq	348	0	-56.92	2 (F)	-62.66	2.4 (TO)	-65.284
RF00010.253.seq	357	4 (C)	-127.74	9 (F)	-123.86	12 (TO)	-130.24
RF00018.2.seq	360	4 (C)	-66.7	10 (F)	-65.2	13.2 (TO)	-71.3
RF00011.18.seq	382	3.6 (C)	-112.06	8.8 (F)	-110.14	14.2 (TO)	-123.94
RF00024.16.seq	451	2 (C)	-127.58	2.8 (F)	-126.16	6.4 (TO)	-129.52
Averages	191.86	0.66	-51.06	1.57	-50.99	2.13	-52.41