Rado Numbers and SAT Solvers

Lamina Zaman
University of Windsor

Curtis Bright
University of Waterloo

I. Introduction

In this report, we present a SAT-based approach to prove some previously unknown values for 3-colour and 3-variable Rado numbers. Additionally, we establish new bounds for certain Rado numbers as well as verify some known bounds previously reported in the literature, specifically those presented in [1], using SAT solvers.

We begin by introducing Rado numbers and SAT solvers. We then describe our methodology for generating Rado numbers, followed by an explanation of how new bounds were discovered and existing bounds were verified. Finally, we provide a summary of our results.

II. Background

A. Rado Numbers

For a given linear equation $\mathcal{E}$, the k-colour Rado number $R_k(\mathcal{E})$ is the smallest number n such that for every k-colouring of [1, n], there exists a monochromatic solution to the equation. If there is a k-colouring of [1, n] with no monochromatic solution to $\mathcal{E}$, we say that $R_k(\mathcal{E})$ is infinite.

B. Satisfiability (SAT) Problems

Before explaining satisfiability (SAT) problems, it is helpful to define the following:

- **Variable:** A Boolean variable (say x) can either be true or false.
- **Literal:** A literal is either a Boolean variable (x) or its negation $\bar{x}$.
- **Clause:** A disjunction of literals.
- **Conjunctive Normal Form (CNF):** A formula expressed as a conjunction of clauses.

A SAT problem asks whether there exists an assignment of true and false values to all variables such that the entire formula evaluates to true. If such an assignment exists, the formula is satisfiable, otherwise, the formula is unsatisfiable.

Many combinatorial problems, including the computation of Rado numbers, can be reduced to a SAT problem. A computer program, known as a SAT solver, can then determine whether or not the formula is satisfiable in an efficient manner.

III. Generating Rado Numbers

We have computed $R_3(ax + ay = bz)$ for $1 \leq b \leq 15$ and $1 \leq a \leq 30$ and given the results in Table 1, which can be found in the appendix. Altogether, the total CPU time on unsatisfiable instances was 21,790 seconds, and the total CPU time on satisfiable instances was 13,516 seconds, amounting to 9.81 total hours. A maximum of 4.3 GiB of memory was used across all instances.

Chang et al. [1] provided the values of $R_3(ax + ay = bz)$ for $1 \leq a, b \leq 10$ and also for $3 \leq a \leq 6$ and $11 \leq b \leq 20$. We have extended them for $1 \leq a \leq 15$ and $1 \leq b \leq 25$ and reported these numbers in Table 2, which can also be found in the appendix. Altogether, the CPU time amounted to 294.05 hours, and a maximum of 26.3 GiB of memory was used across all instances.

A. Technologies Used

For this section of the project, we primarily used Python combined with PySAT [2] for incremental SAT solving and Kissat [3] or CaDiCaL [4] for non-incremental solving.

Initially, we used CaDiCaL integrated with PySAT due to its incremental solving capabilities which enable the solver to reuse learned information across instances. Our approach starts with a counter $n = 1$. For each n, we add the corresponding clauses to the solver. If the instance is satisfiable, we increment n and continue incrementally. This process repeats until we encounter the first unsatisfiable instance, at which point the corresponding n is identified as the Rado number.

Once a Rado number was determined (or suspected to be known based on the patterns we observed), we transitioned to using a non-incremental solver and generated two instances: one for $n - 1$, the last satisfiable instance, and one for n, the first unsatisfiable instance. We generated clauses in DIMACS format, wrote them to two CNF files, and used Kissat to solve both instances. To be consistent in our timings, all instances were ultimately solved non-incrementally using Kissat. Our largest instance, the unsatisfiable instance showing $R_3(15x + 15y = 8z)) = 97875$, contained 293 625 variables and 255 884 401 clauses and took 58.8 hours to solve.

B. The SAT Encoding

Given positive integers n and k and a linear equation $\mathcal{E}$, we now construct a formula in conjunctive normal form that is satisfiable if and only if there exists a k-colouring of $[1, n]$ avoiding monochromatic solutions of $\mathcal{E}$. Therefore, if our formula is satisfiable, then $R_k(\mathcal{E}) > n$. If our formula is unsatisfiable, then $R_k(\mathcal{E}) \leq n$.

Variables: Variables are denoted by $v_{i,j}$ for $1 \leq i \leq k$ and $1 \leq j \leq n$, such that $v_{i,j}$ is true if and only if colour i is assigned to integer j. There are nk variables in the formula.

Similar to [5], our formula consists of the following clauses: positive, negative, and optional.

- **Positive Clauses:** Positive clauses ensure that every number is assigned at least one colour.

$$(v_{1,j} \vee v_{2,j} \vee \cdots \vee v_{k,j}), \quad \forall j \in [1, n]$$

- **Negative Clauses:** Negative clauses ensure that there are no monochromatic solutions to $\mathcal{E}$. For each colour $i \in [1, k]$ and for each solution $(x_1, x_2, \ldots, x_m)$ to $\mathcal{E}$, the clause appears as follows:

$$(\bar{v}_{i,x_1} \vee \bar{v}_{i,x_2} \vee \cdots \vee \bar{v}_{i,x_m})$$

- **Optional Clauses:** Optional clauses ensure that every number is assigned at most one colour.

$$\bigwedge_{1 \le i_1 < i_2 \le k} (\bar{v}_{i_1,j} \vee \bar{v}_{i_2,j}), \quad \forall j \in [1, n]$$

Symmetry Breaking: Since there is no distinction between colours 1 to k, we want the solver to avoid searching for different permutations of the same colouring. So, in addition to positive, negative, and optional clauses, we add symmetry breaking clauses to the solver, reducing the search space by a factor of $k!$ (or $3!$, in our case).

In order to do this, we ensure that colours are introduced in ascending order.

- The unit clause $v_{1,1}$ ensures that colour 1 appears in the first position.
- The clause $(\bigvee_{i=1}^{j-1} \bar{v}_{1,i}) \vee \bar{v}_{3,j}$ for $j = 2, \ldots, R_1(\mathcal{E})$ ensures that colour 1 appears before colour 2.

C. Generating Integer Solutions

Generating integer solutions to create negative clauses is often the most time-intensive part of our calculations, taking longer than the actual SAT solving process. So, we present a way to efficiently generate all integer solutions (x, y, z) within the interval $[1, n]$ for $ax + by = cz$.

We start by iterating i from 1 to n and constructing three linear Diophantine equations

$$\begin{aligned} ax &= cz - by \quad \text{where } x = i, \\ by &= cz - ax \quad \text{where } y = i, \\ cz &= ax + by \quad \text{where } z = i. \end{aligned}$$

The goal is to independently find solutions for each equation.

- Step 1 (Solving a linear Diophantine equation). To begin, we solve one of the above equations (say $ax + by = ci$) for unknowns (x, y) using the Extended Euclidean Algorithm, providing a particular integer solution (x_0, y_0). The general solution is

$$x = x_0 + \frac{bk}{\gcd(a, b)} \quad \text{and} \quad y = y_0 - \frac{ak}{\gcd(a, b)},$$

  where k is a parameter that runs over all integers to generate all solutions.
- Step 2 (Restricting to the interval $[1, n]$). Since we are only interested in solutions within $[1, n]$, we impose the constraints

$$1 \le x_0 + \frac{bk}{\gcd(a, b)} \le n \quad \text{and} \quad 1 \le y_0 - \frac{ak}{\gcd(a, b)} \le n.$$

  These inequalities provide upper and lower bounds on k, ensuring that both x and y remain in the valid range.
- Step 3 (Iterating over valid values of k). Once the feasible range for k is determined, we iterate over all valid values of k and solve for x and y at each step, adding (x, y, i) to our list of solutions.
- Finally, we repeat these steps for the remaining two equations $ai = cz - by$ and $bi = cz - ax$.

IV. Some Bounds on Rado Numbers

A. Some New Bounds

For coprime integers a and b, the data from Table 1 reveals the following patterns.

b	Range of a	$R_3(ax + by = bz))$
2	$[7, 30]$	$a^3 + a^2 + 5a + 1$
3	$[4, 30]$	$a^3 + a^2 + 7a + 1$
4	$[7, 30]$	$a^3 + a^2 + 9a + 1$
5	$[7, 30]$	$a^3 + a^2 + 11a + 1$
6	$[11, 30]$	$a^3 + a^2 + 13a + 1$
7	$[11, 30]$	$a^3 + a^2 + 15a + 1$
8	$[13, 30]$	$a^3 + a^2 + 17a + 1$
9	$[14, 30]$	$a^3 + a^2 + 19a + 1$
10	$[17, 30]$	$a^3 + a^2 + 21a + 1$
11	$[17, 30]$	$a^3 + a^2 + 23a + 1$
12	$[19, 30]$	$a^3 + a^2 + 25a + 1$
13	$[20, 30]$	$a^3 + a^2 + 27a + 1$
14	$[23, 30]$	$a^3 + a^2 + 29a + 1$
15	$[26, 30]$	$a^3 + a^2 + 31a + 1$

The formulas presented above were derived using curve fitting methods. Our analysis began by identifying patterns within each column of data. Some numbers exhibited a constant difference, suggesting a linear relationship, while others showed a constant third difference, indicating a cubic relationship. We constructed systems of linear and cubic equations based on these observations. We then applied Gaussian–Jordan elimination to solve these systems, obtaining the best-fitting formulas describing the sequence and ultimately arriving at the following conjecture.

Conjecture 1. *For coprime positive integers a and b with $a^2 + a + b > b^2 + ba$ and $a > b \ge 3$, $R_3(ax + by = bz) = a^3 + a^2 + (2b + 1)a + 1$.*

Using the same method as stated above, we derived the following conjecture and observation for Table 2.

Conjecture 2. *For coprime positive integers a and b, if a is odd, then for $3 \leq a < b \leq 2a-1$, $R_3(ax+ay=bz)) = a^3b$.*

Observation 1. *For coprime positive integers a and b, with $a < b \leq 2a-1$*

$$R_3(ax+ay=bz) = \begin{cases} a^3b/2 & \text{if } a \in \{6,10,14\}, \\ a^3b/4 & \text{if } a \in \{12\}. \end{cases}$$

B. Some Existing Bounds

For this section of the project, we verify some theorems presented in Chang, De Loera, and Wesley's work [1] using the methods outlined in their paper.

In addition to the previous technologies mentioned, we used the Computer Algebra System (CAS) known as SymPy [6] for symbolic computations and the Satisfiability Modulo Theories (SMT) solver known as Z3 [7] for boundedness checks.

C. Methodology

To prove these theorems using a SAT solver, we must modify our original encoding. Instead of assigning colours to the integers in a fixed range $[1, n]$, we assign colours to a set of symbolic polynomials (S) with parameters a, b, and c (the coefficients of $\mathcal{E}$).

Consider the following example: $R_2(x+(a-2)y=z) = a^2-a-1, \quad \forall a \geq 3$.

- Let $S = \{1, a-1, a, a^2-2a+1, a^2-a-1\}$.
- Suppose 1 is red.
- $(1, 1, a-1)$ is a solution, so $a-1$ is blue.
- $(a-1, a-1, a^2-2a+1)$ is a solution, so a^2-2a+1 is red.
- $(a^2-2a+1, 1, a^2-a-1)$ is a solution, so a^2-a-1 is blue.
- $(1, a, a^2-2a+1)$ and $(a-1, a, a^2-a-1)$ are solutions, so a cannot be either color.

D. Finding Polynomials

We start by initializing a seed set of polynomials (S_0) and an auxiliary gap set (G_0). The function `FindPolynomials` iterates through these sets, using the polynomials in G to generate additional polynomials that are added to S, which is the final output.

As noted in [1], `FindPolynomials` is heuristic in nature and does not guarantee that S leads to an unsatisfiable instance. Consequently, the seed set, gap set, and the number of iterations (`max`) of `FindPolynomials` may require multiple adjustments.

`FindPolynomials` also incorporates a boundedness check implemented by another function, `BoundedIntegerPolynomial`. Given a polynomial with integer coefficients, the function returns true if and only if the polynomial is bounded between the variables `lower` = 1 and `upper` = $R_k(\mathcal{E})$.

E. Results

Here, we present our results for each of the following theorems (from [1]).

Theorem 1. *For all $m \geq 3$, $R_3(x-y=(m-2)z) = m^3 - m^2 - m - 1$.*

Result	`UNSAT`
`max`	1
$\lvert S\rvert$ (our work)	189
$\lvert S\rvert$ (prior work [1])	685
Time to Construct S	10.00 s
Clause Generation Time	4.72 s
Solving Time	0.01 s
Number of Clauses	6473

Theorem 2. *For all $a \geq 3$, $R_3(ax-ay=(a-1)z) = a^3+(a-1)^2$.*

Result	`UNSAT`
`max`	1
$\lvert S\rvert$ (our work)	155
$\lvert S\rvert$ (prior work [1])	1365
Time to Construct S	15.20 s
Clause Generation Time	9.59 s
Solving Time	0.00 s
Number of Clauses	5189

Theorem 3. *For coprime a and b with $b \geq 1$, $a \geq b+2$, $R_3(ax-ay=bz) = a^3$.*

Result	`SAT`
`max`	1
$\lvert S\rvert$ (our work)	123
$\lvert S\rvert$ (prior work [1])	40645
Time to Construct S	3.96 s
Clause Generation Time	8.20 s
Solving Time	0.00 s
Number of Clauses	1365

The satisfiable result indicates that we need to modify one of S_0, G_0, or `max`.

V. Conclusion

In this work, we explored the computation of 3-colour Rado numbers for certain 3-term linear equations. Our SAT-based approach enabled us to extend the known values of $R_3(ax+by=bz)$ and $R_3(ax+ay=bz)$, computing previously unknown Rado numbers and formulating new conjectures based on observed patterns. Additionally, we verified some theorems presented in Chang et al.'s work [1] using the methods outlined in their paper.

References

[1] Y. Chang, A. De, and W. J. Wesley, "Rado Numbers and SAT Computations," arXiv (Cornell University), Jul. 2022, doi: https://doi.org/10.1145/3476446.3535494.

[2] A. Ignatiev, A. Morgado, and J. Marques-Silva, "PySAT: A Python Toolkit for Prototyping with SAT Oracles." Accessed: Dec. 17, 2025. [Online]. Available: https://alexeyignatiev.github.io/assets/pdf/imms-sat18-preprint.pdf

[3] M. J. H. Heule, M. Iser, Matti Järvisalo, and M. Suda, "Proceedings of SAT Competition 2024: Solver, Benchmark and Proof Checker Descriptions," University of Helsinki, 2024. https://researchportal.helsinki.fi/en/publications/proceedings-of-sat-competition-2024-solver-benchmark-and-proof-ch/ (accessed Dec. 17, 2025).

[4] Computer Aided Verification. Springer Science+Business Media, 2024. doi: https://doi.org/10.1007/978-3-031-65627-9.

[5] M. Heule, "Schur Number Five," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, Apr. 2018, doi: https://doi.org/10.1609/aaai.v32i1.12209.

[6] A. Meurer et al., "SymPy: symbolic computing in Python," PeerJ Computer Science, vol. 3, p. e103, Jan. 2017, doi: https://doi.org/10.7717/peerj-cs.103.

[7] C. R. Ramakrishnan and Jakob Rehof, Tools and Algorithms for the Construction and Analysis of Systems. 2008. doi: https://doi.org/10.1007/978-3-540-78800-3.

$a \backslash b$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	14	14	27	64	125	216	343	512	729	1000	1331	1728	2197	2744	3375
2	43	14	31	14	125	27	343	64	729	125	1331	216	2197	343	3375
3	94	61	14	73	125	14	343	512	27	1000	1331	64	2197	2744	125
4	173	43	109	14	141	31	343	14	729	125	1331	27	2197	343	3375
5	286	181	186	180	14	241	343	512	729	14	1331	1728	2197	2744	27
6	439	94	43	61	300	14	379	73	31	125	1331	14	2197	343	125
7	638	428	442	456	470	462	14	561	729	1000	1331	1728	2197	14	3375
8	889	173	633	43	665	109	644	14	793	141	1331	31	2197	343	3375
9	1198	856	94	892	910	61	896	896	14	1081	1331	73	2197	2744	125
10	1571	286	1171	181	43	186	1190	180	1206	14	1431	241	2197	343	31
11	2014	1508	1530	1552	1574	1596	1618	1584	1575	1580	14	1849	2197	2744	3375
12	2533	439	173	94	2005	43	2053	61	109	300	2024	14	2341	379	141
13	3134	2432	2458	2484	2510	2536	2562	2588	2574	2530	2541	2544	14	2913	3375
14	3823	638	3039	428	3095	442	43	456	3207	470	3113	462	3146	14	3571
15	4606	3676	286	3736	94	181	3826	3856	186	61	3795	180	3835	3836	14
16	5489	889	**4465**	173	**4529**	633	**4593**	43	**4657**	665	**4576**	109	**4602**	644	**4620**
17	6478	**5288**	**5322**	**5356**	**5390**	**5424**	**5458**	**5492**	**5526**	**5560**	**5594**	**5424**	**5447**	**5474**	**5475**
18	7579	1198	439	856	**6355**	94	**6427**	892	43	910	**6571**	61	**6409**	896	300
19	8798	**7316**	**7354**	**7392**	**7430**	**7468**	**7506**	**7544**	**7582**	**7620**	**7658**	**7696**	**7488**	**7518**	**7530**
20	10141	1571	**8541**	286	173	1171	**8701**	181	**8781**	43	**8861**	186	**8941**	1190	109
21	11614	**9808**	638	**9892**	**9934**	428	94	**10060**	442	**10144**	**10186**	456	**10270**	61	470
22	13223	2014	**11287**	1508	**11375**	1530	**11463**	1552	**11551**	1574	43	1596	**11727**	1618	**11535**
23	14974	**12812**	**12858**	**12904**	**12950**	**12996**	**13042**	**13088**	**13134**	**13180**	**13226**	**13272**	**13318**	**13364**	**13110**
24	16873	2533	889	439	**14665**	173	**14761**	94	633	2005	**14953**	43	**15049**	2053	665
25	18926	**16376**	**16426**	**16476**	286	**16576**	**16626**	**16676**	**16726**	181	**16826**	**16876**	**16926**	**16976**	186
26	21139	3134	**18435**	2432	**18539**	2458	**18643**	2484	**18747**	2510	**18851**	2536	43	2562	**19059**
27	23518	**20548**	1198	**20656**	**20710**	856	**20818**	**20872**	94	**20980**	**21034**	892	**21142**	**21196**	910
28	26069	3823	**22933**	638	**23045**	3039	173	428	**23269**	3095	**23381**	442	**23493**	43	**23605**
29	28798	**25376**	**25434**	**25492**	**25550**	**25608**	**25666**	**25724**	**25782**	**25840**	**25898**	**25956**	**26014**	**26072**	**26130**
30	31711	4606	1571	3676	439	286	**28351**	3736	1171	94	**28591**	181	**28711**	3826	43

Table 1
$R_3(\mathcal{E}(3, 0; a, b, b))$ for $1 \le b \le 15$ and $1 \le a \le 30$. The previously unknown values are presented in boldface, and the underlined entries correspond to equations whose coefficients are not coprime.

$b \backslash a$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	14	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	1	14	243	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
3	54	54	14	384	2000	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
4	∞	1	108	14	875	243	4459	∞	∞	∞	∞	∞	∞	∞	∞
5	∞	105	135	180	14	864	3430	3072	12393	∞	∞	∞	∞	∞	∞
6	∞	54	1	54	750	14	3087	384	243	2000	**27951**	∞	∞	∞	∞
7	∞	455	336	308	875	756	14	1536	8748	7500	**23958**	**10368**	**54925**	∞	∞
8	∞	∞	432	1	1000	108	2744	14	8019	875	**21296**	243	**48334**	4459	**97875**
9	∞	∞	54	585	1125	54	3087	1224	14	6000	**18634**	384	**41743**	**30184**	2000
10	∞	∞	1125	105	1	135	3430	180	7290	14	**17303**	864	**37349**	3430	243
11	∞	∞	2019	847	1958	1188	**3773**	**1672**	**8019**	**5500**	14	**6048**	**35152**	**24696**	**77625**
12	∞	∞	∞	54	2400	1	**4116**	54	108	750	**15972**	14	**32955**	3087	875
13	∞	∞	∞	1710	3445	1963	**4459**	**1456**	**9477**	**6500**	**17303**	**5616**	14	**21952**	**60750**
14	∞	∞	∞	455	3675	336	1	308	**10206**	875	**18634**	756	**30758**	14	**57375**
15	∞	∞	∞	5408	54	105	**6615**	**2760**	135	54	**19965**	180	**32955**	**20580**	14
16	∞	∞	∞	∞	5725	432	**7616**	1	**11664**	1000	**21296**	108	**35152**	2744	**54000**
17	∞	∞	∞	∞	8330	4743	**10064**	**3825**	**12393**	**8500**	**22627**	**7344**	**37349**	**23324**	**57375**
18	∞	∞	∞	∞	12069	54	**10962**	585	1	1125	**23958**	54	**39546**	3087	750
19	∞	∞	∞	∞	16397	6726	**14782**	**4332**	**16853**	**9500**	**25289**	**8208**	**41743**	**26068**	**60750**
20	∞	∞	∞	∞	∞	1125	**14700**	105	**19080**	1	**26620**	135	**43940**	3430	108
21	∞	∞	∞	∞	∞	455	54	**6699**	336	**12579**	**27951**	308	**46137**	54	875
22	∞	∞	∞	∞	∞	**2019**	**20580**	**847**	**25047**	**1958**	1	**1188**	**48334**	**3773**	**74250**
23	∞	∞	∞	∞	∞	**19056**	**27853**	**8556**	**32453**	**17250**	**35926**	**9936**	**50531**	**31556**	**77625**
24	∞	∞	∞	∞	∞	∞	**28956**	54	432	**2400**	**39600**	1	**52728**	**4116**	1000
25	∞	∞	∞	∞	∞	∞	**40163**	**10200**	**42975**	105	**47850**	**12850**	**54925**	**34300**	135

Table 2
$R_3(\mathcal{E}(3,0;a,a,b))$ for $1 \leq a \leq 15$ and $1 \leq b \leq 25$. The previously unknown values are presented in boldface, and the underlined entries correspond to equations whose coefficients are not coprime.