Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 4** : 1–9.
ISSN-0037-8712
doi:10.5269/bspm.79959

Collatz Rendezvous Model (CRM): A Novel Approach to Agent Synchronization

Vidyashree H. R.*, Lakshminarayana S., Ajith S.

ABSTRACT: The deterministic rendezvous problem is a fundamental challenge in multi-agent systems, requiring to meet at a common location using predefined movement rules. In this paper, we introduced the Collatz Rendezvous Model (CRM), a novel approach that leverages the well-known Collatz process to guide agents towards synchronization. Each agent follows an independent Collatz sequence, iterating through transformation rules until they converge at a common value, typically 1. We develop a computational framework to simulate CRM using Python and Tkinter, providing real-time tracking of agent movements. Through extensive experimentation, we analyze the number of steps required for rendezvous, identifying patterns in synchronization behaviour based on different initial positions. The study reveals that CRM exhibits deterministic yet unpredictable convergence patterns, making it a unique mathematical tool for modelling agent rendezvous. We also discuss potential applications in distributed computing, swarm intelligence and self-organizing systems.

Key Words: Collatz Rendezvous Model (CRM), agent synchronization, deterministic rendezvous, multi-agent systems, Collatz process, computational modelling.

Contents

1. Introduction

1.1. The Rendezvous Problem

Rendezvous problems in game theory involve two or more players attempting to meet in a structured environment using predefined strategies. These problems are crucial in various applications, including robotics, search-and-rescue operations and network synchronization. Traditional rendezvous strategies rely on predetermined movement patterns, probabilistic models or decision-theoretic frameworks to minimize the expected meeting time [7, 18]

Definition 1.1: The Rendezvous problem can be described as follows:
Given two or more agents moving within a specified environment (e.g., a graph, a plane or a network), find a strategy that ensures they meet at the same location in a finite amount of time, despite limited knowledge about their initial positions or movement capabilities. [18]

* Corresponding author.

Submitted November 07, 2025. Published February 03, 2026
2020 *Mathematics Subject Classification:* 11-04, 68R10, 91-10.

1.2. Variants of the Rendezvous Problem

There are several versions of the problem depending on different constraints:

1. Symmetric vs. Asymmetric Rendezvous:
    - Symmetric: Both agents follow the same environment rules and have identical initial conditions.
    - Asymmetric: Agents may have different speeds, knowledge or constraints.
2. Deterministic vs. Probabilistic Approaches:
    - Deterministic: Agents follow predefined algorithms ensuring rendezvous under all conditions.
    - Probabilistic: Agents use random movements or probabilistic strategies to increase the chances of meeting.
3. Synchronous vs. Asynchronous Rendezvous:
    - Synchronous: Agents move at the same time and follow synchronized steps.
    - Asynchronous: Agents may move at different times, making coordination harder.
4. Rendezvous in Different Environments:
    - Graph-based Rendezvous: Agents move along a network or graph structure.
    - Geometric Rendezvous: Agents move in continuous space (e.g., the Euclidean plane).
    - Rendezvous with obstacles: The environment contains barriers that complicate movement.

1.3. Challenges in the Rendezvous Problem

The rendezvous problem presents significant challenges, especially in unknown environments, where agents must meet despite limited initial information. One key difficulty is the lack of a common reference frame; agents may not know their initial separation or relative positions. This problem is further complicated when asynchronous movement is involved, meaning agents operate on independent time schedules, making synchronization harder. Obstacles in the environment can also hinder direct movement, requiring complex navigation strategies. Additionally, in dynamic environments, external factors like moving obstacles or communication delays can affect rendezvous efficiency [1-5]
Another major challenge is designing optimal movement strategies that minimize expected meeting time, especially under adversarial conditions. If agents follow deterministic paths, an adversary could predict and disrupt their meeting. The trade-off between exploration and efficiency is crucial, as searching too broadly increases time, while being too constrained may prevent rendezvous. In asymmetric scenarios, where one agent has more information or mobility than the other, designing fair and efficient protocols becomes difficult. Additionally, in practical applications like robotics and sensor networks, energy constraints impose further limitations. Addressing these challenges requires blending techniques from game theory, optimization and distributed computing to ensure reliable and efficient rendezvous [7, 9, 16, 18]

Definition 1.2: The Collatz conjecture: The Collatz conjecture, also known as $3x + 1$ problem, states that any positive integer will eventually reach 1 when iteratively transformed using the rule: if the number is even, divide it by 2; if odd, multiply by 3 and add 1 [6, 17, 19, 20].
In this paper, we introduce a novel approach based on the Collatz conjecture, a well-known problem in number theory. By using the Collatz sequence as a movement strategy, players iteratively determine their positions using the sequence's recursive transformation rules.

The movement strategy follows:

$$P(t+1) = \begin{cases} \dfrac{P(t)}{2}, & \text{if } P(t) \text{ is even} \\ 3P(t) + 1, & \text{if } P(t) \text{ is odd} \end{cases}$$

Despite its apparent simplicity, the Collatz sequence exhibits complex iterative behaviour and has been studied in various mathematical and computational contexts. In this work, we explore how the Collatz process can be applied as a deterministic rendezvous strategy for autonomous agents, leading to the development of **the Collatz Rendezvous Model (CRM).**

1.4. Collatz Rendezvous Model (CRM)

A Collatz Rendezvous Model is a novel framework that integrates principles from the rendezvous problem with the mathematical structure of the Collatz Conjecture. It explores how two or more agents can efficiently meet (rendezvous) using movement rules inspired by the Collatz sequence, a famous unsolved problem in number theory.

Features of CRM:

1. Rule-based Movements
   - Each agent follows a variation of the Collatz function to determine its next position.
   - Movements are dynamic and depend on whether the agent's position follows an 'even' or 'odd' state.
   - This introduces unpredictability yet maintains a structured approach to convergence.
2. Asynchronous and Symmetric Rendezvous
   - Agents can move asynchronously (without strict timing constraints) while still achieving rendezvous.
   - The approach remains fair by treating agents symmetrically, ensuring equal likelihood of meeting.
3. Efficiency in Search Strategy
   - The Collatz-inspired movement strategy avoids redundant searches, minimizing the expected time to rendezvous.
   - The non-linear nature of the sequence helps agents explore more locations efficiently.
4. Graph and Euclidean Space Applications
   - CRM can be applied to discrete settings (e.g., graphs, networks) as well as continuous environments (e.g., geometric rendezvous)
   - The structure adapts well to real-world applications like robotic coordination and network synchronization.

2. Related Work

The study [3] explores the rendezvous problem where two friends, separated in a building or shopping mall, must meet as quickly as possible without prior agreement. A simple strategy of randomly choosing a location at each step results in an expected meeting time if n steps. The rendezvous problem on a line where two players start at a random distance apart, with an unknown probability distribution but an expected initial distance bounded by a constant with a maximum velocity 1 is studied in [5].

The paper [1] examines the rendezvous problem where two players move in the plane, aiming to minimize their first meeting time through cooperation. Here three scenarios are analyzed such as unknown vector difference, known distance-unknown direction and asymmetric information. The study provides results for each case, exploring strategies to optimize meeting time under this varying uncertainty [1]. The rendezvous problem is studied on another scenario, randomised initial placement, where one player's starting position is chosen equiprobably from a finite set of points including the conditions, known distance-unknown direction and asymmetric information [2].

The markstart rendezvous search, a new framework where players can leave marks at their starting points, potentially providing crucial information for the rendezvous problems is introduced in [4].

Symmetric rendezvous on a line becomes more manageable in the markstart setting, while symmetric rendezvous on a line (with initial distance chosen via a convex distribution) remains easier to analyze in the original framework. The paper [4] also presents results for markstart rendezvous on complete graphs and for cases where the players initial distance follows an unknown probability distribution.

The deterministic rendezvous problem in an unknown anonymous connected graph, where two mobile agents with distinct identifiers must meet at some node, is addressed. The fast deterministic algorithm is proposed that efficiently guide the agents to a successful rendezvous, despite lacking prior knowledge of the graph structure [9].

In the geometric scenario, the algorithm ensures that agents with different compasses and units of length can have rendezvous almost everywhere. A deterministic protocol is proposed in [13], which ensures gathering in finite time, under the condition that robots share a common orientation. The study concludes that orientation is as powerful as instantaneous movements for achieving gathering.

The rendezvous problem for two mobile agents with distinct labels navigating an unknown, connected graph or an unknown connected terrain is investigated in the plane [8]. The paper [14] studies the gathering problem for mobile agents with distinct identifiers in an unknown, anonymous, connected network, proving that it is no harder than the rendezvous problem. The deterministic rendezvous problem is designed for arbitrary startups in rings and general networks, to measure efficiency through the steps taken from the last agent's startup until rendezvous. The results establish deterministic rendezvous as feasible even with arbitrary startup delays [14].

The gathering problem is addressed in [12], in which identical mobile robots in the plane, operating in Look-Compute-Move cycles without direct communication, must gather at a common point in finite time without external control. This work considers a weaker setting where robots have no memory of past cycles, no common coordinate system and no constraints on timing or execution order.

The rendezvous problem in undirected graphs, where two mobile agents with distinct labels starts at different nodes and must meet, is been examined in [16]. The study distinguishes between two scenarios: weak, where agents can meet at a node or on an edge and strong, where agents meet at a node and do not detect encounters on edges. A deterministic polynomial-time algorithm is presented that guarantees rendezvous with runtime depending on graph size, the smaller agent label and the largest edge traversal time [16].

The synchronous rendezvous problem is explored in graphs where two mobile robots (A and B) must meet at the same time and location while facing constraints. Three rendezvous models such as edge monotonic, node inclusive and blind rendezvous models with different movement restrictions, was introduced [10].

A new variant called the Deterministic treasure hunt problem is introduced, which extends the rendezvous problem to scenarios where one agent must locate a stationary target. This leads to the study of strongly universal traversal and exploration sequences, which are enhanced versions of existing universal sequences used for graph traversal [7].

The paper [11] examines the rendezvous problem for two identical, anonymous mobile robots, A and B, in an arbitrary undirected graph. Unlike most previous work, where robots have access to the same nodes and edges, here each robot can only access specific parts of the graph, though they receive full map in advance. The research explores three variants of the problem, differing in the degree of movement restrictions imposed on the robots, in both synchronous and asynchronous computational models. For each variant and model, the feasibility of rendezvous is analyzed and algorithms are proposed whenever possible.

The paper [18] provides a comprehensive survey of deterministic strategies for the Rendezvous problem, where two or more mobile agents (robots) must meet in an environment, such as a graph-based network or a geometric plane. The study primarily focuses on cases where agents cannot see each other before meeting and cannot leave marks to facilitate navigation. Key algorithmic approaches, complexity bounds and feasibility conditions are discussed, with an emphasis on deterministic methods that break symmetry to ensure the successful meeting [18].

The paper [15] investigates deterministic distributed rendezvous for two mobile agents navigating an infinite path (a graph where nodes are labelled with distinct positive integers). The rendezvous time is measured as the number of rounds from the earliest agent's start until the meeting. It is analyzed in three different scenarios: exact position knowledge, partial knowledge and minimal knowledge. The results also extended to finite paths and cycles.

3. Methodology

3.1. Problem Formulation

The Collatz Rendezvous Model is a deterministic rendezvous process where two agents, starting from different initial positions, follow independent Collatz sequences until they reach a common value. The objective is to analyze the trajectory of each agent and evaluate the number of steps required for synchronization.

3.2. Simulation Framework

To implement CRM, we develop a Python-based simulation with a graphical interface using Tkinter. The framework allows the user to:

- Input the initial values for two agents.
- Compute their trajectories using Collatz iterations.
- Visualize the movement of agents step-by-step.
- Track step count and convergence patterns.

3.3. Pseudocode for CRM

The pseudocode outlines a Collatz Rendezvous Model simulation where two agents follow the Collatz sequence rules to determine if they can converge to the value 1 simultaneously. The simulation consists of several key functions:

- Collatz_step(n): This function computes the next step in the Collatz sequence. If the number is even, it is divided by 2; otherwise, it is multiplied by 3 and incremented by 1.
- Simulate_rendezvous(agent1, agent2, max_steps=10000): This function simulates the movement of two agents starting from user provided initial values. It initializes step counting and stores the movement paths of both agents. The algorithm iterates through the sequence, updating each agent's value according to the Collatz step function. If both agents reach 1 at the same step, it announces success and visualizes their paths. If the maximum step count is reached without convergence, it reports failure and still visualizes the paths.
- Visualize_paths(path1, path2, steps): This function extracts and plots the movement trajectories of both agents. It uses matplotlib to create a graph with labels, legends and a grid to show their respective paths over time.
- Start_simulation(): This function handles user input for initial values of both agents and starts the simulation by calling simulate_rendezvous().
- Tkinter UI setup: A Graphical User Interface (GUI) is designed using Tkinter, which includes input fields for agent values, a button to start the simulation and a text area to display the stepwise progress. The event loop ensures smooth user interaction.

Pseudocode:
Input: Initial values for Agent 1 and Agent 2.
Output: Steps taken for rendezvous and visualization of movement.

- Define function collatz_step(n):
  If n is even, return $n/2$
  Else, return $3n + 1$

- Define function simulate_rendezvous(agent1, agent2, max_steps=10000):

  1. Initialize steps=0

  2. Store initial positions: path1←[agent1], path2←[agent2]

  3. Display starting positions: Agent 1=agent1, Agent 2=agent2

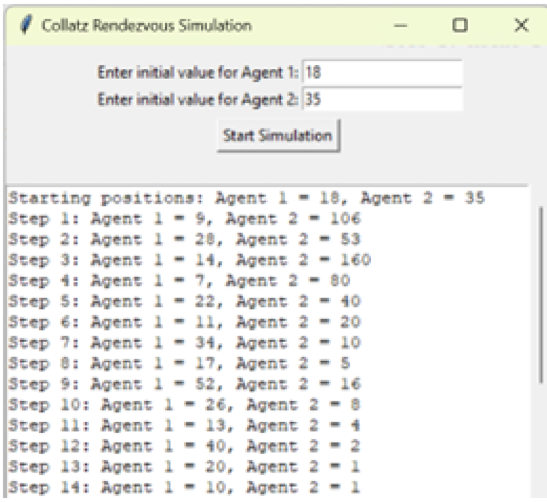
  4. While steps¡max_steps:
     If agent1 is not 1:
         -Update agent1 using collatz_step(agent1)
         -Append new agent1 to path1
     If agent2 is not 2:
         -Update agent2 using collatz_step(agent2)
         -Append new agent2 to path2
     Display step steps+1: Agent 1 =agent1, Agent 2 =agent2
     If agent1 ==1 and agent2 ==1:
         -Display 'Both agents reached 1 after steps+1 steps!'
         -Call 'visualize_paths(path1, path2, steps+1)'
         -Return
     Increment steps

  5. If max_steps reached:
         -Display 'Max steps reached, rendezvous failed'
         -Call 'visualize_paths(path1, path2, steps)

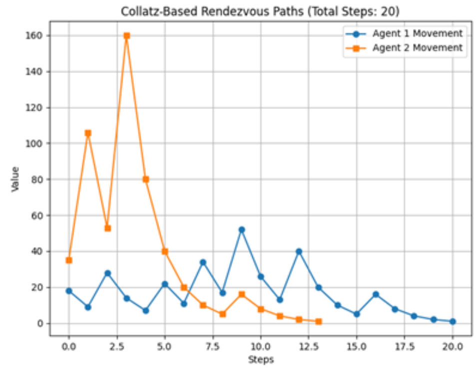
- Define function visualize_paths(path1, path2, steps):
    -Extract x1, y1 from path1
    -Extract x2, y2 from path2
    -Plot trajectories of Agent 1 and Agent 2
    -Add labels, title, legend and grid
    -Display the plot

- Define function start_simulation():
    -Read user input values x1 and x2.
    -Clear previous output
    -Call simulate_rendezvous((x1,), (x2,))

- Setup Tkinter UI:
    -Create main window with title 'Collatz Rendezvous Simulation'
    -Create input fields for Agent 1 and Agent 2 initial values.
    -Add 'Start Simulation' button to trigger 'start_simulation()'
    -Create a scrolled text widget for displaying results
    -Start the Tkinter event loop



(a) CRM (18, 35)



(b) Graphical representation of CRM (18, 35)

Figure 1: Example: CRM (18, 35)

Figure 1(a) shows an example of two agents with their initial starting points, 18 and 35, meeting at 1, successfully attaining rendezvous through CRM. We observe that both agents A1 and A2 meet at 1 or attain rendezvous after 20 steps (Figure 1(b)). Likewise, we can represent how multiple agents can meet by following a deterministic path as Collatz.

4. Results and Analysis

The Collatz Rendezvous Model is tested for various initial values and the results demonstrate how agents following the independent Collatz sequences eventually synchronize at a common value, typically 1.

4.1. Observations:

- Guaranteed Synchronization: Regardless of initial values, all agent pairs eventually reach a common meeting point, confirming the deterministic nature of CRM.
- Variable Step Count: The number of steps required for synchronization varies significantly, depending on whether the starting values are odd or even.
- Intermediate Synchronization Points: Some agent pairs meet at intermediate values before converging to 1, indicating partial rendezvous before final synchronization

4.2. Synchronization Step Analysis

Table 1 shows that any agent starting from any positive number eventually reach 1 or meets other agents for attaining Rendezvous.

Initial Values (A1, A2)	Steps Taken to Meet	Meeting Point
(6, 15)	16	1
(7, 20)	21	1
(12, 27)	51	1
(25, 32)	95	1
(19, 23)	42	1

Table 1: Synchronization Step Analysis

5. Conclusion

In this study, we introduced the Collatz Rendezvous Model (CRM), a novel approach to agent synchronization based on the iterative rules of the Collatz process. The model explores how agents starting from different initial values follow their independent Collatz sequences until they synchronize at a common value, 1. Unlike the traditional methods that rely on predefined paths on randomization, CRM ensures progressive convergence through a sequence of transformations that guide agents toward a common meeting point.

The CRM combines elements of number theory, agent-based modelling, and computational mathematics, offering a unique lens to study deterministic yet unpredictable synchronization process. This framework addresses key challenges such as unknown starting positions, asynchronous movement and varying agent speeds, making it applicable to both theoretical and real-world rendezvous scenarios.

Acknowledgments

The authors would like to thank the referees for their valuable comments and suggestions, which helped improve the quality and clarity of this paper.

References

1. E. Anderson, S. Fekete, *Two-dimensional rendezvous search,* Oper. Res. **49** (2001) 107–118.
2. E. Anderson, S. Fekete, *Asymmetric rendezvous on the plane*, in: Proc. 14th Annual ACM Symp. on Computational Geometry (1998) 365–373.
3. E. J. Anderson, R. R. Weber, *The rendezvous problem on discrete locations*, Journal of Applied Probability, **27**(4) (1990) 839–851.
4. V. Baston, S. Gal, *Rendezvous search when marks are left at the starting points*, Nav. Res. Logist. **48** (2001) 722–731.
5. V. Baston, S. Gal, *Rendezvous on the line when the players' initial distance is given by an unknown probability distribution*, SIAM Journal on Control and Optimization, **36**(6) (1998) 1880–1889.
6. J. Bendegem, *The Collatz Conjecture: A Case Study in Mathematical Problem Solving*, Logic and Logical Philosophy, **14** (2005), doi: 10.12775/LLP.2005.002.
7. S. Bouchard, Y. Dieudonné, A. Pelc, F. Petit, *On deterministic rendezvous at a node of agents with arbitrary velocities*, Information Processing Letters, **133** (2018) 39-43.
8. J. Czyzowicz, A. Labourel, A. Pelc, *How to meet asynchronously (almost) everywhere*, ACM Trans. Algorithms **8** (2012) 37.
9. A. Dessmark, P. Fraigniaud, A. Pelc, *Deterministic Rendezvous in Graphs*, ESA 2003, Lecture Notes in Computer Science, vol. 2832, Springer, Berlin, Heidelberg.
10. A. Farrugia, L. Gasieniec, L. Kuszner, E. Pacheco, *Deterministic Rendezvous in Restricted Graphs*, SOFSEM 2015, LNCS 8939, (2015) 189–200.
11. A. Farrugia, L. Gasieniec, L. Kuszner, E. Pacheco, *Deterministic rendezvous with different maps*, Journal of Computer and System Sciences, **106** (2019) 49-59.
12. M. Cieliebak, P. Flocchini, G. Prencipe, N. Santoro, *Distributed computing by mobile robots: Gathering*, SIAM J. Comput. **41** (2012) 829–879.
13. P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer, *Gathering of asynchronous robots with limited visibility*, Theor. Comput. Sci. **337** (2005) 147–168.
14. D. Kowalski, A. Malinowski, *How to meet in anonymous network*, Theor. Comput. Sci. **399** (2008) 141–156.
15. A. Miller, A. Pelc, *Fast Deterministic Rendezvous in Labeled Lines*, Proceedings of the 37th International Symposium on Distributed Computing (DISC 2023), Available: https://arxiv.org/abs/2311.12976.
16. A. Ta-Shma, U. Zwick, *Deterministic Rendezvous, Treasure Hunts, and Strongly Universal Exploration Sequences*, ACM Transactions on Algorithms, **10**(3) (2014) 12:1–12:15.
17. R. Kosova, R. Kapçiu, S. Hajrulla, A. Kosova, *A Review of Mathematical Conjectures: Exploring Engaging Topics for University Mathematics Students*, International Journal of Advanced Natural Sciences and Engineering Researches, **7** (2023) 180–186. doi: 10.59287/as-ijanser.581.
18. A. Pelc, *Deterministic Rendezvous Algorithms*, Université du Québec en Outaouais, 2023. Available: http://arxiv.org/abs/2303.10391v1.

19. S. S. Saab, *On the Convergence of the Collatz Conjecture Sequence*. Accessed: May 20, 2024. Available: https://www.researchgate.net/publication/379956173_On_the_Convergence_of_the_Collatz_Conjecture_Sequence.

20. S. Omran, H. Elshehabey, On the Collatz Conjecture, 2024. doi: 10.13140/RG.2.2.21071.50082.

Vidyashree H.R.,
Department of Mathematics,
School of Applied Sciences, REVA University,
Bengaluru, Karnataka, India.
E-mail address: vidyashree848@gmail.com

and

Lakshminarayana S.,
Department of Mathematics,
School of Applied Sciences, REVA University,
Bengaluru, Karnataka, India.
E-mail address: narayan_sl@reva.edu.in

and

Ajith S.,
S-Vyasa Deemed to be University (School of Advanced Studies),
Bengaluru, Karnataka, India,
E-mail address: ajith.sdl@gmail.com