

Estructura del documento

Manuel Olmos Antillón¹, Ximena Serna Mendoza², Guillermo Ian Barbosa Martínez², Ismael Posadas Pichardo³, and Paolo Medrano Magaña⁴

¹Ingeniería en Sistemas Computacionales, Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey

²Ingeniería en Ciencia de Datos y Matemáticas, Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey

³Ingeniería en Nanotecnología, Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey

⁴Ingeniería en Biotecnología, Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey

24 de octubre de 2024

Resumen

here you have to write the abstract

Índice

1. Introducción	3
2. PEAS	4
2.1. Performance	4
2.2. Environment	4
2.2.1. ¿Accesible o inaccesible?	5
2.2.2. ¿Determinista o No determinista?	5
2.2.3. ¿Episódico o No episódico?	5
2.2.4. ¿Estático o dinámico?	5
2.2.5. ¿Discreto o continuo?	6
2.3. Actuators	6
2.4. Sensors	7
3. Representaciones	8
4. Modelos generados	8
5. Evidencia de funcionamiento	8
6. Documentación del Programa en Python	8
6.1. Librerías	9
6.2. Funciones	10
6.2.1. <i>init_env (width, height)</i>	10
6.2.2. <i>square_coordinates (top_corner, bottom_corner, circular)</i>	10
6.2.3. <i>generate_water_chunks (env, water_chunks_sizes, percentages)</i>	11
6.2.4. <i>generate_obstacles (env, obstacles, percentages)</i>	11
6.2.5. <i>surrounding (env, coordinate)</i>	12
6.2.6. <i>generate_terrain (env, max_terrain_height)</i>	12
6.2.7. <i>generate_border (env)</i>	13
6.2.8. <i>remove_border (env)</i>	13
6.2.9. <i>environment (width, height, water_chunks_sizes, w_sizes_percentage, obstacles, o_size_percentage, max_terrain_height, add_border)</i>	14

6.2.10. <i>paint_env (env, show_symbols, rover_pos)</i>	15
7. Descripción Formal del Problema	16
8. Búsquedas	16
8.1. Informadas	16
8.1.1. <i>Greedy Search</i>	16
8.1.2. Búsqueda de Costo Uniforme	16
8.1.3. Búsqueda con A*	16
8.2. No Informadas	16
8.2.1. Anchura	16
8.2.2. Profundidad	16
8.3. Locales	16
8.3.1. <i>Hill Climbing</i>	16
8.3.2. <i>Hill Climbing with Random Restarts</i>	17
8.3.3. <i>Simulated Annealing</i> ó Recocido Simulado	17
9. Análisis de Búsquedas	18
9.1. Informadas y No Informadas	18
9.1.1. Mapa 1	18
9.1.2. Mapa 2	18
9.1.3. Mapa 3	18
9.2. Locales	18
9.2.1. <i>Hill Climbing</i>	18
9.2.2. <i>Hill Climbing with Random Restarts</i>	18
9.2.3. <i>Simulated Annealing</i> ó Recocido Simulado	18
10. Conclusiones	18
A. Nombre del Apéndice	18
B. Referencias	18

1. Introducción

Este reporte representa la culminación de un proyecto dedicado a modelar en *Python* al *Rover Perseverance* de la NASA [14], enfocado en su exploración en busca de agua en Marte[7]. A lo largo de diversas entregas,

hemos abordado la adaptación de conceptos clave de la inteligencia artificial, como el modelo PEAS, y la aplicación de una variedad de algoritmos de búsqueda, tanto informada como no informada, para analizar su eficiencia en un entorno simulado. Se ha abordado desde la revisión inicial de la misión del *Perseverance* y la descripción de su entorno, hasta el análisis detallado de algoritmos de búsquedas informadas (*greedy*, costo uniforme y A^*), no informadas (anchura y profundidad) y locales (*hill climbing*, *hill climbing with random restarts* y *simulated annealing* ó recocido simulado [3]), y se ha explorado cómo este agente inteligente aborda su tarea en un ambiente desafiante como el marciano. En este informe, se consolidarán los hallazgos y conclusiones obtenidos a lo largo de cada etapa de este proyecto, ofreciendo una visión integral del desempeño y las capacidades del *Rover Perseverance* [12] en su búsqueda de agua en Marte.

2. PEAS

2.1. Performance

Como se comentó anteriormente, en esta situación el agente es un robot explorador conocido como *Rover Perseverance*. De acuerdo con la NASA, esta entidad es capaz de recolectar muestras de roca y regolito con base en evaluaciones de sus características químicas, físicas, minerales y orgánicas en búsqueda de signos de vida microbiana antigua[8, 5]. De esta manera, se espera que las almacene y, posteriormente, las deposite sobre superficie marciana para ser recuperadas en una misión futura para ser analizadas en grandes laboratorios aquí en la Tierra.

2.2. Environment

El ambiente del robot es el cráter de Jezero en Marte. Es una región cuyo entorno antiguo, según la NASA, pudo haber favorecido la vida microbiana, lo que podría indicar la posibilidad de que existiera la presencia de agua o viscosidad, así como otros factores como la presencia de otros nutrientes, la humedad de la región misma, junto con otras características físicas, químicas, minerales, etc. Además, el cráter, como se ha podido observar en los registros de las recolecciones de la misión, cuenta con rocas marcianas que pueden ser útiles para análisis exhaustivos posteriores.

2.2.1. ¿Accesible o inaccesible?

85 % accesible: el *Perseverance* cuenta con tecnología de punta. Más adelante se hablará sobre esto. Bajo la lupa geológica, algunos factores potenciales para el estudio de vida en el cráter de Jezero, como la composición y estructura internas de la región, son límites naturales del *Rover* debido a la complejidad de estos instrumentos y herramientas que ya de por sí en nuestro planeta representan. Esto va de la mano también con el acceso más microscópico a las estructuras físicas, químicas, etc. de las formas de vida a estudiar. Esto se puede justificar argumentando que el objetivo del *Rover* es, ante todo, la recolección de estos “artículos” para una futura misión de recuperación de estos materiales para un análisis posterior y mucho más sofisticado con herramientas tecnológicas en laboratorios mucho más revolucionarias.

2.2.2. ¿Determinista o No determinista?

60 % determinista: Dados los conocimientos actuales sobre Marte, así como las determinadas rutinas o actividades de exploración sobre el cráter de Jezero específicamente, se pueden determinar numerosos estados futuros e importantes en el cráter. No obstante, es precisamente este factor de incertidumbre, así como comportamientos o fenómenos del planeta que aún podrían ser desconocidos para nosotros[15]. Además, eventos como las conocidas tormentas de polvo contribuyen a la aleatoriedad a la que se enfrenta el *Rover* en este ambiente.

2.2.3. ¿Episódico o No episódico?

No episódico: El cráter de Jezero en Marte no cuenta con procesos bien definidos por inicio o final. A pesar de que pueden tener lugar numerosas acciones y percepciones por parte del agente, su entorno no le permite establecer períodos bien definidos y conocidos, más allá de procesos de exploración, análisis y descubrimiento que tampoco tienen momentos con inicio o fin bien constituidos en el entorno.

2.2.4. ¿Estático o dinámico?

95 % estático: a pesar de la aleatoriedad e incertidumbre que aún puede rodear a la región a explorar, por lo general se entiende como monótono y

desértico al entorno que encierra al cráter. Dada la escasa actividad geológica, la falta de evidencia sólida de la presencia de agua en la actualidad y las bajas temperaturas del planeta, sugieren que en realidad tanto el cráter como el planeta presentan muy poca actividad. La incertidumbre radica principalmente en los mencionados fenómenos del planeta rojo que aún podría ser desconocidos para nosotros o el agente, así como los conocidos como las tormentas de polvo.

2.2.5. ¿Discreto o continuo?

Continuo: Es irónico que, pese al ya mencionado carácter desértico de Marte, como se ha comentado anteriormente, existan procesos ya conocidos. En esta ocasión, hay uno en particular que indica una producción constante de un gas vital para procesos geológicos. Se trata del metano que, aunque no es prueba contundente de la existencia de vida microbiana, sugiere junto con otros factores de disciplinas científicas mencionadas como la geología, la química y la física, actividad en el planeta. Esta actividad, además de la generación de gas metano, puede ir desde los climas de Marte, su recepción de peligrosa radiación cósmica[6], etc. sugieren la carencia de un número o patrón bien definido de acciones y, ahora con el agente, percepciones que pueden verse en el planeta rojo.

2.3. Actuators

- Ruedas para moverse y desplazarse a lo largo del cráter.
- Trípode integrado a la cámara que le permite moverse también y tener vista panorámica.
- Láser ultravioleta, integrado al receptor SHERLOC para hallar químicos.
- MOXIE[10], que produce oxígeno a raíz del dióxido de carbono de la atmósfera del planeta. Es también sensor, ya que debe detectar el químico y transformarlo.
- Láseres para disparar a rocas marcianas bien identificadas.
- Tubos de ensayo de titanio que almacenan rocas marcianas para análisis posteriores.

- Taladro de perforación para horadar la superficie marciana y depositar los tubos.
- Brazo que cuenta con el taladro y le da dirección a este, además de que permite sujetar cuidadosamente las muestras para almacenarlas, depositarlas, etc.
- Chasis que sirve para guardar los tubos que almacenen muestras importantes.

2.4. Sensors

- SuperCam[13], un cámara con tecnología de punta que permite al robot ver a increíbles distancias identificar químicos.
- Micrófono, que permite no solo que el robot escuche, sino que registre en audio procesos como al robot mismo desplazándose, el aire del planeta, los impactos del láser a rocas marcianas, etc.[11]
- SHERLOC[4] y el láser ultravioleta que tienen incorporado. A pesar de que le permite al agente actuar de determinada manera para buscar objetos, también le permite sentirlos, detectarlos, y obtener información del entorno con base en ello.
- RIMFAX[9]: un espectrómetro con rayos X que, prácticamente, permite al *Rover* ver bajo la superficie con un georadar en busca de características geológicas.
- PIXL[2]: otro espectrómetro con rayos X y cámara que permite al agente medir la composición química de las rocas a una escala muy fina e incluso tomarles foto a estas y a las texturas del suelo marciano.
- MOXIE:[10] su papel como sensor radica en la detección de dióxido de carbono. Por otro lado, como se comentó en el apartado anterior, su rol como actuador es producir oxígeno propio. Es una prueba para futuros exploradores.

Elemento	Símbolo
Agua	—
Obstáculos	*
Niveles topográficos	1, 2, 3, 4, 5, 6
Límite del terreno	#
Rover	+

Cuadro 1: Representaciones de los elementos en el mapa.

3. Representaciones

4. Modelos generados

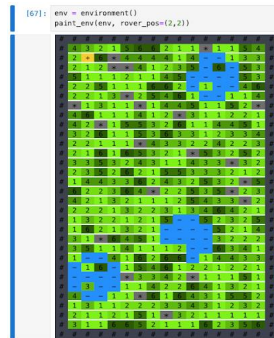


Figura 1: Primera prueba de generación de un mundo aleatoria para el *Rover Perseverance*

5. Evidencia de funcionamiento

6. Documentación del Programa en Python

El código *environment* contiene una serie de funciones que permiten crear un mapa con cuerpos de agua, obstáculos y terreno de variadas alturas de manera aleatoria, buscando generar una representación plausible de un entorno natural.

Las funciones implementadas otorgan al usuario la libertad de especificar la

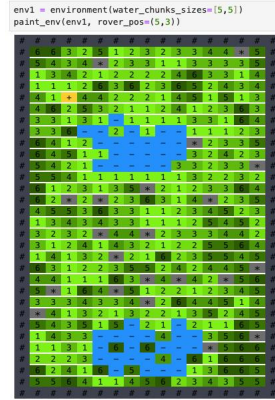


Figura 2: Segunda prueba de generación de un mundo aleatoria para el *Rover Perseverance*

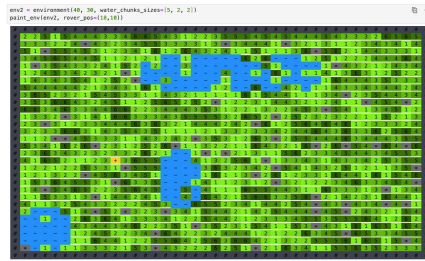


Figura 3: Tercera prueba de generación de un mundo aleatoria para el *Rover Perseverance*

cantidad y tamaños de los cuerpos de agua, cantidad de obstáculos, altura máxima de terreno, etc.

6.1. Librerías

random [1] Para la generación de coordenadas y dimensiones aleatorias apegadas a los requisitos proporcionados para el reto.

SimpleAI [16] Para ejecutar algoritmos de búsqueda informadas, no informadas y locales para el agente diseñado.

6.2. Funciones

6.2.1. *init_env (width, height)*

Recibe

- *width (int)*: Anchura del mapa.
- *height (int)*: Altura del mapa.

Regresa

- *env* (lista de listas de *strings*): Mapa con elementos vacíos.

Descripción Inicializa el entorno natural de la anchura y altura determinada con strings con un espacio.

6.2.2. *square_coordinates (top_corner, bottom_corner, circular)*

Recibe

- *top_corner* (tupla): Coordenada de una matriz.
- *bottom_corner* (tupla): Coordenada de una matriz
- *circular* (booleano): Determina si se debe regresar un patrón circular dentro del recuadro generado entre ambas esquinas.

Regresa

- *coordinates* (lista de tuplas): Lista con coordenadas dentro del recuadro formado por la coordenada *top_corner* y la coordenada *bottom_corner*.

Descripción Obtiene las coordenadas de una matriz que se encuentran dentro del recuadro formado por las esquinas proporcionadas, y elimina las esquinas del recuadro en caso de querer un patrón circular.

6.2.3. *generate_water_chunks (env, water_chunks_sizes, percentages)*

Recibe

- *env* (lista de listas de *strings*): Entorno natural vacío.
- *water_chunks_sizes* (lista de *ints*): Lista con tamaños de cada cuerpo de agua.
- *percentages* (booleano): Determina si los tamaños vienen en porcentaje del mapa o número de casillas.

Regresa

- *None*: Modifica *env* directamente.

Descripción Divide las celdas en n secciones, donde n representa los cuerpos de agua deseados, y selecciona aleatoriamente una celda dentro de cada una de estas secciones. A partir de esta celda de origen, se eligen aleatoriamente una anchura y altura, con las cuales se selecciona otra celda dentro de la matriz. Luego, se obtienen las coordenadas dentro del rectángulo formado por ambas celdas y se seleccionan k coordenadas (donde k es el número de bloques de agua) de esta lista. Estas coordenadas seleccionadas se modifican posteriormente al símbolo del agua en el entorno *env*.

6.2.4. *generate_obstacles (env, obstacles, percentages)*

Recibe

- *env* (lista de listas de *strings*): Entorno natural con cuerpos de agua o vacío.
- *obstacles* (*int*): Tamaño de los obstáculos.
- *percentages* (booleano): Determina si el tamaño viene en porcentaje del mapa o número de casillas.

Regresa

- *None*: Modifica *env* directamente.

Descripción Genera una lista con las celdas vacías del entorno, y selecciona k coordenadas (donde k es el número de obstáculos) de esta lista. Estas coordenadas seleccionadas se modifican posteriormente al símbolo de obstáculos en el entorno *env*.

6.2.5. *surrounding (env, coordinate)*

Recibe

- *env* (lista de listas de *strings*): Entorno natural.
- *coordinate* (tupla de *ints*): Coordenada dentro de la matriz.

Regresa

- *surround_boxes* (lista de *strings*): Lista con los elementos de las celdas que rodean a alguna celda, incluyendo a sí misma.

Descripción Genera una lista con los elementos de las celdas alrededor de alguna celda en la matriz. La función está adaptada para funcionar en esquinas y aristas, donde los alrededores son incompletos.

6.2.6. *generate_terrain (env, max_terrain_height)*

Recibe

- *env* (lista de listas de *strings*): Entorno natural con cuerpos de agua y obstáculos.
- *max_terrain_height* (*int*): Entero que determina el nivel máximo de altura generada en el terreno.

Regresa

- *None*: Modifica *env* directamente.

Descripción Itera sobre las celdas de un entorno, saltándose aquellas que ya contienen agua o son obstáculos. Cuando encuentra una celda vacía, obtiene las celdas que la rodean utilizando la función *surrounding*. Si entre estas celdas cercanas se encuentra agua, se cuadriplica la probabilidad de que la altura del terreno sea 1 durante la selección aleatoria de alturas. En caso contrario, calcula el promedio de alturas de las celdas vecinas y triplica la probabilidad de que la altura generada aleatoriamente esté dentro del rango de la media - 1 a la media + 1. Una vez ajustadas estas probabilidades, se elige una altura con los nuevos pesos y se modifica la coordenada actual con esa altura.

6.2.7. *generate_border (env)*

Recibe

- *env* (lista de listas de *strings*): Entorno natural.

Regresa

- *None*: Modifica *env* directamente.

Descripción Genera un borde alrededor del entorno dado, simulando un contorno.

6.2.8. *remove_border (env)*

Recibe

- *env* (lista de listas de *strings*): Entorno natural.

Regresa

- *None*: Modifica *env* directamente.

Descripción Elimina el borde alrededor del entorno dado, en caso de que este exista.

6.2.9. *environment (width, height, water_chunks_sizes, w_sizes_percentage, obstacles, o_size_percentage, max_terrain_height, add_border)*

Recibe

- ***width (int)***: Anchura del mapa.
- ***height (int)***: Altura del mapa.
- ***water_chunks_sizes (lista de ints)***: Lista con tamaños de cada cuerpo de agua.
- ***w_sizes_percentage (booleano)***: Determina si los tamaños de los cuerpos de agua vienen en porcentaje del mapa o número de casillas.
- ***obstacles (int)***: Tamaño de los obstáculos.
- ***o_size_percentage (booleano)***: Determina si el tamaño de los obstáculos viene en porcentaje del mapa o número de casillas.
- ***max_terrain_height (int)***: Entero que determina el nivel máximo de altura generada en el terreno.
- ***add_border (booleano)***: Determina si el entorno generado contiene una barrera de contorno.

Regresa

- ***env (lista de listas de strings)***: Entorno natural con cuerpos de agua, obstáculos y variadas alturas de terreno.

Descripción Genera un entorno con cuerpos de agua, obstáculos y terreno de altura variable, utilizando los siguientes parámetros predeterminados: un ancho de 15 bloques, una altura de 30 bloques, tres cuerpos de agua que representan el 3 % del mapa, obstáculos que representan el 5 % del mapa, y una altura máxima de terreno de 6 bloques. Estos valores pueden modificarse según la preferencia del usuario.

6.2.10. *paint_env* (*env*, *show_symbols*, *rover_pos*)

Recibe

- *env* (lista de listas de *strings*): Entorno natural.
- *show_symbols* (booleano): Determina si se muestran los símbolos de los elementos en la impresión.
- *rover_pos* (tupla de *ints*): Coordenada de ubicación del *Rover*. Coordenada predeterminada que se encuentra fuera del entorno.

Regresa

- *None*: Imprime el entorno natural de manera estética.

Descripción Imprime una versión pintada del entorno natural, utilizando códigos ANSI para dar formato a la impresión en la terminal. Los elementos del entorno, como el agua, los obstáculos y las diferencias de altura, se representan con colores de fondo diferentes para que sean fácilmente observables.

```

Tiempo en encontrar agua: 0.12 s

Estado final: (9, 1)
Numero de pasos en solución: 10

Camino a la solución:
0: (None, (1, 1))
1: (('Down', (1, 0)), (2, 1))
2: (('Down', (1, 0)), (3, 1))
3: (('Down', (1, 0)), (4, 1))
4: (('Down Left', (1, 1)), (5, 2))
5: (('Down', (1, 0)), (6, 2))
6: (('Down Left', (1, 1)), (7, 3))
7: (('Down Right', (1, -1)), (8, 2))
8: (('Down', (1, 0)), (9, 2))
9: (('Up Right', (-1, -1)), (8, 1))
10: (('Down', (1, 0)), (9, 1))

```

Figura 4: Descripción de la ruta del *Rover* en el Mapa 1 con Búsqueda por Anchura.

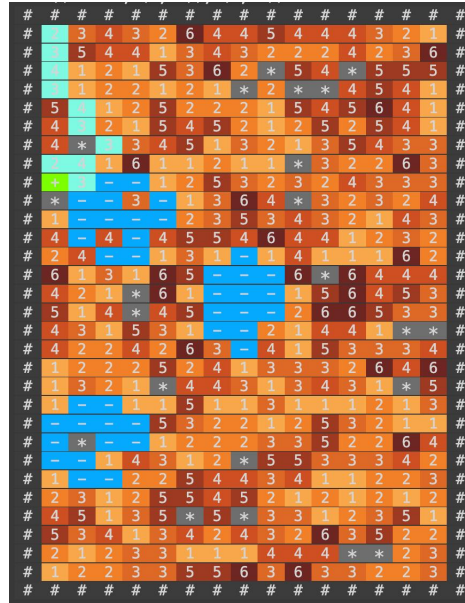


Figura 5: Visualización de la ruta del *Rover* en el Mapa 1 con Búsqueda por Anchura.

7. Descripción Formal del Problema

8. Búsquedas

8.1. Informadas

8.1.1. *Greedy Search* 16

8.1.2. Búsqueda de Costo Uniforme

8.1.3. Búsqueda con A*

8.2. No Informadas

8.2.1. Anchura

Altura mínima

8.3.2. *Hill Climbing with Random Restarts*

Altura máxima

Altura mínima

8.3.3. *Simulated Annealing* ó Recocido Simulado

Altura máxima

Altura mínima

9. Análisis de Búsquedas

9.1. Informadas y No Informadas

9.1.1. Mapa 1

9.1.2. Mapa 2

9.1.3. Mapa 3

9.2. Locales

9.2.1. *Hill Climbing*

9.2.2. *Hill Climbing with Random Restarts*

9.2.3. *Simulated Annealing* ó Recocido Simulado

10. Conclusiones

A. Nombre del Apéndice

B. Referencias

Referencias

- [1] Alimova Rayhon Abdugafforovna. Modules and functions in the python programming language. *Texas Journal of Multidisciplinary Studies*, 18:49–54, 2023.
- [2] Abigail C Allwood, Joel A Hurowitz, Benton C Clark, Luca Cinquini, Scott Davidoff, Robert W Denise, W Timothy Elam, Marc C Foote, David T Flannery, James H Gerhard, et al. The pixl instrument on the mars 2020 perseverance rover. *arXiv preprint arXiv:2103.07001*, 2021.
- [3] Rafael E Banchs. Simulated annealing. *Research progress report, on Time Harmonic Field Electric Logging Austin, University of Texas at Austin*, 1997.

- [4] Luther Beegle and Rohit Bhartia. Sherlock: an investigation for mars 2020. In *EGU General Assembly Conference Abstracts*, pages EPSC2016–11215, 2016.
- [5] JF Bell, JN Maki, GL Mehall, MA Ravine, MA Caplinger, ZJ Bailey, S Brylow, JA Schaffner, KM Kinch, MB Madsen, et al. The mars 2020 perseverance rover mast camera zoom (mastcam-z) multispectral, stereoscopic imaging investigation. *Space science reviews*, 217:1–40, 2021.
- [6] Cristina Campillo de Andrés. Arquitecturas desenchufadas. utopías contemporáneas. marte. el viaje no es el problema. 2021.
- [7] Tomás del Campo Muñoz et al. Clasificación y cuantificación de espectros raman de la misión mars 2020 de la nasa: un caso de estudio con olivinos. 2022.
- [8] Betzabe Natali Carlos Chavez. Factores de riesgo que influyen en el crecimiento bacteriano (*escherichia coli* y *staphylococcus aureus*) en trucha arcoíris (*oncorhynchus mykiss*) comercializado en los principales mercados de huánuco–2023. 2023.
- [9] Svein-Erik Hamran, Tor Berger, Sverre Brovoll, Leif Damsgård, Øystein Hellenen, Mats Jørgen Øyan, Hans Erik Amundsen, Lynn Carter, Rebecca Ghent, Jack Kohler, et al. Rimfax: A gpr for the mars 2020 rover mission. In *2015 8th International Workshop on Advanced Ground Penetrating Radar (IWAGPR)*, pages 1–4. IEEE, 2015.
- [10] Jeffrey A Hoffman, Michael H Hecht, Donald Rapp, Joseph J Hartvigsen, Jason G SooHoo, Asad M Aboobaker, John B McClean, Andrew M Liu, Eric D Hinterman, Maya Nasr, et al. Mars oxygen isru experiment (moxie)—preparing for human mars exploration. *Science Advances*, 8(35):eabp8636, 2022.
- [11] Timothy Leighton. Thoughts on the sounds of mars from nasa’s mars perseverance. *Acoustics Bulletin*, 45(3):22–28, 2021.
- [12] Steels Luc. Cooperation between distributed agents through self-organization. *Decentralized AI*, pages 175–196, 1990.
- [13] Sylvestre Maurice, Roger Craig Wiens, Pernelle Bernardi, Philippe Caïs, S Robinson, T Nelson, Olivier Gasnault, J-M Reess, Muriel Deleuze,

- Fernando Rull, et al. The supercam instrument suite on the mars 2020 rover: Science objectives and mast-unit description. *Space Science Reviews*, 217:1–108, 2021.
- [14] Seth R Meeker, Matthew Noyes, Hong Tang, Garreth Ruane, Camilo Mejia Prada, Eduardo Bendek, Wesley Baxter, Brendan Crill, AJ Eldorado Riggs, Phillip K Poon, et al. The twin decadal survey testbeds in the high contrast imaging testbed facility at nasa’s jet propulsion laboratory. In *Techniques and Instrumentation for Detection of Exoplanets X*, volume 11823, pages 322–334. SPIE, 2021.
- [15] Eulogio Pardo-Igúzquiza and Juan José Durán Valsero. Geología de marte: un planeta fósil. 2020.
- [16] Nimit Thaker and Abhilash Shukla. Python as multi paradigm programming language. *International Journal of Computer Applications*, 177(31):38–42, 2020.