

Luc Steels' Mars Explorer Simulation With Collaborative Transport

Amable José Valdés Cuervo, *amable.valdes.cuervo@alumnos.upm.es*

I. INTRODUCCIÓN

En este documento presento una mejora del experimento de Luc Steel [1]; una simulación de un enjambre de robots moviéndose por la superficie de marte recogiendo muestras de minerales (u otros posibles elementos).

Mi mejora consiste en la modificación de alguna de las reglas de los robots que interactúan con los minerales recogiéndolos y guardándolos en su base.

Estas modificaciones en las reglas me han permitido añadir transporte colaborativo para minerales que pesen demasiado para ser transportados por un solo robot, permitiendo que varios robots interaccionen entre sí para mover minerales más pesados.

II. SIMULACIÓN BASE

Para comenzar este trabajo lo primero que realicé fue una simulación del experimento de Luc Steel [1]. Para ello he creado un programa en JavaScript que dibuja sobre un canvas HTML nuestros agentes y los elementos con los que interaccionarán.

La simulación consiste en lo siguiente: en el entorno se encontrarán dispersos diversos robots que se moverán de manera aleatoria por el mapa. Una vez un robot sienta con sus sensores un mineral se acercará a él y lo cogerá para acto seguido ir a la nave nodriza con él y depositarlo allí. En el trayecto desde la posición del mineral hasta la nave nodriza el robot depositará en el suelo 2 rastros. Si un robot encuentra 1 rastro lo cogerá y se moverá en dirección contraria a la nave nodriza, de esa manera podrá encontrar más rastros o más minerales.

Me he visto obligado a relajar el problema quitando posibles obstáculos presentes en el entorno (riscos, acantilados, terreno irregular...). El comportamiento de evitar obstáculos estaba presente en el paper original, pero implementarlo llevaría mucho tiempo y un control más exhaustivo de las colisiones entre elementos del canvas, por lo que he decidido relajar el problema eliminando obstáculos.

En la Figura 1 muestro los distintos elementos que pueden verse en la simulación, siendo los robots nuestros agentes que interactuarán con el resto de los elementos modificando su entorno.



Fig. 1. Elementos de nuestra simulación: a) La nave nodriza. b) los rastros "migas" que van dejando los robots. c) los robots. d) los minerales.

En el experimento de Luc Steel [1] los robots usaban una *subsumption architecture* para poder trabajar con naturalidad.

La *subsumption architecture* fue propuesta por Rodney Brooks en [1]. Consiste en descomponer el comportamiento de un robot en subcomportamientos, los cuales son organizados de manera jerárquica en capas (*layers*). Cada capa recibe la información de los sensores del robot y la jerarquía de capas permite decidir cual de los subcomportamientos se ejecutará, pudiendo este anular subcomportamientos que se habrían ejecutado en capas más inferiores o no. Un ejemplo de esta arquitectura puede verse en la Figura 2.

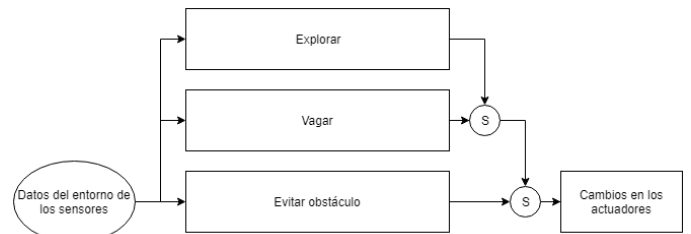


Fig. 2. Ejemplo de la subsumption architecture.

Para la simulación base he seguido las reglas ya propuestas en el paper original a excepción de la regla que evitaba obstáculos, siendo las reglas finales:

1. IF carrying samples AND at the base THEN drop samples
2. IF carrying samples and NOT at the base THEN drop 2 crumbs AND travel up gradient
3. IF detect a sample AND NOT at the base THEN pick sample up
4. IF sense crumbs THEN pick up 1 crumb AND travel down gradient
5. IF true THEN move randomly

III. RESULTADOS SIMULACIÓN BASE

La simulación base funciona perfectamente.

Tarde o temprano los robots encuentran todos los minerales del entorno siguiendo las 5 reglas presentadas anteriormente. En el tiempo en el que se transportan todos los minerales los robots crean caminos artificiales de migas de pan que conectan la nave nodriza con los clúster de minerales, caminos los cuales son destruidos cuando un clúster de minerales deja de proveer material.

En la Figura 3 puede observarse una captura de pantalla de la simulación.

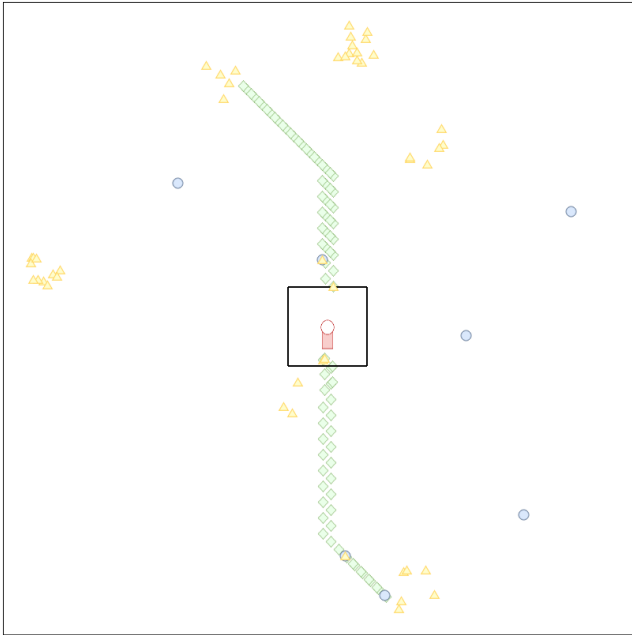


Fig. 3. Captura de la simulación base sobre la que trabajaré.

IV. AMPLIACIÓN: TRANSPORTE COLECTIVO

Uno de los primeros problemas que podríamos encontrarnos en un entorno real sería la situación en la que un robot no podría él solo transportar un mineral desde su ubicación original hasta la nave nodriza debido al tamaño y peso del mineral.

Basándome en la premisa de que existiesen este tipo de minerales más pesados y en el paper de SWAP ROBOTS [3] he programado (siguiendo la misma arquitectura que en el trabajo original) una nueva simulación en la que los robots ahora interaccionan entre sí, comunicándose para poder transportar de manera colectiva dos robots un mineral pesado.

He asignado un peso a los minerales y he representado este peso en su imagen mediante su tamaño; un mineral grande va a ser más pesado, mientras que un mineral pequeño va a ser menos pesado. Un ejemplo del distinto tamaño que pueden tener ahora los minerales puede observarse en la Figura 4.

Para permitir que los robots puedan comunicarse, me he inspirado en [3]; ahora los robots emitirán una leve señal. No tan potente como la nave nodriza, pero sí para que en sus inmediaciones puedan pedir ayuda para mover un mineral. En la figura 4, de nuevo, pueden verse las modificaciones que hemos hecho en los sprites para esta versión y el aspecto de un robot que emite una señal de ayuda para mover un mineral.

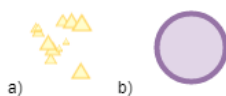


Fig. 4. Nuevos sprites para esta versión: a) Minerales con distinto tamaño. b) Robot emitiendo señal de ayuda.

Para poder realizar esta tarea hay que añadir un nuevo comportamiento a la jerarquía de layers del robot y modificar

alguna ya existentes. Nuestras reglas finales son las siguientes:

1. IF carrying samples AND at the base THEN drop samples
2. IF carrying samples AND NOT at the base THEN
 - a. IF you are the principal transporter THEN drop 2 crumbs
 - b. travel up gradient
3. IF detect help signal AND NOT at the base THEN help other robot
4. IF detect a sample AND NOT at the base THEN
 - a. IF you have enough force THEN pick the sample AND stop signal (if it was on) ELSE start the help signal
5. IF sense crumbs THEN pick up 1 crumb AND travel down gradient
6. IF true THEN move randomly

La primera modificación importante es en la regla 4: ahora cuando un robot encuentre un mineral y vaya a intentar transportarlo puede que el mineral sea demasiado pesado, y entonces el robot comenzará a emitir una leve señal de ayuda para los robots cercanos indicando el mineral sobre el que necesita ayuda.

La segunda modificación importante es que hemos añadido una nueva regla en la jerarquía, la regla 3. Esta regla permite que cuando un robot este vagando por el entorno y encuentre a otro robot que esté emitiendo una señal de ayuda vaya en su auxilio. Esta regla es más importante que la regla 4 pues nos interesa que los robots ayuden a otros robots antes que coger otro mineral (que también podrían ser pesado) y, de esta manera, ir aumentando el número de migas del rastro y atrayendo nuevos robots que podrían ayudar en caso de otro mineral pesado en el clúster. Esta regla es, a su vez, menos importante que la regla 2, pues si el robot ya está transportando un mineral, no puede ayudar a otro robot cuya señal haya escuchado, ya que está ocupado.

Para el transporte colaborativo he programado el siguiente comportamiento: cuando el robot que estaba emitiendo la señal de ayuda encuentra otro robot el nuevo robot se aproxima y el otro apaga la señal de ayuda. Los dos robots giran alrededor del mineral y buscan la posición con la que optimizarían las fuerzas de empuje. Una vez tengan una posición que permita el transporte los robots comienzan a empujar hasta que llegan a la base.

También he realizado una pequeña modificación en la regla 2. Si ambos robots sueltan 2 migas el camino que se generaría tardaría mucho más tiempo en destruirse cuando ya no quedasen minerales en el clúster, un tiempo perdido. Para solventar esto cuando un grupo de robots está transportando un mineral pesado, solo el robot que encontró el mineral (el que emitió la señal por primera vez) será el que soltará el rastro.

V. RESULTADOS AMPLIACIÓN: TRANSPORTE COLECTIVO

La simulación funciona perfectamente.

Todos los minerales son recogidos tarde o temprano en el entorno y llevados a la nave nodriza. Las 6 reglas expuestas anteriormente son robustas y permiten un transporte sin incidentes. En la Figura 5 puede verse una captura de pantalla del transporte colectivo de varios minerales siendo transportados a la nave nodriza.

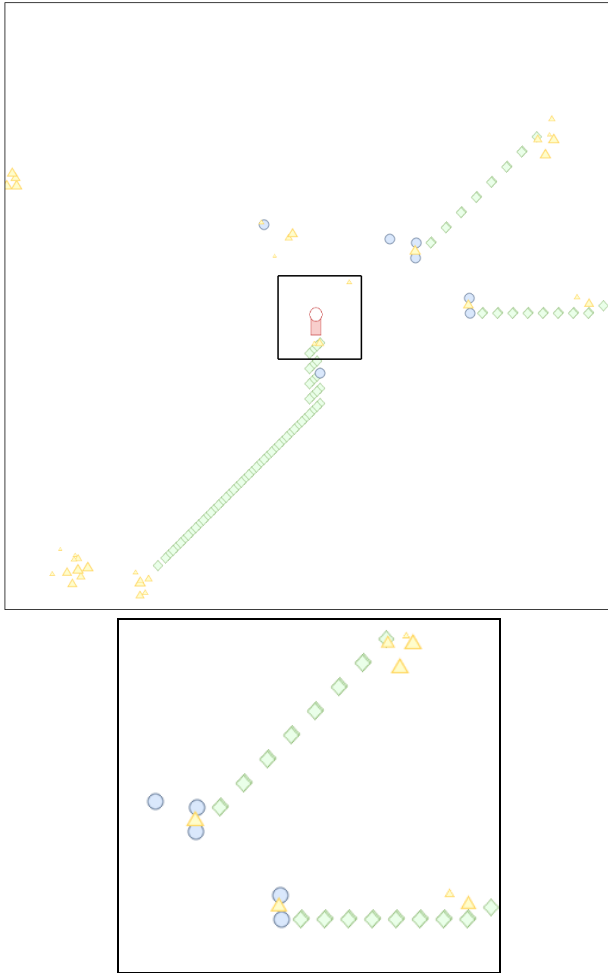


Fig. 5. Captura de la simulación con transporte colectivo y ampliación de la zona donde se pueden observar 2 grupos de 2 robots moviendo 1 mineral.

Una vez realizada esta ampliación encontré un pequeño problema con este enfoque: El *problema de la cena de los filósofos* (o, también llamado, *dining philosophers problem*). Podría darse el caso de que los robots cogieran todos y cada uno de ellos un mineral pesado y todos ellos comenzasen a enviar una señal de ayuda, y de esta manera todo el sistema quedaría bloqueado, esperando eternamente todos y cada uno de ellos a que otro robot les ayudase a mover su mineral.

Como ejemplo de esto en la Figura 6 podemos ver un robot bloqueado, sin nadie cerca para ayudarlo. El resto de los robots tardarán mucho tiempo en, de manera aleatoria, moverse hacia ese lugar.

Esta situación puede solucionarse fácilmente añadiendo un tiempo máximo de emisión de señal y después de exceder ese

tiempo, anulando la posibilidad de coger minerales pesados centrándose solo en los pequeños durante un corto periodo de tiempo (para no volver a caer otra vez en el mismo mineral y volver a emitir la señal). No obstante, lo que de verdad me preocupa de esta versión del simulador es el excesivo tiempo de espera para que llegue otro robot, y me cuestioné si se podría hacer algo para mejorar este tiempo.

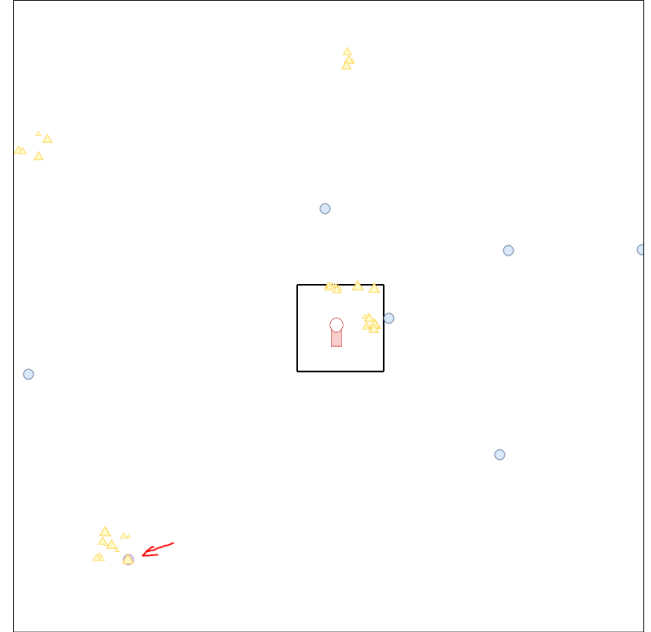


Fig. 6. Captura de la simulación con transporte colectivo en la que puede verse un robot alejado del grupo principal esperando por ayuda que tardará mucho tiempo en llegar.

VI. AMPLIACIÓN: TRANSPORTE COLECTIVO CON CAMINO PREVIO

Para solventar el problema presentado anteriormente y disminuir el tiempo en el que los minerales pesados son transportados e ideado, bajo la versión de la ampliación del transporte colaborativo mostrada anteriormente, una nueva versión en la que el robot que encuentre un mineral pesado, pero no tiene rastros cerca que ayuden a otros robots ha llegar hasta él, genere un rastro primero antes de coger y emitir la señal de ayuda.

De esta manera, con respecto a los sprites no he realizado ninguna modificación, pero si que he modificado las reglas del sistema:

1. IF need to warn others AND NOT at the base THEN drop 3 crumbs AND travel up gradient
2. IF need to warn others AND at the base THEN stop warn others
3. IF carrying samples AND at the base THEN drop samples
4. IF carrying samples AND NOT at the base THEN
 - a. IF you are the principal transporter THEN drop 2 crumbs
 - b. travel up gradient
5. IF detect help signal AND NOT at the base THEN

- help other robot
6. IF detect a sample AND NOT at the base THEN
 - a. IF you have enough force THEN pick the sample AND stop signal (if it was on) ELSE
 - i. IF you have crumbs close THEN start the help signal ELSE warn others
 7. IF sense crumbs THEN pick up 1 crumb AND travel down gradient
 8. IF true THEN move randomly

La primera modificación importante consiste en añadir una nueva sentencia IF en la regla 6; en el caso de que un robot haya encontrado un mineral demasiado pesado para ser cargado entonces el robot comprueba si quedan migas cerca de él. Si no hay un rastro cerca, entonces el robot no enciende la señal de ayuda y en su lugar decide viajar a la nave nodriza para avisar a otros de su situación.

Es entonces cuando entran en juego las nuevas reglas 1 y 2; con la necesidad de avisar a otros robots de un mineral cercano activa y lejos de la base la regla 1 se activa y el robot viaja hacia la nave nodriza, no sin antes dejar 3 migas. Es importante que deje 3 migas, ya que una será para que el propio robot vuelva sobre sus pasos y otra para que un robot cercano se acerque a ayudar. De esta manera aún quedaría una miga de pan para otro robot en el caso de necesitar más ayuda o para conservar el camino por la probabilidad de encontrar más minerales cerca de ese clúster.

La regla 2 se activa si el robot tiene esa necesidad de avisar a otros robots activa y se encuentra en la base. El robot entonces parará su función de avisar al resto de la presencia del mineral y volverá sobre el rastro construido esperando que otro robot encuentre, ahora más fácilmente, el rastro que le guíe al mineral pesado.

VII. RESULTADOS AMPLIACIÓN: TRANSPORTE COLECTIVO CON CAMINO PREVIO

Con estas simples 3 modificaciones (modificación en la regla 6, añadir regla 1 y añadir regla 2) he mejorado la simulación presentada anteriormente.

He comparado el tiempo que tarda una simulación generada con un número aleatorio de minerales en posiciones aleatorias en recogerlos todos y llevarlos a la base con las dos ampliaciones. He registrado los tiempos resultado en la Tabla 1.

Ambas simulaciones tardan una media de alrededor de 2 minutos (120 segundos) en encontrar todos los minerales del entorno y llevarlos a la base, mientras que la 1ª ampliación tarda ligeramente más que la 2ª ampliación.

La 2ª ampliación tarda, de media, ligeramente menos en resolver el problema: unos 10 segundos menos. No obstante, su verdadero potencial se encuentra en que el uso de las reglas de la 2ª simulación disminuye la probabilidad de que un robot quede bloqueado sin incluir un apagado de señal de ayuda a

los pocos minutos si no se encuentra un robot cercano que pueda ayudar a mover los minerales.

Iteración	1ª Ampliación		2ª Ampliación	
	Minerales	Tiempo	Minerales	Tiempo
1	57	142,717	53	107,406
2	40	94,705	44	121,906
3	32	104,507	42	104,116
4	35	141,306	44	109,806
5	49	112,705	42	129,804
6	59	127,608	54	115,005
7	57	121,408	49	100,207
8	47	121,411	57	122,308
9	54	148,509	37	89,204
10	47	134,307	55	127,206
Media	47,7	124,9183	47,7	112,6968

Tabla. 1. Tiempo en recoger todos los minerales de algunas simulaciones.

VIII. POSIBLES MEJORAS IMPLEMENTABLES

Sería interesante ver como funciona la nueva simulación junto a la existencia de obstáculos. En el paper [4] se organizan a varios robots para que puedan atravesar obstáculos junto a transporte colaborativo, y sería interesante añadirlo a este proyecto.

REFERENCIAS

- [1] L. Steels, "Cooperation between distributed agents through self-organisation," in *Intelligent Robots and Systems' 90. Towards a New Frontier of Applications*, Proceedings. IROS'90. IEEE International Workshop on. IEEE, 1990, pp. 8–14.
- [2] R. A. Brooks, "A Robust Layered Control System for a Mobile Robot", *Robotics and Automation*, IEEE Journal of; Mar 1986, pp. 14 – 23, vol. 2, issue 1.
- [3] R. O'Grady, C. Pinciroli, R. Groß, A. Christensen, F. Mondada, M. Bonani, and M. Dorigo. Swarm-bots to the rescue. In G. Kampis, I. Karsai, and E. Szathmry, editors, *Advances in Artificial Life. Darwin Meets von Neumann*, volume 5777 of LNCS, pages 165–172. Springer, 2011. ISBN 978-3-642-21282-6.
- [4] R. O'Grady, R., Groß, R., Christensen, A.L., Dorigo, M.: Self-assembly strategies in a group of autonomous mobile robots. *Autonomous Robots* 28(4), 439–455 (2010).