Machine learning for energy efficient modular robots self-reconfiguration

Baptiste Buchi[‡], Hakim Mabed[†], Frédéric Lassabe[*], Jaafar Gaber[*]

[‡]Université de Lorraine, CNRS, INRIA, LORIA, F-54000 Nancy, France
baptiste.buchi@univ-lorraine.fr

[*]UTBM, CNRS, Institut FEMTO-ST, F-90010 Belfort, France
{firstname.lastname}@femto-st.fr

[†]Université de Franche-Comté, CNRS, institut FEMTO-ST, F-25200 Montbéliard, France
hmabed@femto-st.fr

***Abstract*—While Modular Robots Systems (MRSs) are getting more popular, the challenge of effective self-reconfiguration of the robots persists. Actually, many distributed algorithms were proposed for the Modular Robots Self-Reconfiguration problem (MRSR), however, the efficiency of each algorithm according to the different cases is rarely studied. In a precedent work, we proved the relevance of using an Artificial Neural Network (ANN) component to select the most adapted distributed algorithm according to the current scenario. The proposed approach suffers, as all the other approaches in the literature, ignore the state of charge of the robots. Consequently, an MRSR algorithm may not achieve the MRS reconfiguration if the robots' batteries are exhausted before the end of the algorithm. In this paper, we propose an ANN system that selects the most suited MRSR algorithm according to the MRSR scenario and the initial state of charge of the robots. The results of the tests show an accuracy of 97%.**

***Index Terms*—Self-reconfiguration, modular robots system, distributed algorithm, programmable matter, machine learning, convolutional neural network, energy aware.**

I. Introduction

Programmable matter is one of the most challenging facets of the robotics field in general and modular robotics in particular. The programmable matter aims to design tiny robotics components that act like atoms or molecules in physical objects. These tiny robots have the ability to move with or without the help of the other robots in order to change their topology or form a given shape. The applications of such technologies are extremely wide [1] from transportation [2] and rescuing [3] to area exploration [4], and tracking [5].

A. Modular robots systems

We call Modular reconfigurable robots (MRR) or Metamorphic Modular Robots a set of identical robots of small size, able to move around each other to change the topology of the network itself [6], [7]. Over the years, many platforms modular robots were proposed by the research community [8], [9], [10]. These MRR systems differ according to their spatial organization (2D 4-lattice, 2D 6-lattice, 3D 12-lattice, etc.), and motion mode (rolling, translation, jump, rotation, etc.).

In this paper, we study the 2D-Catom platform [11] representing cylindrical robots piled over a horizontal plan and moving by rolling around the neighboring robots within a vertical plan space (see figure 2). However, the used methodology can be extended to any other platform.

B. Self-reconfiguration modular robots

We define the self-reconfiguration problem (SRMR) as the definition of mechanisms that allow a set of modular robots to change their global shape from the current one (initial shape) to a target shape [12].

Many methods were proposed to solve SRMR problem that could be classified according to the use or not of a centralized unit for establishing the reconfiguration plan. In centralized approaches, a master unit, with total knowledge of the initial reconfiguration of the modular robots, schedules the sequence of moves to perform by each robots to reach the final shape. In the distributed approaches, the robots run a local identical program defining the local motion rules used by each robot. While centralized approaches lead to huge spacial and computational complexity, distributed approaches suffer from variable quality according to the processed SRMR scenario.

To address the limitations of both centralized and distributed approaches, we proposed in a previous work a combination of the two approaches using Artificial Intelligence [13]. To that end, a CNN module analyses the current SRMR problem and then selects the most suitable distributed algorithm among a set of distributed methods: Cylindrical-Catoms Self-Reconfiguration (C2SR) [14] and Translation Based Self Reconfiguration (TBSR) [15]. Once the best algorithm is selected, the centralized unit send the identifier of the selected distributed algorithm to all modular robots and the distributed algorithm is then run.

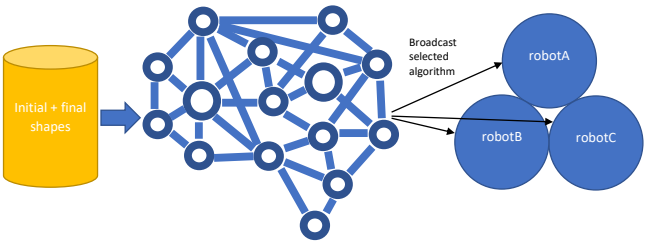

Fig. 1. Hybrid self-reconfiguration process: first the ANN selects the preferred distributed algorithm then the identifier of the algorithm is sent to all robots in order to be executed.

C. Contributions

The impact of the residual energy over the modular robots on the self-reconfiguration algorithm is ignored. Indeed, the convergence of the robots to the final shape depends on the state of charge of their batteries. Once the battery is exhausted, the robots can not move and the algorithm may not converge. To the best of our knowledge, the approach proposed here is the first attempt to study the impact of the energy state of charge on the feasibility and efficiency of shape-shifting algorithms. The proposed method lays on a new hybrid approach that uses an artificial neural network to ensure the energy efficiency of the reconfiguration.

In the remainder of the paper, we start with an overview of energy awareness in the Self-Reconfiguration Modular Robots literature. Then in section III, we describe the energy model of 2D-Catom robots according to the nature of the motion and the number of produced connections/disconnections. In section IV, we detail the structure of the used Artificial Neural Network, in particular the inputs and outputs formats. Section V describes the experiments and the analysis of the results. Conclusion and future works perspectives are given in Section VI.

II. Related Works

Energy awareness becomes one of the major concerns in ICT fields over the last 10 years. Energy consumption in distributed systems in general and telecommunication networks, in particular, concentrates a tremendous amount of research works [16], [17], [18].

The energy issue has been from a very early stage identified as one of the major metrics in the evaluation of modular robots reconfiguration efficiency [19]. However, earlier works on self-reconfiguration of modular robot systems ignore the energy issue [20], [21]. Later, many recent works deal with the optimization of the energy consumption of the robots [22], [23], [24], [25]. Major works in this field focus on the reduction of the total amount of energy needed to achieve the system's reconfiguration [24], which corresponds to the total number of moves made by all robots. Few works tried to consider more relevant metrics for evaluating energy consumption. In [15], authors proposed a self-reconfiguration algorithm that tries to spread the reconfiguration effort over the robots to reduce the deviation of the number of moves made by each one.

Whatever the modular robots hardware platform, the self-reconfiguration algorithms in the literature are all characterized by the following three drawbacks:

- The energy consumption modeling of the robot's elementary motion is basic and considers that all motions spend the same amount of energy [26]. However, the energy cost of an elementary motion depends on the number of involved robots, i.e. the number of disconnections and connections to achieve, and the nature of the motion (going up, going down, rolling over, sliding under). These parameters are depicted in figure 2, which shows 6-lattice modular robots called 2d-Catoms. For example, the red robot rolls over three robots and then disconnects from robots C and D and reconnects with D (using another connector) and E. Figure 3 gives an exhaustive illustration of the different types of motion.
- Self-reconfiguration algorithm assumes that modular robots are, initially, equally charged. Therefore the self-reconfiguration ignores the disparity between different robots' motion capacities. The battery expiration is assumed as a part of the fault tolerance process and should be managed separately by curative mechanisms.
- The battery depletion is not envisaged during the self-reconfiguration procedure. Indeed the self-reconfiguration algorithm assumes that robots are continuously powered by any external continuous current such as wire [27], harvesting system [28], or power transfer between robots [29]. Otherwise, we assume that the reconfiguration process is interrupted until the modular robots are recharged.

This paper presents, to the best of our knowledge, the first attempt to take into account the state of charge of the modular robots in the self-reconfiguration procedure. The objective is to provide a centralized pre-computing procedure that determines the most suitable self-reconfiguration algorithm to use according to the initial and final shapes and the initial residual energy of each robot.

III. Energy cost model for 2D-Catom modular robots

2D-Catom robots are still in the prototype and demonstration stage. Therefore, a machine learning process based on real-world experiments on self-reconfiguration scenarios requires a high number of modular robots, excessive time duration, and a high risk of error. A first simulation-based phase is more efficient to train an artificial intelligence in order to determine the best suitable self-reconfiguration algorithm according to the initial and final shapes.

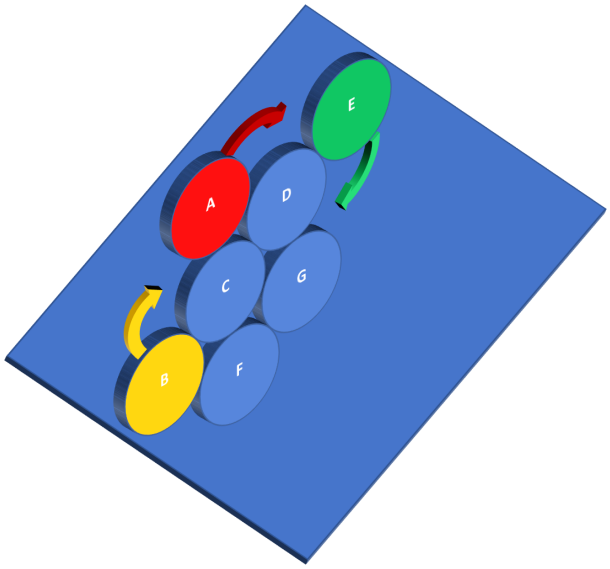

Fig. 2. Birds-eye view on 6-Lattice modular robots (cylinders) deployed over a horizontal plane. The motions of three robots (red, green, and orange) are given by three arrows of a corresponding color. The energy cost of each robot is measured according to the number of disconnections and connections to achieve, and the nature of the motion.

As it is still early in the production of Catoms, the obtained results couldn't be subjected for comparison with real experiments to experiment and verify the algorithm's effectiveness. VisibleSim [30] is a framework for creating behavioral simulators for distributed lattice-based modular robotic systems including 2D-Catoms. VisibleSim was chosen for its fidelity, the simulator is updated frequently by the same teams behind the Catoms [31]. However, for the needs of our study, we need to involve an energy cost model that simulates the energy consumption and the variation of the battery level of the modular robots.

In this section, we describe the energy model we used to take into account energy constraints in modular robots' self-reconfiguration process. We, first, present the data that model the energy constraints then we expose the energy patterns used in our simulations.

A. Energy model

The energy consumption of modular robots depends on the nature of the achieved motion and the number of disconnections and connections done with the neighboring robots. Therefore, a robot may consume energy even when it is still immobile. This is the case when the robot serves as a pivot for the motion of another robot or when the robot disconnects or connects to the moved robot. **We assume that before moving, a robot checks at the same time if its residual energy and the residual energy of involved robots are sufficient.**

To implement that new constraint, we associate the following values to each robot:

- e_i represents the actual battery's residual energy of the robot i.

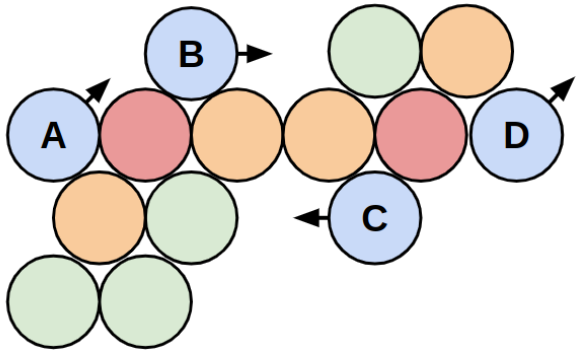

Fig. 3. Energy cost of a move depends on the nature of the move: simple move, climbing, move without support, and climbing without support

- e^{max} represents the maximum energy capacity of robots' batteries. This value is assumed equal for all the robots (robots are homogeneous).
- c^{mr} designates the energy cost required for receiving a message from a neighboring robot.
- c^r defines the energy cost required to perform an elementary move/rotation. An elementary move consists in rolling by 60◦ around another robot called "pivot".
- m_b and m_r define two factors applied to c^r to assess the additional energy cost when the elementary move corresponds to climbing (m_r) or moving without a robot below (m_b) (without support, the robot needs more energy to prevent a fall).
- T defines the time, given in seconds, required to achieve an elementary move.
- ci defines the energy spent when a robot stays in the same position (idle mode) during T seconds without exchanging messages and with only turned-off electrodes (no neighbor).
- c_e represents the cost of turning on or off one single electrode

Figure 3 depicts the four kinds of moves that a robot may perform. Robot B performs a simple move that costs c_r. Robot A climbs which costs $c_r{*}m_r$. Module C moves without support which costs $c_r{*}m_b$. Module D climbs without support and consumes $c_r * m_r * m_b$. Therefore, the energy cost spent by a robot i to perform a move is computed as follows:

$$c_i^{motion} = \begin{cases} 0 & \text{immobile} \\ c^r & \text{simple move} \\ m_r \times c^r & \text{climb} \\ m_b \times c^r & \text{move without support} \\ m_b \times m_r \times c^r & \text{climb without support} \end{cases} \quad (1)$$

In the simulations, we have fixed m_r to 1.5, making climbing consumes 50% more energy than a simple horizontal move. We have set m_b factor to 2.0, which means that moving without below support costs 2 times more than a simple move.

Equation 2 computes the amount of energy spent by a robot during T seconds, C_i^T, according to the robots' mobility, communications, and connections.

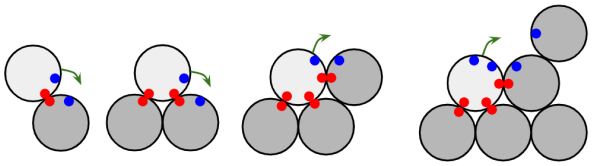

Fig. 4. Number of electrodes used in different situation : (a) : 4 different electrodes used, 2 to activate and 2 to deactivate (b) 6 electrodes (c) 8 electrodes and (d) 10 electrodes

$$C_i^T = ci + c_i^{motion} + l \times c^e + m \times c^{me} + n \times c^{mr} \quad (2)$$

l, m and n represent respectively the number of activated electrodes, the number of sent messages, and the number of received messages during the period T. Figure 4 shows how many electrodes are used for different kinds of moves. In hypothetical case where a robot is isolated, n will equal 0 and C will equal c_i.

We used the data provided in [32] and [33] to set the values of all the variable with realistic values. With this methodology, c_i is worth 286 nanoWatt (nW) and c_e is 533 nW. A robot with a full battery will have 7,4 μWh. The cost for a robot to send a message to a neighbor is negligible when compared to a single move, the orders of magnitude of these two costs are so large (and even larger with the full battery) that we chose to ignore the cost of the messages in the simulation.

B. Energy pattern

We defined 15 different energy patterns to use, pictured in Figure 5. Each pattern describes the cartography of the robots' state of charge before the self-reconfiguration. The initial positions of the robots are modeled by a square grid. The color of a cell depicts the initial state of charge of the robot at this position (the position may be initially empty). The gray level refers to the residual energy level. Darker colors mean exhausted batteries and light points represent the charged batteries. Those patterns are applied to every self-reconfiguration scenario (initial and final shapes). Therefore, the used energy pattern defines the initial values of e_i for every robot.

In Figure 6, we give an example of MRSR problem represented by an image. Black cells represent the positions which do not belong to the initial shape or the final shape. The other cells are colored in RGB mode. Final positions are colored in red ($RGB =< 255, 0, 0 >$), while initial positions are colored by $RGB =< 0, 255, X >$. The blue level, X, depicts the state of charge of the robot located at the associated position. If the position is both initially and finally occupied, the position will take the color $RGB =< 255, 255, X >$, where X is the initial state of charge of the immobile robot.

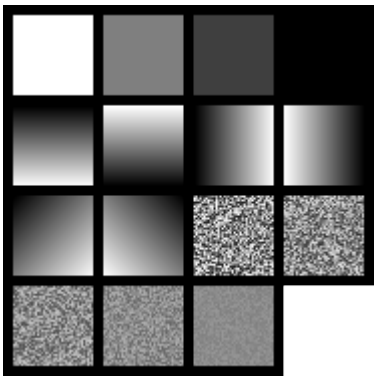

Fig. 5. The 15 energy patterns describing the initial residual energy cartography over the initial positions. A white pixel means a fully charged battery (100%) a black pixel means a dead battery. The patterns are: all robots are fully charged, all robots are mid-charged, all robots are charged at 25%, all robots are charged at 0%, horizontal gradient from top to bottom and from bottom to top, vertical gradient from left to right and from right to left, diagonal gradient from top-left to bottom-right and from top-right to bottom-left, and finally, 5 random patterns with residual energy respectively varying in [10%-100%], [20%-90%], [30%-80%], [40%-70%],[50%-60%].

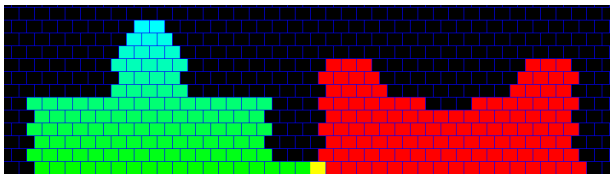

Fig. 6. Example of a scenario, here the initial network has the bump1 shape, with the energy pattern n°6 (gradient, the top robot have the more energy that those on the bottom) into the vase shape.

IV. Neural network

A. Dataset

The scenarios dataset are generated using 12 different classes of shape composed of 120 robots. Figure 7 illustrates these different classes of shape. Using these shapes as initial and final shapes, we obtained 140 different scenarios. For each scenario, we applied the 15 energy patterns described above on the initial shape to obtain 2096 self-reconfiguration scenarios.

B. Convolutional Neural Network

1) Inputs: The objective of the neural network module is to analyze the self-reconfiguration problem in order to determine which distributed algorithm is better. As explained above, the self-reconfiguration problem is modeled by an RGB picture that codes the positions of the initial and the final shape as well as the initial energy of the robots (see Figure 6). Our dataset scenarios represent 2096 different images of $40px \times 30px$, one image by scenario.

2) Structure: In this section, we detail our Neural Network-based modular robot Self-Reconfiguration called CNN. Due to the nature of the modular robot self-reconfiguration problem, it is obvious that geometrical relations between the occupied positions in the initial and the final shapes play a key role in the problem characterization. It

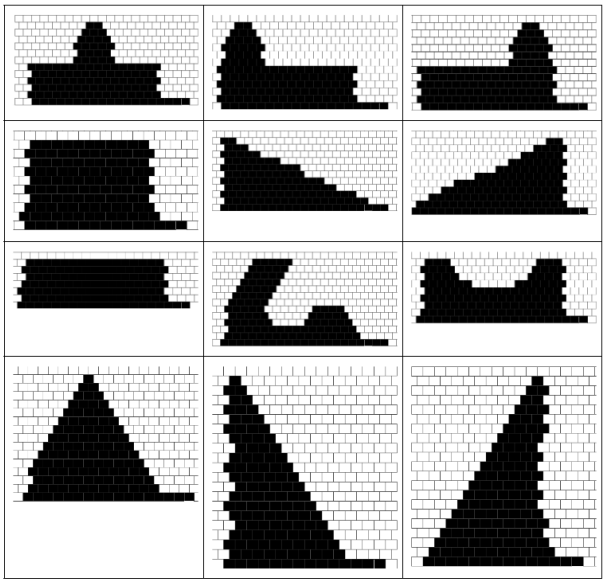

Fig. 7. The 12 shapes defined to be initial and/or goal shape, each composed of 120 robots.

is therefore legitimate to consider a class of ANN adapted to this kind of problem. Convolutional Neural Networks (CNN) are widely used for image and video processing due to their ability to identify the spatial relation between pixels [34], [35]. Therefore, we use a CNN to identify the best MRSR algorithm by analyzing the image that represents to the self-reconfiguration scenario as illustrated in Figure 6.

Figure 8 gives the details of the adopted CNN architecture. Mainly, CNN is based on four cycles of convolution and pooling then 3 layers of classification. The last layer represents the output layer composed of two nodes respectively associated to the TBSR and C2SR scores.

3) Outputs: Energy optimization raises a major question about how to evaluate the energy efficiency of a MRSR algorithm. In this study, we identified 3 criteria to judge the energy efficiency of a self-reconfiguration algorithm. This study represents the first time advanced energy efficiency criteria are used for MRSR problem including an energy model for the different kinds of moves.

1) Sum of the energy used by all the robots for the SR.
2) Max value of energy used by one of the robots for the SR.
3) Ratio of energy used on the energy at the start of the self-reconfiguration.

We conceived three CNN modules according to the considered energy efficiency criterion. The CNN module returns two scores for the respective distributed algorithms according to the given scenario represented by an RGB picture. The distributed algorithm with the highest score is then considered the suitable algorithm.

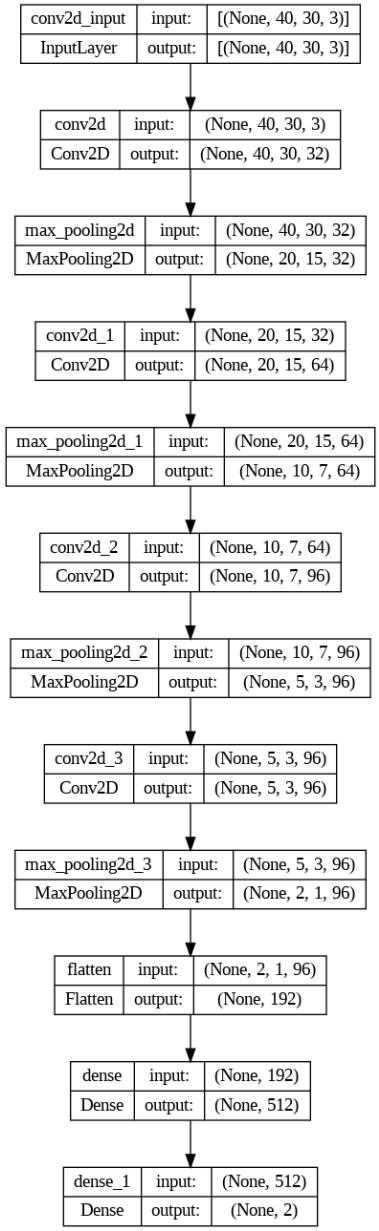

Fig. 8. CNN architecture

V. Experiments and analysis

A. Training step

The training and validation cycles of the CNN module are achieved using respectively 1088 scenarios and 504 scenarios. The last 504 scenarios are used for the test phase and correspond to the results exposed here below.

B. Experimental results

First of all, Figures 9, 10, and 11 show that TBSR outperforms C2SR for all scenarios according to both the maximum energy consumption (MAX) and the maximum ratio of energy consumption criteria (RATIO). However, C2SR performs better in major cases when the total energy consumption criterion

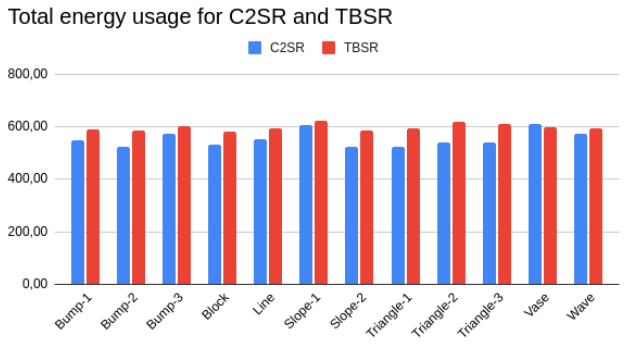

Fig. 9. Total energy used for each goal shape.

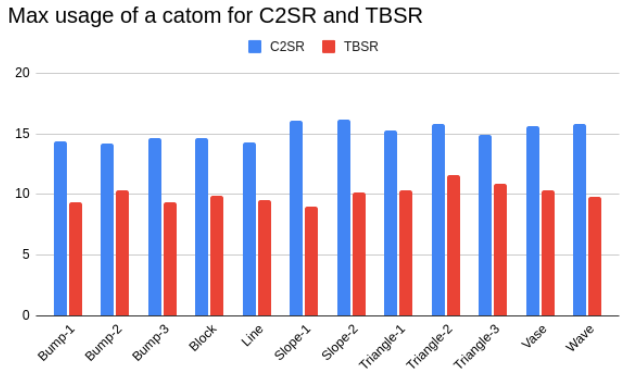

Fig. 10. Max energy used by a robot for each goal shape.

(TOTAL) is considered. This result was predictable since TBSR was initially proposed to guarantee the fair distribution of effort over the robots during the self-reconfiguration.

In figure 12, we show an example of the self-reconfiguration scenario from a Bump shape to a Triangle where all the robots are initially fully-charged. The example shows that TBSR improves energy efficiency according to MAX and RATIO criteria while it decreases energy efficiency

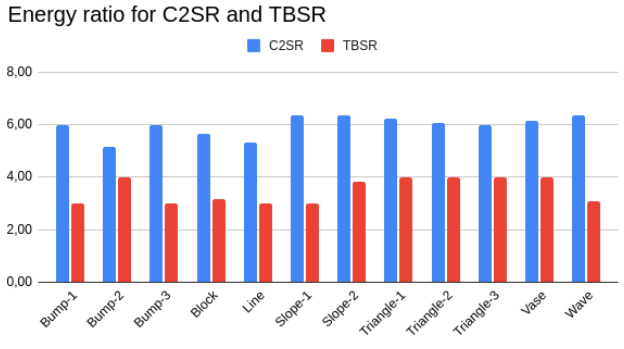

Fig. 11. Ratio of energy used for each goal shape.

Bump1 to Triangle1 with template 100%		
C2SR		
TOTAL	MAX	RATIO
527,30819	17,2343	7
	TBSR	
TOTAL	MAX	RATIO
588,65	10,314	4,00

Fig. 12. Result for a single case : from Bump1 to Triangle1 with template 0 (100% energy).

Average for all 2096 scenarios		
C2SR		
TOTAL	MAX	RATIO
554,04	15,16	5,97
	TBSR	
TOTAL	MAX	RATIO
597,76	10,06	3,51

Fig. 13. Average of results for all scenarios.

regarding the TOTAL metric.

In figure 13, we show the average Total energy used, max energy used and the ratio of energy used of TBSR and C2SR for each criteria. In average, even with the biggest shapes, the robots didn't use a lot of energy.

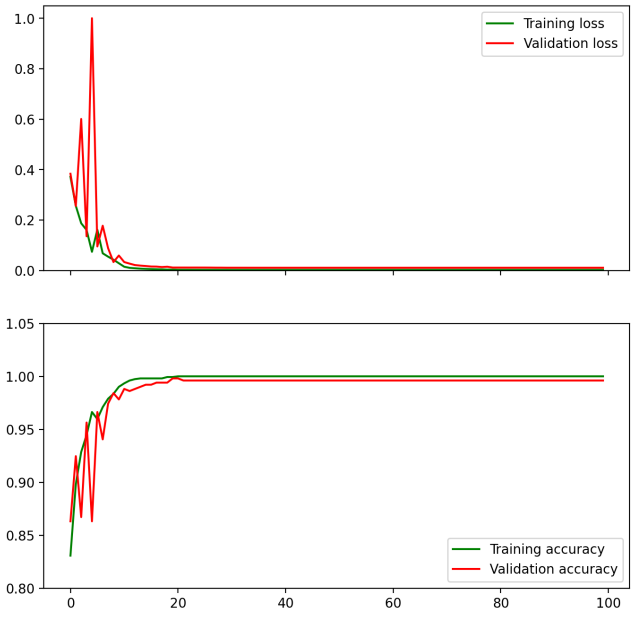

Fig. 14. Training and validation metrics

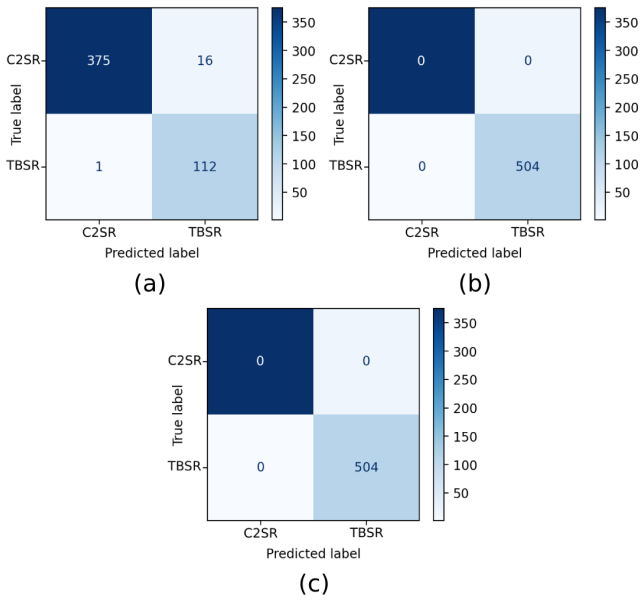

Fig. 15. Confusion matrix for (a) TOTAL (b) MAX and (b) RATIO

C. Analysis

Concerning the performances of the three CNN modules, i.e. the ability of the CNN to identify the truly suitable algorithm, Figure 15 gives the confusion matrices of the conceived neural networks over the test set. The Figure shows that three CNN modules provide respectively 97.88%, 100%, and 100% of correct results according to TOTAL, MAX, and RATIO criteria. The training and validation curves for the accuracy and loss, Figure 14, show that the CNN can quickly generalize the link between the data and the label.

The output for the best suited algorithm can be different for Total energy used, Max energy used and Ratio of energy used. While our tool can be used to find the algorithm for each one of these criteria, when they gave different results, it will be up to the user to choose the best approach to the problem between the three implemented as some situation would require to minimize the total energy usage when other will focus on the ratio of energy used.

VI. Conclusion & Perspectives

Several technological barriers have been overcome in the quest for programmable matter. However, the energy capacity of micro-robots to move has been largely ignored or at least crudely addressed.

In this paper, we present a first attempt at fine modeling of energy consumption of micro-robots according to the nature of the movement. This modeling is integrated into the problem of selecting a self-reconfiguration algorithm according to different criteria for evaluating energy efficiency. The resulting auto-configuration approach represents an advantage over the methods proposed in the literature. Indeed, the prior use of an artificial neural network to determine the best self-reconfiguration algorithm makes it possible to optimize the energy cost of the self-reconfiguration. This optimization leads to prolonging the service life of the robots and thus of the system.

The results obtained show that the CNN is able at rates greater than 95% to determine the best-distributed algorithm to use. Even if in terms of total energy consumed the compared approaches remain identical, the improvement in terms of maximum energy per robot or in energy ratio is significant. Therefore, robots are more able to regenerate their battery through harvesting systems.

References

[1] H. Ahmadzadeh, E. Masehian, and M. Asadpour, "Modular robotic systems: Characteristics and applications," *Journal of Intelligent & Robotic Systems*, vol. 81, pp. 317–357, 2016.

[2] R. Gross, E. Tuci, M. Dorigo, M. Bonani, and F. Mondada, "Object transport by modular robots that self-assemble," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 2558–2564, 2006.

[3] J. Gonzalez-Gomez, J. Gonzalez-Quijano, H. Zhang, and M. Abderrahim, "Toward the sense of touch in snake modular robots for search and rescue operations," in *Proc. ICRA 2010 Workshop "Modular Robots: State of the Art*, pp. 63–68, 2010.

[4] M. Hancher and G. Hornby, "A modular robotic system with applications to space exploration," in *2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06)*, pp. 8 pp.–132, 2006.

[5] A. Romanov, M. Romanov, S. Manko, M. Volkova, W.-Y. Chiu, H.-P. Ma, and K.-Y. Chiu, "Modular reconfigurable robot distributed computing system for tracking multiple objects," *IEEE Systems Journal*, vol. PP, pp. 1–12, 05 2020.

[6] M. Yim, P. White, M. Park, and J. Sastra, *Modular Self-Reconfigurable Robots*, pp. 5618–5631. New York, NY: Springer New York, 2009.

[7] A. Brunete, A. Ranganath, S. Segovia, J. P. de Frutos, M. Hernando, and E. Gambao, "Current trends in reconfigurable modular robots design," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, p. 1729881417710457, 2017.

[8] M. E. Karagozler, *Design, fabrication and characterization of an autonomous, sub-millimeter scale modular robot*. PhD thesis, Carnegie Mellon University, 2012.

[9] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *2012 IEEE International Conference on Robotics and Automation*, pp. 3293–3298, 2012.

[10] J. W. Romanishin, K. Gilpin, and D. Rus, "M-blocks: Momentum-driven, magnetic modular robots," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4288–4295, 2013.

[11] M. E. Karagozler, "Design, Fabrication and Characterization of an Autonomous, Sub-millimeter Scale Modular Robot," 1 2012.

[12] S. Chennareddy, A. Agrawal, and A. K.R., "Modular self-reconfigurable robotic systems: A survey on hardware architectures," *Journal of Robotics*, vol. 2017, pp. 1–19, 03 2017.

[13] F. Witz, B. Buchi, H. Mabed, F. Lassabe, J. Gaber, and W. Abdou, "Deep learning for the selection of the best modular robots self-reconfiguration algorithm," in *2022 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, 2022.

[14] A. Naz, B. Piranda, J. Bourgeois, and S. C. Goldstein, "A distributed self-reconfiguration algorithm for cylindrical lattice-based modular robots," *2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*, pp. 254–263, 2016.

[15] B. Buchi, H. Mabed, F. Lassabe, J. Gaber, and W. Abdou, "Translation based self reconfiguration algorithm for 6-lattice modular robots," in *2021 20th International Symposium on Parallel and Distributed Computing (ISPDC)*, pp. 49–56, 2021.

[16] P. Czarnul, J. Proficz, A. Krzywaniak, *et al.*, "Energy-aware high-performance computing: survey of state-of-the-art tools, techniques, and environments," *Scientific Programming*, vol. 2019, 2019.

[17] O. Kanoun, S. Bradai, S. Khriji, G. Bouattour, D. El Houssaini, M. Ben Ammar, S. Naifar, A. Bouhamed, F. Derbel, and C. Viehweger, "Energy-aware system design for autonomous wireless sensor nodes: A comprehensive review," *Sensors*, vol. 21, no. 2, p. 548, 2021.

[18] A. Merlo, M. Migliardi, and L. Caviglione, "A survey on energy-aware security mechanisms," *Pervasive and Mobile Computing*, vol. 24, pp. 77–90, 2015.

[19] A. Pamecha, I. Ebert-Uphoff, and G. S. Chirikjian, "Useful metrics for modular robot motion planning," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 4, pp. 531–545, 1997.

[20] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-tran: Self-reconfigurable modular robotic system," *IEEE/ASME transactions on mechatronics*, vol. 7, no. 4, pp. 431–441, 2002.

[21] A. Sproewitz, A. Billard, P. Dillenbourg, and A. J. Ijspeert, "Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture," in *2009 IEEE international conference on robotics and automation*, pp. 4259–4264, IEEE, 2009.

[22] L. Furno, M. Blanke, R. Galeazzi, and D. J. Christensen, "Self-reconfiguration of modular underwater robots using an energy heuristic," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6277–6284, IEEE, 2017.

[23] E. Guan, Z. Fu, W. Yan, D. Jiang, and Y. Zhao, "Self-reconfiguration path planning design for m-lattice robot based on genetic algorithm," in *Intelligent Robotics and Applications: 4th International Conference, ICIRA 2011, Aachen, Germany, December 6-8, 2011, Proceedings, Part II 4*, pp. 505–514, Springer, 2011.

[24] H. Lakhlef, J. Bourgeois, H. Mabed, and S. C. Goldstein, "Energy-aware parallel self-reconfiguration for chains microrobot networks," *Journal of Parallel and Distributed Computing*, vol. 75, pp. 67–80, 2015.

[25] A. Majed, H. Harb, A. Nasser, and B. Clement, "Run: a robust cluster-based planning for fast self-reconfigurable modular robotic systems," *Intelligent Service Robotics*, pp. 1–11, 2023.

[26] H. Sun, M. Li, J. Song, and Y. Wang, "Research on self-reconfiguration strategy of modular spherical robot," *International Journal of Advanced Robotic Systems*, vol. 19, no. 2, p. 17298806221081665, 2022.

[27] M. Ohira, R. Chatterjee, T. Kamegawa, and F. Matsuno, "Development of three-legged modular robots and demonstration of collaborative task execution," *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3895–3900, 2007.

[28] S. Dolev, S. Frenkel, M. Rosenblit, R. P. Narayanan, and K. M. Venkateswarlu, "In-vivo energy harvesting nano robots," in *2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE)*, pp. 1–5, 2016.

[29] B. T. Kirby, M. Ashley-Rollman, and S. C. Goldstein, "Blinky blocks: a physical ensemble programming platform," in *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, (New York, NY, USA), pp. 1111–1116, ACM, 2011.

[30] B. Piranda, "Visiblesim: Your simulator for programmable matter," in *Dagstuhl Seminar 16271 (2016)*, vol. 6, (Wadern, Germany), pp. 12 – 12, jul 2016.

[31] "Programmable matter."

[32] Y. Peng, G. Carichner, Y. Kim, L.-Y. Chen, R. Tribhout, B. Piranda, J. Bourgeois, D. Blaauw, and D. Sylvester, "A 286nw, 103v high voltage generator and multiplexer for electrostatic actuation in programmable matter," in *IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits) (VLSI 2022)* (cf2b24b7-0569-4bc5-bbc3-24a72233f0d5 editor.pdf, ed.), no. 21903283, (Honolulu, United States), pp. 158 – 159, jul 2022.

[33] M. E. Karagozler, "Design, Fabrication and Characterization of an Autonomous, Sub-millimeter Scale Modular Robot," 1 2012.

[34] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental concepts of convolutional neural network," in *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, pp. 519–567, Springer, 2020.

[35] R. Haridas and J. L, "Convolutional neural networks: A comprehensive survey," *International Journal of Applied Engineering Research*, vol. 14, p. 780, 02 2019.