Integrated Modeling and Planning for On-Orbit Assembly of Large Space Structures with Mobile Crawling Robots

Alexandre Albore
DTIS, ONERA, Université de Toulouse
F-31055 Toulouse, France
alexandre.albore@onera.fr

Mathieu Rognant
DTIS, ONERA, Université de Toulouse
F-31055 Toulouse, France
mathieu.rognant@onera.fr

ABSTRACT

Large-scale space structures have taken center stage as the future of space exploration, paving the way for ambitious endeavors like sprawling solar power stations, intricate telescopes, and giant space stations. This program lies not in manual assembly, but in empowering space robots to autonomously build these structures through plans provided by dedicated on-orbit assembly planning algorithms.

In this paper we describe an approach to large-scale space structures assembly frameworks, while keeping in mind the many benefits associated to the usage of autonomous crawling mobile robots.

We thus describe an assembly approach and model on which we are developing our research. Initially, all the building elements (beams and nodes) are stored in the payload fairing. These are deployed bit by bit building the structure on which the robots themselves will evolve until reaching the final configuration. We represent the assembling problem as an automated planning instance, where several structural constraints dictate the actions available for execution. The automated planner algorithm provides the steps required to achieve the final, deployed structure.

We implemented a first version of the planner that provides an assembly plan, i.e. a sequence of actions to move the elements from the payload fairing to the deployed structure.

This model helps to illustrate the feasibility of the approach and the benefits of having AI tools for autonomous assembling robots in space.

KEYWORDS

On-orbit assembly, automated planning, crawling robot

ACM Reference Format:
Alexandre Albore and Mathieu Rognant. 2024. Integrated Modeling and Planning for On-Orbit Assembly of Large Space Structures with Mobile Crawling Robots. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 6 pages.

1 INTRODUCTION

Deployable techniques are commonly accepted by the engineering community as the conventional technique for fielding space systems. However, their complexity grows with the size of the structures and is limited by rocket capacity. It is widely accepted that, as the size of space structures increases, the complexity of deployments will become too great and the risk of failure will increase [2, 5, 17]. On-orbit assembly offers a more efficient launch process, a safer assembly process, and the added benefit of reusing autonomous mobile space robot systems for repairs, expansion, inspection, and maintenance of large space structures. This approach enables the construction of truly massive structures. [17, 34].

In comparison with deployable techniques, on-orbit assembly using robots that crawl along the truss offers a number of advantages in terms of resilience and adaptability. Crawling robots can navigate around unexpected obstacles and environmental changes, which is a significant improvement over the rigid folding sequence required by foldable structures [21]. Their modular design allows collaborative work on different structure sections simultaneously, increasing efficiency and providing redundancy in the event of robot failure.

Autonomous mobile robots are well-suited for on-orbit repair and maintenance tasks due to their mobility and intelligence. They can access and repair areas inaccessible to astronauts or fixed tools, reducing human risk and dependency. Equipped with sensors and computing capabilities, robots can diagnose problems, plan repairs and learn from experience, minimizing reliance on ground control. Furthermore, the capability to carry out repairs swiftly and effectively minimizes the downtime of critical space infrastructure[1].

Efficient assembly planning should consider a number of factors: payload design and robotic system should be integrated as a single, complete system because they are interdependent [21]. Virtual reality technology allows people to interact with the assembly process virtually, allowing for more efficient planning of the assembly process [23, 26], but this approach is more suitable for small structures where the plan is synthesized by human expert operators. Auction algorithms based on consensus bundle algorithms are used to solve complex on-orbit assembly task assignments, and local consensus algorithms are proposed for dynamic tasks requiring real-time reassignments [32].

Several approaches have been proposed for automated planning [12] in space assembly. For large-scale assembly, automated planning is used to coordinate multiple, heterogeneous robots.[29].

Path planning for fleets of assembling robots has been considered by works on “Equilibrium Shaping” [13], a method to produce a swarm in which each agent is preassigned to a particular place in the final formation, and where a a global swarm behavior is achieved without robot communication, using neighbor sensing to avoid collisions.

Foust et al., proposed a decentralized auction algorithm with a trajectory planner, implementing model predictive control using sequential convex programming [6]. This distributed guidance and control scheme aims at combining a heterogeneous swarm of component satellites into a large satellite structure.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.

Rodríguez et al. [27] generate assembly and reconfiguration plans for modular structures by calling a motion planner using a symbolic representation of preconditions and logical constraints in PDDL. This results in a sequence of high-level actions that can be autonomously executed by a robot. The planner uses breadth-first search to find an optimal sequence of actions, even if it uses heuristics to select preferred paths, and limits the branching factor in order to speed up the search. In this architecture, as in the present work, each robot is an autonomous agent consisting of a planning layer that determines how to achieve high-level goals.

CPGA – Continuous path generation algorithm – is a variant of the branch-and-bound algorithm and the modified ant colony algorithm that has been used in [28] for the on-orbit assembly path planning process. The major objective of CPGA is to converge rapidly to a local minimum solution of the problem to reduce the calculation cost and optimization time.

Martínez-Moritz et al. [20] use an automated task planner to plan the pick up, move and place operations for mirror tiles in a large spatial structure. The state space is searched using a variant of the depth-first search algorithm that prunes dead-end states. Execution is then monitored to ensure that certain physical constraints are met (e.g., that the trajectory avoids collisions or exceeds the physical limits of the equipment).

We adopt an automated planning approach that generates an ordered sequence of actions for the robots to build the final structure. In order to both avoid dead-end states, and to perform an efficacious search, we use a heuristic search algorithm in the space of the structure configurations. The planner software includes connectivity constraints, communication between elements, and reachability of components throughout the assembly process.

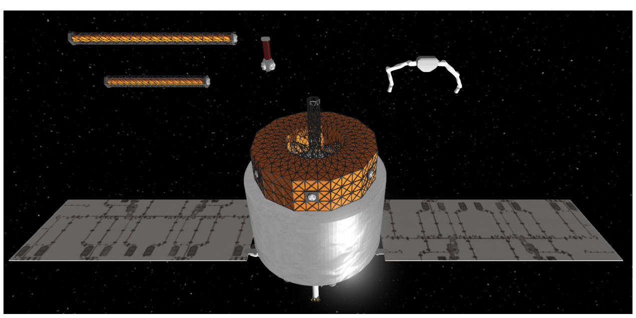

Figure 1: System components: beams, node, crawling robot, spacecraft

2 USE CASE

On-orbit assembly of large structures holds relevance for future missions such as solar power stations, orbital stations, or very large antennas for communication needs. The robotization of these assembly operations becomes all the more accessible as the technological maturity of standard interfaces advances [18, 21]. These interfaces enable couple mechanically as well as transmit electrical power, data and thermal control [33]. This technology has been considered in particular for the assembly of a large optical surface composed of identical tiles using crawling robots [4]. In this paper, we propose to study the prospects offered by crawling robots for the assembly of large structures composed of heterogeneous elements equipped of these interfaces.

To evaluate our approach, and the planning algorithm described in section 4, we considered the recent on-orbit demonstration DOLCE conducted by Caltech University [10]. The DOLCE (Development of the Deployable on-Orbit ultraLight Composite Experiment) is a project that aims to advance technology for space solar power. It focuses on developing lightweight composite materials for deployable structures in space. The ultimate goal is to enable more efficient and cost-effective deployment of solar power systems in orbit. The designed deployed structures collect solar energy and directly convert it in-situ into microwaves for transmission to the ground. This feature simplifies constraints on the support structure and, in particular, the level of electrical power to be transmitted via standard interfaces. We consider the assembly of a support structure that connects a group of nodes equipped with DOLCE modules in a hexagonal pattern. By scaling the current demonstrators, we can estimate that the volume required to hold an active triangular equilateral solar panel of two meters would be a cylinder with a diameter of 0.25 m and a height of 0.55 m. Adding three Hotdock connectors [16] to the base of this module to connect it to the structure, increases the total height of one node to 0.7 m. The resulting model and its components are shown in Figure 1.

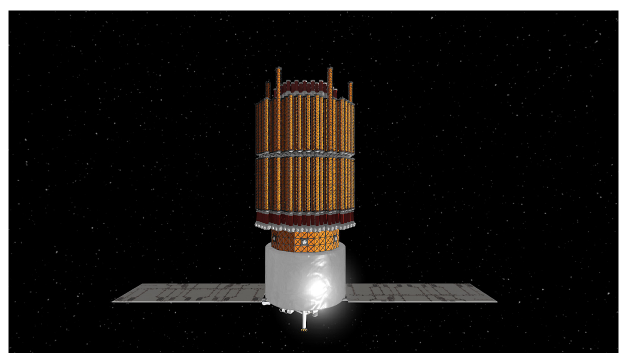

Figure 2: Use case scenario in stacked configuration

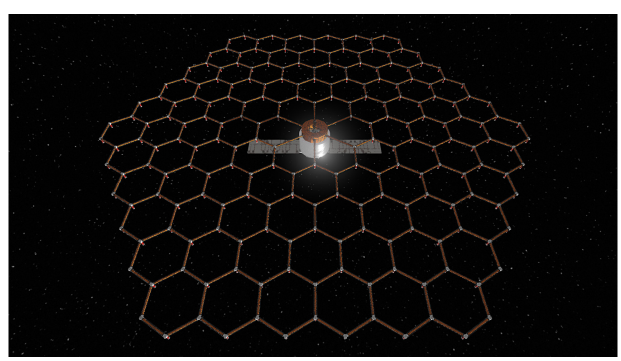

Figure 3: Use case scenario in deployed configuration

To connect these nodes to each other and to the supporting satellite, composite lattice beams fitted with Hotdock connectors at their ends can be used. In this study, two lengths of beams are required: 3 m beams for the first nodes connected to the satellite and 2 m beams between each node. The diameter of the beams is estimated at 0.25 m.

For the crawling robots, a design similar to the MIRROR project [4] can be used, i.e. a torso equipped with two arms with seven degrees of freedom. The end of each arm is equipped with grippers to allow locomotion at any point of the structure, while the torso can carry either a beam or a node deposited by one of the arms (cf. Figure 6). Finally, for the satellite, we chose the same space-bus as the one used in the H2020 PULSAR project [15], with the payload replaced by a truss structure fitted with 6 Hotdocks.

With an arrangement of 2 layers of nodes and 2 layers of beams in a triangular pattern (cf. Figure 2), an assembly of 300 beams and 210 nodes can be placed under the fairing of the Ariane 6 launcher, while complying with mass and volume constraints (see Ariane 6 User's Manual). This arrangement constitutes an initial operating point that can be improved by optimizing the mass of the beams and using a dedicated knapsack problem solver. When deployed, these elements give a total surface area of 42 m in diameter. The deployed structure is depicted in Figure 3.

3 SYSTEM MODELING

The structure described in the previous section, made of different interconnected elements (nodes and beams), and being assembled by crawling robots, is then modeled as a graph on which planning agents evolve, performing assembly plans.

3.1 Structure modeling

Lattice structures are an effective solution for creating large-scale infrastructures with a good mass-rigidity ratio. The modularity [14] of this type of structure also facilitates reconfiguration operations during maintenance, making them highly suitable for robotic assembly. These structures are made by assembling a set of beams according to a specific geometry that provides rigidity.

The proposed concept involves planning tasks for an assembly consisting of truss beams with standard interfaces at their ends and attachment nodes also equipped with standard interfaces, supporting a payload such as a solar panel or deployable antenna. This paper only considers flat structures for deployment, but the concept can be easily extended to three dimensions.

The configuration of the state of the system during the assembly operation can be described in the form of a graph, where the nodes represent the centers of mass (CoM) of the bodies and the edges the mechanical links that connect them. Figure 4 shows the graph of a simple structure arranged in a hexagonal lattice. The blue and orange points represent respectively the nodes associated with the CoM of the beams and the fixing nodes. The green and red points represent the nodes associated with the CoM of the standard interfaces (red active interfaces, green passive interfaces). The gray segments correspond to the edges representing the rigid fasteners, and the green segments the removable fasteners between the connectors.

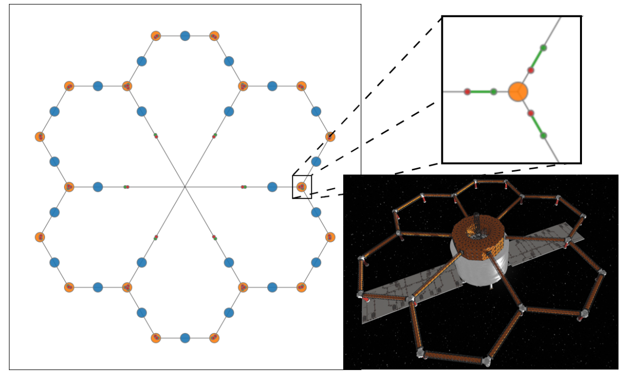

Figure 4: Graph of a deployed planar hexagonal structure

This graph representation also applies to the initial geometric configuration of the structural elements. Figure 5 shows the arrangement of the structural elements prior to deployment, taking into account the space constraints and mass distribution under the launcher fairing. To optimize the storage of these elements, we have chosen a triangular pattern, and a layered distribution.

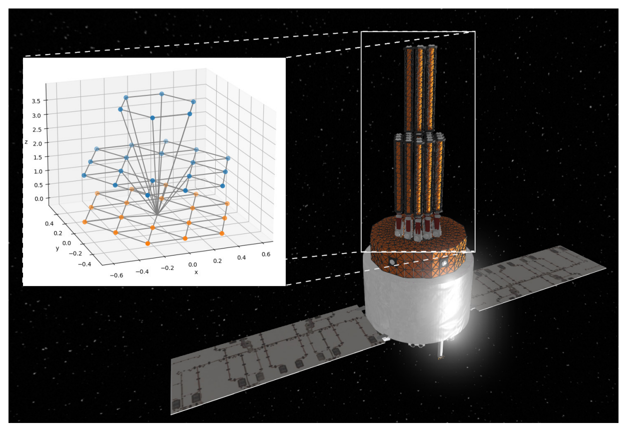

Figure 5: Stacked configuration in a triangular pattern of the structure in Figure 4

3.2 Agent

Crawling robots are systems that adhere to or grasp the structure and move around. Here we assume that the robot does not need any special element to anchor itself to the structure, similar to the Skyworker developed at Carnegie Mellon University [31]. This concept enables small robotic systems to move structural elements within a large work volume. However, the progress of the structural assembly process directly affects the range of possible actions. Elements that are movable and the truss beams along which the agents can crawl evolve dynamically depending on the structure being assembled. Based on the current state of the system, it is then necessary to determine the paths that agents can take and the elements that can be manipulated, in order to identify possible actions.

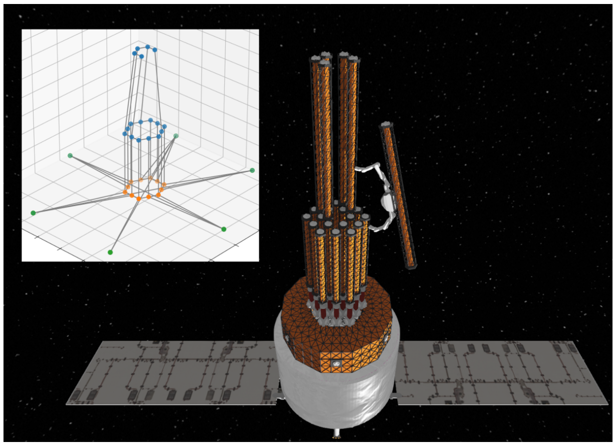

Figure 6: Graph of the paths and reachable positions for a crawling robot moving a beam from the top layer

In Figure 6 is shown the graph of the paths going trough the reachable elements of the payload. Green points represent the reachable positions, the edges of the graph are the paths that the crawling robot can take.

4 NOMINAL ASSEMBLY PLAN COMPUTATION

Automated planning (sometimes called AI planning) [12] is a field of artificial intelligence concerned with automatically finding a sequence of steps that can achieve a specific goal in a particular environment. It involves using algorithms to search through possible sequences of actions to find one that leads to the desired outcome.

Automated planning has been used in several robotics scenarios, including space exploration [9, 17, 34] and logistics industrial robotics [3, 19], mainly representing an effective approach to autonomous behavior, integrating action selection and sensors responses [11].

4.1 Planning problem modeling and solving

A planning agent uses the model of the environment and the system to reason about possible courses of action and how they will affect the state of the world.

In this work we embrace this approach to autonomous behavior. We propose a domain-specific deterministic planner for assembling large-scale space structures. Each state is a graph representing the lattice structure, labeling nodes with their capacity to host new connections, their structural characteristics, and their distance from the center of the structure. From this representation, we can also represent the available paths and reachable elements. Depending on which element is deemed reachable, structural constraints and applicable actions are dynamically updated during the search for a solution plan, since adding an element to the structure may allow to move to a new position and thus reach other nodes. Removing an element may result in certain paths being blocked.

In our implementation of the search algorithm, we adopt a lazy representation of the search nodes, trading memory for time overhead. We keep in memory the sequence of actions π leading to the current state s, and we build the correspondent graph of the structure by simply progressing the root search node s_0 through the plan π. A search node thus represents a state s and corresponds to the sequence of actions π from the initial state s_0 to s. This means that states are not computed explicitly, but represented implicitly in the plan prefix π.

Algorithm 1 Search Algorithm

```
1:  function Search(start, goal)
2:  openList ← priority queue containing start with priority 0
3:  closedList ← empty map
4:  h_init ← heuristic(start)
5:  while openList is not empty do
6:      current ← openList.pop()
7:      cost ← closedList[current]
8:      if cost < current.planCost then
9:          continue
10:     end if
11:     if current is goal then
12:         return extract_solution(current)
13:     end if
14:     operators ← generateOperators(current)
15:     for each op in operators do
16:         child ← op.apply(current)
17:         if (child ∉ closedList or
                child.planCost < closedList[child]) then
18:             closedList[child] ← current
19:             Add child to openList with priority being the plan cost
20:         end if
21:     end for
22: end while
23: return Failure: No plan found
24:
25: function extract_solution(current)
26: plan ← empty list
27: while current do
28:     plan ← current.action
29:     current ← current.parent
30:     plan.reverse()
31: end while
32: return plan
```

The planner performs a best first search based on states ordered by a priority value correspondent to their plan cost value, calculated by adding the cost of reaching the current state from the initial situations plus the heuristic value for reaching the goal situation. The heuristic is calculated as the sum of the distances of the stacked elements from the base of the structure, considering the beams that can be moved, plus the distance needed to reach the remaining target deployed positions. This approximation simplifies the process of shortest path computation in the graph structure, which would require performing an A* search or a Dijkstra algorithm for each applicable action. At equivalent plan costs, nodes with lower heuristic value are preferred.

Algorithm 1 shows the A* search performed. It follows the usual algorithm, with a couple of variations. After initializing the open and closed lists, at line 6 the node with the lowest priority value is taken from the open list. We only expand the node if its associated

Instance	# Beams	# Nodes	Time (s)	Plan length
Small	3	3	0.02	6
Medium	24	18	5.8	42
Large	66	48	160	114
Huge	300	210	16.000	510

Table 1: Planning time and assembly plan length in increasing size instances of the large-scale assembly problem

value is the lowest cost known for this state (lines 8-9). Otherwise, it means that we've already found a cheaper path after creating this node (line 18) and hence can disregard it. If the state corresponding to the current node is the desired goal situation, the solution plan is returned at line 12 by calling the function `extract_solution`. At line 14 we perform a delayed generation of the applicable operators, because this planning problem has a very high branching factor: we then prefer to evaluate the list of applicable actions, and the successor states, only when necessary. Function `generateOperators` does exactly that: it checks which are the movable beams and nodes, which paths are available to the mobile agents, and generates the list of applicable operators for the *current* state. We expand child nodes when the state has not been seen before, or when we find a cheaper path to the child state than before (line 17).

Returned plans are a sequence of elements displacements, from the initial stacked position in the nose cone to the final deployed configuration. These plans are intended to be generated off-line, with the assembly plan uploaded while on orbit.

These plans are intended to be generated off-line, with the assembly plan uploaded while on orbit. Even though, due to the elevate branching factor of the problem (between 10^3 and 10^4 for the big instances), sub-optimal search can be challenging.

Table 1 below shows the results for our planner on benchmarks of increasing size, in order to evaluate the order of magnitude of the instances being solved by a single agent. Tests have been run on a Quad-Core Intel Xeon CPU @ 3.60GHz with 16GB RAM. The length of the plans corresponds to the total number of elements to be displaced, and hence, to the size of the structure. It is clear from these results that elaborating plans for medium or large structures is still feasible from the time point of view, and the memory point of view. Bigger instances terminate in *time out* (with a time limit set to 20.000 s), but the size of the nodes in the open list requires also additional memory in these cases.

Said that, bigger instances of more than 800-900 elements are unrealistic because the number of elements would not fit stacked in the nose cone of the current launchers, e.g. Ariane 6, and multiple rockets deliver payloads would be assembled sequentially.

In our architecture, we contemplate generating solutions seeing each robot as an autonomous agent, executing an assembly plan. Splitting a "Huge" instance of the problem among six "Large" planning agent instances is totally scalable. However, in this first development phase, we implemented this solution with the agents having secluded activities, each being able to reach a predetermined portion of the structure. This separation of the robot activities is sub-optimal, and the extension of this approach to a decentralized architecture featuring collaborative robots is discussed below in the Conclusion section.

5 CONCLUSION

We have presented an approach to model large-scale space structures, and to generate an on-orbit assembly plan, focusing on the usage of AI planning for autonomous mobile crawling robots. The modeling is specific for assembly made by mobile robots, as the structure description, and the dynamic selection of a feasible path along the traversable truss beams are bound to this peculiar approach. A domain-dependent AI planner greedily searches the state space to synthesize an assembly plan, respecting the structural constraints during the search.

The preliminary results of this research seem to indicate scaling-up issues for the largest instances of the satellite structure only, due to the lazy state representation adopted, which requires repeatedly applying back-and-forth sequences of actions from the search root node. However, these issues are only relative as 1) the assembly plan can be generated offline without any issue, and 2) splitting the problem between independent autonomous mobile agents would drastically reduce the overhead.

On the search algorithm side, possible improvements could be achieved by refinement of the state representation, using more accurate heuristics for action selection (e.g. counting the number of actions necessary to reach, disconnect from its neighbors, and move an element), or assessing the plan quality by measuring execution time and actual distance traveled by the mobile agent.

On the model side, future work is to differentiate reachable elements that can only support crawling robots but not be movable to a new position. Connecting constraints, depending on the heterogeneity of the nodes and beams, e.g. male/female interfaces, are used for determining the initial graph, but are not translated into assembling constraints for the final structure.

Distributing the assembly task among independent autonomous mobile robots is the approach that aims at having robots working as a team [24, 29]. Multi-agent tasks can greatly benefit from intelligent cooperation between team agents and can achieve performance close to the theoretical optimum [8]. Thus, to perform large-scale assembly tasks, the robots must coordinate their actions. This would be efficient in the structure modeling described in this paper, as the displacement of elements by one agent would create movement support for the other agents, and taking advantage of redundancy would speed up assembly or maintenance tasks.

This research direction is supported by recent advances in agent task allocation in the context of collaborative heterogeneous robots, where a decentralized approach simultaneously allocates and decomposes high-level tasks among various agents [22]. Coordination of assembly agents allows for resource sharing, rather than the separation of individual workloads, and may support specific actions, such as jointly disconnecting elements to make them movable.

Other physical constraints will limit the total maneuver time of the crawling robots, which is dependent on the path followed, the motion of the payload, and the motion of the robots themselves. Excessive velocity of the robot CoM can affect the attitude stability of the satellite. Such constraints imply the adoption of a temporal planning paradigm [7, 25, 30] to express action durations, temporal constraints between agents, and time synchronization.

ACKNOWLEDGMENTS

This work has been funded by ONERA PRF ASTROLABE.

REFERENCES

[1] David L Akin and Mary L Bowden. 2002. EVA, robotic, and cooperative assembly of large space structures. In *Proceedings, IEEE Aerospace Conference*, Vol. 7. IEEE.

[2] Jeremy Banik. 2016. Realizing large structures in space. In *Proc., Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2015 Symp.* 55–62.

[3] Stefan-Octavian Bezrucav and Burkhard Corves. 2022. Modelling automated planning problems for teams of mobile manipulators in a generic industrial scenario. *Applied Sciences* 12, 5 (2022), 2319.

[4] Mathieu Deremetz, Maxence Debroise, Shashank Govindaraj, Alexandru But, Irene Sanz Nieto, Marco De Stefano, Hrishik Mishra, Bernhard Brunner, Gerhard Grunwald, Maximo Roa, et al. 2022. Demonstrator design of a modular multi-arm robot for on-orbit large telescope assembly. In *16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2022)*. ESTEC Noordwijk, The Netherlands.

[5] John Dorsey, William Doggett, Robert Hafley, Erik Komendera, Nikolaus Correll, and Bruce King. 2012. An efficient and versatile means for assembling and manufacturing systems in space. In *AIAA SPACE 2012 Conference & Exposition*. 5115.

[6] Rebecca C Foust, E Sorina Lupu, Yashwanth K Nakka, Soon-Jo Chung, and Fred Y Hadaegh. 2020. Autonomous in-orbit satellite assembly from a modular heterogeneous swarm. *Acta Astronautica* 169 (2020), 191–205.

[7] Maria Fox and Derek Long. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research* 20 (2003), 61–124.

[8] Roee M Francos and Alfred M Bruckstein. 2023. On the role and opportunities in teamwork design for advanced multi-robot search systems. *Frontiers in Robotics and AI* 10 (2023).

[9] Yang Gao and Steve Chien. 2017. Review on space robotics: Toward top-level science through space exploration. *Science Robotics* 2, 7 (2017), eaan5074.

[10] Eleftherios Gdoutos, Charles F Sommer, Alan Truong, Alexander Wen, Antonio Pedivellano, Kanthasamy Ubamanyu, Richard G Madonna, and Sergio Pellegrino. 2022. Development of the Deployable on-Orbit ultraLight Composite Experiment (DOLCE) for the Space Solar Power Project Demonstration Mission. In *AIAA SCITECH 2022 Forum*. San Diego, CA, 1266.

[11] Hector Geffner and Blai Bonet. 2022. *A concise introduction to models and methods for automated planning*. Springer Nature.

[12] Malik Ghallab, Dana Nau, and Paolo Traverso. 2016. *Automated planning and acting*. Cambridge University Press.

[13] Dario Izzo and Lorenzo Pettazzi. 2005. Equilibrium shaping: distributed motion planning for satellite swarm. In *Proc. 8th Intern. Symp. on Artificial Intelligence, Robotics and Automation in space*, Vol. 25.

[14] Benjamin Jenett, Christine Gregg, Daniel Cellucci, and Kenneth Cheung. 2017. Design of multifunctional hierarchical space structures. In *2017 IEEE Aerospace Conference*. IEEE, 1–10.

[15] Sofiane Kraïem, Mathieu Rognant, Jean-Marc Biannic, and Yves Brière. 2023. Dynamics and robust control of a space manipulator with flexible appendages for on-orbit servicing. *CEAS Space Journal* 15, 5 (2023), 681–700.

[16] Pierre Letier, Torsten Siedel, Mathieu Deremetz, Edgars Pavlovskis, Benoit Lietaer, Korbinian Nottensteiner, Máximo Alejandro Roa Garzon, Juan Sánchez Garcia, Javier Luis Corella, and Jeremi Gancet. 2020. Hotdock: Design and validation of a new generation of standard robotic interface for on-orbit servicing. In *71st International Astronautical Congress, IAC 2020*. IAF.

[17] Delun Li, Lou Zhong, Wei Zhu, Zhipeng Xu, Qirong Tang, and Wenhao Zhan. 2022. A survey of space robotic technologies for on-orbit assembly. *Space: Science & Technology* (2022).

[18] Marc Manz, Sebastian Bartsch, Romain Caujolle, Torsten Vogel, Mark Shielton, Elie Allouis, Stefan Gornig, Francisco Javier Colmenero, Sebastian Torralbo, Marko Jankovic, et al. 2022. Robotic architecture and operational concept for in-space assembly and servicing missions. In *16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2022)*. ESTEC Noordwijk, The Netherlands, 1–2.

[19] Andrea Marrella. 2019. Automated planning for business process management. *Journal on data semantics* 8, 2 (2019), 79–98.

[20] Juan Martínez-Moritz, Ismael Rodríguez, Korbinian Nottensteiner, Jean-Pascal Lutze, Peter Lehner, and Máximo A. Roa. 2021. Hybrid Planning System for In-Space Robotic Assembly of Telescopes using Segmented Mirror Tiles. In *2021 IEEE Aerospace Conference (50100)*. 1–16. https://doi.org/10.1109/AERO50100.2021.9438399

[21] Katherine McBryan. 2020. Comparison between stationary and crawling multi-arm robotics for in-space assembly. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1849–1856.

[22] Antoine Milot, Estelle Chauveau, Simon Lacroix, and Charles Lesire. 2021. Market-based Multi-robot coordination with HTN planning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2606–2612.

[23] P O'B Holt, James M Ritchie, Philip N Day, John EL Simmons, Graham Robinson, George T Russell, and FM Ng. 2004. Immersive virtual reality in cable and pipe routing: design metaphors and cognitive ergonomics. *J. Comput. Inf. Sci. Eng.* 4, 3 (2004), 161–170.

[24] Lynne E Parker. 1998. ALLIANCE: An architecture for fault tolerant multirobot cooperation. *IEEE transactions on robotics and automation* 14, 2 (1998), 220–240.

[25] Damien Pellier, Alexandre Albore, Humbert Fiorino, and Rafael Bailon-Ruiz. 2023. HDDL 2.1: Towards Defining a Formalism and a Semantics for Temporal HTN Planning. In *Proceedings of the International Workshop of Hierarchical Planning (ICAPS)*. Prague.

[26] James M Ritchie, Richard G Dewar, and John EL Simmons. 1999. The generation and practical use of plans for manual assembly using immersive virtual reality. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 213, 5 (1999), 461–474.

[27] Ismael Rodríguez, Adrian S Bauer, Korbinian Nottensteiner, Daniel Leidner, Gerhard Grunwald, and Máximo A Roa. 2021. Autonomous robot planning system for in-space assembly of reconfigurable structures. In *2021 IEEE Aerospace Conference (50100)*. IEEE, 1–17.

[28] Yuchen She, Shuang Li, Bin Du, and Kai Cao. 2018. On-orbit assembly mission planning considering topological constraint and attitude disturbance. *Acta Astronautica* 152 (2018), 692–704.

[29] Reid Simmons, Sanjiv Singh, David Hershberger, Josue Ramos, and Trey Smith. 2001. First results in the coordination of heterogeneous robots for large-scale assembly. In *Experimental Robotics*, Vol. VII. Springer, 323–332.

[30] David E. Smith, Jeremy Frank, and William Cushing. 2008. The ANML Language. In *ICAPS Workshop on Knowledge Engineering for Planning and Scheduling*.

[31] Peter J Staritz, Sarjoun Skaff, Chris Urmson, and William Whittaker. 2001. Skyworker: a robot for assembly, inspection and maintenance of large scale orbital facilities. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, Vol. 4. IEEE, 4180–4185.

[32] Xiaoqiang Yu, Jifeng Guo, and Hongxing Zheng. 2019. Extended-cbba-based task allocation algorithm for on-orbit assembly spacecraft. In *2019 IEEE International Conference on Unmanned Systems (ICUS)*. IEEE, 883–888.

[33] Mehmed Yüksel, Marko Jankovic, Wiebke Brinkmann, and Christian Schoo. 2022. A methodology for electro mechanical evaluation of multi functional interconnects for on-orbit servicing demonstration. In *16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2022), ESTEC Noordwijk, The Netherlands*.

[34] Enyang Zhang, Huayang Sai, Yanhui Li, Xiangyang Sun, Tao Zhang, and Zhenbang Xu. 2023. Modular robotic manipulator and ground assembly system for on-orbit assembly of space telescopes. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* (2023).