Logistics Management for Parking Multiple Cask Plug and Remote Handling Systems in ITER

Rodrigo Ventura[b,*], João Ferreira[a], Iulian Filip[c], Alberto Vale[a]

[a] *Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade Técnica de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal*

[b] *Laboratório de Robótica e Sistemas em Engenharia e Ciência, Instituto Superior Técnico, Universidade Técnica de Lisboa, Av. Rovisco Pais 1, 1049-001, Lisboa, Portugal*

[c] *Faculty of Mechanical Engineering - Technical University Gheorghe Asachi of Iasi, 61 Dimitrie Mangeron Bldv, Iasi - 700050, Romania*

Abstract

During operation, all maintenance inside the reactor building at ITER (International Thermonuclear Experimental Reactor) has to be performed by remote handling, due to the presence of activated materials. Maintenance operations involve the transportation and storage of large, heavyweight casks from and to the tokomak. The transportation is carried out by autonomous vehicles that lift and move beneath these casks. The storage of these casks face several challenges, since (1) the cask storage area is limited in space, and (2) all casks have to be accessible for transportation by the vehicles. In particular, casks in the storage area may block other casks, so that the former has to be moved to a temporary position to give way to the latter. This paper addresses the challenge of managing the logistics of cask storage. In particular, we propose an approach to (1) determine the best position of the casks inside the storage area, and to (2) obtain the sequence of operations required to retrieve and store an arbitrary cask from/to a given storage place. A combinatorial optimization approach is used to obtain solutions to both these problems. Simulation results illustrate the application of the proposed method to a simple scenario.

Keywords: ITER, Remote Handling, Cask and Plug Remote Handling System, Logistics, Storage Location Assignment

1. Introduction

The ITER (International Thermonuclear Experimental Reactor) is a joint international research project, aiming at the demonstration of the technological feasibility of fusion power as an alternative and safe power source. The Cask and Plug Remote Handling System (CPRHS) provides the means for the remote transfer of (clean/activated/contaminated) in-vessel components and remote handling equipment between the Hot Cell Building (HCB) and the vacuum vessel in Tokamak Building (TB) through dedicated galleries, as illustrated in Figure 1.

There are different CPRHS configurations, each defined according to the required activity. The largest CPRHS has dimensions 8.5m x 2.62m x 3.62m (length, width, height) and is entrusted with the transportation of heavy (total weight up to 100 tons) and highly activated components [4]. The CPRHS comprises three sub-systems: a cask envelope containing the load, a pallet that supports the cask envelope and the Cask Transfer System (CTS). The CTS acts as a mobile robot, provides the mobility for the CPRHS and can be decoupled from the entire system. The kinematic configuration, first proposed in [9], endows it with the required flexibility to navigate autonomously or remotely controlled, in the cluttered environments of the TB and the HCB.

*Corresponding author. Tel.:+351 218 418 195; Fax:+351 218 418 291.

Email addresses: `rodrigo.ventura@isr.ist.utl.pt` (Rodrigo Ventura), `jftferreira@ipfn.ist.utl.pt` (João Ferreira), `ifilip@gmail.com` (Iulian Filip), `avale@ipfn.ist.utl.pt` (Alberto Vale)

During the reactor's operation, the in-vessel components, such as the blankets that cover the vacuum vessel, are expected to become activated. When such components have to be removed for disposal, operations are to be carried out by the CPRHS, which is required to dock in pre-defined locations, the

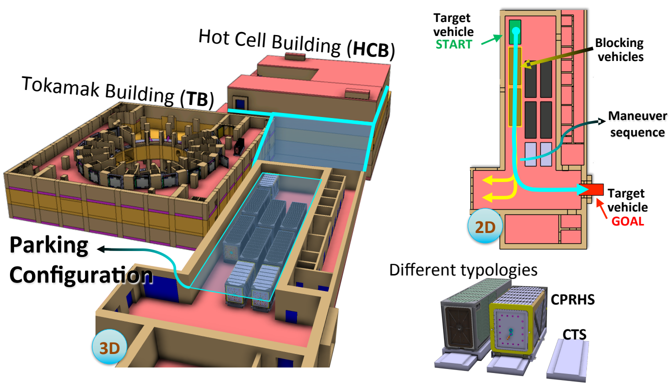


Figure 1: Illustration of the ITER scenario: from left to right, 3D view of the environment, showing in particular the Hot Cell Building (HCB), the Tokomak Building (TB), and the parking area; 2D view of the cask stored in this area; the different typologies of the Cask and Plug Remote Handling System (CPRHS), together with the Cask Transfer System (CTS).

vacuum vessel port cells, located on the three levels of TB: divertor, equatorial and upper level. Then, the components are transported to the HCB for operations of diagnose and refurbishment or disposal of activated material. Hence, the CPRHS must dock at the docking stations through port plugs interfaces or park in parking areas at the different levels of the HCB.

This paper addresses the problem of managing the parking area for the CPRHS, hereby designated simply by *casks*. We assume the designation of specific parking areas. However, given the space constraints in ITER, casks cannot be arbitrarily positioned, since when packing the casks along the available space, casks may block each other. This raises two challenges, that are addressed in this paper: first, since parked casks may block each other, how to retrieve on such blocked casks, and second, since different casks may have differing usage patterns, how to determine the location of each cask, within the parking area, such that the most used ones are more easily retrieved. In other words, the first problem can be framed as a planning problem, where blocking casks are moved to temporary positions, in order for the CTS to have access to the target cask. The second problem concerns the optimal arrangement of casks within the parking area, taking into account their usage frequency.

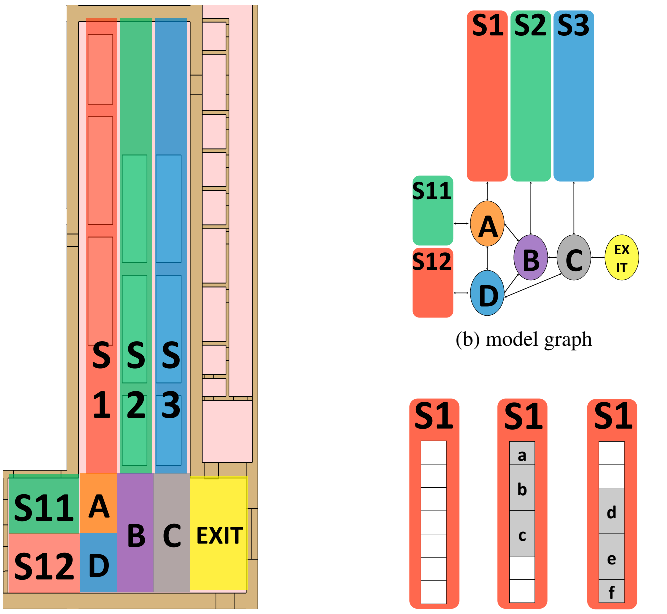
The problem of determining the best arrangement of items within a storage area is generically designated as *storage location assignment problem* in the Operational Research literature [1, 5]. Early work has focused on the research of various assignment policies [6]. More recent research has focused on class-based storage location assignment, employing branch and bound methods [8].

This paper is organized as follows: Section 2 presents the formal problem statement, followed by the proposed solution in Section 3. Experimental results illustrating the approach are presented in Section 4, followed by the conclusions and future work in Section 5.

2. Problem Statement

To formulate the problem we start by making a few simplifying assumptions about the scenario. Let us designate *operational area* the space available for both storing and moving casks around.

Our first assumption is that this operational space can be modeled by a graph $\mathcal{K} = \langle \mathcal{X}, \mathcal{E} \rangle$, where $\mathcal{X}$ is a set of nodes, representing physical locations in the environment, and $\mathcal{E}$ is a set of edges, each one denoting a feasible direct navigation path for the CTS between the corresponding pair of nodes (see Figure 2 for an example, [2, 3]). Three types of nodes are considered: *stacks* are nodes that allow the storage of one or more casks in line along their length, *crossings* are transit nodes where a CTS can navigate among two arbitrary adjacent nodes, and a special node denoted *exit point*, representing the exit of the storage area. Thus, the set of nodes is partitioned into a disjoint union of three subsets, $\mathcal{X} = \mathcal{S} \cup \mathcal{C} \cup \{\varepsilon\}$, where $\mathcal{S}$ and $\mathcal{C}$ are the sets of stacks and crossings, while ε is the exit point. In the example illustrated in Figure 2, $\mathcal{S} = \{\mathrm{S1}, \ldots, \mathrm{S3}, \mathrm{S11}, \mathrm{S12}\}$, $\mathcal{C} = \{\mathrm{A}, \ldots, \mathrm{C}\}$, and $\varepsilon = \mathrm{D}$. Each stack $s \in \mathcal{S}$ is modeled as an ordered set of N_s cells of fixed size (e.g., Figure 2(c)), say



(a) storage area (b) model graph (c) example occupancy

Figure 2: Graph model of an example scenario: (a) storage area divided in cells (S1 to S5 are stacks and A to D are transit areas; (b) corresponding model graph, where rounded rectangles denote stacks and circles denote crossing nodes; (c) examples of occupancy of a stack of 7 free slots. Casks a and f have a size of 1 slot, while casks b–e have a size of 2.

$s = [1 \cdots N_s]$. Without loss of generality, the leftmost cell 1 of a stack s is the stack exit, *i.e.,* the side where casks enter or leave the stack.

Given two arbitrary stack nodes that are connected in the graph, we assume that there is a feasible trajectory allowing a cast to navigate among these stacks. This trajectory consists in the concatenation of the edges connecting the two stack nodes.

Let $\mathcal{A}$ denote a set of casks. The length of each cask is assumed to be an integer multiple of the cell size. This multiple is denoted L_a, for $a \in \mathcal{A}$.

As its name suggests, storage to and retrieval from a stack of a cask is performed in a Last-In-First-Out (LIFO) fashion. In other words, we have that: (i) only the cask closer to the stack exit is accessible for transportation, and (ii) provided that there is enough free cells adjacent to the stack exit, a newly stored cask will occupy some of these free cells (being the amount equal to the cask size).

Given this model, together with the stated cask movement constraints, we can formulate the logistics problem in the following way:

1. the problem of storing (or retrieving) casks, corresponding to moving a single cask $a \in \mathcal{A}$ from (or to) the exit point ε to (or from) a storage position k within a stack $s \in \mathcal{S}$;
2. the problem of determining, given a cask $a \in \mathcal{A}$, the best stack s and position k within the stack, to park the cask, provided its usage rate.

The second problem arises since different locations of a given cask inside the storage area imply different amounts of operations in order to retrieve it, depending on whether this cask is blocked by others.

Formulating the problem this way allows the application of standard combinatorial search methods. The followed approach is detailed in the next section.

3. Proposed solution

Having the problem stated as above, solutions can be found using standard combinatorial search methods. Let us first discuss the approach to the problem of finding the best sequence of operations to move casks from or to the storage area, followed by the optimal cask storage position problem.

3.1. Cask movement planning

Given a cask a, its location l_a can be either in a stack or in a transit node. Thus, the set of possible locations is the disjoint union between crossing nodes and pairs (stack, cell), *i.e.,* $l_a \in C \cup \{(s,k) \mid s \in \mathcal{S}, k \in \{1\ldots,N_s\}\}$. The locations of all casks $\{l_a\}$ form a state where heuristic search can be used to obtain optimal solutions: given an initial and a goal state, the solution corresponds to the sequence of operations that, when applied in sequence, lead from the initial state to the goal. Given a state, the set of operators considered corresponds to all feasible actions that can be performed to a single cask. In particular, for any cask a, if it is in a stack s, it can be moved to any adjacent position, including adjacent free cells and, if it is located in cell 1, the crossing node x for which there is an edge (s,x). Otherwise, if it is in a crossing node x, it can move to either an adjacent node x' for which there is a node (x,x'), or the cell 1 in case there is an edge (x,s) to a stack s and it is free.

Note that the concept of free cell has to defined taking into account the length of each cask: a cell k in stack s is *free* with respect to a cask a if, and only if, the set of cells occupied by all other casks is disjoint to the set of cells occupied by a when $l_a = (s,k)$. Formally, this means that $\{k,\ldots,k+L_a\}\cap\{i,\ldots,i+L_b\} = \emptyset$ for all casks $b \in \mathcal{A}$ such that $l_b = (s,i)$.

To solve the problem of taking a given cask a from storage to exit ε, the proposed solution employs IDA* to obtain the optimal solution. IDA* is a memory efficient version of the well-known A* algorithm, while maintaining the optimality property of the solution [7]. The state space comprises the locations of all casks. At each state there is a designated active cask, which is initially set to a. Search proceeds by recursively expanding successor states. These successors consist of all possible movements of the active cask to a neighbor position (either an adjacent stack cell or graph node). If a neighbor is occupied by a different cask, a subgoal is created in which the active cask is set to the blocking cask and the search tree branches over all free positions among all stacks.

This method allows one to compute the optimal sequence of steps to get an arbitrary cask out of the storage area. Thus, given one configuration of the storage area, we can associate to each cask a cost, corresponding to the effort necessary to transport it to the exit. The cost reflects the total amount of steps involved in the solution, including a penalty for having to move blocking casks to temporary positions.

3.2. Cask storage location determination

Given one assignment of casks to stacks and positions, the method described above allows the computation of the cost of moving each individual cask to the exit, including the cost of moving all other blocking casks to temporary positions. All possible cask arrangements within the parking area form a state space. A given state can be modeled as a function f from the casks to locations within the stacks,

$$\begin{array}{cccc} f & : & \mathcal{A} & \longrightarrow & \mathcal{S}\times\mathbb{N} \\ & & a & \longmapsto & (s,k) \end{array} \tag{1}$$

where s and k are the stack and position within the stack for a given cask a.

We model cask usage as a rate, equivalently (appart from a scale factor) to a probability of usage. This allows us to construct a global cost functional $J(f)$, set to the weighted average of the cask retrieval cost, where the weights are the usage probabilities,

$$J(f) = \sum_i p_i\, c_i(f) \tag{2}$$

where, for a cask i, $p_i \in [0;1]$ is the usage probability and $c_i(f)$ is the cost of moving this cask to the exit, given the arrangement defined by f. This is equivalent to the expected value of the retrieval cost.

We proceed by determining, employing combinatorial search, what is the cask arrangement within the parking area, such that the global cost function is minimized. A simple branch-and-bound algorithm is used to obtain this global minimum.

4. Experimental Results

To evaluate the approach, the algorithms described in the previous section were implemented. The test scenario used was the one depicted in Figure 2, together with three casks, designated 1 to 3.

For an example initial configuration, Figure 3 displays the sequence of moves necessary to take cask 2 from the stack, to the exit node. Note that, since cask 2 is blocked by cask 1, the latter has to be moved to a temporary position before the former can be retrieved from the stack.

Concerning the optimal cask placement within the parking area, the optimal configuration found is shown in Figure 4.

5. Conclusions and Future Work

This paper presented a method to address two problems in logistics, concerning the management of casks within the ITER: (i) determining the best sequence of operations to retrieve a single cask from the storage area, possibly unblocking the path by moving other casks to temporary positions, and (ii) determining the best storage locations for casks, such that the expected cost of retrieval is minimized. This retrieval cost depends on the cask and on its cost to move it to the exit (including getting

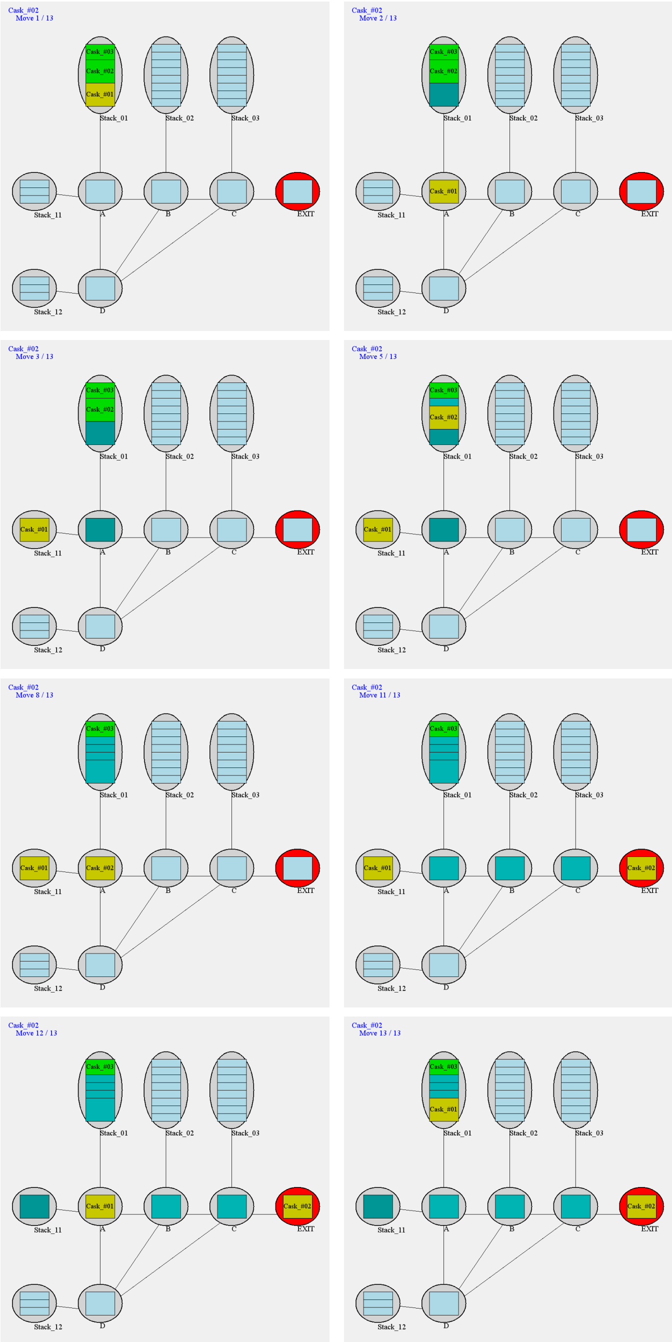


Figure 3: Results for the cask movement planning problem, showing some intermediate steps in the sequence of operations to move cask 2 from stack 1 to the exit (leftmost node).

blocking casks out of the way). Expectation is computed using usage rate as probabilities of picking up casks for retrieval.

A simple proof-of-concept implementation was devised, and results for a simplified ITER-like scenario are presented. However, due to combinatorial nature of the problem, the temporal complexity grows exponentially. Future work will address this issue along two possible ways: (i) more efficient optimization methods, exploiting the structure of the problem, and (ii) approximation methods that, while not guaranteeing optimality,

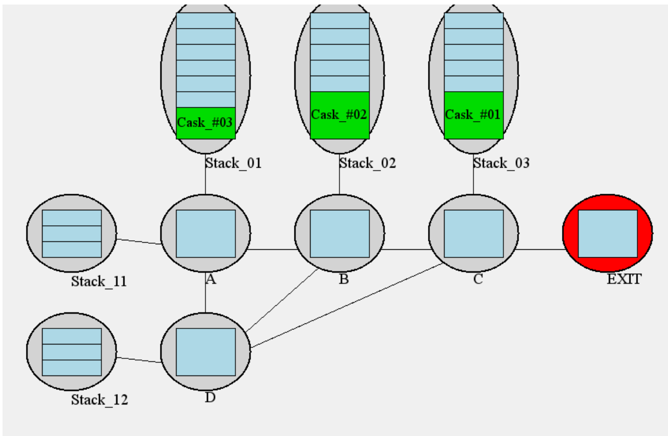


Figure 4: Optimal solution for the cask configuration problem, given the graph shown in Figure 2 and three casks.

may provide satisfiable solutions within a reasonable execution time.

Acknowledgments

The work was supported by FCT in the frame of the Contract of Associate Laboratories of Instituto de Plasmas e Fusão Nuclear/IST (Pest-OE/SADG/LA0010/2011) and Laboratório de Robótica e Sistemas em Engenharia e Ciências/IST. The views expressed in this publication are the sole responsibility of the authors. F4E is not liable for the use which might be made of the information in this publication.

References

[1] J.P. van den Berg, W.H.M. Zijm, “Models for warehouse management: Classification and examples”, Int. J. Production Economics 59, pp. 519–528, 1999.

[2] D. Fonte, F. Valente, A. Vale and I. Ribeiro, ”A Motion Planning Methodology for Rhombic-like Vehicles for ITER Remote Handling Operations”, Proc. of the 7th IFAC Symp. on Intellig. Autonomous Vehicles, Italy, 2010.

[3] D. Fonte, F. Valente, A. Vale, I. Ribeiro, ”Path Optimization of Rhombic-Like Vehicles: An Approach Based on Rigid Body Dynamic”, Proc. of the 15th IEEE Int. Conf. on Advanced Robotics, 106-111, Estonia, June 2011.

[4] C. Gutiérrez, C. Damiani, M. Irving, J. Friconneau, A. Tesini, I. Ribeiro and A. Vale, ”ITER Transfer Cask System: status of design, issues and future developments”, Proc. of the 9th Int. Symp. on Fusion Nuclear Energy, China, 2009.

[5] J. Gu, M. Goetschalckx, L. F. McGinnis, “Research on warehouse operation: A comprehensive review”, European Journal of Operational Research 177, pp. 1–21, 2007.

[6] W. H. Hausman, L. B. Schwarz, S. C. Graves, “Optimal Storage Assignment in Automatic Warehousing Sistems”, Management Science 22(6), pp. 629–638, 1976.

[7] R. E. Korf, “Depth-First Iterative-Deepening: an Optimal Admissible Tree Search”, Artificial Intelligence 27, pp. 97–109, 1985.

[8] V. R. Muppani, G. K. Adil, “A branch and bound algorithm for class based storage location assignment”, European Journal of Operational Research 189, pp. 492–507, 2008.

[9] I. Ribeiro, P. Lima, R. Aparício and R. Ferreira, ”Conceptual Study on Flexible Guidance and Navigation for ITER Remote Handling Transport Casks”, Proc. of the 17th IEEE/NPSS Symp. on Fusion Engineering, pp. 969–972, San Diego, USA, 1997.