IJCAI-17
MELBOURNE

August 19-25, 2017

# HUMAN-ROBOT ENGAGEMENT

## in the Home, Workplace and Public Spaces

## ORGANIZING COMMITTEE

Mary-Anne Williams, University of Technology Sydney and CodeX Stanford University

Benjamin Johnston, University of Technology Sydney and d.school Stanford University

William Judge, Commonwealth Bank of Australia

Amit Kumar Pandey, Softbank Robotics Europe

Meg Tonkin, University of Technology Sydney

Xun Wang, University of Technology Sydney and Commonwealth Bank of Australia

Jonathan Vitale, University of Technology Sydney

UTS CommonwealthBank SoftBank Robotics

# PROGRAM

8:30 Introduction: Mary-Anne Williams (UTS) and William Judge (CBA)

9:00 Invited Speaker: Holly Yanco, University of Massachusetts Lowell USA

10:00 Morning Tea

10:30 Research Paper Presentations and Discussion

Hanan Rosemarin, John P. Dickerson and Sarit Kraus
*Learning to Schedule Deadline- and Operator-Sensitive Tasks*

Frank Dignum, Virginia Dignum and Catholijn M. Jonker
*Ethical Considerations by Personal Assistants for Life*

Ashley Kelso
*Managing Legal Risks from the Societal Integration of Robots*

Elisa Cucco, Michael Fisher, Louise Dennis, Clare Dixon, Matt Webster, Bastian Broecker, Richard Williams, Joe Collenette, Katie Atkinson and Karl Tuyls
*Towards Robots for Social Engagement*

Xiaoyu Ge, Jochen Renz, Nichola Abdo, Wolfram Burgard, Christian Dornhege, Matthew Stephenson and Peng Zhang
*Stable and Robust: Stacking Objects using Stability Reasoning*

11:30 Invited Speaker: Amit Pandey, Softbank Robotics

12:30 Lunch

14:00 Learning from Human-Human Engagement - Peter and Sohie Sardi

15:00 Ideation and Prototyping Session: designing for human-robot engagement

16:00 Afternoon Tea

16:30 Team Presentations

17:00 Workshop Close

# CONTRIBUTED RESEARCH PAPERS

# Learning to Schedule Deadline- and Operator-Sensitive Tasks [*]

**Rosemarin, Hanan** and **Dickerson, John P.** and **Kraus, Sarit**

## Abstract

The use of semi-autonomous and autonomous robotic assistants to aid in care of the elderly is expected to ease the burden on human caretakers, with small-stage testing already occurring in a variety of countries. Yet, it is likely that these robots will need to request human assistance via teleoperation when domain expertise is needed for a specific task. As deployment of robotic assistants moves to scale, mapping these requests for human aid to the teleoperators themselves will be a difficult online optimization problem. In this paper, we design a system that allocates requests to a limited number of teleoperators, each with different specialities, in an online fashion. We generalize a recent model of online job scheduling with a worst-case competitive-ratio bound to our setting. Next, we design a scalable machine-learning-based teleoperator-aware task scheduling algorithm and show, experimentally, that it performs well when compared to an omniscient optimal scheduling algorithm.

## 1 Introduction

Deploying semi-autonomous and autonomous robotic assistants to aid in caring for the elderly is expected to ease the burden on human caretakers. In Japan, for example, the Health, Labor, and Welfare Ministry predicts a shortfall of $380{,}000$ nursing and elderly care workers by 2025, with similar projected imbalances between supply and demand in other developed nations; thus, this problem is timely [Kaneko *et al.*, 2008]. Indeed, robotic helpers have already been deployed in small-stage testing in a variety of countries, including Japan, Italy, and Sweden [Leiber, 2016].

Yet, it is likely that these robots will need to request human assistance—for example, for teleoperation—from time to time. Beyond healthcare, automobile manufacturer Nissan recently announced its plan to augment autonomous vehicle technology with a crew of on-call, remote human "mobility managers" [Nowak, 2017]. As deployment of semi-autonomous robots moves to scale, mapping these requests for human expertise to the teleoperators themselves will be a difficult online optimization problem.

This paper presents a framework for the online allocation of requests to a limited number of *specialized* teleoperators, each of whom have different levels of expertise for types of requests. We generalize a recent state-of-the-art online scheduling algorithm [Lucier *et al.*, 2013] to our setting and test its performance relative to an omniscient offline algorithm. We draw on work in the information retrieval literature to present a novel machine-learning-based method for matching the best job to a specific server at a specific time. We show experimentally that this algorithm performs quite well, beating an adaptation of the closest prior state-of-the-art online scheduling algorithm.

### 1.1 Related Work

Our problem can be seen as a type of job scheduling, which is a classical problem in computer science and operations research. In our case, the users' tasks are the jobs and the teleoperators are the machines or servers. We believe our motivation—that of assigning human teleoperators with specific skills to tasks—pushes us to address a novel version of this problem. We briefly overview recent related work at the current research horizon in this space and detail how our work is different; we direct readers interested in a complete history of job scheduling to work by Pinedo [2015].

Zheng and Shroff [2016] work in a setting where jobs arrive online, and give some partial value for partial execution. Doucette *et al.* [2016] address assigning jobs to agents in an online fashion, and also with preemption of previously allocated jobs in a distributed setting. Neither address jobs' preference for specific servers (as we will, where a job completed on a preferred servers yields greater utility), nor servers' heterogeneous completion rate for a job type. Most related to our work, Lucier *et al.* [2013] look at online allocation of batch jobs with deadlines to identical servers; we generalize their model to a setting with heterogeneous servers and where the jobs have preferences over servers.

From a learning theory point of view, some recent work takes a regret-minimization approach to online job scheduling [Even-Dar *et al.*, 2009]; however, that work is motivated by allocating users/connections to different links via a load

[*] This research was supported in part by a grant from the Ministry of Science Technology, Israel the Japan Science and Technology Agency (jst), Japan.

balancer and assumes that no knowledge of the job's runtime is known ahead of time (as in our case). Rather, the job's runtime is known once it is assigned to a handler. From an applied machine learning point of view, job scheduling with a classification component has recently gained attention [Tripathy *et al.*, 2015; Panda *et al.*, 2015]; most of this work focuses on offline scheduling of jobs with dependencies and deadlines, while we focus on online scheduling of independent jobs. Gombolay *et al.* [2016] take a reinforcement learning approach to the apprenticeship problem, that is, learning human-quality heuristics; they do this by way of a pairwise ranking function, as we do, but their setting is not online.

From the operations management point of view, Pérez *et al.* [2013] focus on the nuclear medicine application area, and take a two-stage stochastic IP approach to scheduling patients that arrive with multi-step tests, e.g., a patient arrives with three tests that have to be performed sequentially, but an individual job cannot be paused once it has started. In their model, once a patient's jobs are scheduled (in the future), they cannot be changed, a constraint we do not have. Anderson [2014] provides state-of-the-art techniques for scheduling residents in hospitals under various constraints; we direct the reader to his work for an in-depth survey of such approaches. We note that our proposed model would be useful in a setting such as scheduling residents to hospitals, and can be seen as addressing a version of that problem.

### 1.2 Our Contributions

This paper presents a machine-learning-based approach to a novel generalization of a classical problem in computer science and operations research. Motivated by the increasing presence of semi-autonomous robots that need to "call out" to human teleoperators, we address the online job scheduling problem where jobs have preferences over which server (teleoperator) completes them, and teleoperators have varying skill levels for completing specific classes of jobs. We extend a recent model of online job scheduling to this setting, give a competitive ratio for a simple generalization of an algorithm in that space, and then present a sophisticated machine-learning-based approach to scheduling jobs. We draw on intuition from the information retrieval literature to learn a ranking function of jobs for servers. We validate our approach in simulaton and show that it outperforms a generalization of the state-of-the-art algorithm for our setting.

## 2 A Model for Scheduling Jobs with Preferences to Heterogeneous Servers

In this section, we formalize our model. It generalizes a recent model due to Lucier *et al.* [2013].

### 2.1 Our Model

Lucier *et al.* [2013] work in a setting where jobs $j \in \mathcal{J}$ arrive online at time $a_j$ with a deadline $d_j$ indicating the last time period at which a job can be completed, and a processing time $p_j$ indicating a base level of resource consumption. Upon completion, jobs yield a value $v_j$. Their model assumes all servers are identical; we will change this later.

They provide an online algorithm for this setting that aims to maximize the total value of completed jobs, and prove a lower bound (worst-case competitive ratio) on the performance of the proposed online scheduling algorithm, by ordering the jobs according to their *value-density*–for a job $j$, defined to be $\rho_j = \frac{v_j}{p_j}$, the ratio of value to processing time. They allow scheduling to occur only when a new job arrives or when a job completes execution. Additionally, server-affinity is assumed; that is, when a task is scheduled to a specific server it will not "migrate" to another server, even when the job is preempted and other servers are idle.

Their scheduling algorithm also relies on three concepts, which we will also use in our generalization of that model. For a given job $j \in \mathcal{J}$, let the $s_j = \frac{d_j - a_j}{p_j}$ be the minimum *slack* necessary for a task to be accepted, which is the ratio of the available time for the task to its processing time. This is compared against a global slack parameter $s$, a hyper-parameter to any scheduling algorithm. Similarly, let $W_j^{-\mu}$ be the time interval $\{a_j, \ldots, d_j - \mu p_j\}$ and $A^{-\mu}(t) = \{j \in \mathcal{J} \mid t \in W^{-\mu}\}$ the set of jobs at time $t$ with a remaining execution window of $\mu$ times the processing time $p_j$. Finally, define a preemption threshold $\gamma$; a job $j_2$ will preempt another job $j_1$ only if the ratio of their value-densities is greater than $\gamma$, i.e., $\rho_{j_2} > \gamma \rho_{j_1}$.

The principles of attaining value only from fully completed jobs and continuing execution on a single server fit well with the requirements of our use cases, including teleoperators assisting elderly patients, or humans assisting semi-autonomous vehicles. However, we note that in our setting, not all servers (teleoperators) are equally skilled. That is, a registered nurse may be quite skilled at helping a geriatric human perform a life task, but less skilled at teleoperating a car through a snowstorm. Furthermore, it may be the case that a geriatric human would get greater value from interacting with the registered nurse than with the incliment-weather-trained driver. Thus, we extend the model of Lucier *et al.* [2013] with the notion of non-identical servers and job preferences, by adding the following attributes:

1. We categorize jobs into discrete *types* $\tau$.
2. Each server $i$ has a scalar *efficiency* $\eta_\tau^i \in (0, 1]$ for each job type $\tau$. The efficiency accounts for the varying proficiency of the servers for the different types of jobs, and modifies the actual execution time of a job of type $\tau$ according to its original processing time, such that $p'_j = \frac{p_j}{\eta_\tau^i}$.
3. Each job $j$ expresses a scalar *preference* for each server $i$, defined as $\psi_j^i \in (0, 1]$. This preference modifies the value gained by completion of the job, $v'_j = \psi_j^i v_j$.

Table 1 summarizes the notation that we use from Lucier *et al.* [2013], as well as the notation we introduced to create our new model.

### 2.2 A Simple Scheduling Algorithm

Given this generalized model, how should we allocate arriving jobs to servers? Similarly, if a job completes on a server, which queued job should be allocated to that newly-idle server? In Section 3, we present a sophisticated machine-learning-based approach to answer these questions; however, first, we generalize a recent state-of-the-art scheduling algorithm, again due to Lucier *et al.* [2013], to our model.

| Symbol | Description |
|---|---|
| $a_j$ | arrival time |
| $d_j$ | job completion deadline |
| $p_j$ | nominal processing time |
| $v_j$ | value received upon job completion |
| $\rho_j$ | value-density, ratio of $v_j$ to $p_j$ |
| $s_j$ | slack of a job |
| $s$ | global slack parameter |
| $W_j^{-\mu}$ | time interval $[a_j, d_j - \mu p_j]$ |
| $A^{-\mu}(t)$ | the set of jobs $j$ at time $t$ with availability at least $\mu$ times $p_j$ |
| $\gamma$ | preemption threshold between jobs |
| $\tau_j$ | job type |
| $\eta_\tau^i$ | efficiency of server $i$ for job type $\tau$ |
| $\psi_j^i$ | preference of job $j$ for a server $i$ |

Table 1: Notation.

First, for any job $j$ and server $i$, define the *server-dependent* value-density $\dot{\rho}_j^i = \rho_j \psi_j^i \eta_\tau^i$, where $\tau$ is the type of job $j$. This is a straightforward adaptation of the value-density metric to the case of heterogeneous servers (via the $\eta_\tau^i$ multiplier) and job preference over servers (via the $\psi_j^i$ multiplier). We then adapt the scheduling algorithm of Lucier *et al.* [2013] to account for the varying nature of the servers by using the server-dependent value-density, and by comparing that value-density difference between the value-density of a candidate job on a specific server and the value-density of running job on that server (zero for idle servers) when making a preemption decision. That algorithm, for multiple servers, is given below as Algorithm 1.

**Algorithm 1** Adapted Online Job Scheduling Algorithm

```
Event Type 1: A job $j$ arrived at time $t = a_j$.
  1. calculate delta value-density ($\rho$) for servers:
     $\forall i \in servers, \Delta\rho_i^j = \rho_i^j - \rho_i$
  2. choose server with highest change to value-density
     $i = \arg\max_i \Delta\rho_i^j$
  3. call the threshold preemption rule (i,t)
Event Type 2: A job $j$ completes on server $i$ at time $t$.
  1. Resume execution of the preempted job $j$ with highest
     server-dependent value-density $\dot{\rho}_j^i$ among any job preempted on $i$
  2. Call the threshold preemption rule below with server $i$
     and time $t$
Threshold Preemption Rule ($i$, $t$):
  1. Let $j$ be the job currently being processed on server $i$
  2. Let $j^* = \arg\max_j \{\dot{\rho}_{j^*}^i \mid j^* \in A^{-\mu}(t)\}$
  3. If $(\dot{\rho}_{j^*}^i > \gamma\dot{\rho}_j^i)$: preempt $j$ in favor of $j^*$ on server $i$
```

In practice, the performance of Algorithm 1—which we call the value-density algorithm for scheduling, or VDAS—can be tuned according to the specific distribution incoming jobs by conducting a grid search on the hyperparameters such as $\mu$, $\gamma$, and the slack $s$. We do just this in our experimental Section 4, to ensure the algorithm's competitiveness given our simulation's parameterization. Next, in Section 3, we design a machine-learning-based approach to solving our online scheduling algorithm and show that, in practice, it outperforms the algorithm above.

# 3 Learning to Schedule

In this section, we describe a method that learns to place jobs on servers, based on features of both the incoming job and idle servers, but also more global features like the state of all assignments and historical preemption. Indeed, we try to learn an optimal scheduling function, defined against an (unattainable) gold standard omniscient offline scheduling algorithm, as described in Section 3.1. We use that algorithm to generate training data to fit a comparator network [Rigutini *et al.*, 2011] that ranks placement decisions, described in Section 3.2. Building on this, Section 3.3 gives RANKING, our learning-based online scheduling algorithm.

## 3.1 Gold Standard: Optimal Scheduling Function

Our goal is to use machine learning methods to learn a good scheduling function—in this case, one that is as close as possible to an optimal offline scheduling algorithm. We start by solving the optimal offline scheduling problem on small-sized scenarios and recording the scheduling decisions; we use this as target labels for our training data during a supervised learning phase discussed in the following section.

Although the optimal offline scheduling is known to be NP-hard [Pinedo, 2015], we scaled the problem so that it could be solved within reasonable time with a MIP solver [Gurobi Optimization, 2016], using 40 jobs of 3 types scheduled to 4 servers with tight timing constraints (to reduce the number of decision variables). We solved over a thousand such scenarios, under constraints that ensure feasibility:

1. *capacity*: only one task is executed at a time on each of the servers;
2. *affinity*: a task can only be executed on a single server;
3. *demand*: a task can either be completely scheduled to satisfy its processing demand or not scheduled at all;
4. *scheduling window*: a task can only be executed between its arrival and deadline; and
5. *event based scheduling*: scheduling and preemption can only occur when a new task arrives or completes.

In order to minimize unnecessary affinity constraints, arriving jobs which are not scheduled are kept in an "unassigned pool" which can be scheduled to any of the servers.

## 3.2 Learning to Rank & Learning to Schedule

We now draw on intuition from the information retrieval literature to learn a ranking function that will be incorporated into a scheduling algorithm which is described in Section 3.3,.

We note that scheduling decisions involve choosing the "best" job for a specific server, and choosing the "best" server for a specific job. Complications in this space include deciding on which features to use, how to quantify the quality of a specific job-server match, and that the number of jobs and servers involved in each scheduling decision is different—thus, it is difficult to train a function with variable-sized input.

Yet, this sort of task is common in information retrieval, where documents need to be ranked according to their match to a given query. Ranking documents shares the complexities enumerated above, including the presence of a variable number of documents per query as well as unknown ranking function. With this in mind, we apply the *cmpNN* architecture [Rigutini *et al.*, 2011] to our domain, and use it to learn a pairwise comparison function of two scheduling options.

The cmpNN architecture is an artificial neural network based on two shared layers which are connected anti-symmetrically. The input to the network consists of two vectors of equal size, and the output consists of two neurons which stand for $[x \succ y, y \succ x]$. This architecture has the following properties:

1. *reflexivity*: for identical input vectors, the network produces identical output (regardless of input ordering); and
2. *equivalence*: if $x \succ y$ then $y \prec x$ and vice versa. More precisely, swapping the input vectors results in swapping of the output neurons: $[o_1, o_2] = f(\vec{x}, \vec{y}) \iff [o_2, o_1] = f(\vec{y}, \vec{x})$.

The only attribute missing to make this network an ideal comparator is *transitivity*, i.e. ensuring that if $x \succ y$ and $y \succ z$ then $x \succ z$, but as we will demonstrate this shortcoming does not limit the network's ranking ability in real world scenarios.

**The Network.** We extended the architecture in two ways.

1. *Deeper network*: the original network used a single hidden layer, which did not train well on our data. Our network uses three hidden layers of decreasing width, while maintaining the shared layer architecture at each hidden layer. The dimension of the first hidden layer is derived from the dimension of the input vectors: $h_{1,dim} = 2^{\lceil \lg(x_{dim}) \rceil + 6}$, with successive layers "shrinking" by a factor of two. The activation of the first two hidden layers is $\tanh$ and the third and fourth layers have a ReLU activation.
2. *Probabilistic output*: the two output neurons of the original architecture are connected to a softmax activation, this provides a probabilistic measure for the comparison, i.e. what is the probability that $x \succ y$. Moreover, this enables using the categorical-crossentropy loss function which improves the learning convergence

The network architecture is shown in Figure 1. The symmetric nature of the network is built by sharing weights as can be demonstrated for the connection between the input and the first hidden layer:

$$\vec{w}^1_{i,1} = w(\vec{X} \to H_{1,1}) = w(\vec{Y} \to H_{1,2})$$
$$\vec{w}^2_{i,1} = w(\vec{X} \to H_{1,2}) = w(\vec{Y} \to H_{1,1}).$$

The bias term of both parts of the first hidden layer is also shared. Thus, the two output vectors of the first hidden layer are:

$$\vec{v}_{1,1} = \tanh(\vec{w}^1_{i,1} \cdot \vec{X} + \vec{w}^2_{i,1} \cdot \vec{Y} + \vec{b}_1)$$
$$\vec{v}_{1,2} = \tanh(\vec{w}^1_{i,1} \cdot \vec{Y} + \vec{w}^2_{i,1} \cdot \vec{X} + \vec{b}_1)$$

The rest of the layers share weights and connections in a similar fashion with their appropriate activation functions.

**The Features.** We used a set of features that combine a description of the candidate job as well as that of the server; this way, a single comparator network can be used to compare jobs for a given server, and to compare servers for a given job. (Due to space, we omit the list of features.)

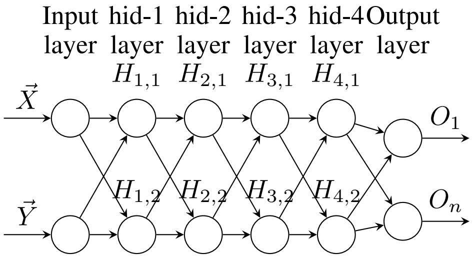


Figure 1: Pairwise comparator scheduling network.

The combined job/server feature vector enables to perform the two type of comparisons we initially desired:

1. ranking two servers ($i_1$, $i_2$) for a given job ($j$): rank$([j, i_1], [j, i_2])$; and
2. ranking two jobs ($j_1$, $j_2$) for a given server ($i$): rank$([j_1, i], [j_2, i])$.

Training samples can be taken by analyzing the optimal scheduler decision for each of the two types of scheduling events:

1. On arrival of a new job $j_a$:
   - If the job $j_a$ gets scheduled:
     (a) Compare new job $j_a$ with all other jobs—preempted $\mathcal{P}^i$ or unassigned $\mathcal{U}$—on the selected server $i$, requiring $\forall j_k \in \mathcal{P}^i \cup \mathcal{U}, [j_a, i] \succ [j_k, i]$
     (b) Compare new job $j_a$ with selected server $i$, versus other servers $k \neq i$, $[j_a, i] \succ [j_a, k]$
   - If the job $j_a$ does not get scheduled:
     (a) Compare new job $j_a$ against all running jobs, $\forall i \in$ active-servers, $[j_a, i] \prec [j_i, i]$
2. On the completion of a job:
   - If another job $j$ is scheduled:
     (a) Compare job $j$ against other pending and unassigned jobs
     (b) If job $j$ was from the unassigned pool, compare that job against other servers

### 3.3 The RANKING Algorithm

We now present our online job scheduling algorithm that incorporates the comparator network discussed above. We build on Algorithm 1 (without its hyperparameters). The adaptation is given below as Algorithm 2.

At a high level, Algorithm 2 performs as follows. When a job is completed, a pairwise comparison is performed on all jobs which are unassigned or were preempted on this server. The pairwise comparison is akin to the first pass of bubble sort, yielding the top ranking job at the top of the list. Since multiple jobs can be completed at the same time step, we need to accommodate for conflicts, i.e., two servers selecting the same unassigned job. Thus, all potential scheduling assignments are saved during this step, and for each conflict (two or more servers selecting a job), we let the job break the tie by comparing two vectors of the same job with the conflicting servers, and the job is removed from the unassigned pool. Servers which "lost" the contentious job, return to the first phase to select another job. The process continues until no more possible matches are available.

**Algorithm 2** The Ranking scheduling algorithm.

```
Event Type 1: Jobs $\{j_k\}$ arrive at time $t = a_j$.
while available servers for unscheduled job $\in \{j_k\}$ do
    1. Calculate top-ranking server for each job;
    2. Resolve multiple assignments to same server according to server's ranking of the jobs;
    3. Schedule job/server pairs;
Event Type 2: Servers $\{i_k\}$ completes its job at time $t$.
while available jobs for idle server $\in i_k$ do
    1. Calculate top-ranking job (among those preempted in this server that are unassigned) for each server;
    2. Resolve multiple assignments to the same job according to that job's ranking of the servers; and
    3. Schedule job/server pairs.
```

Similarly, when a job arrives, it initially builds a list of all candidate servers, composed of the idle servers, and servers whose running job "loses" to the new job ($[j_a, i] \succ [run_i, i]$). As above, multiple jobs can arrive at the same time step, and can request the same server. The conflicts are resolved, this time, from the other side; servers "decide" by comparing the combination of the server with conflicting jobs. This time, jobs which "lost" their requested server return to the first phase of the arrival event.

Next, we compare Algorithm 1 (VDaS) and Algorithm 2 (Ranking) against the offline optimal solution, when available, and against each other on larger simulated instances.

## 4 Experimental Validation

In this section, we compare the performance of the online scheduling VDaS and Ranking algorithms presented in Sections 2 and 3, respectively. To ensure a fair comparison, we performed a standard model selection grid search over the hyperparameters $\mu$ and $\gamma$ for Algorithm 1 (VDaS); we trained the competing Ranking algorithm's comparator network only on "small" scenarios, to be described later. We find that Ranking attains much greater value from completed jobs in the case where servers are homogeneous (§4.1), as well as when the servers are heterogeneously specialized (§4.2), for varying levels of heterogeneity.

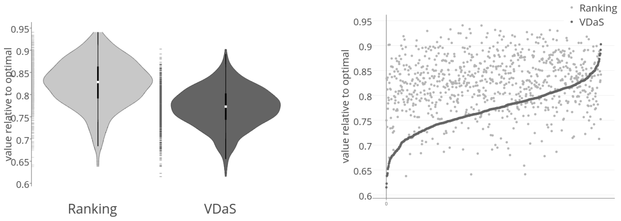
### 4.1 Online Scheduling Performance

We begin by comparing both algorithms in a simulation involving jobs arriving in an online fashion to a set of servers. The evaluation metric is the total value attained from completed jobs of random scenarios. In our simulation, a job $j$ arrives randomly with processing demand drawn uniformly at random $p_j \in [5, 31]$, slack $s_j \in [1.5, 4.0]$, value $v_j \in [50, 200]$, and one of three random types $\tau$. The preference of that job $j$ for each server $i$ is drawn uniformly at random as $\psi_j^i \in [0.5, 1]$. Servers $i$ are initialized with a random efficiency value $\eta_\tau^i \in [0.5, 1]$ at the beginning of the simulation for each type $\tau$.

We perform a standard model selection technique for VDaS—a grid search over the relevant hyperparameters $\mu$ and $\gamma$. We also train the comparator network of Ranking only on our smallest simulation, that is, 40 jobs and 4 servers. As we will see, this network generalizes quite well, and the performance of Ranking remains high—much higher than VDaS—during larger simulations.

For smaller simulations, we compare both algorithms' performance against a prescient offline optimal schedule that maximizes value, which is computed by solving a mixed integer linear program (MILP) using the Gurobi optimization toolkit [Gurobi Optimization, 2016]. For larger simulations, this optimal solution is intractable to compute, so we compare the two algorithms only to each other.

We begin with a small simulation: $40$ jobs arriving to $4$ servers. Figure 2a compares both algorithms to the optimal offline solution (value $1.0$); while neither algorithm achieves the omniscient optimum, both perform well. Yet, the mean fraction of optimal achieved by Ranking is over $5\%$ higher than VDaS. Figure 2b provides an alternative view; here, we take each of the over $1000$ runs, sort them by the fraction of optimal achieved by VDaS, and then plot the performance of Ranking on the same seed. While there are times when VDaS outperforms Ranking, the latter algorithm outperforms the former the majority of the time.



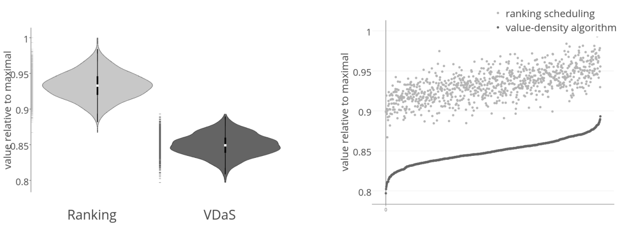
(a) Relative comparison against an offline omniscient schedule. (b) Comparison of VDaS and Ranking on identical runs.

Figure 2: Small test case: $40$ jobs and $4$ servers

When scaling up the scenario size, we no longer have the offline optimal value—solving the offline optimal MILP quickly becomes intractable. The following experiments directly compare the two algorithms, with $1.0$ now representing the highest value achieved by one of the two algorithms.

We now test with $1000$ jobs arriving to $100$ servers. Figure 3a corresponds to Figure 2a, showing the distribution of values achieved by both algorithms. The two algorithms' performances are nearly separated at this point, with Ranking dramatically outperforming VDaS—even thought its internal comparator network was trained on a dramatically simpler scenario. Figure 3b corresponds to Figure 2b; however, on these larger simulations, Ranking always achieves greater aggregate value than VDaS.

A performance gap between the algorithms that grows with the size of the simulation can be explained as follows. As the number of servers increases, the probability of randomly selecting the "correct" server decreases with the number of available servers. The probability of multiple jobs arriving together (or completing together) grows with the number of jobs. The *server-affinity* constraint (which both algorithms obey), in our setting of non-identical servers, incurs a performance penalty for "incorrect" assignments. This was not the case in the homogeneous server work of Lucier *et al.* [2013].



(a) Comparison of VDaS and Ranking. (b) Comparison of VDaS and Ranking on identical runs.

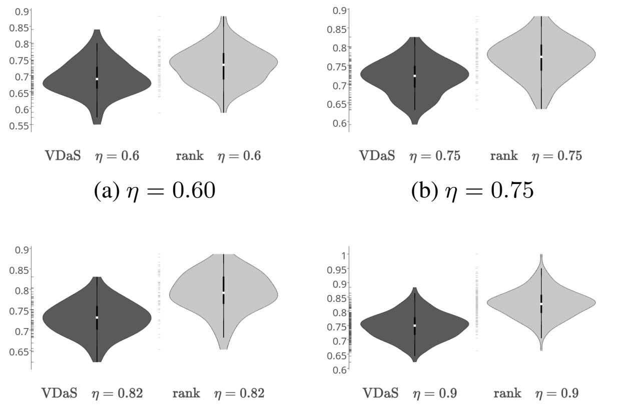
Figure 3: Large test case: $1000$ jobs and $100$ servers

## 4.2 Varying the Expertise of the Servers

Recalling our motivation—specialized human teleoperators providing assistance to the needy—we now test the effect of increased server specialization on algorithm performance in the following two settings:

1. A small group of highly-trained servers with high efficiency, versus a larger group of servers with lower efficiency ($\eta$) over types, where the ratio of the efficiency was tuned to match the change in the number of server, thus, in theory, allowing for similar throughput. In this setting, we *fix* the preferences that each job $j$ has over a server $i$ $\psi^i_j$; this was done to decrease variance and increase the focus on the server's varying efficiency.
2. Two groups of the same number of servers. One group has average efficiency over all job types, while the other group has $1/\#$types servers with high efficiency for a *single* type. We normalize the efficiency parameters to achieve similar throughput and the preference factor that a job has for a server is kept fixed, as motivated above.

Figure 4 demonstrates the first test case, where $4$ groups of servers have efficiencies $\eta \in \{0.60, 0.75, 0.82, 0.90\}$, with a lower number of servers in the groups with higher efficiency. We see that in each setting, Ranking outperforms VDaS, and that the performance grows with the efficiency of the servers only in the Ranking algorithm. Again, this is likely due to the high cost of selecting the "wrong" server.

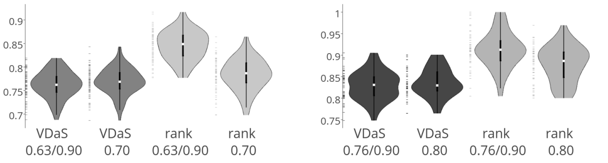


(a) $\eta = 0.60$ (b) $\eta = 0.75$

(c) $\eta = 0.82$ (d) $\eta = 0.90$

Figure 4: Comparing the performance of VDaS and Ranking as the efficiency of servers $\eta$ increases.

We now move to the second test case, where two equally-sized groups have either average but broad efficiency, or high but specialized efficiency. Figure 5 compares the performance of VDaS and Ranking on the group with average but uniform efficiency, the second group of specialized servers. Figure 5a compares both algorithms when the efficiency of the "average" group is $\eta = 0.7$, and the "specialized" group is with efficiencies in $\{0.63, 0.63, 0.9\}$. Figure 5b provides a similar analysis on parameters with lower variance: $\eta = 0.8$ for the average group, and $\{0.76, 0.76, 0.9\}$ for the specialized group. We see that Ranking outperforms VDaS in all the scenarios. Furthermore, and as a testament to the comparator network, Ranking achieves more values as specialization increases, while VDaS does not.



(a) high variance (b) low variance

Figure 5: Comparing the performance of VDaS and Ranking as specialization heterogeneity increases.

# 5 Conclusions & Future Research

Motivated by the increasing presence of semi-autonomous robots that "call out" to human teleoperators, this paper presented a machine-learning-based approach to the online job scheduling problem where jobs (tasks) have preferences over which server (teleoperator) completes them, and teleoperators have varying skill levels at completing specific classes of tasks. We extended a recent model of online scheduling to this setting, and then presented an approach to scheduling tasks that learns a ranking function of jobs for servers. We validated our approach in a simulation; it outperformed a generalization of the state-of-the-art algorithm for our setting.

Future research could consider fairness metrics like "no starvation" and proportional care; this is of independent theoretical and practical interest. Considering more elaborate tiebreaking rules—for example, by drawing intuition from the Hungarian algorithm or stable matching—when a job conflicts with two or more servers might complement fairness or increase overall efficiency. The moral and ethical issues that arise when using autonomous or semi-autonomous help for care or driving [Stock *et al.*, 2016], or AI systems that make decisions autonomously [Conitzer *et al.*, 2017], must be considered.

# References

[Anderson, 2014] Ross Anderson. *Stochastic models and data driven simulations for healthcare operations*. PhD thesis, Massachusetts Institute of Technology, 2014.

[Conitzer *et al.*, 2017] Vincent Conitzer, Walter Sinnott-Armstrong, Jana Schaich Borg, Yuan Deng, and Max Kramer. Moral decision making frameworks for artificial intelligence. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[Doucette *et al.*, 2016] John A Doucette, Graham Pinhey, and Robin Cohen. Multiagent resource allocation for dynamic task arrivals with preemption. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):3, 2016.

[Even-Dar *et al.*, 2009] Eyal Even-Dar, Robert Kleinberg, Shie Mannor, and Yishay Mansour. Online learning for global cost functions. In *Conference on Learning Theory (COLT)*, 2009.

[Gombolay *et al.*, 2016] Matthew Gombolay, Reed Jensen, Jessica Stigile, Sung-Hyun Son, and Julie Shah. Apprenticeship scheduling: Learning to schedule from human experts. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[Gurobi Optimization, 2016] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2016.

[Kaneko *et al.*, 2008] Ryuichi Kaneko, Akira Ishikawa, Futoshi Ishii, Tsukasa Sasai, Miho Iwasawa, Fusami Mita, and Rie Moriizumi. Population projections for japan: 2006-2055 outline of results, methods, and assumptions. *The Japanese Journal of Population*, 6(1), 2008.

[Leiber, 2016] Nick Leiber. Europe bets on robots to help care for seniors. *Bloomberg BusinessWeek*, March 2016. Accessed: 2016-08-23.

[Lucier *et al.*, 2013] Brendan Lucier, Ishai Menache, Joseph Seffi Naor, and Jonathan Yaniv. Efficient online scheduling for deadline-sensitive jobs. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 305–314. ACM, 2013.

[Nowak, 2017] Peter Nowak. Nissan uses NASA rover tech to remotely oversee autonomous car: New Scientist, 2017. [Online; accessed 2017-02-12].

[Panda *et al.*, 2015] Sunita Panda, Pradyumna K. Mohapatra, and Siba Prasada Panigrahi. A new training scheme for neural networks and application in non-linear channel equalization. *Applied Soft Computing*, 27:47 – 52, 2015.

[Pérez *et al.*, 2013] Eduardo Pérez, Lewis Ntaimo, César O Malavé, Carla Bailey, and Peter McCormack. Stochastic online appointment scheduling of multi-step sequential procedures in nuclear medicine. *Health Care Management Science*, 16(4):281–299, 2013.

[Pinedo, 2015] Michael Pinedo. *Scheduling*. Springer, 2015.

[Rigutini *et al.*, 2011] Leonardo Rigutini, Tiziano Papini, Marco Maggini, and Franco Scarselli. SortNet: Learning to rank by a neural preference function. *IEEE Transactions on Neural Networks*, 22:1368–1380, 2011.

[Stock *et al.*, 2016] Oliviero Stock, Marco Guerini, and Fabio Pianesi. Ethical dilemmas for adaptive persuasion systems. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[Tripathy *et al.*, 2015] Binodini Tripathy, Smita Dash, and Sasmita Kumari Padhy. Dynamic task scheduling using a directed neural network. *Journal of Parallel and Distributed Computing*, 75:101 – 106, 2015.

[Zheng and Shroff, 2016] Zizhan Zheng and Ness B Shroff. Online multi-resource allocation for deadline sensitive jobs with partial values in the cloud. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2016.

# Ethical Considerations by Personal Assistants for Life

**Frank Dignum, Virginia Dignum, Catholijn M. Jonker**
f.p.m.dignum@uu.nl, m.v.dignum@tudelft.nl, c.m.jonker@tudelft.nl

## Abstract

Virtual personal assistants, such as Siri, Amazon Alexa, Microsoft Cortana or Google Assistant, are starting to help us in many aspects of our daily lives. We envision that over time agents will be able to combine many of the functionalities that already exist and many others we can now only dream of, and use them in a user-centered way. Although this seems a laudable goal it also raises questions if we imagine this agent to survive and stay with us over our entire life. What happens if the agent knows us better than we know ourselves? It might derive wrong conclusions if it uses data from our teenage exploits or previous marriages. On the other hand it might think it has all relevant information, whereas in reality it might miss some relevant information. So, how will we trust the advice of such an agent? In this vision paper we take stock of these issues and also discuss some consequences for the technical design of such agents.

## 1 Introduction

Advances in Artificial Intelligence and Autonomous Agents make possible the advent of Intelligent Personal Assistants (IPA) that will work with you over many years, learn your preferences and goals and how to adapt to these, and will keep on developing themselves. We define IPA as a digital service looking after a range of your needs. Siri, Amazon Alexa, Microsoft Cortana and Google Assistant are just the beginning of these developments. IPA will possibly exist across platforms allowing it to be accessed from any number of devices, both traditional computing devices, such as laptops or smart phones but also wearables, implants or taking its own physical form such as a robot. These IPAs know so much about you, that they can predict what it is that you would choose (to do) in many situations. Event though this is important for them to be able to do their work correctly, this data can also be the source of undesirable information about about the user and lead to untoward decisions.

Indeed, all is well as long as you are satisfied with the performance of your IPA and can trust the IPA to keep your privacy and work towards your desires. However, what will happen, when technology changes and the IPA needs an upgrade, you no longer can trust your IPA, or due to incapacity or death others will access your IPA?

Moreover, what are the consequences for the design of IPAs given that they have to be able to have life-long interactions? Already now, many issues are raised concerning information stored in social networks such as Facebook or Twitter, where it is virtually impossible to withdraw past information. The Web does not forget. These concerns require not only technological solutions, must must be driven by public debate and informed users, together with regulatory and societal solutions.

Currently, much attention is given to the ethical and societal aspects of intelligent technologies. The IEEE Global Initiative on Ethical Aligned Design of Autonomous and Intelligent Systems brings forward a vision for prioritizing human Wellbeing in AI development, representing the collective input of over one hundred global thought leaders from academia, science, government and corporate sectors in the fields of Artificial Intelligence, ethics, philosophy, and policy[1]. Other initiatives in this area include the 100 Year Study on AI[2] and the Partnership on AI[3]. However, these initiatives are geared to the design and development of (new) AI systems and have not yet given sufficient attention to the issues that emerge from continuous and long-lasting interaction with IPA and other AI systems.

This paper describes a vision of future Intelligent Personal Assistant agents in relation to life changing events of its user, and the possibility of the IPA being misused or hacked. We discuss the questions that these events raise, and possible (unwanted) side effects. The second part of the paper is dedicated to the relation between the techniques and platforms used to develop and maintain these agents and the aforementioned complications. In particular, we discuss the merits and limitations of sub-symbolic machine learning techniques, and symbolic knowledge representation and learning techniques. Furthermore, we discuss the impact of the carrier platforms of these personal assistant agents, distributed platforms, single carriers and backup provisions. The paper concludes with a set of recommendations and guidelines for the development

[1] http://standards.ieee.org/develop/indconn/ec/autonomous_systems.html

[2] https://ai100.stanford.edu/

[3] https://www.partnershiponai.org/

of the personal assistant agents of the future to mitigate the discussed complications in light of the user's intentions and the unintended impact that the IPA might have on the life of others.

## 2 IPA knows and supports you

In the future we will all have IPAs to assist us in our daily lives, work and education. This empowers us and makes our lives run smoothly, as long as nothing goes wrong with the IPA. In this section we consider the kind of information and knowledge that IPAs of the future will collect about you in the course of their work.

In recent research by MediaLab[4], participants indicated what type of services they would like to see performed by IPAs, both at the personal as at the social level. These range from supporting a health condition by monitoring body signs, taking notes and issuing reminders, to alerting to the presence of a friend nearby or managing joint appointments.

Although these answers already indicate a range of services and accompanying information that IPAs should offer they do not seem to include the users preferences and history, while this is something that is already commonplace for Facebook, Google and other social media. We assume that IPAs can use the information collected from the user's web use, but also their actions as observed by devices in the Internet of Things. This includes their Internet use in relation to certain places and situations, health data as sensed day and night by wearables and information about the user that is kept on file at all kinds of organizations, such as tax information, employee file, electronic patient records, school records, and customer privilege cards, just to mention a few. By using sophisticated information integration algorithms, IPA will know about:

- Your preferences for when and where to have what type of meetings
- When to interrupt you given the topic and people involved
- What priority you give to which people / institutions, including all kinds of additional info about these people (positive, neutral and negative)
- How well you stick to your plans, such as quitting smoking or eat healthier
- How you address which people, using what media
- Typical expressions, ways of forming sentences, typical grammatical errors, vocabulary
- Your verbal and non-verbal characteristics, including when you are emotional, when you lie, . . .
- Your socially acceptable hobbies and forms of entertainment; but also your dark secrets

Although these are already interesting features about a person, over time your IPA will abstract from particular features and learn more general ones such as:

- Your norms, values, morals, ethics
- your (daily) habits, including the ones you would like to break or attain
- What makes you happy, angry, sad, . . .
- the ways you would like to be seen / perceived by others, and what is your 'real' self in opposition to the image you would like to present.

Assuming that the technological developments will make it possible to integrate all these types of information into a complex, possibly correct (or maybe not so correct) personality description of the user, then this description should be co-develop with the user. Note that the interaction with the IPA will have an effect on the evolution of the user, so that in effect, we are talking about a co-evolution of the IPA and the user. However, will this image of the user also develop correctly in the case of life changing events? What if a user emigrates, marries or gets children? These events can radically change her behaviour and invalidate a previously built image. Can this be repaired quickly or should the IPA be reset? Does it keep the old data? And, how will that old data affect the current knowledge the IPA holds about the user? In the next section we will discuss some of these issues in more detail.

## 3 Life changing events

At some point in your life, in the future maybe at your birth, you obtain your IPA, you work with it, and at some point you die In the meantime you make or lose friends, start or break relations, hold or lose jobs, and your IPA might be hacked. In this section we discuss the impact of these events on the relation with your IPA and thus, on your life. The questions raised in this section are collated in Section 4 and the technical impact is discussed in Section 5.

### 3.1 Birth, childhood, coming of age

Many examples are already given in the literature of how IPAs can empower people. For example, having an IPA that can help the child finding her way home safely, would enable her to walk to school without human supervision. This can improve the self-worth and sense of autonomy of the child. This might be a reason to give your child an IPA early in life. However, what are other possible consequences of becoming dependent on an IPA so early on in life?
If you get your IPA, to what extent has that already been 'filled' with morals, ethics? Whose morals are these? Normally morals of children are formed by the parents, but do they realize which norms a company might have used to determine the rules of the IPA? What kind of knowledge is already in there? If you would get your IPA at birth, then the first years, the information and knowledge stored in it will be at the direction of people in your environment. You can take no responsibility for what is stored in your IPA at that time, so you cannot be held accountable. On the other hand, years later, you might take your parents or other care takers to court on the basis of evidence stored by your IPA. For example, they hindered your wishes to become a ballerina, thereby blocking you from your intended career as you had no way

[4] https://www.mindshareworld.com/sites/default/files/What_Can_I_Help_You_With_Virtual_Assistant_Report_MindshareUK.pdf

of making up for lost training time in your youth. Such considerations, lead to questions as who is responsible for the information stored in an IPA? At what age or having what kind of mental capabilities should you have to be entrusted with having an IPA? Should the responsibilities and accountability be formally or legally documented? When should this be revisited?

### 3.2 Hacking

The general threat of having your electronic devices and applications hacked also holds for your IPA. In such events the risk is two-fold: exposure and coercion as the well-known one, and manipulation as a new risk. The risk that exists already now, the exposure and coercion risk of hacking refers to the possibility of someone hacking into your systems with the goal of accessing your private information. The hacker can then expose that information in places, to people at times that are harmful to you, but the hacker can also coerce you to do things to prevent the exposure just mentioned. This risk of hacking is well-known and a reason to invest in protection from hacking.

However, with the advent of IPAs that stay with you over long times, IPAs form a kind of extended mind. You might over time come to rely on the IPA to remind you of who the people are that you are meeting at a party, when you met them, what roles and position they hold. Now image your IPA is hacked into, not to reveal your secrets, but to input information into your IPA that enables the hacker to manipulate you. By doing this the IPA might unwittingly be a partner of the hacker in leading you to trust people that you shouldn't, thinking that you have had satisfactory relations or dealings with them before, or that they come recommended by someone that you trust implicitly. This risk might hold for able-minded highly occupied business people and politicians, but is also a real risk for the elderly with failing memories. The crooks that cheat people out of their money gain, by hacking the IPAs of their intended victims, a very convincing ally to achieve their goals.

Whether it is for exposure, coercion, or manipulation, the consequences of having your IPA hacked can be dire. Existing forms of cybersecurity might not be enough, but maybe quantum technology will be a solution, as that can give the means to detect that someone has looked at your IPA. Detecting what was changed and by whom might require techniques not yet envisioned.

Knowing that you IPA has been accessed, but not knowing in what ways it was compromised might have an unstabling impact on the user, especially, if that person knows that his/her own memory is unreliable. Further note, that the impact is worsened by the fact that in some 50 years time we all are so accustomed to relying on our IPA that we might feel then as we would feel know when we, city slickers, would be dropped in the jungle without having our smart phone.

### 3.3 New relationships & partners

IPA can help coordinate with the assistants of other people, helping to schedule social engagements, work commitments, and travel. It could anticipate your needs based on past activities, and coordinate with others to select activities that fit everyone's interests. As your relationship with others deepens, the relationships between your IPAs will also deepen. Your IPA will know a lot about your friends, life-partner and acquaintances. How much should be shared? And who determines what the IPA should keep and what should be discarded after a one-of interaction? Can you indicate that you want to opt-out any recording of interactions with your IPA to be maintained by others' IPAs? What will be the culturally accepted ways of dealing with your partners IPA? Will the idea of 'I have no secrets from you, my partner' extent to the knowledge that is contained in your IPA? Note that this might contain information about previous partners. If you have a devious nature, will your IPA support you in this as well and uphold a socially acceptable fake image of you to your environment?

### 3.4 Broken relationships and Divorce

At times when you are in a dysfunctional relationship you don't want the added burden of having to argue / fight with that person on what your IPAs knows. Can an IPA be coerced by a partner to reveal your whereabouts? Do you have a right to know what the IPA of your partner knows about how your partner feels about you? If either of you decides to break up the relationship, what demands will be made on the future state of knowledge of the IPAs regarding this partner? What should be removed from the IPAs memory?

### 3.5 In Court

How will legal studies and philosophy address the challenges of what the legal status is of the knowledge contained in an IPA? If the divorce goes all the way to court, or if for some other reason you suddenly find yourself in court and demands are made on the contents of your IPA. Should the IPA reveal information damaging to you? Can you be forced to instruct your IPA to remove certain information from its memory? Wouldn't that be the same as having some information be forcibly removed from your own memory? Can the IPA, when in the USA, call on the fifth amendment, being in all intends and purposes an extension of your mind?

### 3.6 Death and inheritance

What happens if you die? Sort of living memory of the deceased? The British tv series 'Black Mirror' episode "Be right back" presents an interesting, horrible example (cf. `https://en.wikipedia.org/wiki/Be_Right_Back`).

How long will the IPA continue to exist? Who will inherit it? Can it be accessed by more than one person / institution? Are copies made (where does that stop)? Can it be re-assigned to another person (see birth)? Does the learning stop when the owner dies (formally stops using it)? Is it possible to stop the learning? Can parts of the memory/knowledge be erased? Who determines that? If the owner gave no explicit instructions then what is kept, destroyed? Is it ethical to erase the whole IPA? Suppose it is decided to erase it, can that really be done in a guaranteed way (copies, distributed storage)? Suppose it is partially erased, to what extent can the erased info be reconstructed, and be self-reconstructed?

Which questions should the IPA (after being inherited) be allowed to answer, which not? To what extent should the IPA be seen as representing the interests of the user, even after that user has died? Should it be able to insist on its own demise if it can argue that that is what its owner would have wanted?

## 4 Concerns and Challenges

Currently the issues related to current and future technology for (intelligent) personal assistant agents, behavioural change systems and persuasive technology, mainly encompass ethical concerns such as privacy Hoven *et al.* [2015], moral responsibility Detweiler *et al.* [2012] and cybersecurity Singer and Friedman [2014], Ethical issues regarding behavioural change systems and persuasive technology consider, for example, that the person using the technology should be informed of and understand and approve of the shifts the technology is trying to achieve in the behaviour of the user Consolvo *et al.* [2009]; Michie *et al.* [2011]; Hartanto *et al.* [2016]. Papers discuss the possibilities of such technology being abused to influence people unwittingly Sunstein [2015]. The possible negative impact of social media on opinion formation of the public is discussed through the phenomenon of information bubbles Liao and Fu [2013]; Pariser [2011].

All these issues also directly reflect on the design of IPAs. Co-evolution is the term used to identify the process in nature in which two or more species interact so intimately that their evolutionary fitness depends on each other. Given the inherent relationship between the IPA and user, a co-evolutionary design approach is needed to ensure that the IPA can evolve as the user does. However, on the one hand, the user will want its IPA to evolve alongside its needs and desires, on the other hand, it may also want the IPA to provide guidance on her development. Social mechanisms such as norms and institutions are the means society uses to stabilize and inform the co-evolution of people and communities. We speculate that similar norms may emerge via a process of evolution amongst IPA and user. To ensure the co-evolution of IPA along the social and individual requirements of its user, a layered normative system can be envisioned, in which higher level norms direct lower level norms King *et al.* [2015]. In this way the user can describe the more general principles that will guide concrete IPA behaviour: With whom does the IPA share information? What knowledge should be forgotten and when should it be forgotten? Which sources (e.g. medical, sensing, mobility, contacts,...) should be combined, when?

Another important concern to be addressed, is the nature of the relationship between IPA and user. As pieces of software, AI systems are basically tools. However, their increased intelligence and inter-ability makes them to be perceived, and subjected to the same social expectations as (human) partners. Following Wallach and Allen [2008], the pathway to engineering IPAs requires the design of operational, functional and full ethical behaviour. Figure 2 describes interaction complexity based on two dimensions, autonomy and social awareness, to classify interactive systems into three basic categories.

Tools, such as a hammer, or a search engine, do not have either autonomy nor social awareness and are not considered to be ethical systems, but in their design the values of their engineers are incorporated. The next type of systems, assistants, have limited autonomy but interact in open environments. Functional morality is sensitive to ethically relevant features of those situations, by hard-wiring ethical responses into the system architecture, resulting in autonomous agents that are better at adjusting their actions to human norms. Finally, full moral agents are able of self-reflection and can reason, argue and adjust their moral behavior. Given the potential evolution of IPAs' capabilities and social role, as described in this paper, they should be seen as *partners* rather than assistants, in the characterization of Wallach and Allen [2008].

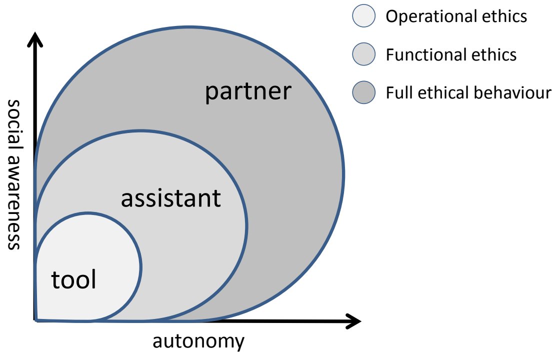


Figure 1: Ethics design stances (adapted from Wallach and Allen [2008])

Methodologies for Design for Values are needed to design IPAs in order to make explicit the values and value priorities of designers and stakeholders Hoven [2005]; Friedman *et al.* [2006]. These methodologies satisfy the following principles: (1) global aims and policies should be explicitly described; (2) enforcement is context based and should to be negotiated between stakeholders; (3) design decisions should be formulated explicitly rather than being implicit in the procedures and objects. All of these have a big impact on the architecture and technical design of IPAs.

## 5 Technical implications

The main technological challenge on the design of IPAs is the uncertainty about the future evolution of platforms and data representation models. Moreover, there is very little research on lifelong use and interaction with digital devices. This poses many questions concerning transferability, backward compatibility, scalability, and data storage. As technology evolves so will the relationship between user and IPA, as will the requirements and intentions for this interaction.

Currently, IPAs knowledge will mainly be encoded as a (deep) neural network, for which no fully satisfactory methods for extracting this knowledge in symbolic form are available. In fact, the statistical models obtained from machine learning algorithms tend to be opaque and monolithic, and therefore it is difficult to understand how results are achieved. To be able to learn, adapt and be accountable for its actions and recommendations, the IPA will need to have a social based model in which social identity, values and norms are explicitly represented. This is a new area of research for which attention was asked as well in Kaminka [2013]; Dignum *et al.* [2014]; Norling [2016] in previous years, which seems to indicate the importance of this area.

Electric Elves, an interesting experiment to build electronic secretaries, see Chalupsky *et al.* [2001], were meant to support humans in daily life activities, 24/7 available and for a long term. These agents were not meant to adapt to the user, but especially meant to take the concerns of the organization into account. In Tambe [2008] the shortcomings of the experiment are discussed: over-generalization, lack of social norm awareness, inadequate handling of privacy issues. Of course we have to realize that this experiment was done more than 15 years ago. More success might be achieved if the experiment would be tried again using deep learning, socially aware agent technology (Corkill *et al.* [2011]; Riemsdijk *et al.* [2015]). Immediate challenges would be how the agent would deal with the tension between employer vs employee interests and concerns. For this, current work on value and norm conflict recognition and conflict resolution is relevant and more research is needed in the implementation of normative conflict reasoning in intelligent systems Vasconcelos *et al.* [2009]; Jiang *et al.* [2014].

## 6 Research Agenda

In the near future the concerns and challenges of Section 4 should be addressed. The key technical solutions will have to focus on dealing with norms, values and ethical dilemmas. In particular, we need knowledge representation languages in which values and norms are key concepts, enabling reasoning about norm conflicts, value-conflicts and priorities on these concepts. We need mechanisms for multi-layer, value-oriented, planning combining long-term aims with short term goals. Ethical dilemmas need to be automatically recognized, and reasoned about. Part of this reasoning should be the ability to recognize that some dilemma cannot be solved by the agent itself, but needs the help of other agents or humans. The need for discussing such dilemmas with others, as well as the requirement that agents should be accountable for their actions, emphasizes the need for transparent deliberation mechanisms to allow for explanation and inspection of the reasoning processes of the agents. Recent work on omniscient debugging Koeman *et al.* [2017] for agent oriented programming languages, are promising technologies to create the technology with which agents will be able to explain its past actions. A related technical challenge is the secure longterm data storage needed for explaining past activities. As agents might live much longer than the humans that initiated these agents, unknown challenges on data governance will have to be addressed from both a technological as well as from a legal perspective. In particular, we think of how to regulate ownership, sharing, withdrawal and replication of data.

Due to the co-evolution process between humans and AI, the human condition, the human value system, and the societal value system will change. The AI systems we develop will have to continuously co-adapt to these changing value-systems. The line of research into the formalization and reasoning with and about value systems and norms needs to be extended. Escalation to meta-levels of reasoning will have to become main-stream technology in which human-AI-interaction will have to focus on the development of both sides through mutual interaction. The current ethical frameworks that underpin these approaches and research lines are inadequate as they are based on thousands of years old value systems in which values are considered to be static and only the emphasis/importance of values change. The nature of the AI - human co-evolution however, makes this static position untenable. We should build on work that investigates how human understanding of values, opportunities, and challenges is mediated by the technologies we use Verbeek [2011]. We believe that this new dynamic ethical framework and in parallel the advanced formalized frameworks for reasoning at various meta-levels with and about values will only be successful when done as an interdisciplinary effort of philosophers, logicians, and researchers in artificial intelligence.

Finally, we need value-oriented design methodologies that support elicitation and inclusion of ethical, societal and legal values through the whole design process. These methodologies can ensure responsible design of IPAs that can be guaranteed to be trusted and accountable for their decisions Dignum [2017].

## 7 Discussion and Conclusions

Based on the questions and technological considerations we come to the following conclusions and considerations, for which we formulated a research agenda in Section 6.

It is clear that we cannot foresee all the consequences of lifelong co-existence of an IPA with a user. This holds even stronger for the continued existence of an inherited IPA that retains (part of) its knowledge about the deceased user and is now owned by the inheritor.

This calls at least for a modularization of the IPA such that parts can be encrypted and others possibly open for specific groups. It should also be possible to explain the behavior of the IPA based on its sources at all times, such that it can be held accountable and verified whether it follows the values and norms it was supposed to adhere to (according to its user). This also plays an important role in the co-evolution of the IPA. If a user wants to add or delete certain sources of information, new sensing devices, or reasoning modules it should be able to check the consequences of these alterations. Explaining what will happen in some use cases will be paramount to achieve this insight (because users are no programmers).

A related question is whether society should create legislation that makes these design decisions for all IPAs legally binding in order to assure accountability? Or should there be tailor-made decisions for each IPA based on the available knowledge about the wishes of the user (as might be stored in the IPA)? However, if the knowledge about the user as stored in the IPA, should be a legal argument in whether or not the IPA should be shut-down or its knowledge about the user (partly) destroyed, then the IPA in its reasoning should follow some form of moral values and ethical principles to ensure that they are always aligned to the user. Furthermore, IPAs should also be equipped with safety constraints.

We argue that these principles will help humans to accept and trust IPAs as they can be perceived to behave as ethically as humans in the same environment. Furthermore, these principles would make it easier for IPAs to determine their actions and explain their behavior in terms understandable by humans.

Something not considered in this article are the future rights of the IPA, which we leave for further exploration.

As a final question, we put to you: Do we want IPAs as sketched in this article? Should we burden ourselves with these moral dilemmas?

## References

Hans Chalupsky, Yolanda Gil, Craig A Knoblock, Kristina Lerman, Jean Oh, David V Pynadath, Thomas A Russ, Milind Tambe, et al. Electric elves: Applying agent technology to support human organizations. In *IAAI*, volume 1, pages 51–58, 2001.

Sunny Consolvo, David W McDonald, and James A Landay. Theory-driven design strategies for technologies that support behavior change in everyday life. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 405–414. ACM, 2009.

Daniel Corkill, Edmund Durfee, Victor Lesser, Huzaifa Zafar, and Chongjie Zhang. Organizationally adept agents. In *COINS2011 Workshop at AAMAS*, 2011.

C. Detweiler, F. Dechesne, K.V. Hindriks, and C.M. Jonker. Ambient intelligence implies responsibility. *Ambient Intelligence and Smart Environments*, 12:33–61, 2012.

F. Dignum, R. Prada, and G.J. Hofstede. From autistic to social agents. In *AAMAS 2014*, May 2014.

Virginia Dignum. Responsible autonomy. In *Proceedings of IJCAI'17*, 2017.

Batya Friedman, Peter H. Kahn, and Alan Borning. Value sensitive design and information systems. *Advances in Management Information Systems*, 6:348 – 372, 2006.

Dwi Hartanto, Willem-Paul Brinkman, Isabel L. Kampmann, Nexhmedin Morina, Paul G. M. Emmelkamp, and Mark A. Neerincx. *Home-Based Virtual Reality Exposure Therapy with Virtual Health Agent Support*, pages 85–98. Springer International Publishing, Cham, 2016.

J. van den Hoven, P.E. Vermaas, and I. van de Poel. *Handbook of Ethics, Values, and Technological Design*. Springer Netherlands, 2015.

J. van den Hoven. Design for values and values for design. *Information Age +, Journal of the Australian Computer Society*, 7(2):4–7, 2005.

Jie Jiang, Huib Aldewereld, Virginia Dignum, and Yao-Hua Tan. Compliance checking of organizational interactions. *ACM Trans. Manage. Inf. Syst.*, 5(4):23:1–23:24, December 2014.

G. Kaminka. Curing robot autism: A challenge. In *AAMAS 2013*, pages 801–804, May 2013.

Thomas C King, Tingting Li, Marina De Vos, Virginia Dignum, Catholijn M Jonker, Julian Padget, and M Birna Van Riemsdijk. A framework for institutions governing institutions. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 473–481. IFAAMAS, 2015.

V. Koeman, K.V. Hindriks, and C.M. Jonker. Omniscient debugging for cognitive agent programs. In *Proceedings of IJCAI'17*, 2017.

Q. Vera Liao and Wai-Tat Fu. Beyond the filter bubble: Interactive effects of perceived threat and topic involvement on selective exposure to information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2359–2368, New York, NY, USA, 2013. ACM.

Susan Michie, Maartje M van Stralen, and Robert West. The behaviour change wheel: a new method for characterising and designing behaviour change interventions. *Implementation Science*, 6(1):1, 2011.

E. Norling. Don't lose sight of the forest. In *AAMAS 2016*, May 2016.

Eli Pariser. *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin, 2011.

M. Birna van Riemsdijk, Catholijn M. Jonker, and Victor Lesser. Creating socially adaptive electronic partners: Interaction, reasoning and ethical challenges. In *Proceedings of the fourteenth international joint conference on autonomous agents and multiagent systems (AAMAS'15)*, pages 1201–1206. IFAAMAS, 2015.

P. W. Singer and Allan Friedman. *Cybersecurity and Cyberwar: What Everyone Needs to Know*. Oxford University Press, 2014.

Cass R Sunstein. The ethics of nudging. *Yale J. on Reg.*, 32:413–591, 2015.

Milind Tambe. Electric elves: What went wrong and why. *AI magazine*, 29(2):23, 2008.

Wamberto W Vasconcelos, Martin J Kollingbaum, and Timothy J Norman. Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 19(2):124–152, 2009.

P.P. Verbeek. *Moralizing Technology: Understanding and Designing the Morality of Things*. Chicago: University of Chicago Press, 2011.

Wendell Wallach and Colin Allen. *Moral machines: Teaching robots right from wrong*. Oxford University Press, 2008.

# Managing Legal Risks from the Societal Integration of Robots

**Ashley Kelso MIEAust***
AustraLaw
Sydney, New South Wales, Australia
ashley@australaw.com.au

## Abstract

This paper discusses the legal risks involved in deploying robots for use by the public, and how those risks can be managed as a design requirement. A recent Queensland case involving an injury inflicted by a robot is analysed to provide practical lessons and insights. The Australian Consumer Law is discussed to highlight the more onerous legal requirements involved when supplying robots for use by the public. The paper then argues that the external 'supervision-based' safety systems used for human-robot interaction in factories will not be feasible for the societal integration of robots. It is asserted that these systems are too resource intensive to be scalable. Instead, it is proposed that legal risks must be managed as part of the design for the societal integration of robots to be viable. The paper then discusses the trade-offs between using machine learning to develop adaptable risk detection and response capability in robots, while retaining the capacity for their decisions to be recorded and analysed for litigation purposes.

## 1 Introduction

The human capacity to 'get stuff done' has always been subject to two main categories of limitations:

1. Physical limitations of speed, strength, skill, and endurance; and
2. Mental limitations of speed, knowledge, creativity, learning, and endurance.

The industrial revolution allowed us to leverage the physical strength, speed and endurance of machines; and, to an extent, to replicate the skill of a human worker by use of physical mechanisms. The next revolution came with computers, whose growing decisional speed, analytical sophistication, and information storage capabilities, have been leveraged with ever increasing success to relieve us from our mental limitations. The next stage has been to combine the two, to allow us to fully delegate the tasks of analysis, decision-making, and the physical-world execution of those decisions by way of robotics.

While robots have been in use for a while now, the extent of the general public's interaction with them has been limited; and mainly in the form of spectacle, gimmickry or variations of a vending machine. Just as engines and computers have now fully integrated their way into the everyday lives of the public, it is now time for robotics to make that leap as well – from relieving industry of the laborious tasks of daily operation, to relieving the public of the laborious tasks of their daily lives.

A robot may be generally defined as a computer capable of executing its decisions in the physical world. It therefore carries with it the ability to inflict physical and psychological harm on the humans it interacts with.

While in industry the use of spatial demarcation between humans and robots, coupled with professional supervision, has been used to manage the inherent risks of robotic errors and human inadvertence, such measures are clearly too resource-intensive to be applied when creating robots for the general public. Instead, the safety measures must be scalable and therefore automated, being an inherent part of the design of these robots. In short, robots must be able to exercise self-restraint and initiative to avoid injury to person and property.

It is self-evident that the social-integration of robots will be brought to a halt if designers do not address two key risks:

1. Social licence to operate: The public must trust robots and actively desire to use them in their day to day lives. The will of the public is necessary for the will of the media, which in turn is necessary for the will of politicians to adapt the law to accommodate the social-integration of robots;
2. Financial ability to operate: Quite simply, injury to members of the public by mass-produced robots brings the capacity for financially crippling litigation. At present, in New South Wales, the 'pain and suffering' damages alone for a single

*The author is a qualified mechatronics engineer and a practicing lawyer in the State of New South Wales, Australia.

personal injury action in negligence can be up to $605,000.[1] By the time economic loss, treatment costs, and legal costs have been accounted for, the financial impact of a single claim can be significant.

This paper aims to offer practical guidance to roboticists on the legal element of robotics design. Whilst the ethics of robots and the decisions they will make is an important topic, the fact remains that robots will be designed or modified by people (just like any technological advancement) to do ostensibly unethical things: terrorist attacks could be carried out with weaponised quad copter drones, chat bots could be used to scam people on a massive scale, and robots could be used to spy on political, legal or corporate opponents. The practical questions then are: how will the consequences of robot-inflicted harm be dealt with by the law? who will be held accountable? And, how can designers guard against their robots falling afoul of the law?

In addressing these issues, this paper will: first, examine how the law currently deals with the actions of robots; second, identify the main legal risks applicable to integrating robots into society; and third, outline how roboticists might employ machine learning to enable the law-abiding functionality of their robots to scale with the growth of their sophistication and increasingly intimate integration into society.

## 2 How the Law Currently Deals with Harm Inflicted by Robots

Hollywood films are awash with examples of robots posing a risk to people, both to their jobs and to their wellbeing. The film Elysium features a scene where Matt Damon's character is terminally injured while working in a factory. The machine he is working on is jammed by a pallet. He is instructed to enter the machine to clear the palate, and when he does the machine starts up again trapping him inside and irradiating him. He was then unsympathetically diagnosed by a medical robot.[2]

Such incidents, however, are no longer confined to the realms of science fiction. The recent case of *Peapell v The Smith's Snackfood Company Limited*[3] involved a person who was injured by a robot in a scenario not dissimilar to the Elysium example above. Examining this case provides a real-world example for roboticists on the important issues to take into account when introducing robots into broader society, such as:

1. Legal responsibility for the actions of robots;
2. Allowing for human inadvertence when designing the safety measures of a robotic system;
3. The serious harm that robots can cause to humans;
4. The need for safety procedures to not just be taught to the human users, but also to be built into the robot itself;
5. The important role that engineers will play as expert witnesses to explain the actions of robots, and the physical consequences of the various 'states' or 'modes' that a robot will enter during the course of operation;
6. The importance of having efficient infrastructure in place to detect risk-inherent robot-human interactions and implement at scale the systems required to resolve those risks by modification, maintenance, or recall; Lest the risks materialise into harm, and the occurrence of prior 'near misses' strengthen a plaintiff's case.

The incident is described at paragraph 1 of the Peapell decision:

> On the 29th of July 2009, Mrs Peapell (the plaintiff) suffered personal injuries while she was carrying out her duties at the defendant's factory as a Packing Machine Operator. The plaintiff was at the time operating the Schubert Multi Pack Machine. During the course of that operation, she was crushed by a robotic arm of that machine while she was clearing jammed cardboard from that machine's 'former', which is situated within the actual confines of the machine.

As a result of this accident the plaintiff suffered injuries, including a hypoxic brain injury which affected her memory.

The plaintiff sued her employer, the Smith's Snackfood Company, in negligence. In short, the elements of negligence are:

1. The defendant owed the plaintiff a duty of care with respect to some source of risk;
2. It was foreseeable to a reasonable person in the defendant's position at the time that the risk, if it materialised, may cause harm to someone in the plaintiff's position;
3. The defendant beached that duty of care by failing to take reasonable care to prevent the risk materialising and causing harm to the plaintiff;
4. The risk materialised and caused harm to the plaintiff.

It is well established at law that an employer owes a non-delegable duty of care to their employees ('non-delegable' means that the employer can delegate the tasks involved in complying with that duty, but they remain liable if harm

[1] *Civil Liability Act 2002 (NSW)*, Part 2, Div 3; *Civil Liability (Non-economic Loss) Amendment Order 2016 (NSW)*.

[2] https://www.youtube.com/watch?v=60V3JFUPvIE

[3] [2016] QDC 265.

results from the negligent execution of those tasks). As part of satisfying this duty of care, an employer must provide its employees with a 'safe system of work' (which includes the safety of the equipment the employee is required to use, the safety of the procedures they are instructed to follow when using that equipment, and supervision to see that the systems are followed). If some part of the employee's job carried the risk of injury, and the employer has not seen to it that this is addressed with appropriate safety measures, the employer will be liable if the risk materialises and causes harm to the employee.

The plaintiff in *Peapell* was successful in establishing that the defendant had breached its duty of care by failing to provide a safe system of work, and that the injuries occurred as a result of this breach. The basis for this was the judge's findings that:

1. The robot jammed after completing a run of Twisties;
2. It was routine for the plaintiff to enter the robot's working space to clear objects that had jammed it;
3. While the sliding door into the area with the robotic arm would lock while the robot was in production mode, it could be easily opened if the stop button was pressed, or if the robot stopped itself – which it would do if it detected a jam;
4. The prescribed procedure did not state that the plaintiff was to press the stop button before entering the robot's working area, as such entering the area only necessitated pressing the stop button if the door wouldn't open. It was the plaintiff's understanding that it was safe to enter the robot's work area so long as it had stopped (whether by the stop button or of it's own volition);
5. When she cleared the jammed cardboard the robot detected this and, as it was not in stop mode, it automatically started again;
6. The plaintiff suffered injury when the robotic arm moved forward and collected her;
7. There were two incidents about twelve months before the subject accident where the robot was able to enter production mode with the access door still open – it having failed to detect this. This contributed to the foreseeability of the subject accident, as it would have identified the need to review the operation of the robot's safety systems;
8. The door detection mechanism was not checked during maintenance inspections between the previous incidents and the subject accident;
9. The robot's printout log, which recorded changes in its modes, was not sufficiently detailed to determine what mode it was in when the accident occurred. This was partly due to ambiguity in the description of events like "Excessive Control Deviation" or "F2028"; it being unclear, even to the engineering expert witness, what these meant.
10. The defendant was found negligent for failing to instruct the plaintiff to always press the stop button before entering the robot's work area. The defendant was also found negligent for failing to ensure and maintain the safe functioning of the access door.

In relation to point 7, evidence was given by two engineers, one stated that this previous incident had occurred because of the manufacturer's incorrect installation of the door detection mechanism. The other stated that it had occurred as a result of a worn part causing the mechanism to operate defectively. The second engineer asserted that standard maintenance procedures ought to have identified the worn part, and that this may have played a role in the subject accident. A significant portion of the judgment was concerned with identifying what modes the robot was in at what times, and how the robot and its safety systems would operate in these different modes. Much of the trial was also taken up with examining the maintenance history of the robot.

While this example concerned an industrial accident, it contains highly relevant insights for those wishing to produce robots that will interact with the public, such as:

1. A plaintiff's lawyers will be uninterested in questions of the robot's personal responsibility for the injury. Instead they will look to hold those responsible who have the assets to pay an award of damages. For example, when a bouncer at a club injures a patron, the lawyers will typically pursue the owner of the club and the security company who was contracted to provide the bouncer's services to the club. Similarly, lawyers will look to sue those who made the robot, and those who caused its operation to involve interaction with the public.
2. It will be advisable set up robust reporting of issues (say, via IoT technology, and web-based user feedback), coupled with the capability to quickly address those issues (for example, Tesla engineers can remotely install software updates and add features without the need to recall the cars to the factory).
3. It will be important to ensure that the robot automatically generates a clear chronological record of is modes, error events, and sensor readings – including any detected attempts to tamper with the robot.
4. It will be important to identify engineers with strong communication skills who can be called on to serve

as expert witnesses. Engineers who become respected by plaintiffs and defendant for their reasonableness and impartiality will be more persuasive in bringing claims to a close privately without the need for public litigation.

5. Roboticists should endeavour to contractually require prospective claimants to engage in confidential dispute resolution processes before a claim can be filed and the complaint become public knowledge.
6. Robots will need to be designed to detect risks and restrain themselves – e.g. detecting a raised heart rate in the user, or the use of a combination of temperature and proximity sensors to foresee collisions with people or animals.

Above all, roboticists will need to accommodate the tendency of people for lapsed judgement and inattention, and should not merely rely on disclaimers or instruction manuals to avoid harm. Even if a law suit is won the trust of the public may be lost, and with it the chances of integrating your robots into society.

## 3 Legal Risks Facing the Societal Integration of Robots

In producing robots for use by the general public, roboticists significantly increase the number of human-robot interactions, decrease the capacity for controlling risk via supervisory means (as would be done in the workplace), expose themselves to more onerous legal requirements, and thereby increase the likelihood of legal claims for damage to person and property. This heightens the need to more actively address the risk of legal claims as an element of the design of robots produced for interaction with the public.

Along with the regulatory questions raised in relation to the use of autonomous vehicles[4] and drones,[5] the societal integration of robots engages the Australian Consumer Law (ACL). While claims can still arise under common law negligence and contract, the ACL reduces the legal hurdles available to roboticist to defend claims through:

1. Strict liability of 'manufacturers' if a person suffers damage to person or property arising from a 'safety defect'; which can be anything that falls below the level of safety that people would generally be entitled to expect from the product.[6] A 'manufacturer' for ACL purposes is anyone involved in achieving the final form and function of the product, or anyone who holds themselves out at the manufacturer or allows their branding to be applied to the product.[7]
2. Preventing manufacturers from 'unfairly' shifting the risks to the consumer. Under the ACL, a term of a contract with a consumer or a small business will be void if it is found to be 'unfair'.[8] Typically, a term will be unfair if it causes a significant imbalance in the power and risk burden of the parties, which is beyond what is necessary in the circumstances to protect the legitimate interests of the party who has the benefit of that term. The ACL provides a number of examples, which include terms that limit the right of the consumer or small business to sue for damage arising from the transaction.[9]
3. Statutorily imposed guarantees as to quality and performance. In consumer contracts the ACL prevents manufacturers from excluding warranties as to fitness for purpose, defects, safety, and durability. These warranties can be extended by any representations or statements made about the product by the supplier, the circumstances in which the purchase was made, and any purpose for which the supplier understands that the consumer intends to use the product for.[10]

Furthermore, the ACL allows for the imposition and enforcement of safety standards for consumer products.[11] The Minister can impose safety standards and ban the supply of consumer products which may cause injury. Breaching a standard or a ban can result in fines and strict liability to compensate consumers who are harmed as a result.

In view of the above, manufacturers and suppliers of robots should avoid placing too much faith in contractual exclusions of liability when making robots for use by the public. If it is assumed that the risk of a claim being made can't be excluded then attention has to be turned to both reducing the risk of harm occurring, and minimising the risk of being held legally responsible for any harm arising from interaction with a robot. This will be achieved by ensuring that designs address:

1. Applicable government and industry standards;
2. The foreseeable sources of risk that arise from the robot alone and in its interaction with its operational environment;

[4] 'NTC Discussion Paper - Clarifying control of automated vehicles - April 2017': http://www.ntc.gov.au/current-projects/clarifying-control-of-automated-vehicles/?modeId=1064&topicId=1166

[5] 'Backyard skinny-dippers lack effective laws to keep peeping drones at bay': https://theconversation.com/backyard-skinny-dippers-lack-effective-laws-to-keep-peeping-drones-at-bay-76580

[6] ACL, s9; Part 3-5.

[7] ACL, s7.

[8] ACL, Part 2-3.

[9] ACL, s25.

[10] ACL, ss54-55.

[11] ACL Part 3-3.

3. Sound training of those facilitating the sale of robots to avoid conduct that could extend the scope of statutory warranties, or cause misunderstanding about how to safely interact with the robot; and
4. Remote logging and resolving of 'near miss' incidents.

The common thread is that the standards and risks should be known or be foreseeable in advance, meaning that they can be addressed as part of the robot's design and maintenance. As the environments in which robots will operate will become increasingly diverse, the challenge will become that of designing robots with an adaptable capacity for risk identification and response; Robots that can identify when the risk or probability of harm to person or property has reached a certain threshold, and calculate a response that will bring that risk back within the tolerance threshold.

## 4 Using Machine Learning to Create Law-Abiding Robots

Attempting to program a robot for every individual source of risk, or risk scenario, will pose a scalability problem as the sphere of their operation, and the public's innovative applications for them, expands. Instead, roboticists will need to start by considering the typical indicators that cause us to foresee that the risk of harm occurring has increased in a situation (e.g. speed, decreasing proximity to other objects, increased friction, unexpected physical resistance etc etc), and enable robots to leverage their sensors to make the same determinations.

From there, the next step will be to train robots via machine learning to again an adaptable capacity for foreseeing risk and calculating the most appropriate response. Indeed we are already starting to see risk avoidance technology in practice in the automotive industry, such as Tesla's collision avoidance technology which has even anticipated collisions between other vehicles.[12]

The flipside to the benefit of applying machine learning to develop law abiding robots, is the requirement that we be able to interrogate them when something does go wrong. In the above case of *Peapell v The Smith's Snackfood Company Limited* it was damaging to the defence's case that they could not prove with sufficient certainty what state the robot was in at the time of the accident. Therefore, they could not effectively use the records generated by the robot to rebut the plaintiff's evidence about whose behaviour was the root cause of the risk that led to the injury – the robot's (and employer's) or the plaintiff's.

It is also likely that if a robot has ostensibly failed to take reasonable care when performing higher order tasks (e.g. a self-driving taxi) that this negligence will be imputed to the manufacturer or owner of the robot, much as the negligence of an employee will be imputed to the employer. Alternatively, the manufacturer may be held liable via warranties or statute-based 'safety defect' grounds.

It will therefore be a design consideration to ensure that the machine learning methods that are applied, allow for the robot to reliably attest to its state and reasoning at the time that the actions in question occurred. Attempting to rebut a plaintiff's evidence by use of empirical evidence about the robot's training and past responses to risk scenarios would be very expensive, time-consuming, and vulnerable to being disallowed on evidence law grounds.

## 5 Privacy

Having discussed the need for robots to be able to log and report on their interactions with their environment (including humans), for the improvement of safety and defence of legal claims, it is important to also address the risk to privacy that this capability creates.

Roboticists should be aware of the new mandatory requirement to report data breaches that will come into effect on 22 February 2018. The *Privacy Amendment (Notifiable Data Breaches) Act 2016* amends the *Privacy Act 1988* to require entities to notify the Australian Information Commissioner and the affected individuals if (paraphrasing):

1. Their data has likely become accessible at some point to an unauthorised party and
2. This access is likely to result in serious harm to whomever the information relates to.

Serious harm is not defined in the Act, but roboticists should work with the assumption that it may come to mean 'material detriment to an individual's interests'.

Importantly, s26WG of the above amendment provides that if the data was stored in a way that would render it unintelligible to an unauthorised party, and it is unlikely that this party could decrypt the data, then notification may not be necessary as serious harm is not sufficiently likely to result. Roboticists should therefore consider both measures to prevent unauthorised access, and also robust methods of encryption (e.g. asymmetric encryption) to address the risk of a data breach.

Such design measures will guard against the risk of fines for non-compliance with the *Privacy Act* (i.e. If an entity neglects to report a data breach), and potential civil claims where data breaches result in financial loss to individuals. Furthermore, these measures will also help sustain the trust

[12] 'Tesla Autopilot predicts crash seconds before it happens' https://www.youtube.com/watch?v=APnN2mClkmk. See also: 'Tesla Autopilot saves lives compilation 2017' https://youtu.be/Ndeb1pMAsh4.

that is necessary for people to continue to see robots as helpers, as opposed to a risk to their privacy.

## 6 Conclusion

This article has aimed to provide practical guidance to roboticists on the legal risks to be addressed when deploying robots for use by the public, and how those risks might be managed via the design process in a scalable fashion. This commenced with an analysis of the Queensland case of *Peapell v The Smith's Snackfood Company Limited* in which a factory worker was injured by a robot. The analysis identified practical considerations, such as designing for human inadvertence; systematic detection, logging, and addressing of 'near miss' incidents; and the importance of a human-readable, autogenerated, time-stamped, log of the robot's state and decisions to rebut evidence about the cause of an accident.

The article then discussed the increased risk of litigation that comes with the social integration of robots. This increased risk arises from the greater number of human-robot interactions, the inability to adequately supervise those interactions (as would be done in a work environment), the increasingly diverse uses and operational environments that people will apply robots in, and the more onerous legal requirements that apply to products and services supplied to consumers.

It was identified that the ACL undermines a number of the methods that businesses typically use to limit their exposure to liability. Critically, it imposes strict liability for 'safety defects', excludes contract terms that 'unfairly' shift risks to the consumer, and statutorily imposes warranties on manufacturers that can be extended by representations at the point of sale.

On that foundation, the article then moved to discuss how the legal risk for roboticists might be addressed in a scalable fashion. It was suggested that where the task is basic and the field of operation narrow and well known, a combination of sensors and if-this-then-that logic would likely suffice. However, as the field of operation for robots expands and the range of possible human-robot interactions grows, machine learning will need to be applied so that robots can develop adaptable risk detection and response functionality.

It was proposed that this functionality should be coupled with the ability to remotely log and resolve 'near miss' incidents and accidents, to increase the pace with which these safety systems develop, and to reduce the risk of these incidents adding support to actions in negligence (as occurred in the *Peapell* case).

It was then argued that it may be a risk to the defensibility of legal actions if the machine learning technology employed can't be interrogated after an accident. This may influence the particular type of algorithms that are used, or require that they be adapted to keep a running log of their decision-making (a kind of computational blackbox).

Finally, the risks to privacy from social robots was discussed. It was identified that in the course of logging and reporting on their daily operations, robots may end up storing or communicating information that could be detrimental to users if accessed by an unauthorised party. It was proposed (in line with an upcoming amendment to the Privacy Act) that roboticists impose security measures to prevent unauthorised access, and to render the data unintelligible to unauthorised parties in the event that it is accessed.

The successful social integration of robots will require an ongoing collaborative effort between the legal and engineering professions. With increased responsibility comes increased risk. Lawyers will need to be skilled at identifying the dominant legal policy issues that will drive the evolution of the law in this area. They will also need to consider how the functionality of robots can be used to strengthen the safety record and legal position of those who deploy robots for public use. Effective communication between the professions will be key to shepherding these innovations into wider social use, while retaining financial viability and social licence to operate.

## References

*Civil Liability Act 2002 (NSW).*

*Civil Liability (Non-economic Loss) Amendment Order 2016 (NSW).*

*Peapell v The Smith's Snackfood Company Limited* [2016] QDC 265.

*Competition and Consumer Act 2010, Schedule 2 (Cth)* ('the Australian Consumer Law').

[Gogarty 2017] Brendan Gogarty, *Backyard skinny-dippers lack effective laws to keep peeping drones at bay,* The Conversation: <https://theconversation.com/backyard-skinny-dippers-lack-effective-laws-to-keep-peeping-drones-at-bay-76580>.

[National Transport Commission 2017] *NTC Discussion Paper - Clarifying control of automated vehicles - April 2017*: <http://www.ntc.gov.au/current-projects/clarifying-control-of-automated-vehicles/?modeId=1064&topicId=1166>.

*Tesla Autopilot predicts crash seconds before it happens*, <https://www.youtube.com/watch?v=APnN2mClkmk>.

*Tesla Autopilot saves lives compilation 2017* <https://youtu.be/Ndeb1pMAsh4>.

# Towards Robots for Social Engagement

**Elisa Cucco, Michael Fisher, Louise Dennis, Clare Dixon, Matt Webster,**
**Bastian Broecker, Richard Williams, Joe Collenette, Katie Atkinson, Karl Tuyls**
Department of Computer Science, University of Liverpool, UK

## Abstract

In this paper we consider the aspects that ensure successful interaction between social robots and people. As such robots are increasingly autonomous, it is crucial that the user can trust their behaviour, and that their decisions are taken within social and ethical requirements. It is important to specify what actions are expected from the robot, verify that the autonomous robot actually achieve these, and validate that the requirements are exactly what the user wants. To this purpose, our activities have been focused on formal verification of autonomous robotics systems, investigating both reliability and robot ethics and deployment of social robots in both constrained and public environments.

## 1 Introduction

Social robots are designed to interact with people in a natural, interpersonal manner, often to achieve positive outcomes across applications such as education, health, quality of life, entertainment, communication, and tasks requiring collaborative teamwork. The long-term goal of creating social robots that are competent and capable partners for people is quite challenging. They will need to be able to communicate naturally with people using both verbal and non verbal signals, in order to engage them not only on a cognitive level, but on an emotional level as well, to provide effective social and task-based support to the users. For this reason their main characteristic is a range of social-cognitive skills to understand human behaviour, and to be intuitively understood by people.

Considering their increasing involvement in social-care and education applications, there is also a growing research emphasis in cognitive Human Robot Interaction on identifying the mental models people use to make sense of emerging robotic technologies and investigating people's reactions to the appearance and behaviours of robots.

As those robots are becoming increasingly autonomous and they are directly interacting with humans it is vital that the user can be assured that those robots are safe, reliable and ethical in order to trust them. Thus, a big concern is not only that ethical and reliable behaviours are met, but also that they can be verified [Dennis *et al.*, 2016a; Charisi *et al.*, 2017].

In this paper we focus on the issues, such as safety, cognitive interaction, and trustworthiness, related to the increasingly common situation in which humans and autonomous robots share an environment. We give an overview of our activities related to this problem and in particular report on a practical human-robot engagement in which we have been involved.

## 2 Social Robots

People are more engaged while interacting with robots that are able to communicate naturally and have some social skills, but it is crucial that they also feel safe.

### 2.1 Human Robot Interaction

Recent advances in physical human-robot interaction have shown the potential and feasibility of robot systems for active and safe workspace sharing and collaboration with humans. This trend has been supported by recent progress in both robotic hardware and software technology that allow a safer human-robot interaction. Thus, by considering the physical contact of the human and the robot in the design phase, possible injuries due to unintentional contacts can be considerably mitigated.

These robot systems include applications such as coworkers (i.e., cooperative material-handling), but also service robots and assistive devices for physically challenged people. Therefore all of them share the common requirement of safe and close physical interaction between human and robot.

While encompassing safety issues based on biomechanical human injury analysis as well as of human movements, human-friendly hardware design and control strategies, learning and cognitive key components have to be developed, in order to enable the robot to predict human motions in real time in an unstructured dynamic environment. Apart from developing the capabilities for interactive autonomy, human safety and physical interaction have to be embedded at the cognitive decisional level as well; thus the robot will be enabled to react or physically interact with humans in a safe and autonomous way. Furthermore, self-explaining interaction and communication frameworks need to be developed to enhance the system usability and interpretability for humans, for example, to communicate whether a situation is safe or dangerous not only with verbal, but also non-verbal communication cues, such as gestures and emotional feedbacks. The key distinctive aspect of human-robot interaction is then the intrinsic dual aspect of physical and cognitive interaction.

**Physical Human-Robot Interaction.** Most work in pHRI (physical Human-Robot Interaction) can be classified across three main categories of interaction: *supportive, collaborative* and *cooperative*. The distinction is marked by the increasing frequency and necessity of physical contact with the robot and level of proximity of the user [Siciliano and Khatib, 2007]. Supportive interactions occur when the robot is not the main performer of the task, but instead provides the human with tools and information to optimize the human's task performance or objectives, for example museum tour guide robots, shopping assistant robot and home-care robots. In this context pHRI typically concerns safety, that is preventing and mitigating the effect of unexpected collisions and performing appropriate proxemic behaviour. To support safety as well as the physical interactions, well-structured robot communication is needed. In collaborative interactions both the human and the robot work on the same task, each separately completing the part of the task best suited to their abilities. In this scenario, the human completes a task requiring human dexterity, while the robot completes the part of the task not well suited to direct human involvement, i.e., repetitive tasks, high force applications, chemical deposition or precision placement. Finally cooperative interactions refer to the extension of cooperative manipulation to include force interaction with humans. The human and the robot work in direct physical contact, or indirect contact through a common object, with cooperative and continuous shared control of the task.

The main solution to make robots physically safer is to pursue a mechanical design that reduces the robot link inertia and weight by using lightweight and highly integrated mechatronics designs. Low inertia and high compliance have become the most desirable features( i.e., the DLR LWR-III [Hirzinger *et al.*, 2001]. However, very compliant transmissions may ensure safe interaction but may be inefficient in transferring energy from actuators to the links for their fast motion. Thus, other approaches to gain performance for guaranteed safety are the intrinsecally elastic robots (VIA- Variable Impedance Actuator method [Tonietti *et al.*, 2005] allows the passive compliance of transmission to vary during the execution of tasks, and the SEA-Series Elastic Actuator method [Pratt and Williamson, 1995] consists in locating the largest actuator at the base of the robot and connecting it through a spring, thus achieving low overall impedance, while small motors collocated at the joints provides high-performance motion).

Haptic sensors are capable of measuring contact and detecting collision, while they are also able to read and display emotion sensed by physical interaction, and can improve also the involvement of the human. Indeed, in human development, touch plays a crucial role in developing cognitive, social and emotional skills, as well as establishing and maintaining attachment and social relationships. Recently, more and more social robots are being equipped with tactile skin, thus allowing the robot to react according to the person touching the robot, or recognize social and affective communicative intent in how a human touch the robot.

**Cognitive Human-Robot Interaction.** A key challenge in robotics is to design robotic systems with the cognitive capabilities necessary to support human-robot interaction. These systems will need to have appropriate representation of the world, the capabilities, expectations and actions of the human and how their own actions might affect the world, their task, and their human partners. Core research activities in this area include the development of representations and actions that allow robots to participate in joint activities with people, a deeper understanding of human expectations and cognitive responses to robot actions and models of joint activity for human-robot interaction [Siciliano and Khatib, 2007].

More specifically research activities in this area include:

- human models of interaction — building an understanding of how people perceive robots and interpret their actions and behaviours, and how these perceptions and interpretations change across contexts and user groups;
- robot models of interaction — the development of models that enable robots to map aspects of the interaction into the physical world and develop cognitive capabilities through interaction with the social and physical environment; and
- models of HRI — creating models and mechanism that guide human-robot communication and collaboration, action planning, and model learning.

Research in cognitive human-robot interaction examines how people, including children and older adults, react to their interactions with social robots. Some approach robots in a scientific-explanatory mode, interpreting a robot's action in an emotionally detached and mechanistic manner, others invest in the interactions emotionally and treat the robots as they were living beings, such as babies or pets [Turkle *et al.*, 2004]. Anthropomorphism, or the attribution of human characteristics to non-human behaviour is an other interesting aspect in HRI research. In [Kiesler *et al.*, 2008] it is shown that people anthropomorphize a physically embodied robot more readily that an on-screen agent, and people behave in a more engaged and socially appropriate manner while interacting with the co-present robot. People also anthropomorphize robots they interact with directly more than they do with robots in general, and with robots that follow social conventions (e.g., polite robots) more than those that do not [Fussell *et al.*, 2008]. Moreover users with low emotional stability prefer mechanical-looking robots to human-like ones [Syrdal *et al.*, 2007]. As might be expected a robot's human-like appearance can have a positive effect on people's propensity towards it but also a too high level of human-likeness may place the robot in an *uncanny valley* [Mori, 1970], which refers to a dip in a hypothetical graph of the relationship between a robot's human-likeness and the human's response, suggesting that a robot that looks like a human, coupled with some remaining non-human qualities, makes users uncomfortable.

## 2.2 Social Robots Interaction

The way a person interacts with a social robot is quite different from interacting with an autonomous robot. Modern autonomous robots are viewed as tools that humans use to perform hazardous tasks in remote environments. However, social robots are designed to engage people in an interpersonal manner in order to achieve positive outcomes in domains such as education, therapy, or health, or task-related work in areas such as coordinated teamwork for manufacturing, search and rescue, domestic chores and more. The development of socially intelligent and socially skilful robots drives research to develop autonomous robots that are natural and intuitive for the user to interact with, communicate with, collaborate with, and teach new capabilities. Dautenhahn's work is among the most consistent concerned with thinking about robots with interpersonal social intelligence where relationships between specific individuals are important [Dautenhahn, 1995; 1997].

Social robots are designed to interact with people in human-centric terms and to operate in human environments alongside people. Their main characteristic is that they engage people, communicating and coordinating their behaviour with humans through verbal, non verbal or affective modalities. Anthropomorphic design principles, spanning from the physical appearance of robots, to how they move and behave, and how they interact with people, are often employed to facilitate interaction and acceptance. For social robots to close the communication loop and coordinate their behaviour with humans, they must also be able to perceive, interpret, and respond appropriately to verbal and non verbal cues from humans.

Depending on different application scenarios, increasing social skills are needed: robots that need to collaborate with humans simply to achieve, or help in a task, do not need to be particular social. On the other hand, robots that serve as companions in the home for the elderly or assist people with disabilities need to possess more social skills, which will make them more acceptable for humans. Without these skills, such robots might not be used and thus fail in their role as an assistant [Dautenhahn, 2007].

To participate in emotion-based interaction, robots must be able to recognise and interpret affective signals from humans, they must possess their internal models of emotions and they must be able to communicate this affective state to others. In particular, social robots need the ability to recognize, understand and predict human behaviour in terms of the underlying mental states such as beliefs, intents, desires, feelings, etc. For instance social robots will need to be aware of people's goals and intentions so that they can appropriately adjust their behaviour to help the human. Furthermore, the behaviour of social robots will need to adhere to people's expectations. They will also need to be able to flexibly draw their attention to what the user is interested in, so that their behaviour and information can be more useful [Siciliano and Khatib, 2007].

Social robots will need to be deeply aware of the user's emotions, feelings and attitudes to be able to prioritize what is the most important thing to do. In general, emotional displays can inform the interpretations about an individual's internal states (agreement or disagreement about a belief, valuing a particular outcome) and therefore help to predict future actions. An increasing number of socio-emotional robots have been designed to realize such functions to facilitate human-robot interactions. Some of these robots have been designed with emotional responses or emotional inspired decision making systems in order to entertain, i.e., AIBO [Fujita, 2004] or Pepper robots. In this way robots handle better human emotional states, and also motivate people toward more effective interactions, which is particular useful in domains such as education, or therapeutic system.

## 3 Human-Robot Engagement

For autonomous systems and social robots to be allowed to share their environment with people, they need to be safe and have to behaves within ethical acceptable limits. One vital aspect to human-robot interaction is *trust*. Indeed, no one will use a robot, or even share the environment with it, if they cannot trust its behaviour. In addition, since autonomous robots need to make decisions, it is crucial to have some ethical principles the robot will use to make such decisions, especially when they concern human safety.

### 3.1 Trust

For the users of a social robot one of the main concerns is that the robot they are interacting with is safe and behaves ethically. *Trust* is the key issue and in order to trust the AI system, the user needs to be informed of all the robot's capabilities. The appearance of trustworthiness might also be an issue, in particular in assisted living technologies. Some concern have been raised related to the impact that such robots can have on elderly [Sharkey and Sharkey, 2012] or children [Matthias, 2011].

Trust also plays a role in choosing an ethical theory to implement in the autonomous robot, even if they are very different. Indeed, trust is a social construct concerned with how the behaviour of the robot appears to the human.

For this reason trustworthiness is considered mainly subjective: a lot of items can change the user's level of trust of a robot, and among them the relationship between trust and harm [Salem *et al.*, 2015]. The concept of trust also involves the robot's reliability and predictability. However, while machine's errors could have an impact on the trust [Salem *et al.*, 2015], also errors occasionally performed by a humanoid robot can increases its perceived human-likeness, and thus, likeability. On the other hand, the nature of the task requested by the robot can affect the users willingness to follow the instructions. People involved in the regulation of the autonomous systems and their integration in the society also need confidence in the system. Finally, developers and engineers need to have confidence in their prototypes as well, and also have the possibility to highlight if there are issues and where they are. Another key requirement for trust is also *transparency*: the human will trust the social robot more likely if he can have some understanding of the robot's action and the reasons for its choices [Charisi *et al.*, 2017].

### 3.2 Robot Ethics

The main concern of robot ethics is to guarantee that autonomous systems will exhibit an ethically acceptable behaviour in all situations in which they interact with human beings. In particular, robot ethics is an applied ethical field whose objectives is to develop scientific/cultural/technical tools that can be shared by different social groups and beliefs. These tools aim to promote and encourage the development of robotics for the advancement of human society and individuals, and to help to prevent its misuse against humankind [Siciliano and Khatib, 2007].

The responsibility for improper or illegal behaviour of the robot can be attributed to the owners, designers, and/or builders of the machines. The question becomes increasingly difficult as the robot become more autonomous and capable of modifying its behaviour through learning and experience, since obviously the behaviour will be no longer based entirely on their original design.

Most of the ethical requirements that the robot has to follow are set by regulatory or standard bodies. In addition, the manufacturers might have built-in more specific ethical codes without contradicting those prescribed by the regulators. Finally the users could decide to add ethical preferences, to make sure that the robot's actions are personally acceptable [Charisi *et al.*, 2017]. Moreover the choices of criteria for a robot to be considered ethical involve the whole of society, therefore *transparency* is of utmost importance.

Finally, while the first concern is to develop robots that behaves ethically in society, it is important also to concern about how the autonomous robot can protect itself against misuse (e.g., taking advantage of the capabilities of the robot to commit criminal acts). Such misuse can be achieved by hacking an existing system or developing an unethical one.

For instance, sophisticated humanoids raise a number of ethical issues, including the following:

- loss of privacy for the human inhabitants, e.g., if the robots are permitted free access to all rooms in a home or if the robot's computer is accessed by hackers;
- ability of the robots to recognize commands that may lead to unethical behaviour;
- rights and responsibilities of the robots, e.g., should they be treated with respect as if they were human;
- emotional relationships, e.g., how a robot should relate to human anger, can a robot be punished for misbehaviour (and if so, how);
- how should a robot react to multiple instructions from different humans.

From the social and ethical standpoint, the assistive robots bear the most sensitive safety and ethical problems (e.g., patients may become emotionally attached to the robots, so that any attempt to withdraw them may cause distress; the robots will not be able to respond to the patient's anger and frustration, such as when a patient is refusing to take medication; a robot may be called by more than one patient and not being able to prioritize the request).

## 4 Our activities

We have focused on the verification of ethical behaviour in autonomous systems, and trustworthiness of social-care robots. Recently our interest in human-robot engagement has been increasing and we also have been involved in a practical case study in cooperation with Tate Modern museum.

### 4.1 Verification of robot ethics

In our society people can trust the decisions of professionals because they are subject to regulations and certification. With autonomous systems, with no human directly in control, ensuring that the system actually matches the required criteria is more difficult. In order to be confident with the robot's behaviour it is crucial to *specify* what actions to expect from the system in particular scenarios, *verify* that the system actually achieves this, and *validate* that the requirements are what the user want [Charisi *et al.*, 2017]. Typically those requirements can be technical, legal or ethical (e.g., never choose to do something dangerous for the user). In particular, is essential that the ethical requirements are certified by a regulator body.

Thus the aim of verification is to ensure that our system meets its requirements. Formal Verification also carries out a comprehensive mathematical analysis of the system to prove whether it corresponds to these formal requirements. By using tools, such as *model checking*, we can prove whether a particular property, that is an expression of the requirements, holds for the model of the system. In this way, the requirements are checked against all possible executions of the system. Verification via model checking is widely used for the analysis of safety and reliability of robotic systems [Dennis *et al.*, 2016b]. We have also recently used formal verification to address ethical issues for autonomous systems [Dennis *et al.*, 2016a; 2015], focusing on the possibility to verify formally whether an autonomous system will behave ethically, given a particular ethical setting.

In work such as [Arkin, 2007; Woodman *et al.*, 2012] the ability of the agent of being also an *ethical governor* has been introduced and verification has been explored in [Dennis *et al.*, 2015]. Such agent will choose the most ethical plan available, allowing unethical choices to occur only when it does not have a more ethical choice.

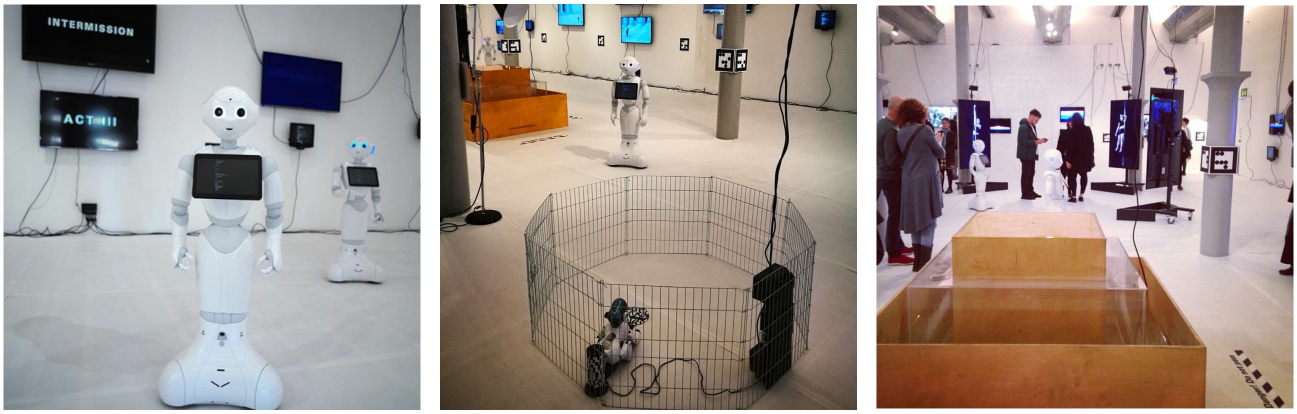
We also have conducted formal verification of an autonomous personal care robot, Care-O-bot, [Dixon *et al.*, 2014; Webster *et al.*, 2015], that is able to autonomously assist a person living within the house. We modelled the robot of Care-O-bot and its environment using Brahms, an high-level multi-agent language. Formal verification was then carried out by translating this to the input language of an existing model checker.

### 4.2 Practical engagement

For our social experiments in interaction between human and robots we started recently to use Pepper Robots, a humanoid robot developed by Aldebaran and Softbank Robotics (see Figure 1).

**Pepper robots.** Pepper is a human-shaped robot, designed mostly to be a companion robot. It is the first humanoid robot capable of recognising the principal human emotions, adapting his behaviour to the mood of his interlocutor, and also learning the user's preferences in order to improve the social interaction.

It can observe human expression by its camera system and identify human voice via its speech recognition system. They respectively enable it to function in a complex environments and to identify movements, and to detect where sounds are



(a) (b) (c)

**Figure 1:** Exhibition at Tate Liverpool. **(a)** Pepper robots. **(b)** Pepper robot and Sony Aibo. **(c)** People moving around robots.

coming from and locate the user's position, while also allowing the robot to identify the emotions transmitted by the user's voice.

Its emotion recognition function render the robot flexible in coping with the situation and interact better and in a more social acceptable way with humans. The constant dialogue between perception, adaptation, learning and choice is the result of what is known as the emotion engine. Furthermore, its anti-collision system (e.g., lasers, infra-red, sonar sensors), enable Pepper to detect both people and obstacles, and therefore to reduce the risk of unexpected collisions.

**Human-robot engagement example.** As an example of human-robot engagement we report on our involvement, as programming team, with an exhibition at the Tate Modern art gallery in Liverpool[1]. The artist, Cecile B. Evans[2] is interested in the increasing influence that new technologies have on the way we feel, and the way we relate to each other. She created a play where the performance is outsourced to two humanoid robots (Pepper) and a robot dog (Sony Aibo), who collaborate with a group of human users appearing on screens (see Figure 1).

In staging this collaboration between humans and robots, Evans hints at the possibility of the technological singularity — the hypothesis that at some point in the near future, artificial intelligence will surpass human intelligence [Bostrom, 2014]. But the work departs from the conventional narrative of "killer robots" and instead imagines a future scenario in which robots and humans will collaborate, working together to fight against external forces. Together, the users and robots navigate a series of events that they learn about through the screens that uncover aspects of the complex relationship between humans and machines.

Also, while the exhibition was running, we were able to collect feedback from visitors: they were mainly feeling comfortable moving around the robots, amazed at how the robots could move naturally, and interested at the idea of robots helping people in a dangerous situation.

## 5 Future work

A significant challenge in using social robots, especially in domestic and social-care environments, is ensuring that the interaction with the human is safe, that the user can trust the robots, and therefore that we can verify and validate that all the ethical requirements are met. We are already working on research fields such as verification and validation, dependability and trustworthiness.

In the near future we are planning to support further this research by utilising a social robot laboratory to investigating the operation of autonomous robotic systems in different physical and virtual environments. In particular the facility will improve our research on how humans and robots interact with each other in a domestic environment (social-care or domestic-assistant scenarios). Another future development would be more focused on the trustworthiness. More in particular, how the trust of the user change if the robot exhibit faulty behaviour, especially in a domestic environment (ongoing work with Kerstin Dautenhahn).

## 6 Conclusions

The future of autonomous robotic systems and their proper integration within our society depends on many different aspects. It is clearly relevant how people perceive the robots and interpret their behaviour. For this reason social robots are provided with increasing social skills.

For autonomous robots to be allowed to share the environment with people they need to be safe and their behaviour has to follow some ethical requirements. Therefore it is important to collect certifications about what to expect from a robot's behaviour, and verify that all these requirements are met. With the increase of autonomy in robotics it is also crucial that the user can trust the robot's behaviour. Indeed, people will never use a social robot, or even share a domestic area with it, if they are not confident that it is behaving safely and that its decisions comply with ethical and social limits.

[1] https:/news.liverpool.ac.uk/2016/10/21/robotics-experts-support-new-tate-liverpool-art-installation/

[2] http://cecilebevans.com/

In order to overcome these issues we have investigated the possibility to use formal verification to guarantee that the autonomous robot is behaving within technical (i.e., safe interaction) and ethical requirements.

**Acknowledgements.** Work described in this paper continues to significantly benefit from collaboration three leading experts: Kerstin Dautenhahn (*social robotics researcher*); Cecile B. Evans (*artist*); and Alan Winfield (*robot ethics researcher*). We thank them all.

## References

[Arkin, 2007] R. C. Arkin. Governing Lethal Behavior: Embedding Ethics in a Hybrid Deliberative/Reactive Robot Architecture. Technical Report GIT-GVU-07-11, Georgia Tech, 2007.

[Bostrom, 2014] N. Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014.

[Charisi *et al.*, 2017] V. Charisi, L. Dennis, M. Fisher, R. Lieck, A. Matthias, M. Slavkovik, J. Sombetzki, A. F. T. Winfield, and R. Yampolskiy. Towards Moral Autonomous Systems. *ArXiv e-prints*, March 2017.

[Dautenhahn, 1995] K. Dautenhahn. Getting to Know Each Other: artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems 16:333–356*, 1995.

[Dautenhahn, 1997] K. Dautenhahn. I Could be You: The Phenomenological Dimension of Social Understanding. *Cybernetics and Systems 28(5):417–453*, 1997.

[Dautenhahn, 2007] K. Dautenhahn. Socially intelligent robots: dimensions of human-robot interaction. *Philosophical Transactions B: Biological Sciences 362(1480):679–704*, 2007.

[Dennis *et al.*, 2015] L. A. Dennis, M. Fisher, and A. F. T. Winfield. Towards Verifiably Ethical Robot Behaviour. *CoRR*, abs/1504.03592, 2015.

[Dennis *et al.*, 2016a] L. A. Dennis, M. Fisher, M. Slavkovik, and M. P. Webster. Formal Verification of Ethical Choices in Autonomous Systems. *Robotics and Autonomous Systems 77:1 – 14*, 2016.

[Dennis *et al.*, 2016b] L. A. Dennis, M. Fisher, N. K. Lincoln, A. Lisitsa, and S. M. Veres. Practical Verification of Decision-Making in Agent-Based Autonomous Systems. *Automated Software Engineering 23(3):305–359*, 2016.

[Dixon *et al.*, 2014] C. Dixon, M. Webster, J. Saunders, M. Fisher, and K. Dautenhahn. "The Fridge Door is Open"–Temporal Verification of a Robotic Assistant's Behaviours. In *Proc. Advances in Autonomous Robotics Systems: 15th Annual Conference, TAROS 2014, pp.97–108* Birmingham, UK, September 1-3, 2014.

[Fujita, 2004] M. Fujita. On Activating Human Communications with Pet-type Robot Aibo. *Proceedings of the IEEE 92(11):1804–1813*, 2004.

[Fussell *et al.*, 2008] S. R. Fussell, S. Kiesler, L. D. Setlock, and V. Yew. How People Anthropomorphize Robots. In *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction*, pages 145–152, 2008.

[Hirzinger *et al.*, 2001] G. Hirzinger, A. Albu-Schffer, M. Höhnle, I. Schäfer, and N. Sporer. On a New Generation of Torque Controlled Light-weight Robots. In *Proc. ICRA, pp3356–3363*. IEEE, 2001.

[Kiesler *et al.*, 2008] S. Kiesler, A. Powers, S. R. Fussell, and C. Torrey. Anthropomorphic Interactions with a Robot and Robot-like Agent. In *Social Cognition 2008*, volume 26, pages 169–181, 2008.

[Matthias, 2011] A. Matthias. Robot Lies in Health Care: when is deception morally permissible? *Kennedy Institute of Ethics Journal 25(2):279–301*, 2011.

[Mori, 1970] M. Mori. Bukimi no tani [the uncanny valley]. *Energy 7(4):33–35*, 1970.

[Pratt and Williamson, 1995] G. A. Pratt and M. M. Williamson. Series Elastic Actuators. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1, pages 399–406 vol.1, 1995.

[Salem *et al.*, 2015] M. Salem, G. Lakatos, F. Amirabdollahian, and K. Dautenhahn. Would You Trust a (Faulty) Robot?: Effects of Error, Task Type and Personality on Human-Robot Cooperation and Trust. In *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp141–148*. ACM, 2015.

[Sharkey and Sharkey, 2012] A. Sharkey and N. Sharkey. Granny and the Robots: Ethical issues in robot care for the elderly. *Ethics and Inf. Technol. 14(1):27–40*, 2012.

[Siciliano and Khatib, 2007] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[Syrdal *et al.*, 2007] D-S. Syrdal, K. Dautenhahn, S. N. Woods, M. L. Walters, and K-L. Koay. Looking Good? Appearance Preferences and Robot Personality Inferences at Zero Acquaintance. In *Proc. AAAI Spring Symposium Multidisciplinary Collaboration for Socially Assistive Robotics, Technical Report SS-07-07, Stanford, California, USA*, pages 86–92, 2007.

[Tonietti *et al.*, 2005] G. Tonietti, R. Schiavi, and A. Bicchi. Design and Control of a Variable Stiffness Actuator for Safe and Fast Physical Human/Robot Interaction. In *Proc. IEEE International Conference on Robotics and Automation (ICRA), pp526–531*. IEEE, April 2005.

[Turkle *et al.*, 2004] S. Turkle, C. Breazeal, O. Daste', and B. Scassellati. Encounters with Kismet and Cog: Children respond to relational artifacts. In *Proc. IEEE-RAS/RSJ Int. Conf. Humanoid Robots*, pages 1–20, 2004.

[Webster *et al.*, 2015] M. Webster, C. Dixon, M. Fisher, M. Salem, J. Saunders, K. L. Koay, K. Dautenhahn, and J. Saez-Pons. Toward Reliable Autonomous Robotic Assistants Through Formal Verification: A Case Study. *IEEE Trans. Human-Machine Systems, 46(2):186-196*, 2016.

[Woodman *et al.*, 2012] R. Woodman, A. F. T. Winfield, C. Harper, and M. Fraser. Building Safer Robots: Safety Driven Control. *Int. Journal of Robotic Research*, 31(13):1603–1626, 2012.

# Stable and Robust: Stacking Objects using Stability Reasoning

**Xiaoyu Ge[1], Jochen Renz[1], Nichola Abdo[2], Wolfram Burgard[2], Christian Dornhege[2], Matthew Stephenson[1]** and **Peng Zhang[1]**

[1] Research School of Computer Science, Australian National University, Canberra, Australia

[2] Department of Computer Science, University of Freiburg, Freiburg, Germany

## Abstract

In this paper, we define a stability reasoning problem that has many potential applications in robotics: given a set of objects, identify a sequence of actions that arranges the objects to form a stable and robust structure. This problem is challenging as for each object there are many possible ways to stack them, and one has to deal with spatial and physical constraints simultaneously.

We formalise this problem as a structure-designing problem based on structural stability and robustness which measures how stable a structure is. We provide a theoretical analysis of the computational complexity of the problem. We propose a structure-designing algorithm with the combination of quadratic programming and qualitative reasoning. We evaluated the method on nontrivial stacking tasks in a simulated environment.

## 1 Introduction

Imagine you want your robot to fetch you several objects from a table. Today's robots would probably bring you one object at a time–if you are lucky. A human, instead, would try to carry as many objects at the same time as possible in order to minimize the back and forth walks. This is what we would like a future robot to be able to do as well, to stack those objects in a way that it can safely transport a number of them at the same time. Transporting a stable stack is harder than just building a stable stack, as the movement likely disturbs the stack. Therefore, we need a robustness measure about how stable a structure is, and need to develop methods that can generate stacks that are stable enough to be transported.

The capability to autonomously build stable and robust structures of a given set of objects is something that is important in a number of application domains, such as warehousing, logistics, construction, or in everyday household settings. Having robots with this capability to augment humans in these domains is therefore highly desirable. Existing applications in these domains can only deal with standardized objects such as pallets or standard sized boxes.

In this paper, we develop a method that allows us to autonomously stack objects of many different sizes and shapes. To achieve this capability two problems need to be solved, namely, *structure designing* and *manipulation planning*, which are at different levels of abstraction. At the high level, structure designing aims at finding an arrangement (including the placing order) of objects so that they form a stable structure suitable for transportation. Manipulation planning deals with low-level details such as how to control the gripper(s) of a robot to manipulate an object to the desirable spatial configuration. In this paper, we focus on the "structure designing" part. This is a very challenging problem for robots. There is an infinite number of ways to stack a set of objects. Not only the placement of objects is crucial, but also the order in which objects are placed, and there has been limited work on verifying the stability of a structure and determining how stable a structure is in this stacking-transporting setting.

To solve the structure designing problem (formally defined later), we develop an effective reasoning mechanism that can reason about the physical and geometrical constraints imposed by the problem domain as well as the requirements of the structure. We propose and implement a method that uses static analysis to determine the stability and robustness of a structure. We formalise the designing problem, provide a theoretical analysis of the complexity of the problem, and develop and evaluate an algorithm that effectively solves it. We evaluate the proposed method in a simulated environment and compare the algorithm with a state-of-the-art pallet stacking algorithm [Schuster *et al.*, 2010] that considers stability.

## 2 Background

While stability analysis [Fahlman, 1974] has been studied as an AI problem since the early 1970s, there has been limited investigation on stability reasoning and structure designing. One relevant problem is 3D bin packing, however, the structural stability is either determined locally [Edelkamp *et al.*, 2014] or not considered. [Ge *et al.*, 2016] solved a visual detection problem in gaming environments by reasoning about the stability of two-dimensional structures. In the area of scene understanding, stability analysis has frequently been applied to guide the segmentation [Jia *et al.*, 2013]. Most of the methods have been tailored to specific domains. In architecture design, [Whiting *et al.*, 2009] proposed a method that can automatically generate stable architectures based on pre-specified grammars (templates of a building) while in our problem domain, we do not specify any template and the method has to identify possible "grammars" by itself. Related research in robotics mainly focuses on manipulation planning. [Toussaint, 2015] proposed a method that identifies a sequence of manipulation actions that stacks by maximizing the height of a structure composed of blocks and cylinders. The stability is quantified using heuristics based on the distance of the objects. [Mojtahedzadeh *et al.*, 2015] tries to identify supporting relations between cargos and determine the order of unstacking. [Schuster *et al.*, 2010] solved a distributor's pallet stacking problem using nest beam search where stability of the stack is optimized.

One related research paradigm is simulation-based reasoning that draws inferences from probabilistic simulations. For example, [Battaglia *et al.*, 2013] predicts the stability of a tower and in which direction it will fall. [Davis and Marcus, 2016] investigated the limitation of this paradigm in automated reasoning. Furthermore, as a simulator only calculates the state of the world using its approximation methods, from which one can hardly understand why a physical phenomenon (e.g. toppling) happens. Without the understanding, it is highly unlikely to make any useful adjustment. Verifying stability with a simulator is also cumbersome even in 2D environments [Stephenson and Renz, 2016], as the time required to wait until the simulated world settles down is often unknown.

# 3 Problem Statement and Modeling

Informally, we solve the following problem: Given a set of objects, use them to build a structure that is stable and robust under certain constraints. We first define the terminology and assumptions of the problem domain; then we formally define structural stability and robustness and formalize the structure designing problem. In the next section we prove its complexity.

## 3.1 Terminology and Assumptions

**Definition 1** (Object)**.** *An* **object** *$o$ is a manipulatable solid rigid physical entity in three-dimensional space. The physical properties of an object we consider include mass and friction. We assume the mass of an object is uniformly distributed.*

**Definition 2** (Structure)**.** *A* **structure** *(or stack) $S_{\boldsymbol{O}}$ is composed of a set $\boldsymbol{O}$ of objects connected to each other through contacts. The contact can be between edges, corners or surfaces of objects. We assume the forces (except the gravity force) will only occur at the contact. We omit the subscript $\boldsymbol{O}$ when it is clear from the context what the objects are. The* **tray** *$Tr$ is the object at the bottom of a structure. We require a structure can only have one tray.*

**Definition 3** (Physical Environment)**.** *The* **ground** *$Gr$ is a flat surface (e.g. the top surface of a table) on which objects can be placed. The ground is not manipulable and always remains static. We assume there is a uniform downward* **gravity** *force in the environment. The direction of the gravity is perpendicular to the ground plane. We denote the gravity force (weight) on a particular object as $\boldsymbol{f^G}$. The* **reference frame** *is a fixed frame of which the xy-plane represents the ground plane and the* origin *is set to an arbitrary point in the ground plane. The z-axis is in the opposite direction of the gravity.*

**Definition 4** (Spatial transformation)**.** *A* **spatial transformation** *$\mathfrak{T} : \mathbb{R}^3 \to \mathbb{R}^3$ maps a point in the reference frame to another point in the same reference frame through a translation and/or a rotation about the origin. $\mathfrak{T}^{(\theta,\hat{a})}$ denotes a rotation of angle $\theta$ around the axis $\hat{a}$. Transforming an object can be viewed as applying the transformation on every point of the object. In the sequel, transforming a structure is equivalent to applying the transformation to every object of the structure.*

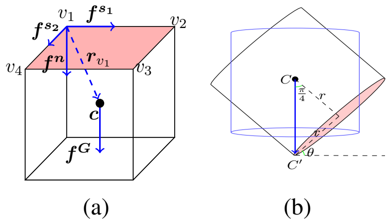


Figure 1: The force modeling at vertex $v_1$. The top face (in red) of the cuboid is contacted with another face (omitted). (b) A created cylinder before (in blue) and after (in black) a rotation of $\theta = \frac{\pi}{4}$. $C'$ is the image of $C$ under the vertical projection to the supporting face. When $\theta \geq \frac{\pi}{4}$, $C'$ will fall outside the supporting face.

## 3.2 Structural Stability

We use the standard definition [Blum *et al.*, 1970] of structural stability, which is based on the concept of static equilibrium. An object at rest is in static equilibrium when the net force ($\sum \boldsymbol{f}_i$) and the net torque ($\sum \boldsymbol{\tau}_i$) of the object equal zero. A structure is stable when each object of the structure is in static equilibrium. Therefore, the static equilibrium of a structure can be expressed as a system of linear equations [Whiting *et al.*, 2009]:

$$\boldsymbol{A_{eq}} \cdot \boldsymbol{x} + \boldsymbol{w} = \boldsymbol{0} \tag{1}$$

$\boldsymbol{w}$ is a vector of weights and external torques on the objects, with $\boldsymbol{w}_i = (\boldsymbol{f_i^G}, \boldsymbol{\tau}_i^{ex})^T$. In most cases there will be no external torque on an object, i.e, $\boldsymbol{\tau}_i^{ex} = \boldsymbol{0}$. $\boldsymbol{x}$ is the vector of unknowns representing the magnitude of the forces within the structure. All forces in $\boldsymbol{x}$ are contact forces given that the forces only occur at the contact surface. To identify the forces on a contact region, we create a triplet of forces $\{\boldsymbol{f^n}, \boldsymbol{f^{s_1}}, \boldsymbol{f^{s_2}}\}$ at each vertex of the contact region: $\boldsymbol{f^n}$ is the normal force perpendicular to the surface of the contact; $\boldsymbol{f^{s_1}}, \boldsymbol{f^{s_2}}$ are static friction forces with the directions along the two edges joining at the vertex (Fig. 1a). The forces are sufficient to model the force distribution across the contact region [Whiting *et al.*, 2009].

$\boldsymbol{A_{eq}}$ is the coefficient matrix for the static equilibrium of a structure:

$$\boldsymbol{A_{eq}} = \begin{array}{c} \\ o_1 \\ o_2 \\ \vdots \\ o_m \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \cdots & c_n \end{array} \\ \begin{pmatrix} \boldsymbol{a}_{11} & \boldsymbol{a}_{12} & \cdots & \boldsymbol{a}_{1n} \\ \boldsymbol{a}_{21} & \boldsymbol{a}_{22} & \cdots & \boldsymbol{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_{m1} & \boldsymbol{a}_{m2} & \cdots & \boldsymbol{a}_{mn} \end{pmatrix} \end{array} \tag{2}$$

Each row $o_m$ in Eq. 2 stands for a particular object while each column $c_n$ stands for a contact between two objects. The entry $\boldsymbol{a}_{ij}$ lists the coefficients of the contact forces acting on $o_i$ if $o_i$ is involved in $c_j$( otherwise $\boldsymbol{a}_{ij} = \boldsymbol{0}$), therefore, $A_{eq}$ is a sparse matrix as each contact ($c_i$) column will only involve at most two non-zero $\boldsymbol{a}_{ij}$ entries.

There are two additional constraints on the forces, namely, the non-negative normal force constraint (C1) and the Coulomb's friction constraint (C2). In our domain, there is no attraction force between objects. Therefore we use C1 to constrain the normal forces to be non-negative so that the forces are always separating two objects away (not attracting the objects together e.g. like glue). C2 approximates the static friction using the Coulomb model. By adding the two constraints to Eq. 1 we have the final static equilibrium equations:

$$\begin{aligned} \boldsymbol{A}_{eq} \cdot \boldsymbol{x} + \boldsymbol{w} &= \boldsymbol{0} & \\ \|\boldsymbol{f}_i^{\boldsymbol{n}}\| &\geq 0 & C1 \\ \|\boldsymbol{f}^{\boldsymbol{s_1}}\|, \|\boldsymbol{f}^{\boldsymbol{s_2}}\| &\leq \mu \|\boldsymbol{f}_i^{\boldsymbol{n}}\| & C2 \end{aligned} \tag{3}$$

We are now ready to formally define structural stability.

**Definition 5** (Structural Stability)**.** *A structure $S_{\boldsymbol{O}}$ is stable when there is a solution to Eq. 3. $\mathfrak{S} : \boldsymbol{S} \rightarrow \{0, 1\}$ is a function that determines the stability of a structure. $\mathfrak{S}(S) = 1$ if $S \in \boldsymbol{S}$ is stable or $\boldsymbol{O} = \emptyset$, otherwise $\mathfrak{S}(S) = 0$.*

In most cases, the number of the unknown forces in $\boldsymbol{x}$ will be greater than the number of the equations, which makes Eq. 3 indeterminate, i.e., there is an infinite number of solutions. A solution to Eq. 3 does not imply the stability of the structure in reality because the solution might not be instantiable with real world physics. However, assuming that the friction model is correct then the inconsistency of the equation implies the instability of the structure [Blum *et al.*, 1970].

By solving Eq. 3, a robot can figure out whether a structure is stable or not. However, a "yes"/"no" answer to stability is insufficient. To safely transport the stacked objects the agent also needs to ensure that the structure is stable enough to be carried, i.e., the robot should be able to tell "how" stable a structure is. In everyday scenarios, there are various factors that will affect the stability of the structure during transportation. For example, the structure can be tilted significantly when the robot moves on uneven ground or when the robot manoeuvres to avoid collisions. Therefore, it is desirable to build structures that can remain stable up to a realistic tilt angle $\theta$.

**Definition 6** (Structure Tilt)**.** *A structure S, rotated by a tilt angle $\theta$ around a tilt axis $\hat{a}$ is denoted as $\mathfrak{T}^{(\theta,\hat{a})}(S)$.*

We quantify the "robustness" of a structure based on the maximum tilt angle we can apply to it without the structure becoming instable. Formally,

**Definition 7.** *Maximum Tilt Angle of S ($\theta_S^{max}$)*

$$\forall \hat{a}, \mathfrak{S}(\mathfrak{T}^{(\theta,\hat{a})}(S)) = \begin{cases} 1, \theta \in [-\theta_S^{max}, \theta_S^{max}] \\ 0, otherwise \end{cases} \tag{4}$$

In this paper, we only use ±X/Y-axis as rotation axes $\hat{a}$ for the evaluation. We now formalize the structure designing problem.

**Definition 8.** *(The Structure Designing Problem (SDP)) Given a set $\boldsymbol{O}$ of $k$ objects on the ground and a tilt angle $\theta \in [0, \frac{\pi}{2})$, is there a sequence of actions $\{\langle \mathfrak{T}_i, o_i \rangle, 1 \leq i \leq k\}$ such that:*

$$\begin{gathered} o_1 \textit{ is the tray} \\ \theta^{max}_{S_{\boldsymbol{O_k}}} \geq \theta \\ \mathfrak{S}(S_{\boldsymbol{O}_{n\in[1,\ldots,k]}}) = 1 \end{gathered} \tag{5}$$

*where $\boldsymbol{O}_n = \{o' | o' = \mathfrak{T}_i(o_i), o_i \in \boldsymbol{O}, i \in [1, \ldots, n]\}$?*

The last condition requires that each intermediate structure during the construction should also be stable. An instance of the SDP problem is denoted as SDP$\langle \boldsymbol{O}, \theta \rangle$.

### 3.3 Structure Designing is Hard

We show that the computational complexity of SDP is NP-hard by reducing the NP-hard circle packing problem [Demaine *et al.*, 2010] to SDP.

**Definition 9** (Circle Packing)**.** *Given a set of circles $\boldsymbol{Q}$ of different sizes and a square $T$, decide whether it is possible to pack the circles in the square so that the circles are inside the square and none of the circles are overlapping.*

**Theorem 1.** *The structure designing problem is NP-hard.*

*Proof.* We show that an instance $\langle \boldsymbol{Q}, T \rangle$ of CPP can be reduced to SDP in polynomial time. We set the tilt angle $\theta$ of the problem to $\frac{\pi}{4}$. For each circle $q_i \in \boldsymbol{Q}$, we create an object $o^{q_i}$ with the shape of a cylinder whose radius $r$ is the same as $q_i$. The height of $o^{q_i}$ is set to $2r \tan\theta = 2r$. The geometry of the object allows the object itself to remain stable with a tilt angle less than $\theta$, when $o^{q_i}$ is standing on its base that is fully contained within a supporting surface. Assuming there is a sufficiently large friction between the contacting faces, the object will only start to topple as the tilt angle becomes greater than $\theta$ when the projection of its centroid falls outside the supporting surface ( Fig. 1b).

We then create an object $o_{tray}$ with the cuboid shape bounded by the squares of the same dimension as $T$. We pick up an arbitrary face $F_{tray}$ of $o_{tray}$ and one arbitrary base from each object $o^{q_i}$, and let the static friction be non-zero only between the tray and the other objects. The friction coefficient is set to a sufficiently large value so that when being tilted $o^{q_i}$ will first start to topple rather than to slide. Following this procedure, the reduction is done in polynomial time with the time complexity $O(|\boldsymbol{Q}| + 1)$.

There will be a solution to SDP$\langle o_{tray} \cup \boldsymbol{O}^{\boldsymbol{Q}}, \theta \rangle$ if and only if there is a solution to $\langle \boldsymbol{Q}, T \rangle$. For the "only-if" part, we observe that to create a structure that can remain stable with the tilt angle $\theta$, the only possible way is to place $o^{q_i}$ immediately on $o_{tray}$. Otherwise, for example, the structure with the cylindrical objects stacked on top of another cannot sustain any tilt because there is no friction between them by the setting. The base of each object has to be fully contained within the surface of tray otherwise the object will start to topple before the tilt angle reaches $\theta$. As the objects are solid, there will be no overlapping between objects. The positions of the base of each object and the supporting face of the tray in the resulting structure is a solution to the original circle packing problem. For the "if" part, a solution to a CPP problem can be straightforwardly transformed to a solution to the SDP problem by placing $O^{q_i}$ with its base at the same position as $q_i$. □

## 4 An Effective structure designing Algorithm

We developed an algorithm that uses *forward search* augmented with *backwards adjustment* to find a solution. Each node of the search tree represents a stack of objects that have been placed. It expands a node by adding an object to the stack. The search starts with selecting the tray according to *tray selection policy*. It always adds an object on top of a supporting face $F_{\uparrow}$ of another object. A node is labelled as a *dead-end* if the stack is unstable. The algorithm checks the stability of the stack by rotating it around the ±X-axis and the ±Y-axis with tilt angle $\theta$. The contact regions can be directly retrieved from the *spatial representation*. When expanding a node, the forward search selects the next unexplored object $o$ using the *object selection policy*, and chooses a *supporting surface ($F_{\uparrow}$)* on which the object is placed with the *placement selection policy*. It uses the *face selection policy* to determine which face $F_{\downarrow}$ of the object will be supported by $F_{\uparrow}$. If there is an existing object in the stack that requires support from its side faces $\boldsymbol{F_{i\leftarrow}}$, the face selection policy will choose placements that make a side face $F_{\rightarrow}$ of $o$ contacts$F_{i\leftarrow}$. It backtracks if $o$ cannot support any of the unsupported side faces. The algorithm will only verify stability when there are no unsupported objects in the stack. Whenever the algorithm comes to a dead-end, it will identify the object that contributes most to the instability of the structure. Then it will backtrack to the node where the object has been added, adjust the pose of the object, and then continue the forward search from that node. The search keeps expanding nodes until it finds a solution or when there are no more expandable nodes.

### 4.1 Spatial representation

We represent the shape of an object as a set of faces (convex polygons). A polyhedron is represented as a set of its bounding faces. The representation of a cylinder (or cone) contains the minimum bounding rectangles of its bases and another four equally-spaced auxiliary polygons at the side of the cylinder. We use the auxiliary polygons to approximate the region where other objects can contact and support the cylinder. The structure of a stack is represented as a directed graph $\langle V, E\rangle$ with each vertex $v$ representing an object $o$ in the stack. There is an edge from $v_1$ to $v_2$ if $o_1$ supports $o_2$ via a top-down surface contact. The edge is labeled as a pair of the contact faces $(F_{o_1\uparrow}, F_{o_2\downarrow})$. The graph is acyclic as we always add an object on top of another. We use a vertex $v_g$ to represent the ground. We define the *support depth* of an object $o$ as the length of the longest path from $v_g$ to $v_o$. We update the support graph whenever an object is added to or removed from the stack.

### 4.2 Forward search

**Face selection policy (FSP).** Given an object standing on its face $F_{\downarrow}$, the *critical angle* at which the object starts to topple is given by $atan(\frac{r_c}{d(c,F_{\downarrow})})$, where $d(c, F_{\downarrow})$ is the distance between the centroid $c$ and its image $c_{F_{\downarrow}}$ of the orthogonal projection onto the plane of $F_{\downarrow}$, and $r_c$ is the *supporting radius* whose length is the distance between $c_{F_{\downarrow}}$ and the closest edge or corner of $F_{\downarrow}$. When $c_{F_{\downarrow}} \notin F_{\downarrow}$, we set the angle to zero.

**Definition 10.** *(Local stability) Given a tilt angle $\theta$, an object o, its supported face $F_{o\downarrow}$ and the supporting face $F_{\uparrow}$ , let $D_o^{\theta}$ be the closed disk of center $c_{F_{\downarrow}}$ and radius* $\tan\theta \cdot d(c, F_{\downarrow})$, *o is locally stable if $D_o^{\theta} \subset F_{\uparrow} \cap F_{\downarrow}$.*

The disk $D_o^{\theta}$ outlines the region where the vertical projection of the centroid will fall into with any tilt angle less than $\theta$. Therefore, $D$ has to be covered by the contact region otherwise, a toppling may happen. We denote the disk resulting from the critical angle as $D_o^{max}$. We sort the faces by their critical angles so that the face ($F_{\downarrow}^{max}$) with maximum critical angle $\phi^{max}$ will be searched first. When $\phi^{max} < \theta$, which means the object (*self-unstable object*) cannot remain stable only with the support from $F_{\downarrow}^{max}$. To support a self-unstable object, we identify the set $\boldsymbol{F_{\rightarrow}}$ of side faces where support should be given when $o$ is standing on $F_{\downarrow}$. $F_{i\rightarrow}$ is the face that shares an edge with $F_{\downarrow}$ and the minimum distance between $c_{F_{\downarrow}}$ and the edge is less than the radius of $D_o^{\theta}$. $F_{\downarrow}^{max}$ refers to the face requring the fewest side supports.

**Tray and object selection policy (OSP).** The algorithm will choose the object that has the largest supporting radius as the tray, and use $F_{tray\downarrow}^{max}$ as the initial supporting surface. The algorithm sorts the set $\boldsymbol{O}$ of the remaining of unplaced objects by making pairwise comparison between them: Given two self-stable objects $o_1$ and $o_2$, $o_1$ will be placed earlier than $o_2$ if putting $o_1$ on $o_2$ is more stable than vice versa. We measure the stability of the structure of $o_1$ on top of $o_2$ by calculating the height of the weighted centroid of the structure:

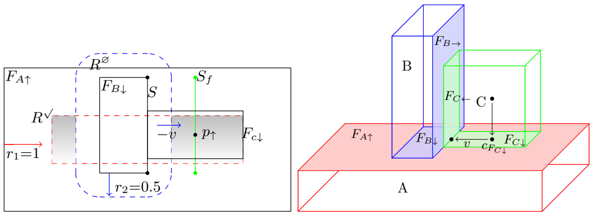
$$H(o_1, o_2) = \frac{w_{o_1} \cdot d_1 + w_{o_2} \cdot (d_1 + d_1' + d_2)}{w_{o_1} + w_{o_2}} \tag{6}$$

where $d_{1/2} = d(c_{o_{1/2}}, F^{max}_{o_{1/2}\downarrow})$ and $d_1' = d(c_{o_1}, F_{o_2\uparrow})$. We write $o_1 \leq o_2$ if $H(o_1, o_2) \geq H(o_2, o_1)$ or $o_2$ is a self-unstable object while $o_1$ not. It can be proven that the relation $\leq$ is a total order on $\boldsymbol{O}$. This ordering will make the algorithm try to place self-unstable objects first. The reason is that those objects are most likely to cause instability of a stack, therefore, we want to secure them first before stacking other objects.

**Placement selection policy (PSP).** If there are no self-unstable objects in the current stack, we choose the flattest face from each of the placed objects as candidates of supporting faces $\boldsymbol{F_{\uparrow}}$ , and sort them in ascending order of their support depth. We prefer supporting faces at a smaller depth because if there is an adjustment required, it will affect fewer objects.

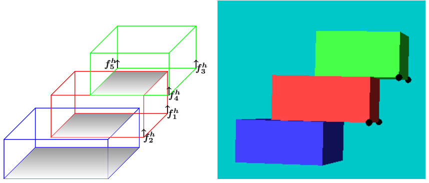
A placement is defined as a pair of a point $p_{\uparrow} \in F_{\uparrow}$ and the orientation of the object. There are infinitely many point locations on $F_{\uparrow}$, and some of the locations are physically or spatially infeasible for placement. We first find the physically-sound region $R^{\surd}$ by insetting $F_{\uparrow}$ with the radius of $D_o^{\theta}$. $o$ is locally stable with any placement of $p_{\uparrow} \in R$. We then obtain the set of objects that are supported by $F_{\uparrow}$ from the support graph, which gives a set $\boldsymbol{F_{\downarrow}}$ of the supporting faces that already contact $F_{\uparrow}$. We offset $F_{i\downarrow} \in \boldsymbol{F_{\downarrow}}$ (Minkowski sum) by the radius of $D_o^{max}$. This forms a spatially infeasible region $R_i^{\varnothing}$ within which any placement of $o$ will cause an intersection with the other placed objects. The final feasible region is given by $R^f = R^{\surd} - \bigcup R_i^{\varnothing}$ (Fig. 2a). We sample $k$ placement points uniformly from $R^f$, and try two orientations of the object by rotating it at the angle of $\frac{\pi}{2}$ and $\pi$ around the direction vector from $c_{F_{\downarrow}}$ to $c$. Given $n$ objects in the stack, we will test at most *2kn* possible placements for each face of the object. To reduce it, we use only the face $F_{\downarrow}^{max}$ for placement.

If there is an unsupported self-unstable object on a face $F_{\uparrow}$, we will pick up an unsupported side face $F_{\leftarrow}$ from it and choose a side face $F_{o\rightarrow}$ from $o$ that shares the same edge $e_o$ as $F_{o\downarrow}$. We align $F_{o\downarrow}$ with $F_{\uparrow}$ and align $F_{o\rightarrow}$ with $F_{\leftarrow}$. We then obtain the line segment $S$ of the intersection between $F_{\uparrow}$ and $F_{\leftarrow}$, and obtain the vector $v$ from $c_{F_{o\downarrow}}$ to its closet point on $e_o$ (see Fig. 2b). We translate $S$ with $-v$, the resulting segment $S^f$ is an outer-approximation of the region of possible placements that make $F_{o\downarrow}$ contact $F_{\uparrow}$ and make $F_{o\rightarrow}$



(a) (b)

Figure 2: (a) The top-down view of a placement selection for $F_{c\downarrow}$ on $F_{A\uparrow}$. The radius $r_1$ of $D_c^\theta$ is 1, the maximum supporting radius $r_2$ of $C$ is 0.5. $R^f$ is the shaded area. We show a placement at $p_\uparrow$ with $F_{C\rightarrow}$ touching $F_{B\leftarrow}$. (b) the resulting stack.



(a) (b)

Figure 3: (a) We show each $\boldsymbol{f_i^h}$ in the stack. The blue object is on the ground. The shaded regions are the contact surfaces. (b) Visualization of $\boldsymbol{f_h^{act}}$ (black dots) found by the algorithm. The red is the saboteur.

contact $F_{\leftarrow}$. We sample k placement points from $S^f \cap R^f$. We test every possible combination of side faces. i.e, Given $n$ unsupported side faces, and $m$ available side faces of $o$, there are at most $kmn$ possible placements. The policy will discard any placement that cause an intersection with other objects.

### 4.3 Backwards adjustment

When verifying the stability of a structure, we aim to find out places where lacking supporting forces can cause instability. Possible places are the unsupported vertex points of $F_{o_i\downarrow}$ of each placed object $o_i$. We add hypothesized normal forces $\boldsymbol{f_i^h}$ at each identified vertex point, which can be viewed as additional contact points at the contact surface (Fig. 3a). We create a quadratic program by adding $\boldsymbol{f_i^h}$ as unknowns to Eq.3 We solve the following quadratic program:

$$\begin{array}{lc} \min_{\boldsymbol{f^h}} & \sum \|\boldsymbol{f_i^h}\|^2 \\ \text{such that} & \boldsymbol{A}_{eq} \cdot \boldsymbol{x} + \boldsymbol{w} = \boldsymbol{0} \\ & \|\boldsymbol{f_i^n}\| \geq 0, \|\boldsymbol{f_i^h}\| \geq 0 \\ & \|\boldsymbol{f^{s_1}}\|, \|\boldsymbol{f^{s_2}}\| \leq \mu \|\boldsymbol{f_i^n}\| \end{array} \tag{7}$$

which minimizes the square sum of the magnitude of the hypothesized normal forces. We say $\boldsymbol{f_i^h}$ is *activated* when $\|\boldsymbol{f_i^h}\| > 0$. The objective value of the program will be zero when a structure is stable without any $\boldsymbol{f_i^h}$ being activated. When Eq.7 has a solution and the objective value is non-zero, we can obtain the set $\boldsymbol{f_h^{act}}$ of the activated forces (Fig. 3b). The *saboteur* is an object that is supported by forces in $\boldsymbol{f_h^{act}}$ and has the lowest support depth (if there are more than one such objects, we choose the one at the lowest depth of the search tree). The algorithm then backtracks to the node where the object has been added and perform the adjustment.

The adjustment is a local optimization procedure that maximizes the area of $D_o^{max} \cap F_\uparrow$ by iteratively translating the object on the plane $P_\uparrow$ of $F_\uparrow$ without intersecting any other objects. To prevent it from over-adjusting an object, once an object $o$ has been adjusted, we label the node of $o$ as a *frozen node*. The algorithm will not add any hypothesized forces at $o$ to Eq. 7 in the later search. A node will be *unfrozen* when it has been visited again via a normal backtrack.

| | | Cuboid | | | Combo.1 | | | Combo.2 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| M1 | 10 | √ | √ | 10 | √ | √ | 3 | 9 | 0.99 | 11 |
| M2 | 10 | 5 | 0.93 | 17 | - | - | - | - | - | - |
| M1 | 15 | √ | 0.99 | 76 | 9 | 0.97 | 53 | 9 | 0.95 | 62 |
| M2 | 15 | 1 | 0.82 | 32 | - | - | - | - | - | - |
| M1 | 20 | 9 | 0.90 | 192 | √ | 0.93 | 166 | 9 | 0.92 | 135 |
| M2 | 20 | 1 | 0.78 | 89 | - | - | - | - | - | - |
| M1 | 25 | √ | 0.92 | 259 | √ | 0.97 | 169 | 8 | 0.96 | 163 |
| M2 | 25 | 0 | 0.97 | 802 | - | - | - | - | - | - |

Table 1: #obj: number of objects #sim number (max=10) of valid solutions in Gazebo. √ indicates all are valid. #sol the average percentage of objects in a detected stacking plan (√ if a complete solution is detected for every instance). t: time usage in seconds. M1: our method. M2: the pallet stacking method.

## 5 Evaluation

We evaluate the method in Gazebo (gazebosim.org) which is a state of the art simulator used in robotics. The algorithm communicates with Gazebo via ROS (ros.org). To test the method, we generate a dataset that will be made public for future benchmarking purposes. The dataset contains 120 scenes with a varied number (denoted as #obj) of objects that have varied shape (cuboid or cylinder), size (large, medium or small), and density. We randomly sample dimensions of the object from pre-specified intervals for each size category. We choose density that approximates the density of wood, water, and stone. A scene contains 20% large-sized objects and 60% medium-sized objects. We prefer more medium-sized objects because small objects have less effect on the stack while having more large objects can substantially reduce the search space (just stack on top of another). Each scene has a static table on which the algorithm is expected to stack the objects. We run the algorithm with PSP sampling 5 possible placements per selection. We use CGAL (cgal.org) for the geometric computation and the quadratic programming. Dealing with friction is not the focus of the paper, therefore we use the same $\mu$ for all the objects and set it greater than the tilt angle $\theta$ so that instability will be caused by toppling rather than sliding (when $\mu < \theta$, objects can be treated in the same way as self-unstable objects). We set the time limit per problem instance to 5 minutes. After the timeout, the algorithm will terminate once it finished expanding the last node and will return the complete solution or if not found, the best partial solution (that stacks the most objects). Therefore, this algorithm is also an anytime algorithm. We verify a solution in Gazebo by placing the object one by one in the same order they were added during the stacking. After the stacking, we tilt the table and the stack around each of the rotation axes at the angle of 0.4 radians (maximum pedestrian ramp slope recommended by ADA [1990] is 0.08 radians). A solution is *valid* if the stack is stable in all cases.

When a scene contains fewer than ten objects, the algorithm can usually find a complete solution (if it exists) within ten seconds. Here we focus our evaluation on harder scenes (#obj $\geq$ 10). We generated three groups of scenes with different

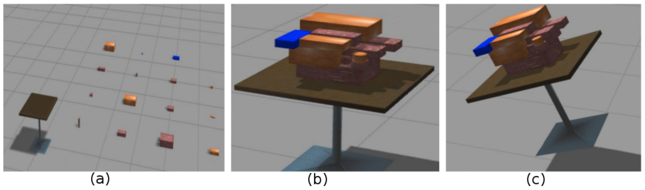


Figure 4: (a): initial state of a scene in Combo.2 with 15 objects. (red: stone, blue: water, yellow: wood). The long slim cylinder is the self-unstable object.(b-c) a completed stack before and after one tilt

combinations of objects, namely, all cuboid (Cuboid). 30% cylinder (Combo.1), and 30% cylinder with self-unstable objects (Combo.2). For scenes that have 25 objects, to make a scene solvable, we added a large cuboid which has the maximum allowed width of the category "large". The result is summarized in Table. 4. We generated ten scenes for each entry of the table. When #obj $\leq$ 30, the method can stack more than 90% of objects within the time limit. It usually stacks fewer objects when self-unstable objects are present. The Cuboid group is harder than other groups. Because cylinders of the same size category often occupy less space than cuboids and a cylinder only has one orientation to test. Therefore, having more cuboids makes PSP spend more time on finding valid placements.

We compare our method $M1$ with the pallet stacking method $M2$ presented in the IROS paper [Schuster *et al.*, 2010]. $M_2$ uses nest beam search with local and global optimization procedures. Since their method is designed for handling cuboid-shaped packages, we run their method on the scenes of the Cuboid group. We set $k_{local}$ = $k_{global}$ = 10 and use resolution of 1cm, which is a standard setting mentioned in the paper. We use OSP to find a tray as the "pallet" for the method. We let $M2$ run until it finishes stacking. The result shows, in most cases, our method outperforms $M2$. For scenes of $\#obj \leq 20$, our method can stack more objects than $M2$. It is because $M_2$ uses beam search that has a high chance of being trapped in local minima if the employed heuristic fails. Since $M2$ discretizes the pallet into a finite set of grids, in each iteration of the search, $M2$ will only consider placement on these grids. Therefore, it can hardly escape from local minima. $M2$ can stack more objects when $\#obj = 25$. It may be because we added the large cuboid to this group of scenes, which makes the setting more like a pallet stacking problem for which $M2$ is designed. In all the cases, the stack created by $M2$ is less stable than $M1$. It implies $M2$ can only deal with vertical stability.

## 6 Conclusion and Future Work

We formalised and solved a structure designing problem that can be applied to various areas where stability reasoning is desired. We believe it is an important first step towards building intelligent systems that can successfully interact with the physical world. Previously, a common solution employed by the robotics community was to check the stability in simulation as anything else would be prohibitively expensive. As most of the solutions found by our method are valid in the simulator, it makes reasoning about structural stability feasible for robotics.

The next step is to extend the method to account for the grippers of a robot, and integrate the method with robot motion planning frameworks. The support graph of a solution can be used as the input to the existing symbolic manipulation planners. Our method can still be useful with concave objects if we can approximate the centroid of objects with reasonable accuracy. To handle uncertainty in visual perception, we will use an outer-approximation (e.g. a minimum bounding polyhedron) of the object for placement selection while only compute the contact surface for regions that are of high certainty.

## References

[ADA, 1990] ADA. Americans With Disabilities Act of 1990 Public Law 101-336, 1990.

[Battaglia *et al.*, 2013] Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

[Blum *et al.*, 1970] Manuel Blum, Arnold Griffith, and Bernard Neumann. A stability test for configurations of blocks. Technical report, MIT, 1970.

[Davis and Marcus, 2016] Ernest Davis and Gary Marcus. The scope and limits of simulation in automated reasoning. *Artificial Intelligence*, 233:60–72, 2016.

[Demaine *et al.*, 2010] Erik D Demaine, Sándor P Fekete, and Robert J Lang. Circle packing for origami design is hard. *arXiv preprint arXiv:1008.1224*, 2010.

[Edelkamp *et al.*, 2014] Stefan Edelkamp, Max Gath, and Moritz Rohde. Monte-carlo tree search for 3d packing with object orientation. In *KI*, pages 285–296, 2014.

[Fahlman, 1974] Scott Elliott Fahlman. A planning system for robot construction tasks. *AIJ*, 5(1):1–49, 1974.

[Ge *et al.*, 2016] Xiaoyu Ge, Jochen Renz, and Peng Zhang. Visual detection of unknown objects in video games using qualitative stability analysis. *TCIAIG*, 8(2):166–177, 2016.

[Jia *et al.*, 2013] Zhaoyin Jia, Andrew Gallagher, Ashutosh Saxena, and Tsuhan Chen. 3d-based reasoning with blocks, support, and stability. In *CVPR*, 2013.

[Mojtahedzadeh *et al.*, 2015] Rasoul Mojtahedzadeh, Abdelbaki Bouguerra, Erik Schaffernicht, and Achim J Lilienthal. Support relation analysis and decision making for safe robotic manipulation tasks. *Robotics and Autonomous Systems*, 71:99–117, 2015.

[Schuster *et al.*, 2010] Martin Schuster, Richard Bormann, Daniela Steidl, Saul Reynolds-Haertle, and Mike Stilman. Stable stacking for the distributor's pallet packing problem. In *IROS*, pages 3646–3651. IEEE, 2010.

[Stephenson and Renz, 2016] Matthew Stephenson and Jochen Renz. Procedural generation of complex stable structures for angry birds levels. In *CIG*, 2016.

[Toussaint, 2015] Marc Toussaint. Logic-geometric programming: An optimization-based approach to combined task and motion planning. In *IJCAI*, 2015.

[Whiting *et al.*, 2009] Emily Whiting, John Ochsendorf, and Frédo Durand. Procedural modeling of structurally-sound masonry buildings. In *TOG*, volume 28, 2009.