Multi-Reward Policies for Medical Applications: Anthrax Attacks and Smart Wheelchairs

Harold Soh
haroldsoh@imperial.ac.uk

Yiannis Demiris
y.demiris@imperial.ac.uk

Dept. of Electrical and Electronic Engineering
Imperial College London, South Kensington Campus
London SW7 2AZ, United Kingdom

ABSTRACT

Medical decisions are often difficult; they involve uncertain information, multiple-objectives and debatable outcomes. In this work, we discuss the application of the multi-reward partially-observable Markov decision process (MR-POMDP) and NSGA2-LS, a hybridised multi-objective evolutionary solver, to two problems in the medical domain: anthrax response and smart-wheelchair control. For the first problem, we use a discrete model and analyse the trade-offs between the best solutions (in the form of finite-state controllers) found by our evolutionary algorithm. For the second, we contribute an extension of our method to the continuous space and optimising recurrent neural networks (RNNs) for use on medical robots such as smart wheelchairs.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; G.1.6 [**Optimization**]: Stochastic programming

General Terms

Algorithms

1. INTRODUCTION

Medical professionals often have to make decisions under uncertainty and with multiple criteria or objectives to consider. For example, during the early stages of a possible anthrax outbreak, information is scarce and available data may be unreliable. If the *only* objective was to minimise the loss of life in the short term, a "trivial" solution would be to spare no expense in distributing antibiotics, shutting down each potential exposure zone and containing all individuals suspected of being infected. However, in the real-world, economic and long-term societal costs are also important; one has to consider the probability that an outbreak really exists and what happens post hoc.

GECCO'11, July 12–16, 2011, Dublin, Ireland.

In medical robotics, we observe a similar general problem when trying to program intelligent controllers for smart wheelchairs; the user's preferences or desires when manoeuvring around his environment are often not known in advance. Furthermore, mechanical and electronic components can fail and the controller has to compensate for slipping wheels and sensor failure. An analogous problem also exists in diagnosis and treatment; tests are not 100% accurate, treatments can be costly and still fail. These examples are emblematic of the difficulties caused by multiple-objectives, incomplete information and ineffective actions, common in the health-care domain.

In prior work [18], we introduced the multi-reward partially-observable Markov decision process (MR-POMDP) as a general framework for modelling such problems. MR-POMDPs offer a convenient "language" for modellers who need to consider *both* multiple objectives and uncertainty. The solution for a MR-POMDP is not a single policy but the Pareto policy set: a set of "best" policies that maximise objectives to varying degrees.

While our previous paper mainly described performance results, this work focusses on MR-POMDPs for two medical problems. The emphasis is on analysing the the Pareto-policy set and solutions found by our evolutionary solver. We first study the multi-criteria anthrax response problem; after a qualitative discussion of the best found solutions, we extract selected candidate policies and simulated their execution on hypothetical anthrax outbreaks. We then discuss the expected trade-offs when applying such policies and considerations when using them as starting-points for designing practical solutions.

Second, we contribute an extension of our method to *continuous state spaces* by evolving recurrent neural networks (RNNs) as controllers for mobile robots, in particular smart wheelchairs. Our goal was to have the robot respond to user-preferences related to driving speed and power-consumption. We evolved non-dominated policies in simulation, transferred them on to a Pioneer P3-AT robot and performed a quantitative analysis on the driving behaviours of selected policies on a "shuttle-run" problem. With a Pareto optimal set of controllers, we envision smart wheelchairs that are able to change their behaviour "on-demand" to adapt to higher-level preferences.

In the next section, we briefly review the MR-POMDP and finite-state controllers (FSCs). In Section 2.4, we discuss our hybridised multi-objective evolutionary algorithm. Our analysis of the multi-criteria anthrax response problem is given in Section 3. Section 4 discusses additional prelimi-

nary results on optimising RNNs for continuous spaces and our proposed policy-switching method. We conclude with final remarks and future work in Section 5.

2. BACKGROUND

In this section, we briefly review the MR-POMDP, finite state controllers and our hybrid multi-objective evolutionary algorithm. We refer readers wanting more detail to [18].

2.1 The MR-POMDP Model

A MR-POMDP models an environment where states are not directly visible to an agent [12]. Instead, the agent makes observations, from which it has to infer actions to take. Depending on the underlying state and the action, the agent receives feedback in the form of a *vector*-valued reward; in standard POMDPs, agents only receive *scalar*-valued rewards.

Formally, a MR-POMDP is a tuple $\langle S, A, Z, T, O, \mathbf{R}, \gamma \rangle$ where S is the set of states, A is the set of possible actions and Z is the set of observations available to the agent. T is transition function $T(s, a, s') = P(s'|s, a)$ which gives the probability of moving to state s' from s given action a and O is the observation function $O(s, z) = P(z|s)$ i.e. the probability of observing z in state s. $\mathbf{R}$ is a vector of reward functions $\mathbf{R} = [R^{(1)}, R^{(2)}, \ldots, R^{(M)}]$ where each $R^{(i)} : S \times A \times S \rightarrow \mathbb{R}$ models the reward for arriving in state s' after executing action a in state s under objective i. Finally, the discount factor, γ $(0 \leq \gamma \leq 1)$ regulates how much future rewards are discounted[1].

Since observations only give partial information about the current state, an agent has to rely on the complete history of its observations and actions. We define a finite history as $h_t = \{a_o, z_1, \ldots, a_{t-1}, z_t\} \in H$ where a_t and z_t are the action and observation at time t respectively. A policy π maps elements of H to actions $a \in A$ (or a distribution of actions in the case of *stochastic* policies). That is, policies tell an agent what to do based on what it has seen up to that point. In this work, we seek policies that maximise the expected cumulative discounted reward for each objective:

$$\max \mathbf{E}\left(\sum_{t=0}^{\infty} \gamma^t R_t^{(i)} \mid \pi, \mathbf{b}_0\right) \text{ for } i = 1, 2, \ldots, M \qquad (1)$$

where $R_t^{(i)}$ is the reward received at time t under reward function $R^{(i)}$, M is the number of objectives and $\mathbf{b}_0$ is a given distribution over the starting states.

2.2 The Pareto Policy Set

With multiple rewards, the value of a given policy is a vector $\mathbf{E}_\pi = [E_\pi^{(i)}]$ where $E_\pi^{(i)}$ gives the value of the policy under reward function $R^{(i)}$. To determine the optimal policy (or policies), we need to be able to compare policy value vectors.

Intuitively, a policy is preferred over another if it possesses a higher value for at least one objective, and is no worse for all others. Formally, a policy π_k *dominates* policy π_l, denoted as $\pi_k \succ \pi_l$, if at least one of its value functions p is strictly better than that of policy l and none of its value functions are worse i.e. $E_{\pi_k}^{(i)} \geq E_{\pi_l}^{(i)}$ for all $i = 1, \ldots, M$ and there exists p such that $E_{\pi_k}^{(p)} > E_{\pi_l}^{(p)}$. In contrast, if $E_{\pi_k}^{(i)} \leq E_{\pi_l}^{(i)}$ for all i and there exists p such that $E_{\pi_k}^{(p)} < E_{\pi_l}^{(p)}$, then we say π_k is *dominated* by π_l, denoted $\pi_k \prec \pi_l$. Otherwise, π_k and policy π_l are *non-dominating*, $\pi_k \sim \pi_l$.

[1]For simplicity, we assume discount factors are equal across rewards

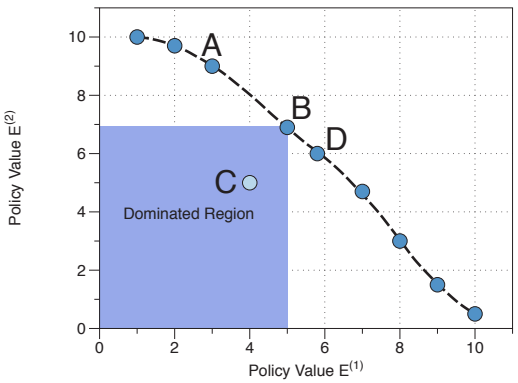

Figure 1: An illustration of dominance on a bi-objective problem. A, B and D are non-dominating solutions on the Pareto optimal front. C is dominated by B and D, but is *not dominated* by A.

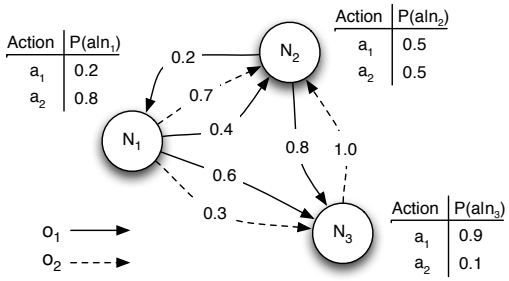

Figure 2: A FSC with three nodes $\{N_1, N_2, N_3\}$, two actions $\{a_1, a_2\}$ and two observations $\{o_1, o_2\}$. For each node, the action taken is governed by the probability distribution $\psi(n, a) = P(a|n)$. The transition from one node to another is dictated by another probability distribution $\eta(n', z, n) = P(n'|z, n)$. In this example, the probability of taking action a_1 in node N_1 is 0.2, after which a transition occurs depending on the observation received. If observation o_1 is received, a transition is made to N_2 with probability 0.4 or to N_3 with probability 0.6.

Given the above definitions, the best policies, $\pi_k^* \in \overline{\mathbf{P}}^*$, are those that are *not dominated* by any other policy; there does not exist π_l such that $\pi_l \succ \pi_k^*$. We call these policies *Pareto-optimal* and $\overline{\mathbf{P}}^*$ is the *Pareto optimal set*. The set of all value vectors for the policies in the Pareto optimal set is called the *Pareto optimal front*, $\overline{\mathbf{E}}^* = \{\mathbf{E}_{\pi*}\}$. Fig. 1 illustrates the concept of dominance and a sample Pareto optimal front.

Given a MR-POMDP, our goal is to find the Pareto optimal set of policies or *Pareto-policy set* (PPS). That said, we have not made clear what form or representation such policies should take. In this work, we begin with policy-search for finite-state controllers (FSCs) and later, consider recurrent neural networks (RNNs).

2.3 Finite State Controllers

A finite state controller (FSC) is a graph-based representation of a policy. Each node (also called a "memory state") dictates an action to take and depending on the observation received, we transition to another node in the graph (which defines the next action to be taken and so on). We work mainly with *stochastic* FSCs where each node defines a probability distribution over possible actions and nodes to transition to. As an example, a three-node FSC is shown in Fig. 2.

2.4 Multi-Objective Hybrid EAs

In [18], we introduced two hybrid multi-objective evolutionary algorithms (MOEAs) based on NSGA2 [16] (as a representative of MOEAs using standard real-coded recombination and mutation operators) and MO-CMA-ES [9], which represented the more recent estimation of distribution (EDA) class of methods. In this work, we focus on the hybridised NSGA2 algorithm (NSGA2-LS) as it was the best performing method in our experiments.

NSGA2-LS is a memetic "steady-state" version of NSGA2 that incorporates a specialised local-search operator for FSCs. Similar to other MOEAs, NSGA2-LS iteratively generates new solutions using the simulated binary crossover (SBX) and polynomial mutation operators [16]. At each iteration, the population (together with the offspring) are sorted using a fast non-dominated sorting algorithm. Each solution is assigned a rank (lower ranks are better) and a second preference criteria, *crowding distance*, which approximates the density of solutions around the individual. NSGA2-LS is *elitist* in that it preserves only the top $|\mathbf{P}|$ solutions in the population $\mathbf{P}$ at each iteration. In addition, NSGA2-LS uses a dynamic operator selection scheme based on operator rewards similar to [1]. Intuitively, we want more successful operators to be used more frequently. The reward given to each operator is a *cost-benefit* ratio where the "benefit" of using a particular operator is defined as the proportion of solutions the offspring is better than *relative to its parent* and the "cost" is simply the processing time used by the operator.

2.4.1 FSC Representation

Each FSC is represented with a vector (genome) $\mathbf{x} \in \mathbb{R}^{|N||A|+|N|^2|Z|}$. There are two segments to this genome for the action selection distribution $\psi(n,a)$ and node transition distribution $\eta(n,z,n')$ respectively. We refer to segments of the genome by $\mathbf{x}^\psi$ and $\mathbf{x}^\eta$. To ensure that probability distributions remained valid and the resulting evaluation function was differentiable, we use the soft-max function, e.g. $\psi(n,a) = P(a|n,\mathbf{x}^\psi) = \exp(\mathbf{x}^\psi[n,a])/Q$ where $\mathbf{x}^\psi[n,a]$ is the associated variable for $\psi(n,a)$ and Q is the normalisation factor.

2.4.2 Hybridisation with Local Search

The FSCs we are attempting to optimise are large in the number of parameters; as stated, the genome consists of $|N||A|+|N|^2|Z|$ real variables. Multi-objective EAs may be slow to converge in such large search spaces and may not find solutions with sufficient precision. A potential solution to this problem are *hybrid* or *memetic algorithms* [13, 14] that combine MOEAs with local-search methods; the intuition being that local-search can quickly locate good solutions that the evolutionary operators can build upon. We hybridised both NSGA2 and MCMA with gradient-based local-search as an operator and to keep our search computationally-feasible, we transformed the MR-POMDP to a POMDP with reward function $R^{\mathbf{w}}$ using a weighted-linear combination approach[2]. We use a weight vector $\mathbf{w}$ drawn from a uniform distribution and apply an efficient conjugate-gradient method [7] until convergence or for a maximum of $5|\mathbf{x}|$ iterations.

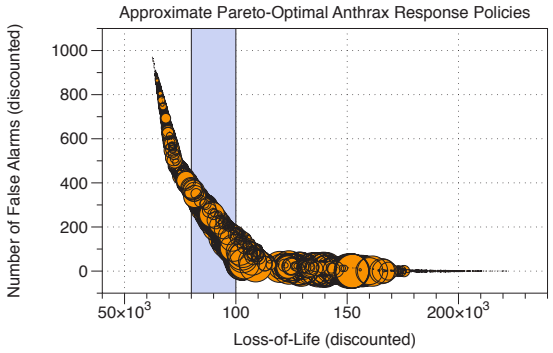

Figure 3: The approximate Pareto policy set for the multi-criteria anthrax problem with three objectives to minimise: the loss of life (x-axis), the number of false alarms (y-axis) and the cost of investigation (size of circle).

3. MULTI-CRITERIA ANTHRAX RESPONSE

The problem of anthrax outbreak detection was formulated as a POMDP by Izadi and Buckeridge [10] alongside public health experts. This POMDP is comprised of six states ("normal", " outbreak day 1" to "outbreak day 4" and "detected") with two observations ("suspicious" and "not suspicious"), four actions ("declare outbreak", "review records", "systematic studies" and "wait") and a relatively complex reward function that combined the economic costs from multiple sources such as productivity loss, investigative costs, hospitalisation and medical treatment. We refer readers to [10] for complete details of the POMDP and the results found using a single reward function.

In our multi-objective formulation, we have three objectives to minimise: loss of life (R^l), number of false alarms (R^a) and cost of investigation (in man-hours, R^m). The loss of life is a cumulative cost based on the number of deaths in the event of a real outbreak. The number of false alarms increased by one when an outbreak was wrongly declared and finally, the cost of investigation is computed based on the man-hours spent on reviewing records and systematic studies (systematic studies were more intensive and hence, more costly).

We evolved solutions using NSGA2-LS (15 runs, 3600 seconds of computational time for each run) and extracted a final non-dominated set as our approximate Pareto policy set (PPS). Note that we do not claim these to be optimal solutions but instead, consider these solutions to be starting points for developing real-world policies.

[2]Transformation details in [18]

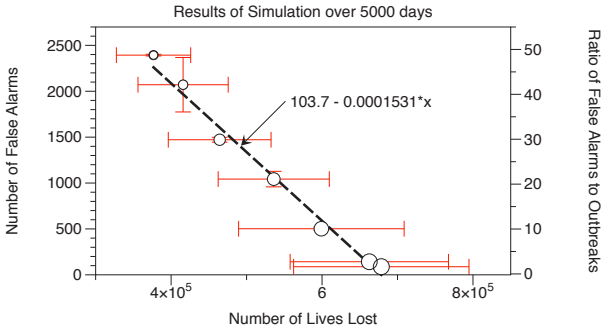

Figure 4: Seven policies from the Pareto policy set simulated over 5000 days. Error bars show the standard deviation across 50 iterations. Size of the bubbles indicate the number of man hours spent in investigative costs. The dashed line is a linear function fitted to the *ratio* of false alarms v.s. the number of lives lost (in thousands) to real outbreaks. We noticed a negative linear relationship between the loss-of-life to the number of false alarms; for every 1000 lives gained, we would have to tolerate an increase of approximately 1.5 in the ratio of false alarms to outbreaks.

3.1 Qualitative Analysis

Let us begin with a qualitative analysis of the PPS (shown in Fig. 3) and discuss the overall trade-offs between the different policies. We observe that the front consists of three "pieces", each with a seemingly linear trade-off between R^l and R^a.

As stated in the introduction, if one only wishes to minimise the loss-of-life, a trivial solution is to declare an outbreak whenever one is asked; no man-hours need to be spared (the number of lives lost is still non-zero as although an outbreak may be declared, the original infected may not survive). One observes this policy at the left extremum of the graph where as expected, the number of false alarms is at a maximum.

At the other end of the Pareto-policy front, we observe another trivial policy: we can get away with declaring no outbreaks (simply by waiting) and thereby reducing the number of false-alarms to zero. But as we might expect, the loss-of-life is at the maximum.

For real-world use, we consider the most interesting part of the PPS is "middle" portion between these two extremes, where some work is required to balance the two objectives R^l and R^a. We can decrease the number of false alarms at the expense of more investigative hours, illustrated by the size of the circles. From the plot, there appears to be a linear increase in the number of hours required to reduce the number of false alarms without drastically increasing the loss of life. Of course, there are trivial policies that simply toss a (biased) coin as to whether to declare an outbreak or not but those do not perform as well as those that make use of our dual tools of systematic studies and record reviews.

3.2 Selected Policy Simulation and Comparisons

We selected seven different policies along the middle portion of the policy front (blue area in Fig. 3) and simulated the policies over the course of 5000 days (similar to [10]). Each simulation was repeated 50 times and the number of lives lost, false alarms and investigative man-hours were averaged over the runs. Fig. 4 shows the results obtained.

As before, we observe a linear relationship between the number of false-alarms and total number of lives lost. A fitted linear function (dashed line) shows that for every 1000 lives gained, we would have to tolerate an increase of approximately 1.5 in the ratio of false alarms to outbreaks. Another detail to consider is that there is a significant variability in the outcomes across the simulations with the same policy. This is to be expected since stochasticity and unknown variables play a significant role. That said, the MR-POMDP model and the PPS allow us to consider these aspects to make informed judgements about which policy would best suit societal values.

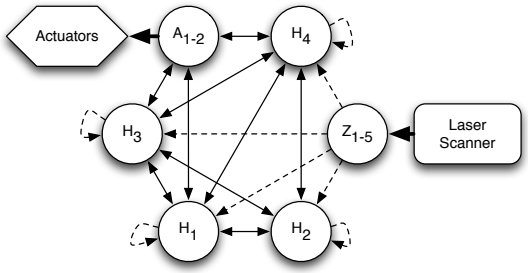

Figure 5: A RNN with four hidden nodes $\{H_1, \ldots, H_4\}$, two action nodes $\{A_1, A_2\}$ and five observational nodes $\{Z_1 \ldots, Z_5\}$. The action and observational nodes represented as single nodes each since the compositional nodes do not interact. The directional dashed lines are associated with a single weight variable where-else the bi-directional solid lines are associated with two weights. Note that hidden nodes also have self-loops.

4. CONTINUOUS SPACES AND POLICY SWITCHING FOR MEDICAL ROBOTS

Moving beyond discrete-space test problems [18], we applied our method to more a complex multi-reward scenario with continuous observation and action spaces. In particular, we were interested in generating "inverse-models" that produce velocity commands (both translational and rotational) in a shared-control system on a smart wheelchair [3, 17]. Prior work focussed on optimising safety but we wished to model situations where users may have two additional, conflicting, goals: speed and power consumption.

4.1 Recurrent Neural Networks (RNNs)

Since FSCs do not generalise easily to continuous domains, we optimised policies in the form of recurrent neural networks (RNNs) with sigmoidal activation functions. RNNs are similar to the canonical feedforward artificial neural networks except that *feedback connections* are present. This allows RNNs to exhibit complex temporal behaviour using internal memory states. An example RNN is shown in Fig. 5.

Both FSCs and RNNs are graph-based policy representations, hence only minor modifications to our MOEA was re-

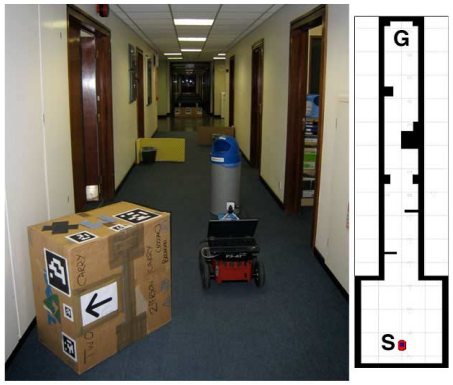

Figure 6: The real-world (left) and simulated (right) obstacle course for the Shuttle-Run Problem. The robot has to race between S and G repeatedly, not bump into obstacles and conserve power. This problem is partially observable with continuous state, observation and action spaces.

quired. Each RNN is a vector $\mathbf{x} \in \mathbb{R}^{|N_H|(|N_H|+|A|+|Z|)}$ where N_H is the set of hidden nodes, A is the set of actions and Z is the set of observations. The most substantial change was in the local-search algorithm; since gradient information was not easily obtained, we opted for a simple greedy local-perturbation search where solutions undergo 100 iterations of polynomial mutation and evaluation. Future work would involve other learning algorithms such as back-propagation through time [15].

For this experiment, we optimised *fully-recurrent* RNNs of up to 4 hidden nodes. Our test platform was Art: a Pioneer P3-AT equipped with a SICK laser. The laser's maximum range was limited to 4 meters and its field of view was divided into five segments with the minimum reading in each segment fed into Art's RNN. The RNN outputs two values: the robot's desired speed and turning rate. Note that Art's translational and turning speed were capped at 0.6m/s and $\frac{1}{4}\pi$rad/s respectively. Given these parameters, $|A| = 2, |Z| = 5$ and $|N_H| = 4$, we have 44 real-valued variables to optimise.

4.2 The Shuttle-Run Problem

Our main aim in this test was to determine if NSGA2-LS was able to successfully optimise RNNs for use in a real-world environment. As a proof-of-concept, we designed a "shuttle-run" problem where the objective was to race to the end of a corridor and back again repeatedly within a specified time period. However, there are two additional considerations: the robot is not allowed to hit any obstacles and power consumption should be minimised.

This problem can be modelled by a MR-POMDP with three reward functions. This first objective is modelled by $R^{(1)}$ whereby Art earned a reward of 1000 each time it reached the (alternating) goal positions. To prevent plateaus on the fitness surface, Art was given an additional $1/(1+d_G)$ reward at the end of each trial, where d_G is the distance to the next goal. The second reward function $R^{(2)}$ modelled driving safety and Art received a penalty of 1000 each iteration it was in contact with an obstacle. The third reward function $R^{(3)}$ modelled power consumption. For this proof-of-concept, we used a simplified power usage model where Art received a negative reward of $\alpha v + \beta\omega$ where v and ω are the robot's translational and angular velocity respectively. In our experiments, we set $\alpha = \beta = 5$. Also, to prevent solutions where the robot simply remains still, a large penalty is given to solutions which do not complete any laps.

To simulate the runs needed to evolve the RNN, we used the Player-Stage framework [6]. The simulated obstacle course is shown in Fig. 6 and each evaluation lasted 20,000 iterations ($\approx$ 15 minutes in real-time).

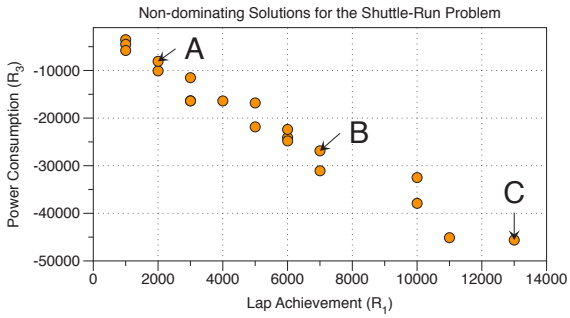

Figure 7: Best solutions for the Shuttle-run problem found after 2.5×10^6 evaluations. As expected, there is a trade-off between the number of laps achieved and power consumption. At the extrema, the robot is able to make 13 laps (with high power consumption) or only a single lap by conserving power.

4.3 Results

The best non-dominating solutions after 2.5×10^5 evaluations are shown in Fig. 7. The (simulated) P3-AT was able to complete a maximum of thirteen laps. Interestingly, the policy with the smallest power use did so not by moving slowly through the course but instead, by coming to a complete stop after finishing a single lap.

We transferred the RNNs onto Art and created a similar (but not identical) real-world obstacle course (Fig. 6). We conducted six three-minute trials using policies A, B and C (Fig. 7). There was little variation between different runs of the same policy and we observed that all three policies were sufficiently general to navigate the real-world course without bumping into obstacles. However, real-world speeds were slower than simulation, possibly due to surface friction.

Policies A, B and C completed an average of 2, 2.6 and 2.83 laps respectively. Policy A took slower, more careful turns and slowed down ahead of obstacles (median speed of 0.23 ms^{-1}). In contrast, C featured more "aggressive" driving with a median speed of 0.42 ms^{-1}. Finally, policy B drove in a balanced manner, with a median speed of 0.34 ms^{-1}. The difference in behaviour can be seen in Fig. 8 showing distributions of translational and turning speeds achieved by policies A, B and C.

4.4 Policy Switching on-Demand

To test policy-switching, we designed a policy-selection mechanism that chooses actions based on user-defined preferences. In this system, the input is fed to a set of non-dominated policies. The policy selector picks the policy with normalised values closest (in terms of Euclidean distance) to the user-defined preference vector $\mathbf{w}$ and sends the associ-

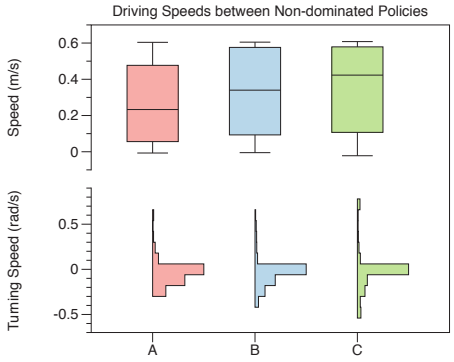

Figure 8: Distributions of driving speeds sampled at 10Hz for policies A, B and C. Policy C achieved the highest median driving speed and featured more "aggressive" driving with quick turns around corners. In contrast, policy A slowed down substantially (even stopping) ahead of obstacles and making turns on the spot. Policy B struck a balance between A and C.

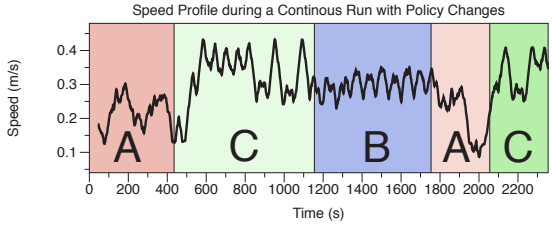

Figure 9: Translational speed profile during the 40-minute continuous run with policy changes. From the (smoothed) speed graph, we observed clear changes in behaviour after changes in policy.

ated action to the actuators. For this experiment, $\mathbf{w}$ was simply the desired trade-off between speed and power consumption (since we disregarded policies which hit obstacles).

We allowed Art to drive continuously in the obstacle course for 40 minutes but policies were changed from A to C to B in the middle of the course at varying intervals. Art completed 33 laps and the (smoothed) velocity profiles in Fig. 9 clearly show a difference in driving behaviour after changes in policy. In a second experiment, we ran Art for 3 minutes but with policies from the entire non-dominated set interleaved at 10 second intervals. In both experiments, policies were successfully changed mid-way through the course without incident (no obstacles were hit), demonstrating that it is possible to adaptively change policies "on-the-fly" based on top-down preferences.

4.5 Transfer to Smart Paediatric Wheelchairs

In the UK alone, there are more than 50,000 disabled children who require mobility assistance [4] and power mobility advocates consider mobility as "an essential component of a child's early intervention program" [2, 11]. In our current research, we are developing a *safe*, paediatric wheelchair we call the Assistive Robot Transport for Youngsters (ARTY) [19] shown in Fig. 10.

In brief, ARTY is a children's powered wheelchair augmented with sensors (both IR and sonar-based) and a mini-PC (the extended capabilities module) as the main computational platform for localisation, obstacle avoidance, path-planning and intention prediction. Thanks to its modular design, ARTY accepts a wide range of input and sensor devices (via CANBus or USB), important for catering to a wide range of disabilities. Using ARTY, we are currently evolving new policies that account for varying disabilities and the three aforementioned objectives of safety, power-consumption and speed, as well as developmental objectives.

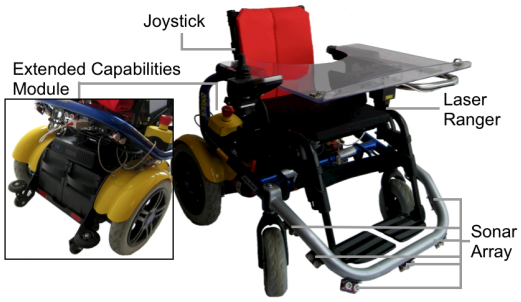

Figure 10: The ARTY Smart Wheelchair.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented the MR-POMDP and the NSGA2-LS algorithm for modelling and solving multiple-objective problems with uncertainty in the medical domain. We demonstrated the applicability of our method to two problems; multi-criteria anthrax response and a shuttle-run problem.

With the the anthrax problem, we illustrated how the Pareto policy set can be used to better understand the trade-offs between policies. We observed that MOEAs are capable of finding both "trivial" policies and complex ones that can be used as the foundation for building real-world solutions. As an extension of our prior work, we optimised RNNs using the Stage simulator and successfully transferred the policies on to a Pioneer P3-AT robot. We observed on a real-world "shuttle-run" experiment that the non-dominated policies expressed qualitatively and quantitatively dissimilar driving behaviours.

An issue with our current MR-POMDP formulation is that we seek to maximise the expected cumulative discounted rewards. This may not be entirely appropriate for certain problems; e.g., is a future life worth less than present one? A potential solution may be to maximise policies using a different criteria such as the cumulative average reward. In addition, we have mainly dealt with small problems; work needs to be done to examine the use of MOEAs and MR-POMDPs for larger, more complex problems and using other policy representations, such as biologically-inspired hierarchical architectures [5] or bayesian-based influence diagrams.

To close, we believe that the MR-POMDP and NSGA2-LS will be useful in a variety of problems in the medical domain (e.g., treatment of ischemic heart disease [8]) to analyse the various factors that come into play during the decision-making process. We envision that MR-POMDPs will give evolutionary algorithmists and medical professionals a common platform for framing their problem, one simple enough

to be computationally tractable but complex enough to handle real-world difficulties.

6. ACKNOWLEDGMENTS

The authors thank members of the BioART lab for their advice and help in the preparation of this manuscript, and Masoumeh Tabaeh Izadi for sharing her POMDP model.

7. REFERENCES

[1] P. A. N. Bosman and E. D. de Jong. Combining gradient techniques for numerical multi-objective evolutionary optimization. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 627–634, New York, NY, USA, 2006. ACM.

[2] C. Butler. *Wheelchair toddlers*, pages 1–6. RESNA, 1997.

[3] T. Carlson and Y. Demiris. Human-wheelchair collaboration through prediction of intention and adaptive assistance. *2008 IEEE International Conference on Robotics and Automation*, pages 3926–3931, 2008.

[4] D. L. Cox. Wheelchair Needs for Children and Young People : a Review. *British Journal of Occupational Therapy*, 66(May):219–223, 2003.

[5] Y. Demiris and B. Khadhouri. Hierarchical attentive multiple models for execution and recognition (hammer). *Robotics and Autonomous Systems*, 54(361-369), 2006.

[6] B. P. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *In Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323, 2003.

[7] W. W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. on Optimization*, 16(1):170–192, 2005.

[8] M. Hauskrecht and H. Fraser. Modeling treatment of ischemic heart disease with partially observable markov decision processes. In *Proc AMIA Symp.*, pages 538–542, 1998.

[9] C. Igel, N. Hansen, and S. Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation*, 15(1):1–29, 2007.

[10] M. Izadi and D. Buckeridge. Decision theoretic analysis of improving epidemic detection. In *AMIA Annu Symp Proc.*, pages 354–358, 2007.

[11] M. A. Jones, I. R. McEwen, and L. Hansen. Use of power mobility for a young child with spinal muscular atrophy. *Physical Therapy*, 83(3):253–262, 2003.

[12] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1995.

[13] J. Knowles and D. Corne. Memetic algorithms for multiobjective optimization: Issues, methods and prospects. In *Recent Advances in Memetic Algorithms*, pages 313–352. 2005.

[14] Y. S. Ong and A. J. Keane. Meta-lamarckian learning in memetic algorithms. *Evolutionary Computation, IEEE Transactions on*, 8(2):99–110, April 2004.

[15] D. Rumelhart, G. Hinton, and R. Williams. *Learning internal representations by error propagation.*, chapter 8. Parallel Distributed Processing. MIT Press, Cambridge, MA, 1986.

[16] M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors. *A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II*, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[17] R. C. Simpson. Smart wheelchairs: A literature review. *Journal Of Rehabilitation Research And Development*, 42(4):423–436, 2005.

[18] H. Soh and Y. Demiris. Evolving policies for multi-reward partially observable markov decision processes (mr-pomdps). In *GECCO '11: Proceedings of the 2011 conference on Genetic and evolutionary computation (to appear)*, 2011.

[19] H. Soh and Y. Demiris. Involving young children in the design of a safe, smart paediatric wheelchair. In *HRI Young Pioneers Workshop*, 2011.