Expensive Multiobjective Optimization for Robotics

Matthew Tesch, Jeff Schneider, and Howie Choset
{mtesch, schneide, choset}@cs.cmu.edu

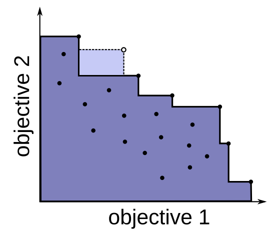
Abstract— **Many practical optimization problems in robotics involve *multiple competing objectives* – from design trade-offs to performance metrics of the physical system such as speed and energy efficiency. Proper treatment of these objective functions, while commonplace in fields such as economics, is often overlooked in robotics. Additionally, optimization of the performance of robotic systems can be restricted due to the *expensive* nature of testing control parameters on a physical system. This paper presents a multi-objective optimization (MOO) algorithm for expensive-to-evaluate functions that generates a *Pareto set* of solutions. This algorithm is compared against another leading MOO algorithm, and then used to optimize the speed and head stability of the sidewinding gait for a snake robot.**

I. Introduction

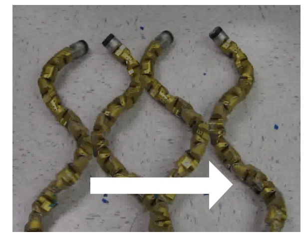
Many problems in robotics inherently require optimization of multiple conflicting criteria, such as the speed of a system and its energy efficiency. Often, it is tempting to simply consider a scalar combination of these criteria, e.g., to optimize a linear combination of speed and efficiency. Unfortunately, such combinations limit the solution to a single point based on preferences implied by the aggregate function. In the *multi-objective optimization* (MOO) community, these multiple objectives are treated explicitly as independent unless the user has a clear preference between them. Instead of a single optimum, this gives rise to a set of *Pareto optimal* solutions, termed the *Pareto set* (see Figure 1(a)).

This full set of solutions is useful in real-world situations. For example, our group as well as several others have shown impressive locomotive capabilities with snake robots (c.f. [1]). Often, these systems are equipped with an on-board camera to allow an operator to teleoperate the robot out of direct line-of-sight. This camera is fixed to the undulating body (often the head) of the robot, which causes difficulty in maintaining situational awareness during locomotion (Figure 1(b)). To mitigate this effect, one can control the robot through small-amplitude low-frequency motions to stabilize the camera. However, such motions also reduce the speed of the system, creating a scenario with two conflicting notions of good performance. In this case, speed may be important when the robot is teleoperated through a wide, easy to comprehend space, but as the robot enters a more complex passage, the stability of the head camera may become paramount. This clearly demonstrates how the relative importance of objectives can change during operation, and hence the need for the full Pareto set of solutions.

Another important consideration when optimizing the performance of a robotic system is that the system can be *expensive* to run – where expense can be measured in terms such as time, financial cost, or use of computational resources. This means that when attempting to optimize the performance of such a system, its use must be budgeted. In this work we consider an *experiment* to be a single objective function evaluation of a set of parameters on the robot itself. In the expensive MOO case, the goal is to reduce the number of experiments needed to find the *Pareto set* of solutions.



(a)



(b)

Fig. 1: **(a)** The set of Pareto optimal solutions are those that are not *Pareto dominated* by any other points in the full set of solutions. The Pareto optimal points in this figure are those which are on the stair-stepped line. The hypervolume of such a set is the volume which is dominated by the Pareto set, shown as the dark shaded region; this is a popular measure of the quality of a multi-objective solution set. The light shaded region is the potential improvement to the hypervolume if the point indicated by the circle were added to the set. **(b)** One problem when teleoperating our snake robots is that the large movement of the head, which houses the camera, can cause operators to become disorientated. We demonstrate the multi-objective optimization methods in this paper to help solve this problem, simultaneously optimizing for the conflicting objectives of head stability and speed.

The expensive nature of objective evaluations requires specialized optimization methods. Methods such as gradient descent or evolutionary algorithms are not designed to limit the number of objective evaluations, but instead to find the optimum quickly assuming nearly free objective evaluations. Whereas most optimization algorithms must choose a next experiment quickly – often milliseconds – the optimization of expensive functions permits algorithms to take more time (usually several seconds) to select an experiment, carefully taking all previous data into consideration.

In this paper we seek to bring the benefits of viewing optimization problems in a MOO framework to the robotic locomotion community. We describe two algorithms for obtaining a Pareto optimal set of solutions while budgeting the number of tests on the robot. Finally, we provide a concrete real world MOO example on the snake robot.

This paper also contains a brief background on successful single-objective optimization methods for expensive system and describes extensions of one of these methods to the multi-objective case. We compare two MOO algorithms on a set of test functions, and finally, we seek to improve the head stability of our snake robot during execution of a sidewinding gait while simultaneously maximizing speed. The novel contributions in this work include description, demonstration, and validation of an MOO algorithm based on the expected improvement in hypervolume [2], the first known application of MOO techniques to the robotics domain, and optimization of the combined locomotive speed and head stability of our snake robot system.

II. Related Work

A. Expensive Black-Box Optimization

Expensive functions, or those for which evaluations take significant resources (time, money, computation, etc.), often also fall in the category of *black-box* functions (those which provide no gradient or derivative information when sampled). Furthermore, these functions need not have guarantees of convexity or linearity; one must search for a global optimum over a function which is in all likelihood nonlinear and non-convex. Optimization of such functions by many standard techniques is ineffective; for example, gradient-ascent approaches would require a number of samples around a sampled point to find a (potentially unstable) approximation to the gradient. For expensive function evaluations this is impractical (especially in higher dimensions) for a single point/gradient sample.

This has motivated the development of a class of "gradient free" optimization techniques; these include local approaches, such as a Nelder-Mead simplex search (c.f. [3]), and global approaches such as genetic algorithms [4] or simulated annealing [5]. Naturally, a globally optimal solution is preferred to a locally optimal one, but unfortunately most methods which search for such global optima require a large number of function evaluations. Again, this is prohibitive if these evaluations are expensive.

To address the particular challenges of expensive black-box optimization, a group of techniques is based on the idea of predicting the entire unknown expensive function from limited sampled data. These *surrogate function*-based sequential experiment selection methods rely on a function regression method, usually a Gaussian process [6], to estimate the true objective function after each subsequent expensive function evaluation. The information provided by the surrogate is then used to intelligently select parameters for future experiments. These regression methods often provide a measure of the uncertainty of their estimate which can be used in conjunction with the estimated objective function value to make more informed decisions. An excellent survey on this subject is given by Jones [7], and more recent work such as [8] continues to demonstrate new applications.

In our previous work [9], we have used these expensive global optimization methods (in particular *efficient global optimization*, or EGO [10]) to optimize the performance of the snake robots discussed in this paper. To this point only single-objective optimization problems have been considered. Locomotion with snake robots serves as an exemplary application of surrogate function optimization techniques because each "experiment" with the snake robot can take several minutes, and provides no gradient information (fitting the "black box" description).

B. Multi-Objective Optimization

The notion of optimality that is embraced in the field of multi-objective optimization is that of a set of *Pareto optimal* solutions. This set, named after economist Vilfredo Pareto, includes all solutions which cannot be improved in one objective without a corresponding decrease in another. In particular, a point in objective space a is said to *dominate* b, written $a \succ b$ or $b \prec a$, if a is at least as good in every objective, and better in at least one. The *Pareto optimal* subset of a collection of points P is $\{p \in P \mid \forall\, q \in P,\ (p \sim q) \vee (p \succ q)\}$. For detailed coverage of these ideas, see [11].

This notion of Pareto optimality allows us to define the best set of parameters given no particular relative importance of the objectives. Once knowledge of this optimal set is obtained, it is straightforward to select the best parameters for any given objective trade-offs or constraints. Finding such optimal sets has been important for a number of real world applications, including modeling grasshopper foraging behavior [12], rehabilitation of water distribution networks [13], design of airfoils [14], and optimization of spacecraft trajectories [15].

However, finding these sets of optimal points requires specialized optimization methods. For cases where the objective functions are linear, the NISE method [16] has been developed to converge quickly on a good approximation of the Pareto set, even in problems of very high dimensions. Multiobjective simplex methods such as [17], which extend the single-objective linear constrained optimization simplex method[1] developed in 1947 [18], provide exact solutions for the Pareto optimal set for linear objective functions.

For nonlinear cases there are also a number of methods; perhaps the most popular is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [19]. This empirically has been shown to produce good results which are well distributed over the *Pareto front* – the Pareto optimal set given in objective space coordinates.

C. Expensive MOO

In the case of expensive optimization for multiple objectives, there is significantly less literature on identifying the Pareto set; the evolutionary methods used in standard MOO are most appropriate when samples are cheap and parameter and objective spaces are very high dimensional. Most expensive MOO approaches attempt to extend successful single objective expensive optimization techniques. One set of techniques create a single aggregate objective function at each step, and choose to optimize this function; this approach is taken by ParEGO [20], where the aggregate is a weighted combination of individual objectives as terms in

[1] Note that this simple method differs from the Nelder Mead constrained nonlinear optimization method.

an augmented Tchebycheff function, and the single objective optimization method used is Jones' EGO algorithm.

Other approaches attempt to work in the full objective space rather than simplifying the problem to one objective. For example, Keane [21] attempts to directly measure the multivariate expected improvement of a point – how much the *hypervolume* of the Pareto front increases (Figure 1(a)). The expression Keane presents is a simplification of the true quantity and only measures improvement as an increase from a single point on the Pareto front; Emmerich et al. [2] redefine this improvement more rigorously (yet are still able to find a closed-form expression) using Lebesgue integration on a partition of the objective space.

D. Snake Robots

This work was motivated by the goal of improving the locomotive capabilities of snake robots, especially those of the latest generation in our lab [22]. These robots have demonstrated impressive locomotive capabilities, and with their small diameter of 10 cm they can fit into small channels and spaces too confined for other mechanisms. Much like their biological counterparts they use cyclic control trajectories called *gaits* to move across relatively regular terrain, such as a flat expanse of grass, a roughly uniform diameter pole, or a regular grid of poles.

Although the space of cyclic controls is infinite, Choset's robots are usually controlled by motions within a finite dimensional constrained control trajectory subspace (the *gait model* described in [1]). This model, defined by

$$\alpha(n,t) = \begin{cases}\beta_{even} + A_{even}\sin(\theta), n = \text{even},\\ \beta_{odd} + A_{odd}\sin(\theta+\delta), n = \text{odd},\end{cases} \qquad (1)$$

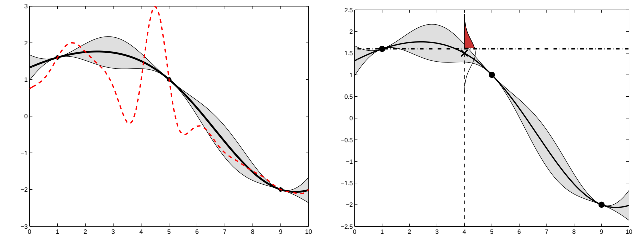
$$\theta = \left(\frac{d\theta}{dn}n + \frac{d\theta}{dt}t\right), \qquad (2)$$

is general enough to command the snake to slither, sidewind, roll in an arc, wrap around a tree or pole in a helix and climb, turn in place, and traverse via many other motions. Similar controllers have been used by other researchers (c.f. [23], [24]). In this paper, we consider the optimization of an augmented sidewinding gait, based this equation with restrictions on the set of free parameters (β, A, etc.).

One difficulty with optimizing an objective like the stability of the head module during a gait is that there is no simple, accurate motion model for these robots due to their frequent collisions with the ground and multiple simultaneous sliding contacts. This requires one to actually run experiments on a physical system to reliably sample such an objective, which motivates the use of expensive optimization techniques described above.

III. Optimization Method Overview

We first describe the single-objective precursors to the MOO algorithms that are the focus of the paper. Although an algorithm based on the optimization of Emmerich et al.'s expression for the expected hypervolume improvement is described below, we also implement the ParEGO aggregate function optimization method for purposes of comparison and verification. We note before continuing that we use the convention of maximizing a reward function rather than minimizing; this differs from much of the optimization literature referenced, but aligns with our goal of improving robot performance.



(a) (b)

Fig. 2: **(a):** A surrogate function (dark line) interpolates sampled points of an unknown underlying function (dotted red line). The surrogate quantifies uncertainty in its prediction, as shown by the shaded region.**(b):** Experiment selection metrics such as probability of improvement and expected improvement consider the predictive distribution at a potential sample point x (vertical line), compared to the best previous sample (horizontal line).

A. Expensive Single Objective Optimization

The expensive single optimization methods that underly this work are *global*, *constrained* methods. Formally, given a parameter space $X \subset \mathbb{R}^n$ and a objective function $f\colon X \to \mathbb{R}$, they search for $\operatorname{argmax}_{x \in X} f(x)$. As it is expensive to obtain samples of $f(x)$, there is a limit to the number of times f can be sampled.

As described above, a common and successful approach to expensive optimization task involves the use of a nonlinear surrogate function fit to the sampled data points, as shown in Figure 2(a). Such algorithms are summarized in the following steps:

1) Sample a set of initial points (often randomly or with a *Latin hypercube* experimental design)
2) Fit a surrogate to the sampled points
3) Select the next sample location by optimizing a metric
4) Repeat steps 2-3 until convergence

The central challenge is the determination of the *fitness metric* which is optimized in step 3 to select subsequent sample locations. This metric must balance exploration (sampling in unknown regions) and exploitation (sampling in known good regions) during the optimization. Neither extreme (choosing the current maximum of the surrogate or sampling purely based on uncertainty, such as [25]) result in methods which converge to the optimum efficiently. To improve the quality of search, the uncertainty of the estimated function value should be used in conjunction with that estimated value.

To this end, simple approaches may select subsequent samples based on a weighted sum of the estimated function and its error (c.f. IEMAX [26]). Another approach that incorporates a natural trade-off is to maximize the *probability of improvement* [27], [28]. For a given test point, the probability of improvement is the integral of the tail of the predictive distribution above the maximum value of the objective found so far (Figure 2(b)). To be successful, these methods require tuning parameters that explicitly balance exploration and exploitation.

We choose to use a more principled method to address this trade-off: the idea of *expected improvement* (EI) [29] popularized by Jones et al.'s *efficient global optimization* (EGO) algorithm [10]. Given a set of sampled values $\tilde{Y}$, the improvement $I(y^*)$ of a new sample at x with the value y^* is the increase in the maximum value of this set with the addition of y^*. The expectation of this quantity over the surrogate function's predictive distribution at x, $p_x(y)$, is given as

$$\begin{aligned} EI(x) &= \mathbb{E}[I(y)] \quad (3) \\ &= \int_{\max(\tilde{Y})}^{\infty} (y - \max(\tilde{Y}))p_x(y)\,dy, \quad (4) \end{aligned}$$

and captures the intuitive idea of selecting the next sample as the point which you expect to most improve your current solution (Figure 2(b)). This statistical measure automatically balances the trade-off between exploration and exploitation without requiring a tuning parameter.

B. Multiple Objective Extensions

As the expected improvement metric has been a success in expensive single-objective optimization, the natural question is whether it can be applied to the MOO case. In order to extend the EI metric, one must first define the notion of 'improvement'. In the single objective case, improvement has a natural definition, because there is already a single objective f which is being optimized. The improvement of a sampled value of y^* over the set of sampled points $\tilde{Y}$ is simply the increase in the maximum value of the resulting set, or

$$I(y^*) = \max(y^* - \max(\tilde{Y}), 0) \quad (5)$$

In multi-objective optimization there is more than one objective, and the solution is not just a single point, but an entire *Pareto set* of points. To measure improvement in such a set, a valid metric must be devised to measure the quality of such a set. One such metric is the set's *hypervolume* [30]. This is the volume in objective space which is Pareto-dominated by at least one point in the Pareto set. A reference point in objective space must be selected to define the lower bounds of this volume; this point should be chosen given prior knowledge of or an educated guess about the minimum possible value of the objective functions (e.g., the net displacement of a gait must always be greater than or equal to 0).

The concept of the hypervolume indicator is illustrated for a two-dimensional objective space in Figure 1(a). This measure has desirable properties; for example it is not affected when a dominated point is added to a set of solutions, and the addition of a non-dominated point always increases a set's hypervolume. Given a method to compute the hypervolume HV of a solution set, improvement can be defined as

$$I(y^*) = \mathrm{HV}(\tilde{Y} \cup y^*) - \mathrm{HV}(\tilde{Y}). \quad (6)$$

Although a simple closed-form solution has been derived for EI in the single objective case, $\mathbb{E}[I(y^*)]$ is not straightforward when the improvement is measured in terms of hypervolume; evaluating this for a given test point requires either a multidimensional numerical integral, or the development of an analytic form for this expectation. Fortunately, Emmerich et al. have provided the outline of a method to compute this quantity in closed form [2]. Because we are working in objective space, this metric considers the joint improvement in all objectives simultaneously, trading off the benefits of sampling a point which might improve one or the other.

Using hypervolume as the indicator of solution set quality and finding an efficient computation of its expectation allows us to use machinery from the single objective case for optimization with multiple objectives. Our multi-objective optimization algorithm using the *expected improvement in hypervolume* (EIHV) metric can be summarized in a form parallel to that of surrogate-based single objective algorithms:

1) Sample the objectives at a set of initial points
2) Fit a surrogate to the sampled points for each objective function
3) Select the next sample location by selecting the sample with the largest EIHV value
4) Repeat steps 2-3 until convergence

C. Limitations and Implementation Details

When implementing surrogate-function based algorithms, a number of issues can arise. Most importantly, these methods rely on the function regression method to provide a reasonable estimate of the objective and the uncertainty of that estimate. We use Gaussian processes (GPs) as this regression method, which can lead to a number of pitfalls. From our experience, we have a number of recommendations. First, the use of an existing package, such as the GPML MATLAB library [31], can greatly reduce initial time and effort of implementation. Next, when fitting a surface, it is important to carefully tune the *hyperparameters* that describe the GP to obtain a realistic and non-trivial fit. As recommended by [6], we find hyperparameters that maximize the log likelihood of the data. This is done via a large number of line search optimizations in the hyperparameter space (using the GPML package's `minimize` function) from hundreds of random seed points, including the best hyperparameter value found in a previous fit.

To further improve the fit and reduce necessary manual involvement with the fitting process, we choose to run this hyperparameter selection process over a number of different sets of covariance functions for the GP (effectively model selection over a number of different function forms for regression). By using the log likelihood of the data as a selection metric, this allows the complexity of the model to match the complexity of the data. Initially, simple covariance functions are chosen; as the number of data points collected increases, the complexity typically grows to match the trends shown by the data.

In addition to obtaining a quality surrogate function fit, it is important to ensure the global maximum is found when optimizing the experiment selection metric. This function is often highly irregular and strongly peaked. Taking the logarithm can ensure a more numerically stable optimization. Also, we advise many random restarts of your favorite built-in optimization algorithm to ensure that the space is well covered. As this optimization determines the quality of the point selected, it is important to spend time on this step, both during implementation as well as when running the code to select experiments.

Finally, the effectiveness of these methods is restricted to fairly low-dimensional spaces. The authors usually work with parameter spaces from 2-8 dimensions, but have had success up to 20 depending on the objective function complexity. The limitations are driven by the ease and robustness of fitting a GP in higher dimensions and the reliability of the optimization of the selection metric in those spaces.

IV. Test Results

The primary motivation for the careful experiment selection methods described herein is the expensive nature of testing the performance of physical robotic systems. Therefore to justify the selection of one algorithm to use for optimization of a physical system, we ran a more extensive comparison on two simple analytic functions. This also allowed us to test algorithm implementations, and ensure they functioned as expected.

Many of the multi-objective test functions in the literature are particularly designed to confound existing multi-objective evolutionary algorithms (MOEAs), and therefore involve large high-dimensional parameter spaces with many separated Pareto set regions. The low dimensional analogues, when they exist, are trivial surfaces that do not provide for a reasonable evaluation of the optimization algorithms we were considering.

Instead, we chose to use a region of the Branin test function, a common benchmark for global single objective optimization from the Dixon-Szegö test problem set [32]. We chose different regions of the Branin function for each objective, and found that the resulting surfaces exhibit qualitative similar properties to those observed for the performance of our robots. These regions, shown in Figure 3, are defined by the following equations, where B is the original Branin function and all inputs are between 0 and 10 inclusive:

$$f_1^1(x) = B(x, 1)/10, \tag{7}$$
$$f_2^1(x) = B(3, x)/10, \tag{8}$$
$$f_1^2(x_1, x_2) = B(x_1, 2 + 0.5x_2)/20 \tag{9}$$
$$f_2^2(x_1, x_2) = B(0.4x_1, 5 + 0.1x_2)/10 \tag{10}$$

In addition to the optimization method described in §III-B, we have chosen a simple popular state of the art optimization method for expensive multi-objective problems called ParEGO [20]. This algorithm takes the approach of, at each iteration, generating a single aggregate objective function. It then reverts to a single-objective experiment selection method (the expected improvement based algorithm EGO, described in §III-A) to choose the next sample. Finally, we also use random experiment selection to provide a baseline from which to measure the importance of any careful experiment selection.

For the simple one-dimensional test function, we ran each algorithm 20 times, each time independently selecting 40 locations at which to sequentially sample the objective functions. The repeated trials are necessary because initial sample location selection is random. For the two-dimensional test function, each algorithm was run 20 times, with 20 sampling locations selected each time.

The resulting algorithm performance is shown in Figure 4. In each case, both ParEGO and the expected improvement in hypervolume were shown to significantly outperform random experiment selection, demonstrating the potential savings when optimizing on expensive systems. The use of EIHV also outperformed ParEGO for either tested setting of ParEGO's s-value (10 and 1000), showing an algorithm with no tuning parameters and good performance. Due to these empirical results, we chose to use EIHV when optimizing multiple objectives on the physical snake robot.

V. Robot Results

After the validation of EIHV in the previous experiments, we used this method to generate a set of Pareto optimal solutions for competing objectives of head stability and speed. In order to do so, we first needed to define our cost functions and the parameter space over which we are optimizing.

The speed objective definition is straightforward – after running the snake for 10 seconds with the gait parameters as specified by the optimizer, we measured the net displacement of the center of mass. The head stability objective was more complicated. To obviate the need for a motion capture lab, we combined several tools that only relied on sensor data from on-board the robot. Intuitively, we wished to capture how much the camera image changed for an operator. Moreover, since the resulting image from a translating camera is less disorienting than a rotating camera, we chose to focus on the latter motion.

We choose a point p at a distance l of 18 inches normal to the center of the camera lens plane as the desired focal point. Using a Kalman filter based sensor-fusion state estimation technique [33], we estimated the motion of this point throughout the gait. However, because we are primarily concerned with orientation, we use a shape-stable body frame termed the *virtual chassis* [34], consider only the orientation component of the virtual chassis' state estimate, and define the point p to be in the world frame assuming the virtual chassis is fixed in position but free rotate in all three dimensions.

More formally, let the $4x4$ homogeneous transform of the estimated orientation of the virtual chassis at the timestep t

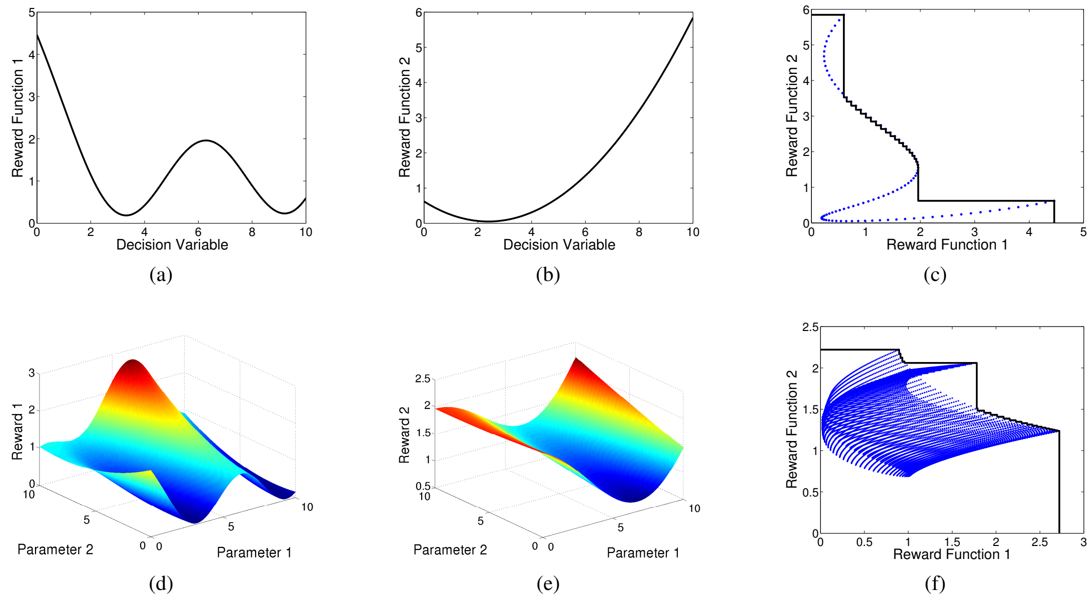


Fig. 3: Competing test functions created from one dimensional slices (**(a)** and **(b)**) and scaled two dimensional regions (**(d)** and **(e)**) of the Branin test functions. The plots in **(c)** and **(f)** show the resulting objective space projection (blue dots) and Pareto frontiers (black lines).

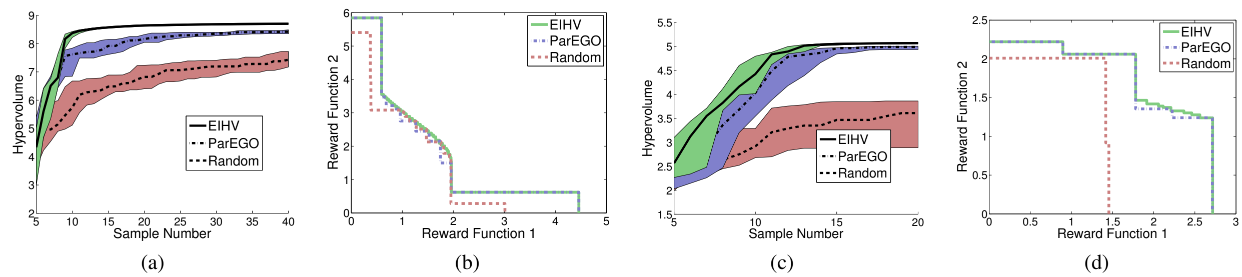


Fig. 4: Results of two different expensive MOO methods – optimization of the expected improvement in hypervolume (EIHV) and ParEGO – and a random experiment selection baseline on the 1-D **(a)** and 2-D **(c)** test functions shown in Figure 3. The line indicates the median hypervolume of the 20 trials, while the shaded regions represent the middle 50% of the trials. Also shown are the resulting Pareto frontiers from each method on each test function (**(b)** and **(d)**). On both functions EIHV outperforms ParEGO, and both methods are significantly better than random, demonstrating the importance of careful experiment selection in expensive optimization.

be defined by $R(t)^{VC}$. Then the position of the head frame in the virtual chassis frame at t is given by the transform matrix $T(t)^0_{VC}$ (where the 0 indicates the 0th snake module). The ray from the head camera module to p is of length l in the z direction, so let $p_0 = [0, 0, l, 1]^T$ be the location of p in the head module frame. Given these variables, we can define p in the fixed-position, free-rotation virtual chassis "world" frame at time t as

$$p(t) = R(t)^{VC} T(t)^0_{VC} p_0. \tag{11}$$

To transform this focal point location into the stability cost, the minimum bounding box for this swept area is calculated. First, the vertical sweep is

$$V = \max_{0<t<T}(p_z(t)) - \min_{0<t<T}(p_z(t)), \tag{12}$$

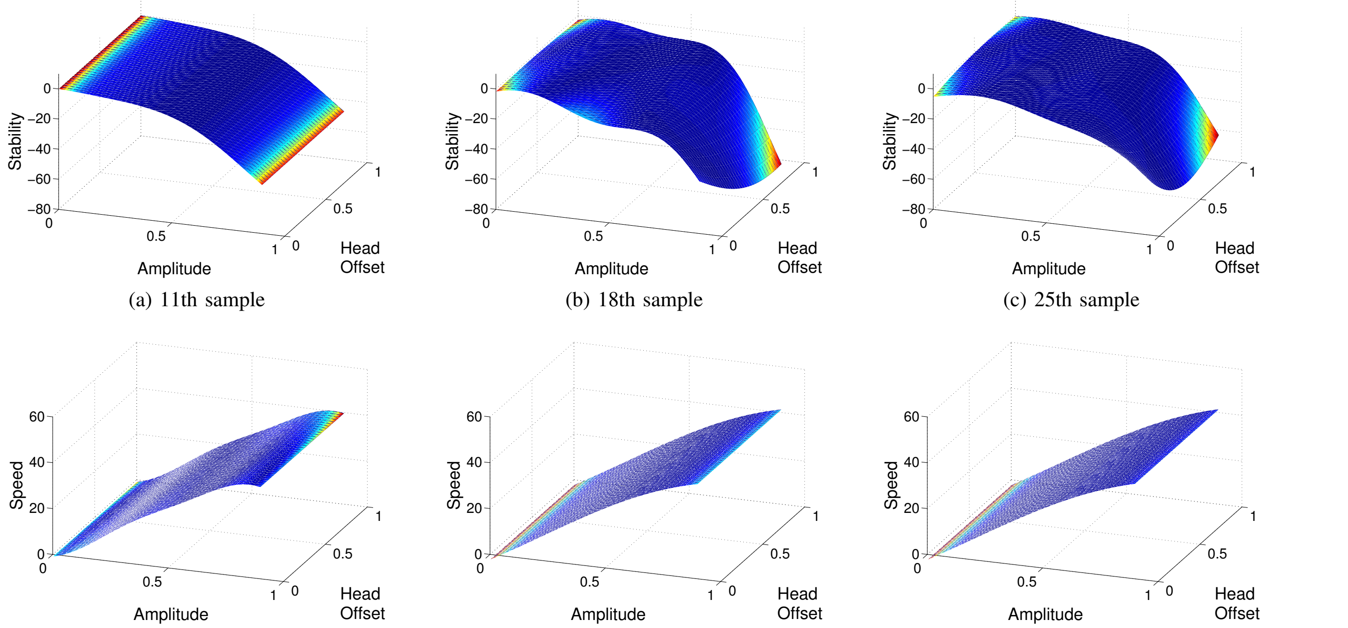
where $p_z(t)$ is the z component of $p(t)$, and T is the total number of timestamps. The horizontal sweep first requires calculation of the total angular sweep from the origin to all $p(t)$ projected onto the $x-y$ plane. This should be the smallest angle for which the interior cone (from the origin) can contain all $p(t)$. This angle is H_θ, and is given in radians. The horizontal sweep is calculated as

$$d(t) = \sqrt{p_x(t)^2 + p_y(t)^2}, \tag{13}$$

$$H = H_\theta \sum_{0<t<T} \frac{d(t)}{T}, \tag{14}$$

where $p_x(t)$ and $p_y(t)$ are similarly the x and y components of $p(t)$, and $d(t)$ is the distance from the origin of $p(t)$ projected into the $x-y$ plane. Finally, the total cost – representing a bounding box of the swept position of the focal point – is given by $V \times H$. In practice we negate this value because our optimization methods maximize rather than minimize.

As we are primarily concerned with speed and head stability, we chose to optimize parameters of an augmented gait model that would add more position control of the head module while still keeping a low-dimensional parameter



(a) 11th sample (b) 18th sample (c) 25th sample

(d) 11th sample (e) 18th sample (f) 25th sample

Fig. 5: After the initial random experiments, the Gaussian process function regression method generates an estimate of the function after each subsequent sample. These surfaces are shown here for three selected iterations, where the top row is regression of the stability objective and the bottom is the regression of the speed objective. The color of the surface is the uncertainty in the estimate (blue regions in the center have low uncertainty, red regions near the edge have high uncertainty).

(a) Point A, 1.25Hz frame rate (b) Point B, 0.4Hz frame rate (c) Point C, 0.4Hz frame rate (d)

Fig. 6: **(a)-(c):** Time lapse images showing the head and snake motion of three different points in the head stability/speed Pareto set. The snake in (a) moves quickly, but the head rotates back and forth. The snake in (b) moves at a slower rate, but with noticeably less head sway. Finally, (c) is a very slow but very head-stable motion. As can be seen in **(d)**, these points span the Pareto front, and none dominate the others. Shown as blue dots are the results of all experimental evaluations during the optimization.

space. We take the basic form of the gait model in Equation (1), restrict it to the sidewinding parameter space as defined in [1], and add an additional offset ϕ for the head module:

$$\alpha(1,t) = \beta_{odd} + A_{odd}\sin(\theta + \delta + \phi) \tag{15}$$

We then optimize over the amplitude and ϕ, with a fixed ratio between A_{odd} and A_{even}.

Finally, when running tests on physical systems, the non-deterministic nature can cause instability in many optimization methods. Although using a GP that can explicitly model this noise allows many surrogate function based optimizers to perform well in the presence of noise, we make two adjustments. First, we run each trial 5 times, averaging the results for each objective and removing outliers. Second, when determining the Pareto set for the purpose of the calculation of EIHV, we use the surrogate's estimate of the objective values for each sampled point. This improves the stability by ensuring that an artificially high sampled value will not too strongly discourage nearby samples.

The process and results of these optimization trials can be seen in Figures 5 and 6, respectively. The optimizer has sampled points which span the trade-off between both objectives, and in particular has found relatively fast motions that have improved head stability. The three montages of robot motion shown in Figures 6(a), 6(b) and 6(c) show how this multi-objective optimization approach generates a range of solutions, whereas the typical approach of aggregating these objectives would only have found one of these solutions.

VI. Conclusions and Future Work

Multi-objective optimization is often a reasonable alternative to creating a single aggregate objective in the case of competing system performance objectives. This is a case which comes up frequently in robotics as well as many other fields such as design, decision theory, and economics. Instead, a Pareto optimal set should be found, which contains all solutions which are not *dominated*, or completely outperformed, by another solution. The generation of Pareto optimal solutions sets is especially difficult when sampling the performance of a system is expensive, but once accomplished these solutions can be selected from to provide real-time trade-offs between objectives.

In this paper, we have created and tested a MOO approach based on maximization of the expected improvement in hypervolume of the Pareto set. We have compared this to a leading MOO algorithm, ParEGO, on multiple test functions. Finally, we have applied the former algorithm to a practical application, the task of finding snake robot gait parameters for fast and head-stable sidewinding. This application required care to reduce the effect of noisy evaluations on the optimization performance as well as the creation of a head-stability cost function from recent state-estimation techniques for the robot.

Future work involves testing these methods with higher dimensional parameter spaces and use of these methods on other robotic systems. In addition, the explicit handling of noisy objective evaluations and guarantees of convergence to a dense covering of the full Pareto optimal solution set are open problems. Both of these should be addressed in order to give potential adopters more confidence in the results of these methods.

References

[1] M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz, and H. Choset, "Parameterized and Scripted Gaits for Modular Snake Robots," *Advanced Robotics*, vol. 23, no. 9, pp. 1131–1158, Jun. 2009.

[2] M. Emmerich and J.-w. Klinkenberg, "The computation of the expected improvement in dominated hypervolume of Pareto front approximations," Leiden Institute for Advanced Computer Science, Tech. Rep. 1, 2008.

[3] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, no. 4, Jan. 1965.

[4] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic Programming: An Introduction*. Morgan Kaufmann, 1997.

[5] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[6] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[7] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, 2001.

[8] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," *ArXiv e-prints*, 2012.

[9] M. Tesch, J. Schneider, and H. Choset, "Using Response Surfaces and Expected Improvement to Optimize Snake Robot Gait Parameters," in *International Conference on Intelligent Robots and Systems*. San Francisco: IEEE/RSJ, 2011.

[10] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, no. 4, 1998.

[11] J. L. Cohon, *Multiobjective Programming and Planning*. Courier Dover Publications, 2004.

[12] K. D. Rothley, O. J. Schmitx, and J. L. Cohon, "Foraging to balance conflicting demands : novel insights from grasshoppers under predation risk," *Behavioral Ecology*, vol. 8, no. 5, pp. 551–559, 1997.

[13] P. B. Cheung, L. F. Reis, K. T. Formiga, F. H. Chaudhry, and W. G. Ticona, "Multiobjective evolutionary algorithms applied to the rehabilitation of a water distribution system: A comparative study," in *Int. Conf. Evol. Multicriterion Optimization*, C. M. Fonseca and E. Al., Eds. Berlin, Germany: Springer-Verlag, 2003, pp. 662–676.

[14] B. Naujoks, L. Willmes, T. Back, and W. Haase, "Evaluating Multicriteria Evolutionary Algorithms for Airfoil Optimisation," *LECTURE NOTES IN COMPUTER SCIENCE*, no. 2439, pp. 841–850, 2003.

[15] V. Coverstone-Carroll, "Optimal multi-objective low-thrust spacecraft trajectories," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 387–402, Jun. 2000.

[16] R. S. Solanki, P. A. Appino, and J. L. Cohon, "Theory and Methodology Approximating the noninferior set in multiobjective linear programming problems," *European Journal Of Operational Research*, vol. 68, pp. 356–373, 1993.

[17] M. Zeleny, *Linear Multiobjective Programming*. Springer-Verlag, 1974.

[18] G. B. Dantzig, "Programming in a Linear Structure," Comptroller, USAF, Washington, D.C., Tech. Rep., 1948.

[19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182—-197, 2000.

[20] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, Feb. 2006.

[21] A. J. KEANE, "Statistical improvement criteria for use in multiobjective design optimization," *AIAA journal*, vol. 44, no. 4, pp. 879–891, 2006.

[22] C. Wright, A. Buchan, B. Brown, J. Geist, M. Schwerin, D. Rollinson, M. Tesch, and H. Choset, "Design and Architecture of the Unified Modular Snake Robot," in *2012 IEEE International Conference on Robotics and Automation*, St. Paul, MN, 2012.

[23] J. Gonzalez-Gomez, H. Zhang, E. Boemo, and J. Zhang, "Locomotion capabilities of a modular robot with eight pitch-yaw-connecting modules," in *9th international conference on climbing and walking robots*. Citeseer, 2006.

[24] A. Kuwada, S. Wakimoto, K. Suzumori, and Y. Adomi, "Automatic pipe negotiation control for snake-like robot," *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, no. 438, pp. 558–563, Jul. 2008.

[25] J. M. Bernardo, "Expected Information as Expected Utility," *The Annals of Statistics*, vol. 7, no. 3, pp. 686–690, May 1979.

[26] A. W. Moore and J. Schneider, "Memory-based stochastic optimization," *Advances in Neural Information Processing Systems*, pp. 1066–1072, 1996.

[27] H. J. Kushner, "A new method for locating the maximum point of an arbitrary multipeak curve in the presence of noise." *Journal of Basic Engineering*, vol. 86, pp. 97–106, 1964.

[28] A. Žilinskas, "A review of statistical models for global optimization," *Journal of Global Optimization*, vol. 2, no. 2, pp. 145–153, Jun. 1992.

[29] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of Bayesian methods for seeking the extremum," *Towards Global Optimization*, vol. 2, pp. 117–129, 1978.

[30] E. Zitzler and L. Thiele, "Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study," pp. 292–304, Sep. 1998.

[31] C. E. Rasmussen and C. K. I. Williams, "Gaussian Processes for Machine Learning." [Online]. Available: http://www.gaussianprocess.org/gpml/code/gpml-matlab.tar.gz

[32] L. Dixon and G. Szego, "The global optimization problem: an introduction," *Towards Global Optimization*, vol. 2, pp. 1 – 15, 1978.

[33] D. Rollinson, A. Buchan, and H. Choset, "State Estimation for Snake Robots," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1075–1080, 2011.

[34] D. Rollinson and H. Choset, "Virtual Chassis for Snake Robots," *IEEE International Conference on Intelligent Robots and Systems (accepted)*, 2011.