AA222 Final Project: Optimization of Bicycle Suspension Tuning

Jeffrey B. Robinson
Master's Degree Candidate, Aeronautics & Astronautics, Stanford University

Bicycle suspension tuning is briefly treated with regard to the effectiveness of Cyclic Coordinate Descent (CCD), representing commonly used uninformed methods, as compared to six other algorithms representing the Direct, Stochastic, and Population-based categories. A two degree of freedom dynamical model incorporating internally resolved ground contact and displacement limits, as well as motion-rate-dependent damping coefficients, is used to approximate the behavior of a bicycle wheel traversing stochastic terrain. Objectives assessing traction and ride comfort are heuristically calculated from model simulation, and a combined obejctive is calculated from a simple weighted sum. Performance of the seven algorithms is compared by objective value obtained after a fixed number of objective evaluations for each of three weighting cases. Generalized Pattern Search (GPS) using basis vectors as the search direction set is found to be the best performing method, and a practical alternative to CCD.

I. Introduction

Bicycle suspension is critical to the ability of the rider to maintain control and comfort when traversing rough ground, particularly in off-road, or "mountain" biking. The suspension system of a bicycle must be able to smooth the effects of large impulses due to trail obstacles or rider input as well as smaller, higher-frequency disturbances due to roughness of the trail surface. In the case of either input, the goals of the suspension are to reduce the magnitude of impulses transferred to the rider - improving comfort - and to maximize contact between the wheels of the bicycle and the ground - improving traction.

Suspension components are often among the most expensive components required to assemble a bicycle, and passive observation of media regarding the mountain bike industry has yielded a general impression that advancement in suspension development is a primary point of conversation and marketing [1, 2]. However, relatively little exists in the way of publicly available information regarding the methodology of tuning bicycle suspension. In practice, the process of tuning bicycle suspension thus often appears to the cyclist as more of an art than a science, and tends to require prolonged effort to achieve noticeable improvement. What little information is available tends to suggest a simplistic approach resembling Cyclic Coordinate Descent (CCD).

The purpose of this project is to explore the applicability of gradient-free optimization methods to the problem of tuning bicycle suspension, to determine whether any are better suited to the task than the typical CCD approach. The primary goal of this work is to compare the effectiveness of CCD, in terms of objective value achieved after a given number of function evaluations, to representatives of the direct, stochastic, and population-based categories of optimization algorithm.

II. Related Work

Little concrete information is available about the engineering efforts made by companies developing mountain bike suspension components. However, it is clear from marketing efforts by suspension component manufacturers that suspension tuning is of significant interest to the industry, with companies offering paid tuning services, smartphone apps, and even aftermarket components to simplify the tuning process [2–6].

In the broader scope of suspension tuning optimization, significant effort has been made for cars, public transportation such as buses, road-going commercial cargo vehicles, and railway vehicles, with [7–10] being some examples of the breadth of the field. In these studies, the suspension systems of multi-wheeled vehicles are analyzed using kinematic models, laboratory testing, and field testing. The methods used to optimize the damping parameters of the suspension systems are in some cases left unclear beyond the definition of the optimization problem, as in [7]. In [8], a multiobjective optimization problem is solved using the `gamultiobj` genetic algorithm in Matlab. In many papers on the subject, such as [10], the object of optimization is an active rather than a passive suspension system, in which cases the optimization algorithm tunes not only the damping parameters of the system but also the control forces applied by the suspension system. There does not seem to be an industry consensus on an optimal method for tuning suspension.

III. Modeling Approach

A. Overview

The suspension system is modeled as a 2 degree of freedom dynamical system in which the bicycle frame and wheel are masses connected by a spring and passive damper. The wheel is connected to the ground by an undamped spring. This model is similar to those used and discussed by [9, 10]. Figure 1 demonstrates the layout of this dynamical model, and Equation 1 summarizes the equations of motion used to simulate it.

The ground surface is generated stochastically - the gradient of the surface elevation at each time step is sampled randomly from a normal distribution with mean $-0.2\ m/m$ and variance $20\ m/m$ to generate a surface of realistic proportions. Several values for these parameters were tested, one example being

(0, 50), which generated exaggerated surface features that forced the bicycle model to spend more time "jumping" than closely following the ground. This ground surface gradient is moderated by a Simple Moving Average (SMA) of previous gradient values over a characteristic scale equivalent to the diameter of the bicycle wheel. The number of time steps included in this moving average is calculated from the characteristic scale, the solver time step, and a specified constant horizontal speed. The scale parameters chosen for this project were a 27.5 inch wheel with a 2.5 inch tire, representative of a typical contemporary mountain bike. These sizes translate to a wheel diameter of 0.711 m and a tire thickness of 0.0635 m. The frame mass was set as 50 kg, half of the sprung mass of a heavy cyclist and frame. The wheel and tire (unsprung) mass was set as 5 kg.

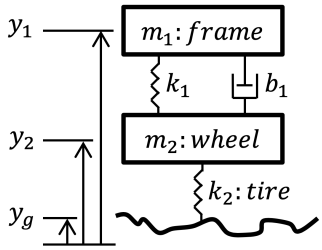

Fig. 1 Suspension Dynamical Model

$$\ddot{y}_1 = -g - \frac{b_1(\dot{y}_1, \dot{y}_2)}{m_1}(\dot{y}_1 - \dot{y}_2) - \frac{k_{1_{mod}}(y_1, y_2)}{m_1}\left[(y_1 - y_2) - (y_{1_0} - y_{2_0})\right] \tag{1a}$$

$$\ddot{y}_2 = -g + \frac{b_1(\dot{y}_1, \dot{y}_2)}{m_2}(\dot{y}_1 - \dot{y}_2) + \frac{k_{1_{mod}}(y_1, y_2)}{m_2}\left[(y_1 - y_2) - (y_{1_0} - y_{2_0})\right] - \frac{k_{2_{mod}}(y_2, y_g)}{m_2}\left[(y_2 - y_g) - (y_{2_0} - y_{g_0})\right] \tag{1b}$$

$$\dot{y}_g^{(i)} = \dot{y}_g' + \frac{1}{n_{\text{scale}}}\sum_{j=i-n_{\text{scale}}}^{i-1}\dot{y}_g^{(j)} \qquad \dot{y}_g' \sim \mathcal{N}(-0.2, 20) \tag{1c}$$

B. Abstractions

One key concern in assessing the realism of this model is the likely interactions of the front and rear wheels of the bicycle through rotation of the frame. In order to circumvent the extra degrees of freedom associated with bicycle frame rotation, as well as the corresponding rotation of the suspension axis of freedom, the rider is assumed to maintain control of the bicycle and stabilize its movements, with even weight distribution between the front and rear wheels. This assumption of stability enables the further assumption that the behavior of each wheel can be independently modeled, since a significant deviation from even weight distribution would likely correspond to a crash.

Another potential issue of abstraction with this model is the potential for a finite-radius wheel to have multiple simultaneous points of contact with sufficiently rough ground, altering the appearance of the ground surface as perceived by the suspension. This concern is addressed by the SMA smoothing applied to the stochastic ground surface, as the characteristic length scale of large-scale terrain roughness is set as the wheel diameter.

C. Spring Constant Multipliers

To account for the displacement limitations of the suspension, its spring constant is modified by a multiplying factor that exponentially increases as suspension displacement reaches its limits. This allows the system to resolve displacement limits internally, although with the drawback of increased model sensitivity to time step resolution, as a slight overstep of the displacement limit will generate a large spring constant. A similar multiplying factor is used for the tire, but at the extension end of its displacement range the tire spring constant is reduced to zero to resolve ground contact. Equation 2 is used to calculate the modified spring constants.

These spring constant multipliers are an attempt to improve on the method employed by [10], in which the displacement constraints were resolved as additional objectives of the optimization problem. Figure 3 demonstrates the effect of these spring constant multipliers, overlaying the multipliers on normalized suspension and tire travel during one 10-second simulation.

$$x_1 = \frac{(y_1 - y_2) - (y_{1_0} - y_{2_0})}{x_{1,max}} \tag{2a}$$

$$k_{1_{mod}} = k_1\left[e^{-10(x_1+1)} + e^{10x_1} + 1\right] \tag{2b}$$

$$x_2 = \frac{y_2 - y_g}{y_{2_0} - y_{g_0}} \tag{2c}$$

$$k_{2_{mod}} = k_2\left[0.5\left[\tanh\left(100\left(1 - x_2\right)\right) + 1\right] + e^{-10x_2}\right] \tag{2d}$$

Equation 2a is the normalized suspension displacement, wherein $x_{1,max}$ refers to the suspension travel, and was set to 0.2 m. Equation 2c is the normalized tire displacement, and $(y_{2_0} - y_{g_0})$ was set as the zero-load tire thickness of 0.0635 m. The coefficient of 10 in each exponential term is the "ramp rate" of the displacement limit, and was chosen as the highest value that would still allow reasonable model stability. The hyperbolic tangent term in Equation 2d zeroes the tire spring constant when the tire is airborne.

Numerical instabilities were encountered somewhat frequently in solving the suspension model for certain design points, i.e. during optimization, and manual adjustment of the ramp rate in the exponential terms indicated an inverse correlation between ramp rate and model stability. These instabilities

were tolerated for this paper, as numerical instability due to the violation of displacement limits can reasonably be taken to imply physical damage to the suspension components in reality.

D. Damping Coefficient

The damping coefficient for the suspension is also adjusted as a function of current suspension motion rate to allow for direction (rebound/compression) and rate (high-/low-speed) dependence. The piecewise definition of the damping coefficient is implemented through a fit of three hyperbolic tangent functions to handle the step-like transitions between the four damping coefficients. Equation 3 is used to calculate the damping coefficient from a given pair of wheel and frame motion rates. Figure 4 shows the damping coefficient function overlaid on suspension motion rate simulation data.

$$\dot{y}_{net} = \dot{y}_1 - \dot{y}_2 \tag{3a}$$

$$\begin{aligned} b_1 = {} & \frac{(b_{1_{CL}} - b_{1_{CH}})}{2} \left[\tanh\left(100\left(\dot{y}_{net} - \dot{y}_{cross,C}\right)\right) + 1\right] \\ & + \frac{(b_{1_{RL}} - b_{1_{CL}})}{2} \left[\tanh\left(100\dot{y}_{net}\right) + 1\right] \\ & + \frac{(b_{1_{RH}} - b_{1_{RL}})}{2} \left[\tanh\left(100\left(\dot{y}_{net} - \dot{y}_{cross,R}\right) + 1\right)\right] \\ & + b_{1_{CH}} \end{aligned} \tag{3b}$$

In Equation 3b, the coefficient of 100 in each hyperbolic tangent term defines the "steepness" of each transition. This coefficient can be reduced to produce more realistically smoothed transitions. The four damping coefficients and damping regime crossover points are discussed further in Section IV.B.

E. Solving

The suspension model is solved as a system of second-order ODEs using the Julia `DifferentialEquations.jl` package, with the `DP5` solver algorithm. This solver was selected as the fastest for this task after measuring the time required to solve the model for a fixed time span with several of the solvers in the package. For each objective evaluation, the model is solved 20 times, each with a 10-second time span, with the random seed used to generate the terrain surface set based on the iteration number out of 20 to ensure repeatability. Figures 2, 3, and 4 show representative output data for a single 10-second run of the model, using the optimal design parameters from the [0.0, 1.0] objective weight case listed in Table 2, with a horizontal velocity of 5 m/s as was used for all cases during optimization. The objectives are calculated as heuristics from the stored state vector data produced by the ODE solver during each simulation. For most design points, the model can be run to evaluate the objectives in less than two seconds, but for some points encountered during optimization, the model can take one or two minutes to solve due to stability issues. Any simulation runs terminated due to instability are discarded and given an infinite objective value, in accordance with the above discussion of the impact of displacement limits on model stability.

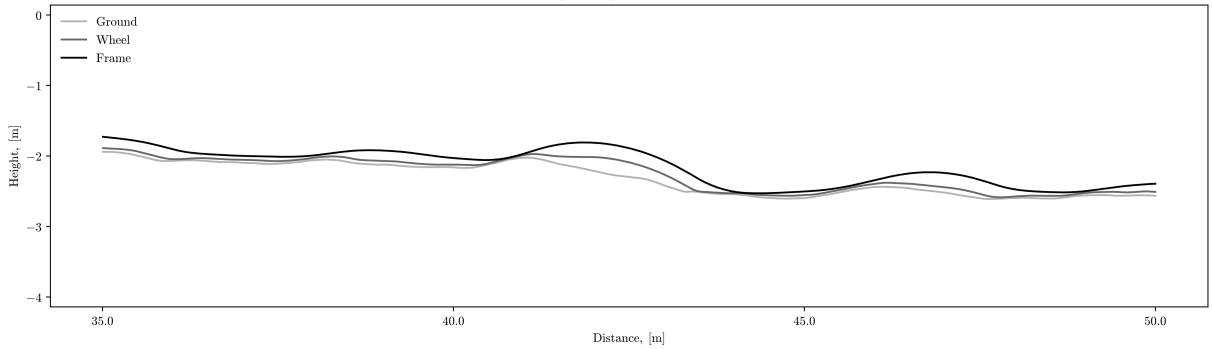

Fig. 2 Example Simulation Data for Third Design Point in Table 2: Ground, Wheel, and Frame Position

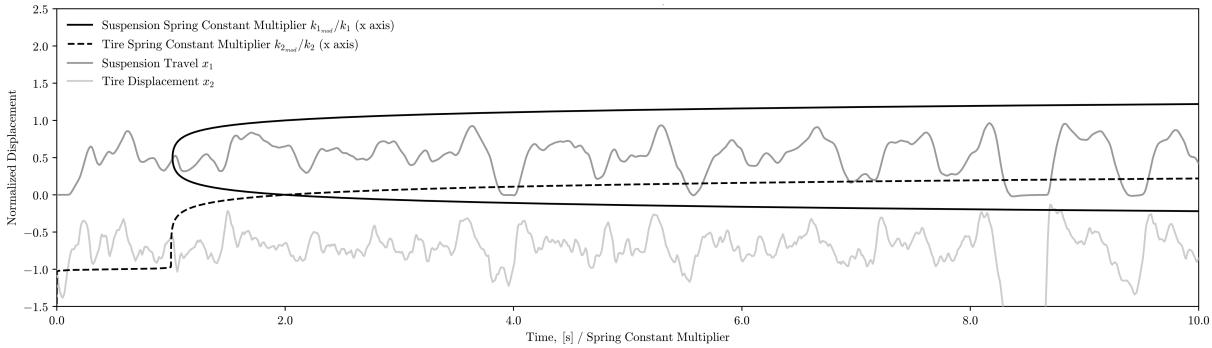

Fig. 3 Example Simulation Data for Third Design Point in Table 2: Normalized Suspension Travel/Tire Displacement

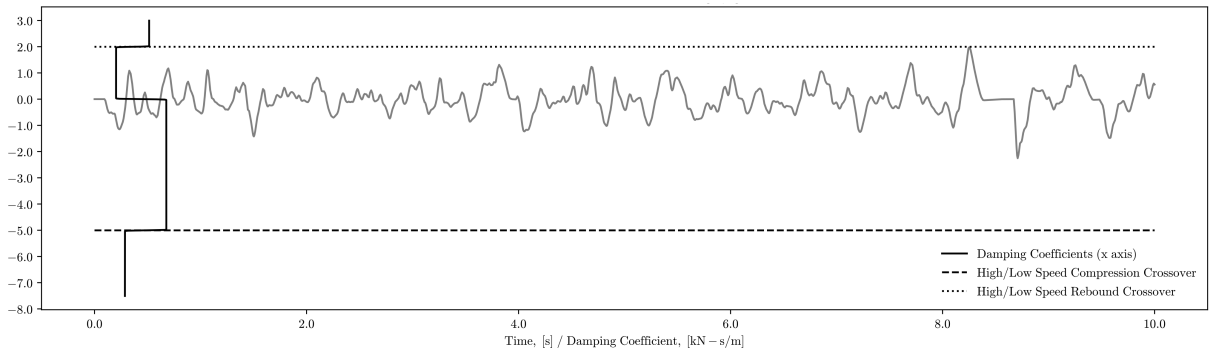

Fig. 4 Example Simulation Data for Third Design Point in Table 2: Net Suspension Motion Rate, [m/s]

IV. Problem Formulation

A. Objectives

The objectives of this optimization problem are "Ground Following" and "Ride Comfort," defined as:

- Ground Following - mean normalized distance between bicycle tire and ground (x_2, Equation 2c)
- Ride Comfort (Transmitted Impulse) - the RMS vertical acceleration of the bicycle frame ($\ddot{y}_1$, Equation 1a)

The minimization of these objectives corresponds respectively to: improved traction between the bicycle tires and ground; and improved rider comfort, and therefore control capability. In assessing the performance of suspension systems for other applications, sources [7, 10] use similar criteria. The Ground Following objective typically produces values of $O(10^{-1})$, and the Ride Comfort objective typically produces values of $O(10^{0})$.

B. Design Space

The design space of this problem consists of the different tuning parameters commonly available on mountain bike suspension:

- spring rate - spring constant of the suspension, adjusted in practice by air pressure or swapping mechanical springs
- tire spring constant - adjusted by air pressure
- compression damping - damping coefficient(s) applied only during the compression stroke of the suspension
- rebound damping - damping coefficient(s) applied only during the rebound, or extension, stroke of the suspension

The *compression damping* and *rebound damping* parameters are often subdivided into "low-speed" and "high-speed" components, where each applies only to compression or rebound events occurring above or below a specified crossover point in terms of suspension motion rate, to better enable the suspension to differentiate its response to "high-speed" phenomena such as surface roughness and to "low-speed" phenomena such as larger obstacles or rider input. The crossover points are also included as parameters in the design space. The design space is thus eight-dimensional:

$$[k_1,\ k_2,\ b_{1_{RH}},\ b_{1_{RL}},\ b_{1_{CL}},\ b_{1_{CH}},\ \dot{y}_{cross,R},\ \dot{y}_{cross,C}]$$

The default values for these parameters were determined through trial and error as the first combination of values that satisfied all constraints and reliably produced stable model results. These values are:

$$\begin{aligned} k_1 &= 6\ kN/m \\ k_2 &= 20\ kN/m \\ b_{1_{RH}} = b_{1_{RL}} = b_{1_{CL}} = b_{1_{CH}} &= 1\ kNs/m \\ \dot{y}_{cross,R} &= 2\ m/s \\ \dot{y}_{cross,C} &= -2\ m/s \end{aligned} \tag{4}$$

As shown, the values for these design parameters are reduced to $O(10^{0})$ or $O(10^{1})$ to allow for unit step size to be of reasonable magnitude to traverse the design space.

C. Constraints

Sign constraints are applied to each parameter, since negative damping coefficients and spring constants are not physically realistic. The signs of the damping coefficient crossover points are also enforced.

$$\begin{aligned} k_1,\ k_2,\ b_{1_{RH}},\ b_{1_{RL}}\ b_{1_{CL}}\ b_{1_{CH}},\ \dot{y}_{cross,R} > 0 \\ \dot{y}_{cross,C} < 0 \end{aligned} \tag{5}$$

The relationships between the spring constant k_1 and each damping coefficient b_{1_i} are also constrained, to avoid the "pogo-stick" effect of low damping, while still allowing the suspension to respond quickly enough to be effective [9].

$$\zeta_i = \frac{b_{1_i}}{2\sqrt{m_1 k_1}} \qquad 0.2 < \zeta_i < 1.0 \tag{6}$$

These constraints are implemented through a sum of quadratic and count penalties in the optimization problem, with a fixed multiplier of 100 for the combined penalty to ensure the penalty magnitude surpasses the magnitude of the objective.

V. Optimization

A. Algorithms

Seven algorithms were used to solve the optimization problem described above: two representing each gradient-free algorithm type (Direct, Stochastic, Population-based), along with CCD as the method frequently intuitively used in practice. All

algorithms used are adaptations of the Julia implementations in [11], and all were modified to terminate after a specified number of objective evaluations. Methods requiring a single start point were initialized with the default design point.

1. Cyclic Coordinate Descent (CCD)

CCD was implemented with Fibonacci Search, using Algorithms 3.1, 3.2, 4.1, and 7.2 in [11]. These algorithms were modified to allow for tracking of function evaluations, such that each `line_search` was allowed 20 function evaluations to be shared between the `bracket_minimum` and `fibonacci_search` stages. The algorithms were also modified to output the best single design point found during the search instead of a bracket. The initial step size `s` and step size expansion factor `k` for the `bracket_minimum` algorithm were set to 0.2 and 2.0 respectively, after some preliminary testing.

2. Generalized Pattern Search (GPS)

GPS was applied to this project with the positive and negative basis vectors as the positively-spanning set of search directions. Algorithm 7.6 in [11] was used, with the addition of differing step sizes for each step direction. The step size reduction factor was kept at the default of 0.5, and the initial vector of step sizes was [2.0, 2.0, 0.5, 0.5, 0.5, 0.5, 1.0, 1.0].

3. Nelder-Mead Simplex (NMS)

NMS was implemented as in Algorithm 7.7 of [11]. The starting simplex for NMS was composed of the default design point and eight other randomly generated design points. Each parameter of each random design point was generated by adding to the default parameter a random fraction of the step size specified for each parameter for GPS above, with random sign.

4. Adaptive Simulated Annealing (ASA)

ASA, which augments Simulated Annealing with the Corana step size update, was selected to represent the Stochastic methods due to its aptitude for noisy objective functions like this problem. ASA was implemented using Algorithms 8.5 and 8.6 from [11]. The `nt` hyper-parameter, the number of step update cycles before temperature reduction, was reduced from 100 to 2. A temperature of 1000 was used, and the default step sizes were the same as used for GPS.

5. Covariance Matrix Adaptation (CMA)

CMA was selected to represent the Stochastic methods due to its highly sophisticated approach, particularly due to its weighting of sampled points in updating its distribution, an advantage over similar but simpler Cross-Entropy Method and Natural Evolution Strategies. CMA was implemented using Algorithm 8.9 from [11], with modifications to track the best design point discovered and sample the mean of the distribution at each iteration. The only hyper-parameter adjusted for CMA was the step-size scalar, which was reduced to 0.8 from the default of 1.0 after some preliminary testing.

6. Particle Swarm Optimization (PSO)

PSO was selected to represent the Population-based optimization algorithms due to its simple and effective distributed search method. PSO was implemented using Algorithms 9.11 and 9.12 from [11]. The initial population for PSO was generated using the same randomized step method used to generate the initial simplex for NMS. The population size for PSO was adjusted between 10 and 20 to maximize the number of iterations possible before reaching the termination condition.

7. Firefly Algorithm

Firefly Algorithm was selected to represent the Population-based algorithms as similar but distinct compared to PSO. Firefly was implemented with Algorithm 9.13 from [11], with the addition of tracking of the best design point. The population for Firefly was generated using the same method as for PSO and NMS. The population size for Firefly was adjusted to allow at least two full iterations of the algorithm, but little room was left for adjustment due to the quadratic dependence of objective evaluation count on population size.

B. Objective Weighting

Since this problem involves two objectives, the performance of the algorithms could be compared when generating a Pareto Frontier or when targeting a specific weighting of the objectives. Since suspension tuning typically resembles the latter, a simple weighted sum was used with specific weightings for the objectives. The three objective weightings considered for this project, corresponding to Ground Following and Ride Comfort respectively, are: $[1.0, 0.0]$, $[0.5, 0.5]$, $[0.0, 1.0]$.

C. Results

Each algorithm was run once for each of the three objective weightings, with the number of function evaluations limited to 640. This number was chosen somewhat arbitrarily to allow for four full iterations of CCD. The weighted objective values resulting from this process are shown in Table 1, as are the best design points produced for each combination of weights in Table 2. Some of the results produced are somewhat unrealistic, for example the extremely high tire spring rate found by CCD for the first case due to the lack of a constraint on tire pressure. Additionally, the instability of the optimum found by NMS for the third weighting combination is somewhat questionable, as a design point that sometimes causes a failure of the model would likely correspond to a physically impractical or dangerous combination of parameters. These results could likely be improved upon with further effort dedicated to hyper-parameter tuning or more widely varied initialization conditions.

Additionally, none of the potential optima made use of the high-speed rebound or compression damping coefficients, since the motion rates generated by the simulations remained generally within the bounds of the crossover points. Future work could investigate improvements to the terrain generation scheme used here to allow for more variation without simply causing jumping, as was observed here for high terrain roughness.

Table 1 Weighted Objective Values

Algorithm	Weightings		
[Mean Rank]	[1.0, 0.0]	[0.5, 0.5]	[0.0, 1.0]
CCD [3.33]	0.0003 [1]	3.773 [6]	6.057 [3]
GPS [2.33]	0.0378 [2]	3.495 [4]	5.781 [1]
NMS [5.33]	0.0433 [3]	3.975 [7]	(unstable)
ASA [5.33]	0.4126 [6]	3.675 [5]	6.344 [5]
CMA [3.33]	0.2389 [5]	3.221 [1]	6.215 [4]
PSO [3.00]	0.1137 [4]	3.301 [3]	5.896 [2]
Firefly [2.66]	0.1137 [4]	3.277 [2]	5.896 [2]

VI. Conclusion

The results obtained from this project suggest that, despite its simplicity, CCD is a moderately effective method for optimizing suspension tuning, even when compared to far more sophisticated algorithms. Out of the seven algorithms tested, its average performance came in fourth, and in the weighting case specifically optimizing for ground following, CCD was able to vastly outperform the other algorithms. A more intriguing outcome of this analysis, however, is that GPS was the highest performing algorithm across all three weighting cases, despite being of similar or lesser complexity than CCD. Although the assumption at the inception of this project was that the greater sophistication and broader sampling inherent in the Stochastic and Population-based algorithms would correspond to a more optimal objective value with the same number of function evaluations, the results do not seem to bear this out clearly. Due to the limited degree of hyper-parameter tuning afforded to each algorithm prior to comparison testing for this project, one possible conclusion is that the algorithms less sensitive to hyper-parameter tuning were more successful. If this hypothesis is valid, then simpler algorithms such as CCD and GPS naturally emerge as better candidates for real-world application than the alternatives, due to their simple implementation, effectiveness in optimizing these objectives, and insensitivity to hyper-parameter tuning.

Table 2 Best Design Points

Weighting	Algorithm	k_1 kN/m	k_2 kN/m	$b_{1_{RH}}$ kNs/m	$b_{1_{RL}}$ kNs/m	$b_{1_{CL}}$ kNs/m	$b_{1_{CH}}$ kNs/m	$\dot{y}_{cross,R}$ m/s	$\dot{y}_{cross,C}$ m/s
[1.0, 0.0]	CCD	9.3482	52448.7	0.929493	1.3534	1.29536	1.18824	2.22229	-0.629508
[0.5, 0.5]	CMA	4.39099	26.2983	0.419942	0.236607	0.713198	0.639914	2.73297	-5.30933
[0.0, 1.0]	GPS	3.55062	27.4778	0.51524	0.203226	0.677425	0.286786	1.99681	-5.00467

References

[1] Cunningham, R., "Seen at Sea Otter 2013 - Zerode G2 Gearbox Bike," , 2013. URL https://www.pinkbike.com/news/Seen-at-Sea-Otter-2013-Zerode-G2-Gearbox-Bike.html.

[2] Vorsprung Suspension, "The Tuesday Tune Ep 21: Low and High-Speed Rebound Adjustments - Video," , 2017. URL https://www.pinkbike.com/news/vorsprung-tuesday-tune-ep-21-low-high-speed-rebound-video.html.

[3] Aston, P., "Fox Introduces New Factory Tuning Program," , 2017. URL https://www.pinkbike.com/news/fox-factory-tuning-program.html.

[4] Vorsprung Suspension, "New Products and Innovative Tuning System from Vorsprung," , 2017. URL https://www.pinkbike.com/news/new-tractive-luftkappe-tuning-systems-from-vorsprung.html.

[5] Formula, "Formula CTS Technology – New Approach to Fork Tuning," , 2017. URL https://www.pinkbike.com/news/formula--cts-technology-a-new-approach-to-fork-tuning.html.

[6] Levy, M., "Cane Creek's Dialed Tuning App - First Look," , 2016. URL https://www.pinkbike.com/news/cane-creek-dialed-tuning-app-first-look-2016.html.

[7] Rakheja, S., Ahmed, A. K. W., Yang, X., and Guerette, C., "Optimal Suspension Damping for Improved Driver- and Road-Friendliness of Urban Buses," *SAE Technical Paper 1999-01-3728*, 1999. doi:10.4271/1999-01-3728.

[8] Johnsson, A., Berbyuk, V., and Enelund, M., "Pareto optimisation of railway bogie suspension damping to enhance safety and comfort," *Vehicle System Dynamics*, Vol. 50, No. 9, 2012, pp. 1379–1407. doi:10.1080/00423114.2012.659846.

[9] Karnopp, D., "Active Damping in Road Vehicle Suspension Systems," *Vehicle System Dynamics*, Vol. 12, No. 6, 1983, pp. 291–311. doi:10.1080/00423118308968758.

[10] Hać, A., "Suspension optimization of a 2-DOF vehicle model using a stochastic optimal control technique," *Journal of Sound and Vibration*, Vol. 100, No. 3, 1985, pp. 343 – 357. doi:10.1016/0022-460X(85)90291-3, URL http://www.sciencedirect.com/science/article/pii/0022460X85902913.

[11] Kochenderfer, M. J., and Wheeler, T. A., *Algorithms for Optimization*, The MIT Press, 2019.