



# Optimizing Robotic Cheetah Leg Parameters Using Evolutionary Algorithms

Maxim Buzdalov<sup>(✉)</sup>, Sergey Kolyubin, Artem Egorov, and Ivan Borisov

ITMO University, Saint Petersburg, Russia  
{mbuzdalov,s.kolyubin,aaegorov,borisovii}@itmo.ru

**Abstract.** We present a new application suitable for evolutionary algorithms: geometry optimization for robotic applications. Our working example is a robotic cheetah leg, which uses simple control algorithms, but accurately crafted and tuned mechanics to maximize motion efficiency. In this paper we aim at tuning its parameters, such that the joints of the leg follow the desired trajectories as close as possible. Optimization is done in two stages involving just two parameters each.

Even this simply-looking problem presents a challenge to evolutionary algorithms, as it is both ill-conditioned and multimodal. However, we show that choosing a better fitness function that captures our desires in a different way can make the problem much easier.

**Keywords:** Robotic leg · Fitness function · Multimodal functions

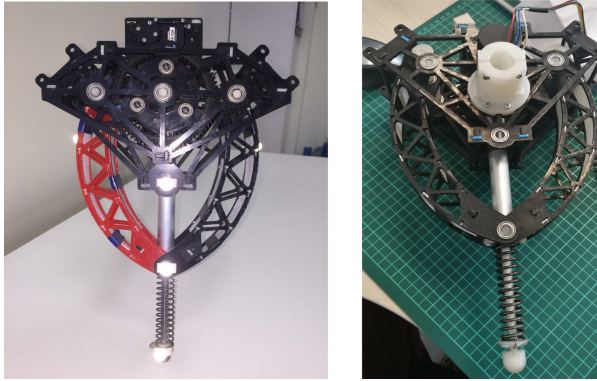
## 1 Introduction

Development of generative (or computational) design methods and algorithms became a hot topic in CAD-system in general, and robotics in particular, several decades ago [2, 9, 23]. However, with recent developments in machine learning and numerical optimization techniques we expect a new wave to come, bringing not only theoretical achievements, but also a number of interesting implementation examples. Modern methods of structure synthesis are used in various robotic tasks in manufacturing, aquaculture and fishery, technical medicine, automated inspection, entertainment and film making, and others [3, 7, 8, 11, 13, 15, 16, 20–22].

On the other hand, working in different application areas requires understanding of specific design requirements and constraints, which will define the optimization problem framework, including cost function selection. For example, synthesis of the mechanism of the robot feeder was carried out in accordance with the condition of minimum acceleration and speed limits of the end-effector [20]. In [11] synthesis of a surgical robot was carried out according to the parameters of the instrument working area, depending on its position for a serial robot. Various methods have been developed and applied for legged robots in particular.

---

Supported by Russian Science Foundation grant (project no. 17-79-20341).



**Fig. 1.** The prototype of the Minitaur mechanism built in the BE2R Lab, ITMO University, based on the concept presented in [17]

In [15] the walking bilateral robot with flexible passive legs has been developed, where each joint has been optimized to ensure smooth movements similar to a human gait. In [8] dynamic synthesis was performed for the jumping robot to reduce the torque on the drivers.

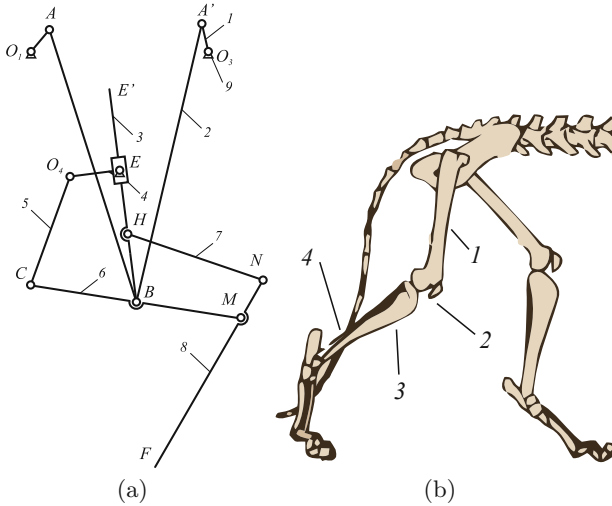
Generative design in general includes topology optimization, nevertheless the problem of robot geometric parameters optimization addressed in this work is an essential part of it. Moreover, for many tasks, especially when we stick to certain manufacturing technologies, doing parametric optimization covers 90% of what is needed and gives more cost-efficient solutions.

This research is motivated by the task to synthesize a structure that would mechanically repeat the desired behaviour of the bio-inspired robotic cheetah leg mechanism (see Fig. 1 for an example of our experimental test-bed). Same goal can be achieved if we leave system design as is and force desired motion by means of control algorithms along, but this approach has its bottlenecks. At first, it is much less energy efficient and requires much more complicated control systems both in terms of hardware (sensors and actuators) and algorithms.

In this sense, our work is related to recent developments of the Disney-Research team in the field of mechanism optimization. Mainly, they are solving animatronics tasks [3, 13, 21], jumping and walking robots projects [5, 14].

In our previous attempt, we used the Pattern Search optimizer [10] offered by the MATLAB environment. Since Pattern Search is an advanced form of continuous local search, which has good convergence properties but a lack of the global search ability, our next step, which we attempt in this paper, is to investigate how global optimizers perform on the same task. For this stage, we chose the representative continuous evolutionary optimizers from the two most prominent families which are known to perform well in various benchmarks: Differential Evolution [18] and Covariance Matrix Adaptation Evolution Strategy, or CMA-ES [6].

We should mention that even though this case study is related to legged robot design, the proposed approach is quite general to be applied for mechanism optimization of various robotic systems. On the other hand, these steps in rigid-body kinematics optimization can be a starting point for further studies including automatic elastic elements allocation and its elasto-static parameters adjustment, which was done manually in authors' preceding work [1].



**Fig. 2.** a) The mechanism of the Cheetah Leg: (1) cranks, (2) connecting rods, (3) crank arm/femur, (4) brick, (5) rocker, (6) connecting rod/tibia, (7) rocker/fibula, (8) output link/ankle, (9) frame, F is a contact point with ground/feet. b) Cheetah skeleton: (1) femur, (2) patella, (3) tibia, (4) fibula

The rest of the paper is organized in the following way. Sect. 2 sets some preliminaries including mechanical design issues to be taken into account, framework of the proposed parametric optimization, and description of the cost function used in our previous studies. Section 3 presents main contribution of this work, which is improving parametric convergence by means of evolutionary algorithms along with a newly suggested cost function. We also put here optimization results together with our observations on what properties the mechanism optimization problem has to make it an interesting challenge for evolutionary algorithms implementation. Section 4 concludes.

## 2 Preliminaries

### 2.1 Leg Mechanism Description

Topology of our robotic leg is inspired by the animal cheetah's muscle-skeletal structure (see Fig. 2(b)). But completely copying the legs' structure makes not

much sense, instead we want a mechanism that will reproduce desired animal-like motions, which is of interest for animatronic designs.

The leg can be divided in three parts: the hip, the knee, and the foot. The Minitaur mechanism (Fig. 2(a)) is a good choice for the robotic hip because of the following reasons. It can produce a large range of different gaits, controlling the difference in the angles of the input links and their speed. It is compact so the height and length of the jump does not depend on the width of the mechanism [17]. The input shafts  $O_1$  and  $O_2$  are rotating cranks  $O_1A$  and  $O_3A'$  to actuate leg mechanism. The mechanism of the Minotaur is responsible for moving the patella (see  $B$  on Fig. 2(a)).

The knee mechanism (see  $EO_4CB$  on Fig. 2(a)) controls the angle between the femur and the tibia within the specified range (see  $E'B$  and 6 on Fig. 2(a)). The knee mechanism is attached to the output link of the Minitaur mechanism (thigh) and to the brick (see  $EH$  on Fig. 2(a)).

The last group of the leg are tibia, fibula and ankle and they execute pantograph functions (see  $BM$ , 7 and 8 on Fig. 2(a)).

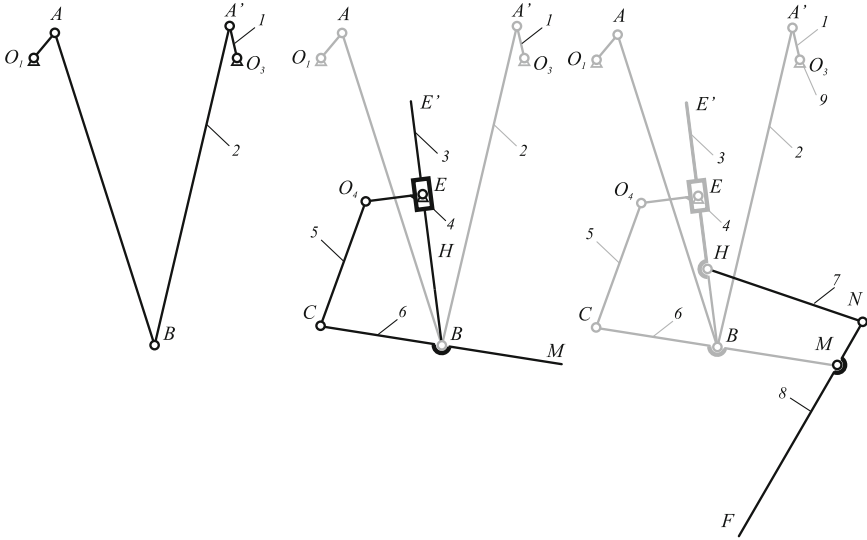
## 2.2 Decomposition and Constraints

Capturing rigid-body kinematics only, the mechanism can be described as a function transforming the input values like rotating cranks' angles to output values like contact point coordinates in Cartesian space. This function exists in the kinematic state space, which can be represented as all possible output values. This space can be narrowed depending on constraints imposed on kinematic parameters of the mechanism.

In our case, this transformation is indeed the forward kinematics nonlinear function, and lengths of the links are its parameters. So, in order to set the optimization problem first we need to decide, which parameters should be *a priori* fixed and which can be optimized, and what are upper and lower bounds for the latter ones. This choice is mostly based on mechanical design limitations.

Since the multi-link mechanism under consideration has quite many parameters that can be optimized, we decided to divide it to subsystems, define constraints and desired behaviour for each one, and therefore split optimization procedure into stages. This decomposition is illustrated in Fig. 3.

In order to save space and do not bother the reader with mechanical details we will not elaborate here on imposed design constraints, but mention that it were reasoned by conditions for the existence of linkages, overall mechanism dimensions' and structural strength's limitations. As the result, for example, for the hip subsystem length's of rods  $AB$  and  $A'B$  (upper bounded by 260 mm) and phase difference between input shafts  $0^\circ < \beta < 180^\circ$  were chosen as parameters for optimization, while the rest of subsystem's geometric parameters were fixed to pre-defined values. The choice of parameters allows to use basic box constraint handling techniques.



**Fig. 3.** Leg mechanism decomposition for optimization, from left to right: the hip, the knee, and the foot structures. Active black indicates a structure to be optimized.

### 2.3 Existing Cost Function

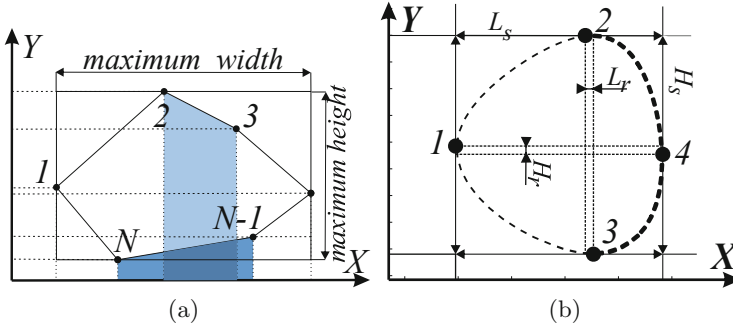
Our goal is to find geometric parameters values such that characteristic points of our mechanism will follow a desired trajectory, while the latter one is defined by certain key features important for locomotion like stride length and height, path curvature, and so on.

Driven by intention to minimize computational load, in our previous study we used the following optimization cost function (see [4] for details). We compared perimeters and areas for the calculated cyclic path and the desired ones as major metrics for curve-fitting. But since these two aggregated measures are not enough to guarantee that curve shapes will be close to each other, we augmented it with matching heights and lengths of a stride, i.e. minimizing distances between most left, right, top, and down points of the desired and calculated trajectories respectively (see Fig. 4(b)). These points were selected as special, because they break the trajectory into phases of jerk, leg pull-up, leg extension, and landing.

To simplify calculations, we approximated path traced by a point as a polygon, specified by pairs of the planar coordinates of its vertices. In this case the area  $S$  and perimeter  $P$  of the polygon were calculated by splitting it into segments, which are defined based on the polygon vertices locations (see Fig. 4(a)):

$$S = S_{N,1} + \sum_{i=1}^{N-1} S_{i,i+1}, \quad P = L_{N,1} + \sum_{i=2}^N L_{i-1,i},$$

where  $S_{i,j} = \frac{(x_j - x_i) \cdot (y_j + y_i)}{2}$  is the area of the trapezoid with indexes  $i$  and  $j$  denote respective points numbering, and  $L_{j,i} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ .



**Fig. 4.** a) A polygon that is constructed from an array of coordinates of a curve. b) 1, 4 are the leftmost and rightmost, and 2, 3 are the highest and lowest trajectory points respectively.  $L_s$  and  $H_s$  are stride length and height respectively

The resulting optimization cost function was calculated as

$$F = (S^* - S)^2 + (P^* - P)^2 + \sum_{i=1}^4 ((x_i^* - x_i)^2 + (y_i^* - y_i)^2),$$

where  $S$ ,  $S^*$ ,  $P$  and  $P^*$  denote the area and the perimeter of the polygons related to the calculated and the desired trajectories respectively (see Fig. 4(b)).

However, as it will be shown later in this paper, such cost function selection has certain disadvantages in terms of parametric convergence, especially when using evolutionary algorithms.

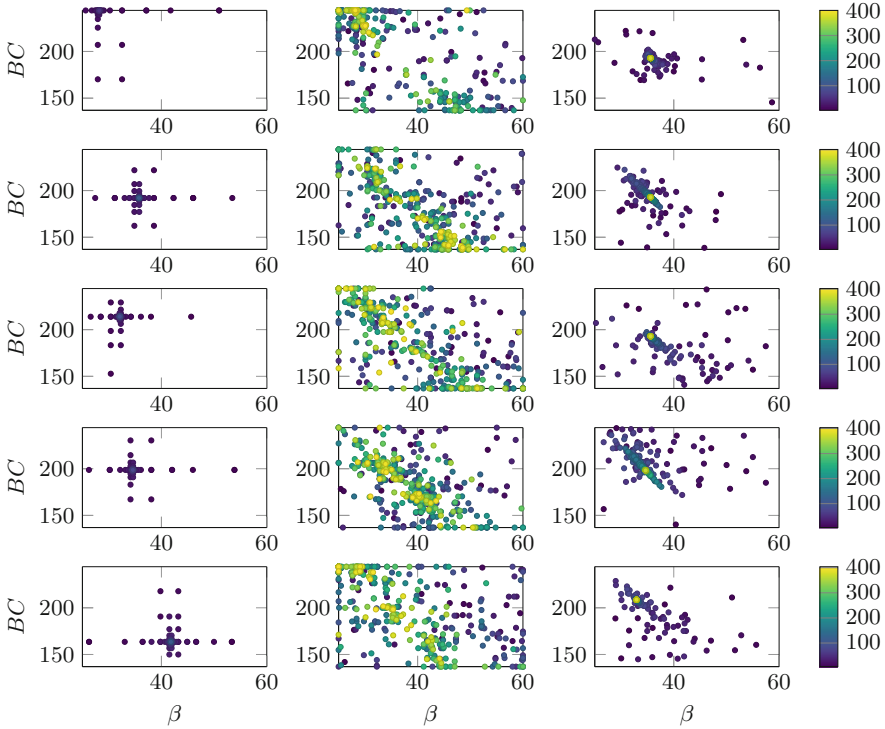
### 3 Experiments and Observations

In our experiments, we consider optimization of the first two stages: the hip (Fig. 3, left) and the knee (Fig. 3, middle). Both of them involve optimizing two variables, which gives an advantage of the convenient visual analysis of the behaviour of optimizers. We used the following optimizers:

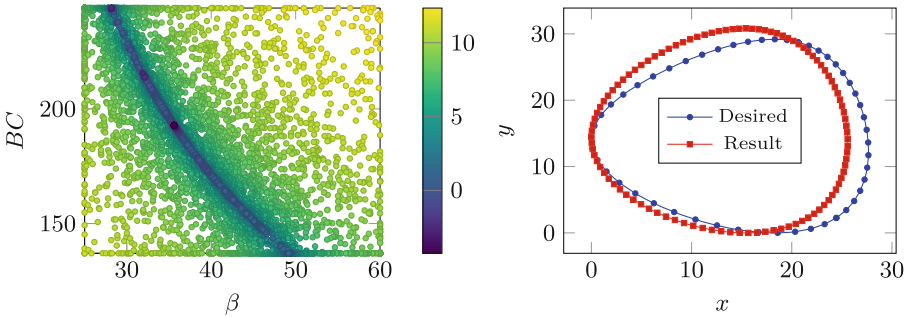
- Pattern Search, proposed in [10]; the MATLAB implementation was used with the following parameters: maximum of 40 iterations (which was enough for complete convergence), mesh size  $10^{-4}$ , mesh contraction factor 0.5, and the use of complete poll.
- Differential Evolution [18]: the DE/1/rand/bin variant with population size 40, crossover probability 0.5, differential weight sampled from [0.5; 1.5].
- CMA-ES [6]: the official implementation<sup>1</sup> with its default settings.

The computation budget was set to 400 for all the methods (however, Pattern Search tends to converge prematurely much earlier). As fitness functions involve modeling and hence are expensive, we performed only 10 runs for each optimizer.

<sup>1</sup> <https://github.com/CMA-ES/pycma>.



**Fig. 5.** Convergence in the search space: Pattern Search (left column), Differential Evolution (middle column) and CMA-ES (right column) on the old function for hip. Runs ranked 1, 3, 5, 7, 9 according to the final fitness value are shown. Point colors show the number of fitness evaluation



**Fig. 6.** Left: The fitness landscape of the old function for hip, with points colored according to the logarithm of fitness minus 8.88. Right: The best found trajectory for hip with fitness 8.8936 according to the old function

The considered problems have only basic box constraints, which Differential Evolution and Pattern Search can handle out of the box, so the only change was

to use  $10^9$  plus the distance to the feasible region as a fitness value for infeasible queries made by CMA-ES.

We discuss the **hip stage** first. Figure 5 displays five out of ten runs for each optimizer, chosen based on the resulting fitness values in order to showcase more of the possible outcomes of these algorithms. The colors in this figure are based on the number of the fitness function evaluations corresponding to each point, so these plots visually illustrate the convergence behaviour of the optimizers. One can see that Pattern Search converges very quickly, CMA-ES tests a larger region of the search space before convergence, and Differential Evolution is still scattered very much across the search space after 10 iterations. More experiments show that, for the latter, a wide range of population sizes from 10 to 40 results in roughly the same behaviour.

The best fitness values out of these ten runs were 8.9644 for Pattern Search, 8.9639 for Differential Evolution and 8.8936 for CMA-ES, and the medians were 10.4478, 9.4342 and 9.5830, respectively. The Wilcoxon rank sum test [12, 19] with the one-sided hypotheses based on these median values yields the following  $p$ -values:

- 0.0144 for Pattern Search vs CMA-ES,
- 0.0446 for Pattern Search vs Differential Evolution,
- 0.5733 for CMA-ES vs Differential Evolution,

which indicates that Pattern Search is worse on this problem than either of the evolutionary algorithms, whereas the latter perform similarly.

Figure 6 presents all the queries made by these algorithms combined on a single plot, where the color represents the logarithm of the fitness value (the darker the color, the better the fitness). As one can see, the fitness landscape of this problem resembles the Rosenbrock function with a number of local optima in the valley. It appears to be quite hard: in our case CMA-ES found a local optimum overlooked by other methods.

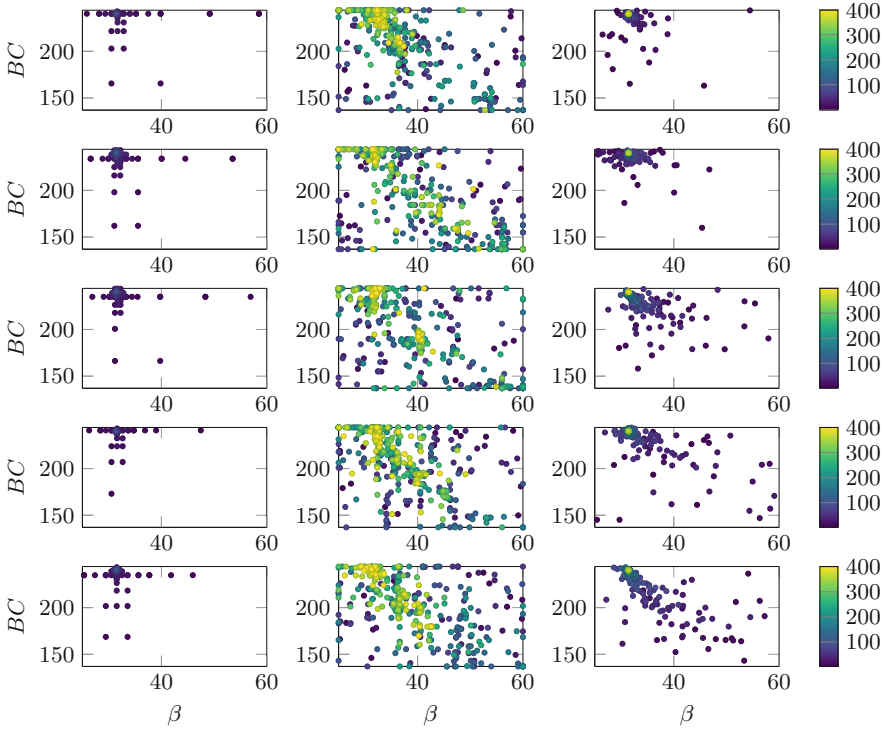
Figure 6 also presents the desired hip trajectory and the best found individual, according to the fitness function. We can see that, although the overall shape is captured correctly, there is still a visible difference between the trajectories.

In an attempt to escape from the complexities imposed by the fitness landscape of the existing fitness function, we have designed a **new one**, which is only marginally harder to compute, but captures the intended qualities better. This function is evaluated as follows:

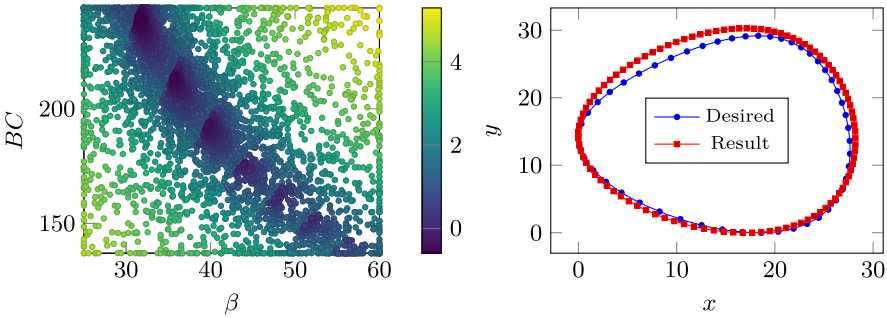
- the desired trajectory and the one generated by the evaluated individual are represented as piecewise linear functions  $D(t)$  and  $A(t)$  of the parameter  $t \in [0; 1]$  returning two-dimensional points, where  $t = 0$  means the first trajectory point,  $t = 1$  the last one, and other values are interpolated linearly;
- the value  $\int_0^1 (D(t) - A(t))^2 dt$  is computed using the two-pointer approach.

For the new fitness function, Fig. 7 and 8 represent the same data as for the original function. Based on the presented scatter plots, the convergence properties of the algorithms appears to remain the same. The best fitness values out of





**Fig. 7.** Convergence in the search space: Pattern Search (left column), Differential Evolution (middle column) and CMA-ES (right column) on the new function for hip. Runs ranked 1, 3, 5, 7, 9 according to the final fitness value are shown. Point colors show the number of fitness evaluation



**Fig. 8.** Left: The fitness landscape of the new function for hip, with points colored according to the logarithm of fitness. Right: The best found trajectory for hip with fitness 0.545307 according to the new function

**Table 1.** Statistical summary of the results for the knee stage experiments

	Pattern search	Differential evolution	CMA-ES
Old function			
Min	22.5552	22.8797	22.5541
Median	33.2892	24.1715	28.1541
New function			
Min	6.3197	6.3180	6.3183
Median	7.9530	6.3197	6.6584

the ten runs are 0.54546 for Pattern Search, 0.56200 for Differential Evolution and 0.54531 for CMA-ES, whereas the respective medians are 0.58083, 0.60336 and 0.57958. The  $p$ -values produced by the Wilcoxon rank sum tests are:

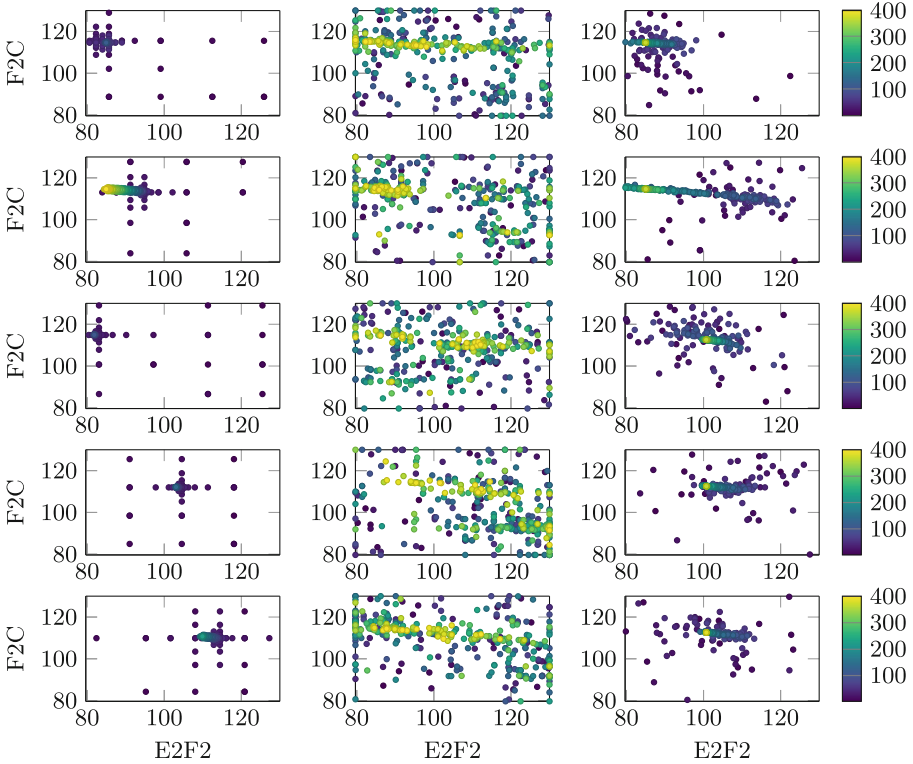
- 0.2644 for Differential Evolution vs Pattern Search,
- 0.09516 for Differential Evolution vs CMA-ES,
- 0.05256 for Pattern Search vs CMA-ES,

so all the algorithms now perform quite similarly. Together with the landscape observation on the left of Fig. 8 this indicates that the fitness landscape appears to be much easier, although still multimodal. One can also see on the right of Fig. 8 that the approximation of the desired trajectory is now much better.

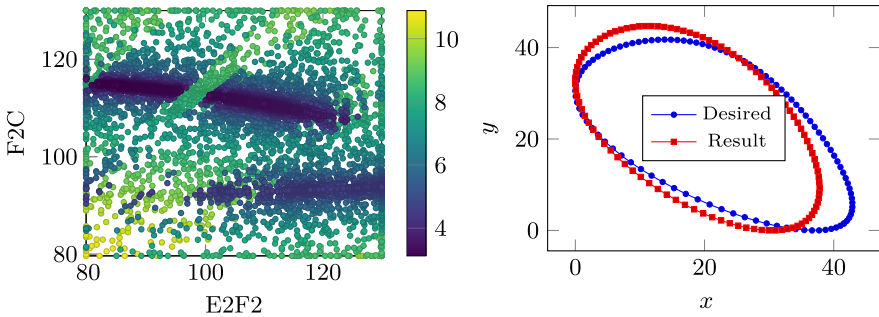
The same experiments with both functions performed for the **knee stage** demonstrate mostly the same trends (Fig. 9, 10, 11 and 12), although the fitness landscapes appear to be easier. Tables 1 and 2 present the basic statistical evaluation. They indicate, in particular, that all the algorithms are quite similar on the old function, whereas Differential Evolution performs somewhat better on the new function.

**Table 2.** The  $p$ -values by the Wilcoxon rank sum test for the knee stage experiments

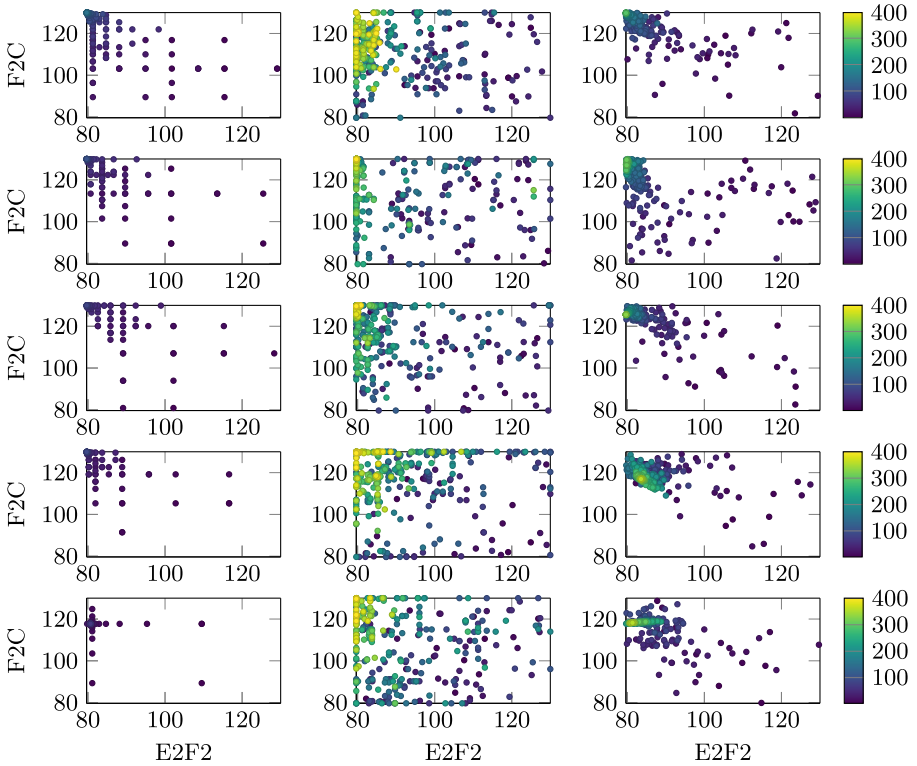
Null hypothesis	P.S. $\geq$ CMA	CMA $\geq$ D.E.	P.S. $\geq$ D.E.
Old function	0.0615	0.1237	0.0828
New function	0.2644	0.0014	0.0057



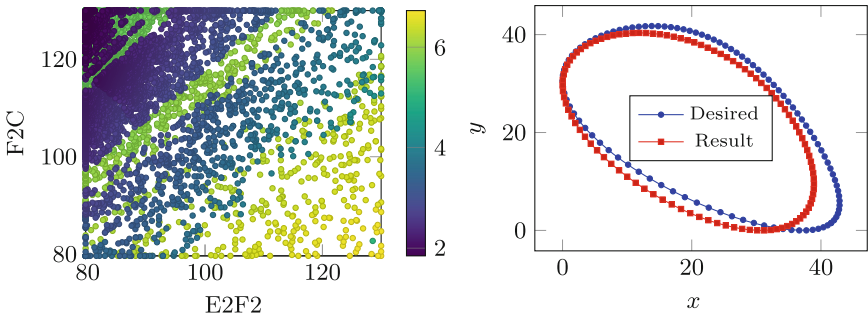
**Fig. 9.** Convergence in the search space: Pattern Search (left column), Differential Evolution (middle column) and CMA-ES (right column) on the old function for knee. Runs ranked 1, 3, 5, 7, 9 according to the final fitness value are shown. Point colors show the number of fitness evaluation



**Fig. 10.** Left: The fitness landscape of the old function for knee, with points colored according to the logarithm of fitness. Right: The best found trajectory for knee with fitness 22.5541 according to the old function



**Fig. 11.** Convergence in the search space: Pattern Search (left column), Differential Evolution (middle column) and CMA-ES (right column) on the new function for knee. Runs ranked 1, 3, 5, 7, 9 according to the final fitness value are shown. Point colors show the number of fitness evaluation



**Fig. 12.** Left: The fitness landscape of the new function for knee, with points colored according to the logarithm of fitness. Right: The best found trajectory for knee with fitness 6.31798 according to the new function

## 4 Conclusion

We have investigated the problem of parameter tuning for the robotic cheetah leg from the viewpoint of evolutionary computation. Regarding the considered methods, both CMA-ES and differential evolution tend to outperform the previously used local search method Pattern Search in terms of solution quality. Which of the considered evolutionary algorithms is better, depends on the particular problem variation in question. Based on the observed scatter plots, CMA-ES is typically better at exploitation, whereas Differential Evolution seems to be stronger at exploration, however, the overall difference is not very large.

The investigation of whether the recent advances in the corresponding algorithmic families have an impact in these problems is a possible future work. We also plan to investigate how the simultaneous optimization of all the parameters of the robotic cheetah leg relates to the already-considered chain of independent optimization runs, as well as to attempt the complete structure synthesis rather than just geometry optimization.

The most insightful part is that a choice of the fitness function, even among those which have the same global optimum and express the same desires, can significantly influence the quality of the solutions and the hardness of the problem. The fitness function used in previous studies appears to be both ill-conditioned and multimodal, which makes the problem especially hard, whereas the new function is much simpler.

## References

1. Borisov, I.I., Larkina, A.E., Egorov, A.A., Kulagin, I.A., Kolyubin, S.A., Stramigioli, S.: Study on elastic elements allocation for energy-efficient robotic cheetah leg. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robotics, pp. 1696–1701 (2019)
2. Cagan, J., Campbell, M.I., Finger, S., Tomiyama, T.: A framework for computational design synthesis: model and applications. *J. Comput. Inf. Sci. Eng.* **5**(3), 171–181 (2005)
3. Coros, S., et al.: Computational design of mechanical characters. *ACM Trans. Graph.* **32**(4), 1 (2013)
4. Egorov, A., Larkina, A., Borisov, I., Kolyubin, S., Stramigioli, S.: Cascaded constrained optimization for cheetah-inspired galloping robot leg mechanism design. In: IFAC World Congress (2020, accepted for publication)
5. Ha, S., Coros, S., Alspach, A., Kim, J., Yamane, K.: Task-based limb optimization for legged robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2016)
6. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.* **9**, 159–195 (2001)
7. Hao, Q., Li, S., Liu, B.: The kinematic performance optimization of a 2-DoF parallel robot based on new variance indices. In: IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (2016)
8. Heab, G., Geng, Z.: Dynamics synthesis and control for a hopping robot with articulated leg. *Mech. Mach. Theory* **46**(11), 1669–1688 (2011)

9. Heisserman, L.: Generative geometric design. *IEEE Comput. Graphics Appl.* **14**(2), 37–45 (1994)
10. Hooke, R., Jeeves, T.A.: “Direct search” solution of numerical and statistical problems. *J. ACM* **8**(2), 212–229 (1961). <https://doi.org/10.1145/321062.321069>
11. Lum, M.J.H., Rosen, J., Sinanan, M.N., Hannaford, B.: Optimization of a spherical mechanism for a minimally invasive surgical robot: theoretical and experimental approaches. *IEEE Trans. Biomed. Eng.* **53**(7), 1440–1445 (2006)
12. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* **18**(1), 50–60 (1947)
13. Megaro, V., et al.: Designing cable-driven actuation networks for kinematic chains and trees. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2017)
14. Megaro, V., Zehnder, J., Bächer, M., Coros, S., Gross, M., Thomaszewski, B.: A computational design tool for compliant mechanisms. *ACM Trans. Graph.* **36**(4), 1–12 (2017)
15. Ogura, Y., et al.: Human-like walking with knee stretched, heel-contact and toe-off motion by a humanoid robot. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2006)
16. Shakiba, M., Shadmehr, M.H., Mohseni, O., Nasiri, R., Ahmadabadi, M.N.: An adaptable cat-inspired leg design with frequency-amplitude coupling. In: *International Conference on Robotics and Mechatronics* (2016)
17. Snippe, M.S.: Cheetah robot leg mechanism: analysis, design and cost of transport. Master’s thesis, University of Twente, Netherlands (2017)
18. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**(4), 341–359 (1997)
19. Wilcoxon, F.: Individual comparisons by ranking methods. *Biometrics Bull.* **1**(6), 80–83 (1945)
20. Xi, Y., Xia, Y., Li, X., Sun, Y., Wang, H.: Kinematic analysis and parameter optimization of six-bar linkage based on ADAMS. In: *International Conference on Mechanic Automation and Control Engineering* (2010)
21. Xu, H., Knoop, E., Coros, S., Bächer, M.: Bend-it: design and fabrication of kinetic wire characters. *ACM Trans. Graph.* **37**(6), 239:1–239:15 (2018). <https://doi.org/10.1145/3272127.3275089>
22. Zhang, J., Jin, Z., Feng, H.: Type synthesis of a 3-mixed-DOF protectable leg mechanism of a firefighting multi-legged robot based on GF set theory. *Mech. Mach. Theory* **130**, 567–584 (2018)
23. Zhou, M., Rozvany, G.I.N.: The COC algorithm, part II: topological, geometrical and generalized shape optimization. *Comput. Methods Appl. Mech. Eng.* **89**(1), 309–336 (1991)