MAS-Bench: Parameter Optimization Benchmark for Multi-agent Crowd Simulation

Extended Abstract

Shusuke Shigenaka
University of Tsukuba
shusuke-shigenaka@aist.go.jp

Shunki Takami
University of Tsukuba
s-takami@aist.go.jp

Shuhei Watanabe
University of Freiburg
watanabs@tf.uni-freiburg.de

Yuki Tanigaki
AIST
tanigaki.yuki@aist.go.jp

Yoshihiko Ozaki
AIST & GREE, Inc.
ozaki-y@aist.go.jp

Masaki Onishi
AIST
onishi-masaki@aist.go.jp

KEYWORDS

Evaluation Tool, Pedestrian Simulator, Parameter Estimation

ACM Reference Format:
Shusuke Shigenaka, Shunki Takami, Shuhei Watanabe, Yuki Tanigaki, Yoshihiko Ozaki, and Masaki Onishi. 2021. MAS-Bench: Parameter Optimization Benchmark for Multi-agent Crowd Simulation: Extended Abstract. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

Multi-agent crowd simulation is generally used to construct an environment suitable for reality by parameter estimation. This estimation is to modify a model defined by human movements and behaviors to fit real-world data by means of statistics or optimization [2, 11]. In the field of traffic engineering, for example, parameter estimation is used to modify speed models and collision models of pedestrians and vehicles [10, 13]. For different simulation settings, it is not easy to obtain original data of each real-world environment. Few existing estimation methods have been fairly evaluated. Therefore, there is a need for a benchmark to discuss the applicability of certain estimation methods to other use cases.

We propose a benchmark based on a multi-agent system (MAS), MAS-Bench, for parameter estimation on a crowd simulation. The benchmark allows its users to challenge parameter optimization of a complicated real-world system easily. As shown in Figure 1, MAS-Bench provides an integrated benchmark to optimize parameters without any initial construction of simulation models or related systems. We hope that MAS-Bench will boost the further improvement of potential optimization methods, mostly applicable to high-dimensional problems and data assimilation methods required in real-time simulation.

MAS-Bench will be publicly available on GitHub[1].

2 PROPOSED BENCHMARK: MAS-BENCH

2.1 Problem Setting of Parameter Estimation

MAS-Bench offers modeling of a real-world crowd routing to exit from a large-scale outdoor festival site. The benchmark handles a problem to minimize the error between an original value O and its simulated value S by adjusting input parameter values. O and S are observable values with pedestrian walking speed from the main festival venue (origin) to the nearest station (destination) and the number of people at the destination. θ is an unobservable parameter set of the Gaussian mixture model (GMM) to estimate the number of people at the origin. The problem is formulated as follows:

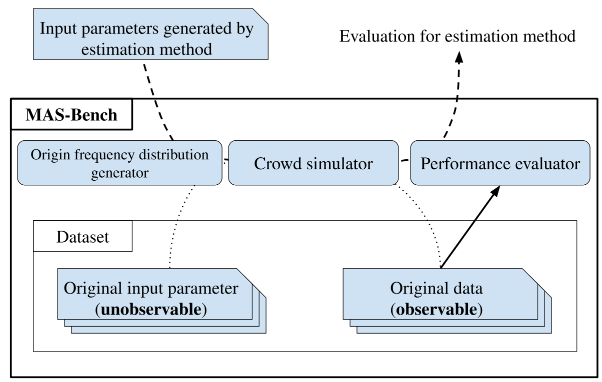

Figure 1: Overview of the proposed MAS-Bench.

$$\theta^* = \mathrm{argmin}_{\theta \in \Theta} \epsilon_{\mathrm{all}}(\theta, O, S) \quad (1)$$

The error ϵ_{all} is computed as the sum of the linear combination of ϵ_{dest} and ϵ_{pac}, where ϵ_{dest} is the error in the number of arrivals, and ϵ_{pac} is the error in the walking speed. Each error is calculated as a root means square error (RMSE).

MAS-Bench evaluates an input set θ of Gaussian distribution parameters: a ratio π, mean μ, and variance σ of the d-th Gaussian distribution. The Gaussian model $f_d(t;\theta_d)$ of the d-th basis of $f(t;\theta)$ can be described as follows:

$$f_d(t;\theta_d) = \frac{C_d \pi_d}{\sqrt{2\pi\sigma_d^2}} \exp\left(-\frac{(t-\mu_d)^2}{2\sigma_d^2}\right) \quad (2)$$

$$f(t;\theta) = \sum_{d=1}^{D} f_d(t;\theta_d) \quad (3)$$

where $C_d = 1/\int_0^T f_d(t;\mu_d,\sigma_d)dt$ denotes the normalization constant, and T denotes the closing time of the simulation.

[1] https://github.com/MAS-Bench/MAS-Bench

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.
© 2021 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

Table 1: Performance evaluation of observable and unobservable data using RMSE (mean and variance values after five iterations).

	2D		5D		8D		17D	
	Observable	Unobservable	Observable	Unobservable	Observable	Unobservable	Observable	Unobservable
RS	10.37 (1.53)	**0.64 (0.12)**	13.97 (3.48)	10.77 (0.98)	17.53 (2.39)	17.37 (0.22)	14.95 (1.85)	16.66 (0.99)
PSO	**6.69 (0.48)**	0.65 (0.21)	8.03 (2.35)	13.34 (5.51)	16.01 (7.23)	16.43 (3.80)	11.06 (0.66)	14.80 (2.01)
CMAES	**6.71 (0.53)**	0.80 (0.40)	**5.42 (1.00)**	6.96 (3.69)	**5.57 (0.43)**	**9.04 (1.90)**	**9.55 (0.53)**	**13.03 (0.77)**
SHADE	8.81 (1.09)	0.94 (0.34)	6.12 (0.49)	**4.37 (1.21)**	**7.35 (1.15)**	**8.29 (2.80)**	12.02 (1.51)	**11.52 (1.06)**
TPE	7.44 (0.61)	**0.61 (0.16)**	**4.90 (0.13)**	**4.35 (1.60)**	7.96 (0.88)	13.38 (3.52)	**9.74 (0.79)**	13.40 (3.16)

2.2 Crowd Simulation

In MAS-Bench, the multi-agent crowd simulator uses actual measurements such as agents' walking flow data with RGB-D cameras and each walking distance between the origin and destination with a handy Global Positioning System (GPS). Both were obtained every 15 minutes on Route 1 (320m), Route 2 (600m), and Route 3 (770m). These different lengths of routes are provided to control the flow on each route and to alleviate congestion. The crowd simulator sets four types of walking agents according to walking speed and roles: Guided Agent as the speed baseline, Busy Agent at 10% faster than the Guided Agent, Slow Agent at 50% slower, and Pacing agent at an average walking speed within one link (about 30m).

2.3 Baseline of Optimization Methods

MAS-Bench provides five optimization methods: Random search (RS), Particle swarm optimization (PSO), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), Success-History based Adaptive Differential Evolution (SHADE), and Tree-structured Parzen Estimator (TPE).

RS is an optimized baseline algorithm to randomly generate an input parameter set, specifically for a high-dimensional and multimodal function [3]. PSO is a typical metaheuristic algorithm to seek an optimal solution according to adaptability to a given environment [6]. CMA-ES is a state-of-the-art method of black-box optimization [7–9] to generate an input parameter set from a multivariate normal distribution of a suitable parameter set. SHADE is also known as a state-of-the-art method of black-box optimization [12], derived from Joint Approximation Diagonalization of Eigen-matrices (JADE) [14]. TPE is a high-performance algorithm adopted for Bayesian optimization by Hyperopt [4] and Optuna [1] to probabilistically determine the next input parameter set from the evaluation history [5].

3 EXPERIMENTS: USE CASE OF MAS-BENCH

This section demonstrates a use case of MAS-Bench parameter optimization with the experimental results.

3.1 Experiment Settings

To compare the performance of optimization methods, we evaluated $((D+1)/3)^2 \times 20$ [input parameter sets] ×40 [generations] where D is the number of dimension (parameters). Each dimension is set based on calculation of [distribution]×[agent type]×[distribution parameter set]. We considered four types of benchmark problems.

- 2D for Guided Agent
- 5D for Guided Agent and Busy Agent
- 8D for one distribution of three agent types
- 17D for two distributions in one agent type

The total number of walking agents is 45,000 for a large-scale crowd problem and 4,500 for a small-scale crowd problem. The number of pacing agents is set to 900 for a large-scale and 90 for a small-scale. The departure agent ratio is 0.01 for both problems. We used a calculator of Intel CPU Core i9-9900K (3.5GHz, 10 cores, 20 threads), which can calculate up to 20 input parameter sets simultaneously per calculation or one minute for an input parameter set.

3.2 Results of MAS-Bench Use Case

Table 1 lists the results of evaluating the small-scale crowding problem (Number of the walking agents 4,500) by dimension, showing the mean and variance values after five iterations of optimization. The best solution among the five optimization methods is shown in red bold, and the next best solution is shown in bold. From the experimental results, for observable data, CMA-ES performed better than the other algorithms. For unobservable data, TPE is better for low-dimensional problems (2D and 5D), and SHADE is better for high-dimensional problems (8D and 17D). For RS and PSO, the results for Observable and Unobservable are both poor except for 2D. This suggests that this benchmark can estimate parameters closer to the original values by minimizing Observable data's error.

4 CONCLUSION

This study proposed MAS-Bench, a parameter optimization benchmark to enable users to evaluate their optimization methods. MAS-Bench provides MAS-simulation modeling of crowd flows from the origin to the destination of the small-scale and large-scale problems. The MAS-Bench crowd simulator controls the crowd flows by varying in a set of input parameters. Users can optimize the unobservable parameter set by applying their optimization methods to MAS-Bench to fairly analyze the candidate methods under the same model and settings.

The present MAS-Bench uses estimates by the simulator as original values. In the future study, we will introduce MAS-Bench using original values and examine the multidisciplinary correlation of real-world phenomena.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-Generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2623—-2631.

[2] Wael K.M. Alhajyaseen and Hideki Nakamura. 2010. Quality of pedestrian flow and crosswalk width at signalized intersections. *IATSS Research* 34, 1 (2010), 35–41.

[3] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research* 13, 1 (2012), 281–305.

[4] James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. 2015. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery* 8, 1 (2015), 014008.

[5] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems* (Sierra Nevada, Spain) *(NIPS'11)*. Curran Associates Inc., Red Hook, NY, USA, 2546–2554.

[6] Mohammed El-Abd and Mohamed S. Kamel. 2009. Black-Box Optimization Benchmarking for Noiseless Function Testbed Using Particle Swarm Optimization. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers* (Montreal QC, Canada) *(GECCO '09)*. Association for Computing Machinery, New York, NY, USA, 2269–2274.

[7] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. 2010. Comparing Results of 31 Algorithms from the Black-Box Optimization Benchmarking BBOB-2009. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation* (Portland, Oregon, USA) *(GECCO '10)*. Association for Computing Machinery, New York, NY, USA, 1689–1696.

[8] Nikolaus Hansen and Stefan Kern. 2004. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In *Parallel Problem Solving from Nature - PPSN VIII* (Birmingham, UK). Springer Berlin Heidelberg, Berlin, Heidelberg, 282–291.

[9] Monte Lunacek and Darrell Whitley. 2006. The Dispersion Metric and the CMA Evolution Strategy. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation* (Seattle, Washington, USA) *(GECCO '06)*. Association for Computing Machinery, New York, NY, USA, 477–-484.

[10] Himangshu Saikia, Fangkai Yang, and Christopher Peters. 2019. Priority Driven Local Optimization for Crowd Simulation. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (Montreal QC, Canada) *(AAMAS '19)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2180–2182.

[11] Hitoshi Shimizu, Tatsushi Matsubayashi, Akinori Fujino, and Hiroshi Sawada. 2020. *Theme Park Simulation Based on Questionnaires for Maximizing Visitor Surplus*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2002–2004.

[12] Ryoji Tanabe and Alex Fukunaga. 2013. Success-history based parameter adaptation for differential evolution. In *2013 IEEE congress on evolutionary computation* (Cancun, Mexico). 71–78.

[13] David Wolinski, S J. Guy, A-H Olivier, Ming Lin, Dinesh Manocha, and Julien Pettré. 2014. Parameter Estimation and Comparative Evaluation of Crowd Simulations. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 303–312.

[14] Jingqiao Zhang and Arthur C Sanderson. 2009. JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation* 13, 5 (2009), 945–958.