Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

SURROGATE ASSISTED CALIBRATION FRAMEWORK FOR CROWD MODEL CALIBRATION

Wenchao Yi

School of Computer Science and Engineering
Nanyang Technological University
50 Nanyang Avenue
639798, SINGAPORE

Jinghui Zhong

School of Computer Science and Engineering
South China University of Technology
No.381 Wushan Lu, Tianhe District
Guangzhou, 510640, CHINA

Singkuang Tan
Wentong Cai

School of Computer Science and Engineering
Nanyang Technological University
50 Nanyang Avenue
639798, SINGAPORE

Nan Hu

Institution of High Performance Computing
Agency for Science Technology and Research
1 Fusionopolis Way, 16-16 Connexis North
138632, SINGAPORE

ABSTRACT

Surrogate models are commonly used to approximate the multivariate input or output behavior of complex systems. In this paper, surrogate assisted calibration frameworks are proposed to calibrate the crowd model. To integrate the surrogate models into the evolutionary calibration framework, both the offline and online training based approaches are developed. The offline training needs to generate training set in advance, while the online training can adaptively build and re-build the surrogate model along the evolutionary process. Our simulation results demonstrate that the surrogate assisted calibration framework with the online training is effective and the surrogate model using artificial neural network obtains the best overall performance in the scenario evaluated in the case study.

1 INTRODUCTION

Crowd simulation and modeling gains a lot of attention in recent years. It has been applied in many applications like evacuation modeling, urban planning, and military training (Duives, Daamen, and Hoogendoorn 2013). The Flow-based approach (Tajima and Nagatani 2001), entity-based approach (Helbing and Molnar 1995), and agent-based approach (Zhong et al. 2015) are the most used methods in crowd modeling. Among them, agent-based methods treat each individual as the separate entity that can move autonomously according to predefined rules. Compared with other methods, agent-based approaches can generate more realistic and customized behavior due to the fact that it considers the local interactions of surroundings.

In the crowd modeling and simulation, one of the most important problems is the model calibration. The parameters of the crowd model need to be fine tuned so that the generated simulation results can fit well with the motion in the real world. Evolutionary algorithms (EAs) offer an efficient way to solve the problems. Johansson, Helbing, and Shukla (2007) presented an evolutionary algorithm to determine the optimal parameter setting of the social force model. In the paper, they presented three specifications of the interaction forces between the pedestrians. The EA is used to find a robust parameter setting for the improved social force models on the tested scenarios. Wolinski et al. (2014) proposed a general optimization framework for parameter estimation in their five agent-based simulation models. Several algorithms are used as the search engine of the framework, among which a combination of genetic and greedy algorithm obtains the best performance. Zhong et al. (2015) proposed differential evolution (DE) algorithm to calibrate the parameter setting of the social force model. Later, they proposed an improved DE algorithm for calibrating the social force model (Zhong and Cai 2015). Their experimental results show that the improvements of DE algorithm are effective in finding better setting of their parameters.

The EA is effective in finding the proper parameter settings, although it may take a long time to find the setting. Rather than improving the EA itself, an effective way is to use surrogate model to build a substitute model for the fitness evaluation. To further extend the EA-based crowd model calibration framework in (Zhong et al. 2015), in this paper we propose two approaches to integrate the surrogate models. The most popular surrogate models include Support Vector Machine (SVM) (Chang and Lin 2011), Artificial Neural Network (ANN) (Yan et al. 2017), Gaussian Process (GP) (Huang et al. 2006, Erickson, Ankenman, and Sanchez 2016). They are used in our proposed frameworks.

Some recent works on surrogate models in the area of simulation optimization focus on exploring combined usage of multiple surrogate models. Chiu et al. (2016) used multi-fidelity models to assist the EA in solving large scale optimization problems. Fan and Hu (2016) proposed a random search algorithm for the simulation optimization using promising area search and surrogate model. The algorithm constructs and optimizes a series of surrogate models that approximate the objective functions on the promising search area iteratively. Some other works focus on improving a specific surrogate model. Meng and Ng (2016) proposed an improved additive global and local GP model that can automatically change between global search and local search for optimization problem. The search region is gradually narrowing down but with certain jumping-out probability. Pedrielli and Ng (2016) used Kriging as the meta-model and trust region method to find the promising solutions instead of the sampling method.

Different from the existing work on surrogate models in the area of simulation based optimization, this paper aims to integrate surrogate models into an EA-based crowd model calibration framework. The potential challenge in embedding surrogate models into the EA-based crowd model calibration framework is that the surrogate model itself needs to be well trained in order to achieve good performance. Hence, we focus on identifying the effective surrogate model as well as the effective way to train the surrogate model to assist the EA in solving the crowd model calibration problem.

The rest of the paper is organized as follows: Section 2 describes the density-based evolutionary framework. Section 3 presents the proposed surrogate assisted calibration framework with an offline or online training. Surrogate models comparison and simulation results of the studied scenario are investigated in Section 4. Section 5 concludes the paper.

2 DENSITY-BASED EVOLUTIONARY FRAMEWORK FOR CROWD MODEL CALIBRATION

In this section, we will briefly introduce the EA-based crowd model calibration process. The related detailed description on the density-based evolutionary framework for the crowd model calibration can be found in Zhong and Cai (2015).

2.1 Distance Measurement for Crowd Model Calibration

The objective of crowd model calibration is to calibrate the agent's model parameters so that the simulation result matches with what is observed in the real scenario. The density measurement reflects the macroscopic level dynamics of the crowd. Hence, this paper focuses on the density-based calibration method to measure the differences between the generated agents' movement and the expected real motion.

In each time-step, the position of each agent is recorded. Then, the recorded density distribution is compared with the expected one at each discrete time-step. The objective is to minimize the sum of the difference between the simulation and the real scenario. As illustrated by Zhong and Cai (2015), the region is divided into $W*L$ grids, where W is the width and L is the length of the region, and the agents' density

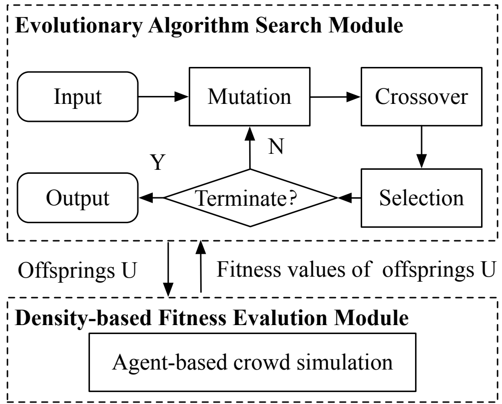


Figure 1: The evolutionary algorithm based search mechanism.

through the simulation in each grid can be calculated as follows:

$$\rho(\chi,t) = \sum_{i=1}^{c} \frac{1}{\pi R^2} * exp(-\frac{||r_i(t)-\chi||^2}{R^2}). \tag{1}$$

where c is the total number of the pedestrians in the observed region. $r_i(t)$ is the position of agent i at the time t and χ is the center of the grid. R is the kernel radius and is set as 2. The distance measurement can then be defined as:

$$Distance = \sum_{t=1}^{T} (\sum_{i=1}^{W*L} (|\rho(\chi_i,t) - \varsigma(\chi_i,t)|)). \tag{2}$$

where $\varsigma(\chi_i,t)$ is the objective behavior at the time t in the grid χ_i; $\rho(\chi_i,t)$ is the density of the agent-based simulation in the grid i at the time t, and T is the simulation duration. The goal is to find the best parameter setting so as to minimize the sum of the differences in all the grids at each time-step between the real motion and the one generated by the agent-based simulation.

2.2 The Evolutionary Algorithm Based Search Mechanism

The EA-based search mechanism is shown in Figure1. The parameters that need to be optimized in the agent-based crowd simulation are set as the input. The objective is to minimize the sum of the differences in all grids as defined in (2). The population is randomly generated within the search region, then mutation, crossover and selection operators are repeated until the stopping criterion is met. Lastly, the best parameter setting of the agent-based crowd model is generated as the output.

The adaptive DE with optional external archive (JADE), which was proposed by Zhang and Sanderson (2009), is one of the most effective DE variants by far. In this paper, the JADE algorithm is set as the evolutionary algorithm search module. The JADE algorithm differs from the classical DE algorithm in its mutation operator and the way to generate the control parameters. The DE/current-to-$p$best/1 operator is used in JADE:

$$v_{i,g} = x_{i,g} + F_i * (x^p_{best,g} - x_{i,g} + x_{r1,g} - x_{r2,g}). \tag{3}$$

where $v_{i,g}$ is the mutation vector. $x^p_{best,g}$ is a randomly chosen individual with top p percent best fitness value. $x_{i,g}$ is the target vector. $r1$ and $r2$ are two randomly generated indices within the population size that are mutually different from i. g represents the current generation. The p value is fixed at 0.1 in the proposed framework and is set as 0.05 in the original JADE algorithm. F_i indicates the scale factor for the current individual i.

The control parameters F (scale factor) and CR (crossover rate) are generated from certain distributions. The mean value of each distribution is generated by weighted linear combination of the mean value of the

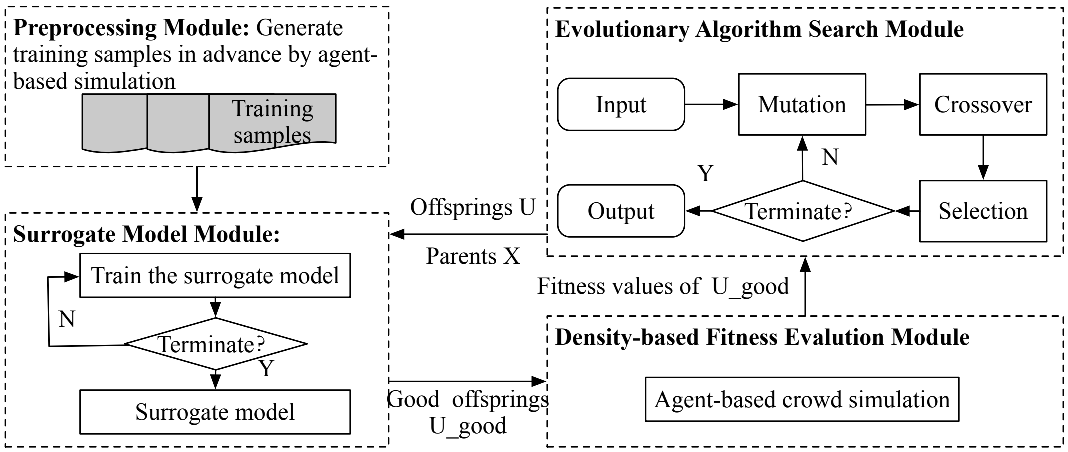


Figure 2: The surrogate assisted calibration framework with the offline training.

successful control parameters from the previous generation and the mean value of each control parameter in the last generation. The detailed description of control parameters F and CR can be found in Zhang and Sanderson (2009). The initial mean values of the two control parameters (i.e., μ_F and μ_{CR}) are both set as 0.5. The weight in calculating the mean value of each distribution (i.e., c) is set as 0.1. The population size (i.e., NP) is set as 30.

3 SURROGATE ASSISTED CALIBRATION FRAMEWORK

The objective of the proposed surrogate assisted calibration framework is to obtain better results with limited simulation runs. In this section, surrogate-assisted calibration frameworks with an offline training or online training are presented. Then, the surrogate models used in the case study will be briefly introduced.

3.1 Surrogate Assisted Calibration Framework with The Offline Training

The surrogate assisted calibration framework with the offline training is presented in Figure 2. Firstly, the training set needs to be generated in advance by agent-based crowd simulation in the preprocessing module. The Latin Hypercube sampling method is used to uniformly generate the training set. The size of the training set is set as 500. Once the surrogate model is built, the trained surrogate model is used first to evaluate the offsprings generated after the crossover operation in the evolutionary algorithm search module. The input of the surrogate model is the parameters that need to be optimized and the output is the predicted density distance. Only the predicted good offsprings are passed to the density-based fitness evaluation module and the agent-based crowd simulation is used to evaluate the actual fitness values. In other words, if the predicted fitness value of an offspring is larger (worse) than target vector (i.e., parent), the offspring will be discarded. In this way, we only use the time-consuming agent-based crowd simulation to evaluate the predicted good offsprings. After that, the selection operator in the evolutionary algorithm search module is used to select better ones between the good offsprings and their corresponding parents. The above evolution process will continue until the stopping criterion is met.

In general, the surrogate model is served as a filter to select only the good offsprings for further fitness evaluation. But, the offline framework needs a preprocessing module that requires to generate the training samples in advance, which incurs additional cost.

3.2 Surrogate Assisted Calibration Framework with The Online Training

To further improve the framework, an online training based surrogate model assisted evolutionary framework is proposed (see Figure 3). Instead of generating the training set at the beginning of the evolutionary process,

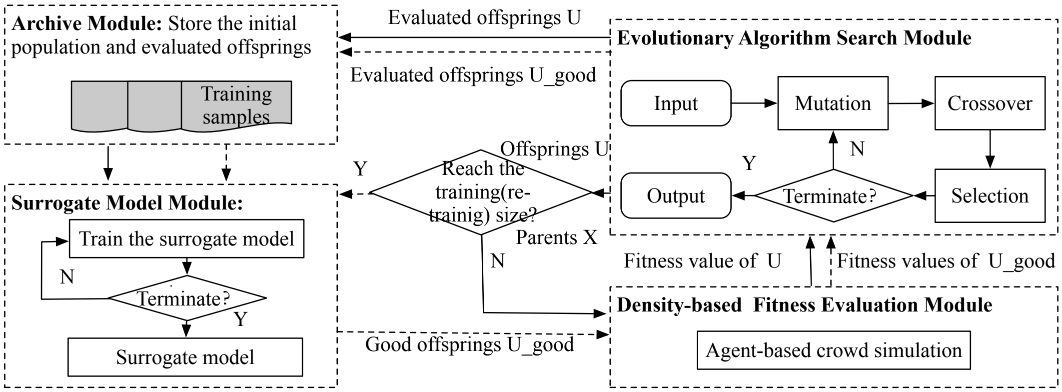


Figure 3: The surrogate assisted calibration framework with the online training.

the online training framework collects the evaluated offsprings into an archive during the evolutionary process in the evolutionary algorithm search module. If the size of the archive is smaller than a predefined training size α, the surrogate model is not trained and built. The agent-based crowd simulation is used directly to evaluate all the offsprings. The solid line in Figure 3 represents the data flow between the modules before a surrogate model is generated. Once the number of evaluated offsprings in the archive reaches α, the surrogate model will be trained and built. Then, it will be used to filter the offsprings as it does in the offline training framework. The re-training process will be activated whenever the number of evaluated offsprings in the archive is increased by α. The dashed line in Figure 3 represents the data flow between the modules after the surrogate model is generated.

3.3 Surrogate Models

The frequently used surrogate models like Kriging (Kleijnen 2009), ANN (Zhang, Patuwo, and Hu 1998), and SVM (Chang and Lin 2011) are chosen in our study. SVM is one of the regression models. It calculates the regression function as follows:

$$f(x) = \sum_{i=1}^{m}((\hat{\lambda}^i)^* - (\lambda^i)^*)k(x, x^p) + b^*. \tag{4}$$

where $k(.,.)$ is a kernel function, $(\hat{\lambda}^i)^*$, $(\lambda^i)^*$, $i = 1, ..., m$, are Lagrange multipliers, which are obtained by solving the optimization problem. b^* is obtained by exploiting the complementarity conditions of the problem. During the training process, $(\hat{\lambda}^i)^*$, $(\lambda^i)^*$, and b^* need to be learned. There are different types of kernel functions, like linear, radial basis function, Gaussian and sigmoid kernels. The radial basis function is set as the kernel in our experiments.

ANN has the input, hidden, and output layers. Weights and bias should be added to each input of a neuron and the activation function should be selected to decide whether the signal will be passed to the next layer or not. The weights and bias should be obtained during the model training process. The ANN with M hidden neurons and K output takes the form:

$$f_k(x) = f_a \sum_{j=1}^{M}(w_{kj}^2 f_a(\sum_{i=1}^{n}(w_{ij}^1 x_i + b_{j0}^1) + b_{k0}^2), k = 1, 2, ..., K. \tag{5}$$

where w is the weight of the input in each layer and b is the bias. f_a is the activation function. The tansig, linear, and sinsig are the most commonly used activation functions. The tansig activation function takes the form:

$$f_a(x) = \frac{2}{1 + exp^{(-2*x)}} - 1. \tag{6}$$

The Kriging method is a representative of GP approaches. It uses the interpolation method in which the interpolated values are modeled by a GP. It treats the interpolated values as a global model with local deviations:

$$f(x) = p(x)\beta + Z(x). \tag{7}$$

where $p(x)$ is a known function, which offers the global approximation of the design space. $Z(x)$ represents the realization of a stochastic process with zero mean and nonzero covariance. β is the regression parameter. The covariance matrix of $Z(x)$ is defined as follows:

$$Cov[Z(x^i), Z(x^j)] = \delta^2 \boldsymbol{R}[R(x^i, x^j)]. \tag{8}$$

where $\boldsymbol{R}$ is the correlation matrix, $R(x^i, x^j)$ is the correlation function between any two of the samples x^i and x^j, and δ is the standard deviation. The correlation function is given by the user and we take the Gaussian correlation function in our experiments, which is defined as follows:

$$R(x^i, x^j) = exp[-\sum_{k=1}^{n} \theta_k |x_k^i - x_k^j|^2]. \tag{9}$$

The key of the Kriging method is to obtain the correlation vector θ, which represents the correlation between different samples.

4 CASE STUDY

In this section, the performance of the proposed surrogate assisted calibration framework with either the offline or online training is investigated. Firstly, the social force model, used as the underlying crowd model, is introduced. Then, the street crossing scenario used in the case study is described. The use of various surrogate models with both the offline and online training are presented. Lastly, the effect of training size α used in the online training is investigated.

4.1 The Social Force Model

The social force model is first proposed by Helbing and Molnar (1995). The method uses virtual forces between agents, obstacles and objectives (i.e., goals) to guide agents' motion. Generally, the model can be described as follows:

$$f = m_i \frac{dv_i}{dt} = f_{i_o} + \sum_{j(j \neq i)} f_{ij} + \sum_{w} f_{iw}. \tag{10}$$

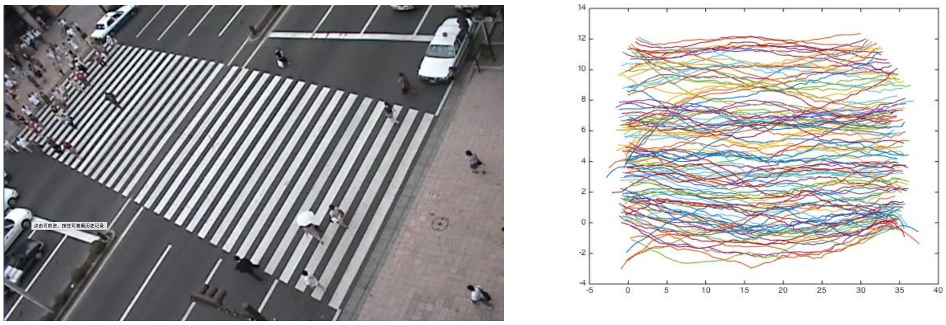
where m_i is the mass and $\frac{dv_i}{dt}$ is the acceleration of the agent i. f_{i_o} represents the attraction force from the objective and f_{ij} is the force between the neighborhood agents. f_{iw} is the repulsive force from the obstacles. The attraction force from the objective can be calculated as follows:

$$f_{i_o} = m_i * \frac{v_i^0(t) e_i^0(t) - v_i(t)}{\tau_i}. \tag{11}$$

where $v_i^0(t)$ is the desired speed and $e_i^0(t)$ is the vector that points to the objective. $v_i(t)$ represents the current velocity of the agent i. τ_i is the time-step that the agent adapts its velocity. The repulsive force between the neighborhood agents can be formulated as follows:

$$f_{ij} = [A * exp(\frac{r_{ij} - d_{ij}}{B}) + k * g(r_{ij} - d_{ij})] * \boldsymbol{n}_{ij} + \kappa * g(r_{ij} - d_{ij}) \Delta v_{ji} \boldsymbol{t}_{ij}. \tag{12}$$

$$\boldsymbol{n}_{ij} = (n_{ij}^1, n_{ij}^2) = (\boldsymbol{r}_i - \boldsymbol{r}_j)/d_{ij}, \quad \boldsymbol{t}_{ij} = (-n_{ij}^2, n_{ij}^1), \quad \Delta v_{ji} = (v_j - v_i) * \boldsymbol{t}_{ij}. \tag{13}$$



(a) Street crossing scenario (b) Extracted pedestrian trajectories

Figure 4: The studied street crossing scenario.

where d_{ij} is the distance between the agent i and agent j. $r_{ij} = r_i + r_j$ is the sum of their radius, and $\boldsymbol{n}_{ij}$ is the normalized vector point from the agent i to agent j. $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ represent the position of the agent i and agent j, respectively. $\boldsymbol{t}_{ij}$ is the tangential direction, and Δv_{ji} is the tangential velocity difference. In addition, the repulsive force between the obstacles and agent can be calculated as follows:

$$f_{iw} = [A * exp(\frac{r_i - d_{iw}}{B}) + k * g(r_i - d_{iw})] * \boldsymbol{n}_{iw} + \kappa * g(r_i - d_{iw})(\boldsymbol{v}_i * \boldsymbol{t}_{iw})\boldsymbol{t}_{iw}. \tag{14}$$

where d_{iw} is the distance between the agent and the obstacle. $\boldsymbol{n}_{iw}$ and $\boldsymbol{t}_{iw}$ are the direction perpendicular and tangential to it. A, B, k, and κ are parameters. The function $g(x)$ is defined as follows:

$$g(x) = \begin{cases} 0 & if\ x < 0. \\ x & else. \end{cases} \tag{15}$$

Hence, the social force model has five parameters: A, B, k, κ, and τ. The ranges of the five parameters that need to be optimized are listed in Table 1.

Table 1: Parameter setting of the social force model.

Parameter	Descriptions	Range
A	Reflects the force of the interaction	[1000,5000]
B	Determines the interaction range	[0.01,1]
k	Determines the obstruction effect of interactions	[10000,300000]
κ	Determines the obstruction effect of interactions	[10000,300000]
τ	Reflects the time to adapt the actual velocity	[0.1,1]

4.2 The Street Crossing Scenario

Figure 4(a) shows the street crossing scenario, where the pedestrians walk across the street. Our objective is to calibrate the parameters to reproduce the behaviors recorded in the video. The video is adopted from Teknomo (2006) with 156 pedestrians and 60 seconds in length. The zebra crossing region is 11 meters×31 meters and the positions of the pedestrians are recorded every 0.5 second. The tracks of their footprints are shown in Figure 4(b).

As a pedestrian's speed varies along the time, we use the average speed as the desired speed. The recorded maximum speed of each pedestrian is set as the maximum speed of the corresponding agent. The first and last position of a pedestrian is set as the origin and destination of the corresponding agent.

4.3 The Evaluation of Surrogate Models

In this section, the performances of the surrogate models are studied. 500 samples (parameter settings with their corresponding function values) from the street crossing scenario are used to train the surrogate models and another 100 samples are used to evaluate the performance of the surrogate models. The mean squared error (MSE) and correlation coefficient R^2 are used to evaluate the surrogate models. The smaller the MSE, the better the corresponding surrogate model. The bigger the R^2, the better the surrogate model. Kriging is coded using MATLAB DACE toolbox. ANN is coded using MATLAB Neural Network toolbox with 5-15-1 layout (tansig is used as activation function and Bayesian regularization back-propagation is used as the training algorithm). SVM is coded using MATLAB LIBSVM 3.22 toolbox (Chang and Lin 2011) (SVM type used is epsilon-SVR and radial basis function is used as the kernel). The comparison results of the surrogate models are shown in Table 2.

Table 2: Comparison between three surrogate models.

Surrogate model	MSE	R^2
Kriging	1.7833E+05	0.6466
ANN	1.2138E+05	0.7598
SVM	1.2951E+05	0.7508

From Table 2, we can see that the ANN model is the best for the testing set on this scenario. The Kriging obtains the worst results mainly due to the over-fitting problem on the training set. The SVM shows slightly inferior results than the ANN. The comparison results suggest that the ANN is the best candidate surrogate model to be integrated with the EA-based calibration framework.

4.4 Simulation Results

Table 3 shows the average and standard deviation errors over twenty-five independent runs of the various calibration approaches. All the approaches are terminated after 1000 fitness evaluations, including the fitness evaluations in the training process of the surrogate model. The offline suffix is used to indicate the surrogate model assisted calibration approaches with the offline training. The surrogate model assisted calibration approaches with the online training is marked with the online suffix. The JADE without suffix represents the approach without the surrogate model. The training size is set as 500 fitness evaluations for the offline training and 300 fitness evaluations (i.e., α) for the online training. One tailed two sample t-test is conducted to identify the significant difference between the JADE-ANN-online and other compared approach. The significance level is set as 0.05. In Table 3, ”+”, ”-” means the JADE-ANN-online approach is significantly better, or worse than the compared approach, respectively. ”=” means there is no significant difference between the two approaches.

Table 3: Simulation results.

Approach	Mean±Std	$p-value$
JADE	1.3956E+03±3.6132E+01[+]	< 0.0001
JADE-Kriging-offline	1.4251E+03±2.7050E+01[+]	< 0.0001
JADE-ANN-offline	1.3900E+03±3.5614E+01[+]	< 0.0001
JADE-SVM-offline	1.4334E+03±3.2612E+01[+]	< 0.0001
JADE-Kriging-online	1.3759E+03±3.3728E+01[+]	0.0025
JADE-ANN-online	1.3480E+03±2.7834E+01	—
JADE-SVM-online	1.3509E+03±2.7834E+01[=]	0.6414

Comparing with the offline training based approaches, the online training based surrogate assisted approaches enable the surrogate model to do adjustment during the evolutionary process. It also can be seen from Table 3 that the online training based surrogate assisted approaches show improvement in mean value, comparing with the JADE and offline training based surrogate assisted approaches. As illustrated in Table 2, the ANN model obtains the best performance amongst various surrogate models in terms of accuracy. Thus, it is not surprising that the JADE-ANN-online obtains the best performance and fastest convergence speed in tested scenario. Among the offline training based approaches, the JADE-SVM-offline and the JADE-Kriging-offline show inferior performances than the JADE. The reason why the JADE-SVM-offline and JADE-Kriging-offline obtain worse performance than the JADE-ANN-offline may be due to less accuracy in the surrogate model as illustrated in Table 2. However, the JADE-SVM-online and JADE-Kriging-online still obtain better results than the JADE algorithm. This shows that the SVM and Kriging surrogate models are still effective when use properly. To further illustrate the effectiveness of the proposed framework, we record the number of the fitness evaluations of various approaches when they reach fitness value of 1400 in Table 4. The maximum number of fitness evaluations is set to 1000 (that is, the evaluation run will terminate if the fitness value does not reach 1400 after 1000 fitness evaluations).

Table 4: Fitness evaluations needed when fitness value reaches 1400.

Approach	Mean fitness evaluations
JADE	920
JADE-Kriging-offline	1000
JADE-ANN-offline	668
JADE-SVM-offline	974
JADE-Kriging-online	920
JADE-ANN-online	538
JADE-SVM-online	728

It can be seen from Table 4 that the JADE-ANN-online approach requires the minimum number of fitness evaluations to reach 1400 fitness value, which indicates that it has the fastest average convergence speed among all the approaches. Comparing to the JADE, it can be seen that the JADE-ANN-offline, JADE-ANN-online, and JADE-SVM-online approaches can reduce the fitness evaluations effectively. Comparing the offline training based approaches with the online ones, the online training based approaches require less number of fitness evaluations. This indicates the proposed online training is effective. Comparing the JADE-ANN-online with JADE-SVM-online, it can be seen that the JADE-ANN-online needs less number of fitness evaluations than the JADE-SVM-online.

To further investigate the convergence speed of the online training based approaches, the convergence curve of the JADE and online training based approaches at the median value of the twenty-five independent runs are presented in Figure 5. From Figure 5, it can be seen that the surrogate-assisted approaches with the online training can achieve better results than the approach without the surrogate model. Among the online training based approaches, the JADE-ANN-online achieves better competitive performance and relatively faster convergence speed comparing to the JADE-SVM-online and JADE-Kriging-online in spite of the difference in initialization.

Simulation results presented in this section further confirmed that the ANN is the most effective surrogate model when compared to the SVM and Kriging for the test scenario.

4.5 Training Size Study for The Online Training Framework

For the online training based surrogate-assisted approaches, the training size (α) needs to be fine tuned. In this experiment, the training size is set as 100, 300, and 500, respectively. An evolution run is terminated after 1000 fitness evaluations. So, the re-training process will be activated 9, 3, and 1 times, respectively. The ANN surrogate model is selected as the representative model to test different training size. Each setting is conducted ten runs and the simulation results are listed in Table 5. One tailed two sample t-test

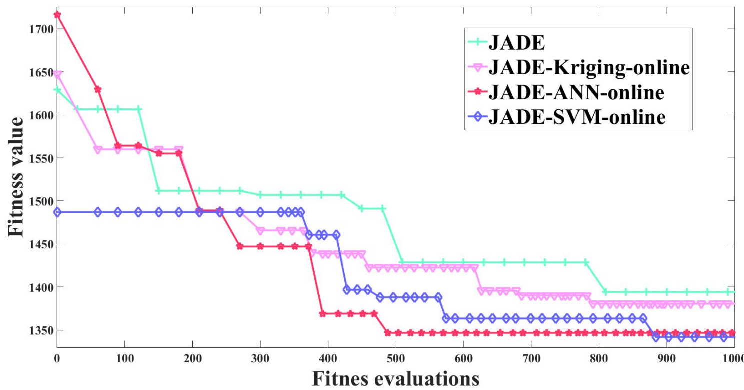


Figure 5: Convergence curve.

is conducted to identify the significant difference between the best online training size and the other two settings. The significance level is set as 0.05. "+" means the best online training size is significantly better than the compared one.

Table 5: Simulation results for the online training based JADE-ANN approaches with different training sizes.

Approach	Mean±std	p-value
JADE-ANN-online-100	1.3832E+03±2.7850E+01[+]	0.0147
JADE-ANN-online-300	1.3496E+03±2.7991E+01	—-
JADE-ANN-online-500	1.3855E+03±3.9902E+01[+]	0.0317

We can see from Table 5 that the training size with 100 fitness evaluations and 500 fitness evaluations achieve similar worse results than the case with training size equals to 300. The reason is that the small training size may lead to an initial surrogate model with less accuracy. The population may move to the wrong direction, resulting in worse performance. For the large training size, the result is mainly due to the converging effect of the EA, which may lead the population to converge to a small region. Although the surrogate model is built and used after 500 fitness evaluations, the population may not jump out of the region and hence result reduction in performance. The training size with 300 fitness evaluations obtains the best performance among the three settings. In this case, the training process of the surrogate model is activated three times when the number of fitness evaluations reaches 300, 600, and 900.

5 CONCLUSIONS AND FUTURE WORKS

In this paper, the surrogate model assisted calibration framework with the offline or online training is proposed for crowd model calibration. Three types of surrogate models are selected for evaluation. With the surrogate model, the agent-based crowd simulation is only used to evaluate the promising offsprings rather than the whole population. Simulation results show that the proposed surrogate assisted calibration framework with the online training is more effective for the tested scenario. Among various approaches, the JADE-ANN-online approach achieves the best performance comparing to other approaches and JADE-SVM-online approach is also competitive but requires more fitness evaluations to reach a desired fitness value. For future research, the use of different fidelity models can be investigated. Low fidelity model can be built to reduce the prediction time while the high fidelity model offers more accurate prediction. Balance can be made between the prediction time and accuracy of the model.

ACKNOWLEDGMENTS

Wenchao Yi, Singkuang Tan, Wentong Cai, and Nan Hu would like to acknowledge the support from the following two grants: i) IHPC-NTU Joint R&D Project on "Symbiotic Simulation and Video Analysis of Crowds"; and ii) AcRF Tier 1 Project on "A Cooperative Co-evolutionary Framework for Data-driven Model Calibration". Jinghui Zhong is supported by the National Natural Science Foundation of China (Grant No. 61602181) and Fundamental Research Funds for the Central Universities (Grant No. 2017ZD053).

REFERENCES

Chang, C.-C., and C.-J. Lin. 2011. "LIBSVM: a Library for Support Vector Machines". *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chiu, C.-C., S. Zhang, J. T. Lin, L. Zhen, and E. Huang. 2016. "Improving the Efficiency of Evolutionary Algorithms for Large-scale Optimization with Multi-fidelity Models". In *Proceedings of the 2016 Winter Simulation Conference*, edited by R. S. T. M. K. Roeder, P. I. Frazier and S. E. C. E. Zhou, T. Huschka, 815–826. Piscataway, New Jersey, USA: Institute of Electrical and Electronics Engineers, Inc.

Duives, D. C., W. Daamen, and S. P. Hoogendoorn. 2013. "State-of-the-art Crowd Motion Simulation Models". *Transportation Research Part C: Emerging Technologies* 37:193–209.

Erickson, C., B. E. Ankenman, and S. M. Sanchez. 2016. "Comparison of Gaussian Process Modeling Software". In *Proceedings of the 2016 Winter Simulation Conference*, edited by R. S. T. M. K. Roeder, P. I. Frazier and S. E. C. E. Zhou, T. Huschka, 3692–3693. Piscataway, New Jersey, USA: Institute of Electrical and Electronics Engineers, Inc.

Fan, Q., and J. Hu. 2016. "Simulation Optimization via Promising Region Search and Surrogate Model Approximation". In *Proceedings of the 2016 Winter Simulation Conference*, edited by R. S. T. M. K. Roeder, P. I. Frazier and S. E. C. E. Zhou, T. Huschka, 649–658. Piscataway, New Jersey, USA: Institute of Electrical and Electronics Engineers, Inc.

Helbing, D., and P. Molnar. 1995. "Social Force Model for Pedestrian Dynamics". *Physical Review E* 51 (5): 4282.

Huang, D., T. T. Allen, W. I. Notz, and N. Zeng. 2006. "Global Optimization of Stochastic Black-box Systems via Sequential Kriging Meta-models". *Journal of Global Optimization* 34 (3): 441–466.

Johansson, A., D. Helbing, and P. K. Shukla. 2007. "Specification of the Social Force Pedestrian Model by Evolutionary Adjustment to Video Tracking Data". *Advances in Complex Systems* 10:271–288.

Kleijnen, J. P. 2009. "Kriging Metamodeling in Simulation: a Review". *European Journal of Operational Research* 192 (3): 707–716.

Meng, Q., and S. H. Ng. 2016. "Combined Global and Local Method for Stochastic Simulation Optimization with an AGLGP Model". In *Proceedings of the 2016 Winter Simulation Conference*, edited by R. S. T. M. K. Roeder, P. I. Frazier and S. E. C. E. Zhou, T. Huschka, 827–838. Piscataway, New Jersey, USA: Institute of Electrical and Electronics Engineers, Inc.

Pedrielli, G., and S. H. Ng. 2016. "G-STAR: a New Kriging-based Trust Region Method for Global Optimization". In *Proceedings of the 2016 Winter Simulation Conference*, edited by R. S. T. M. K. Roeder, P. I. Frazier and S. E. C. E. Zhou, T. Huschka, 803–814. Piscataway, New Jersey, USA: Institute of Electrical and Electronics Engineers, Inc.

Tajima, Y., and T. Nagatani. 2001. "Scaling Behavior of Crowd Flow Outside a Hall". *Physica A: Statistical Mechanics and its Applications* 292 (1): 545–554.

Teknomo, K. 2006. "Application of Microscopic Pedestrian Simulation Model". *Transportation Research Part F: Traffic Psychology and Behaviour* 9 (1): 15–27.

Wolinski, D., S. J Guy, A.-H. Olivier, M. Lin, D. Manocha, and J. Pettré. 2014. "Parameter Estimation and Comparative Evaluation of Crowd Simulations". *Computer Graphics Forum*:303–312.

Yan, F., Z. Lin, X. Wang, F. Azarmi, and K. Sobolev. 2017. "Evaluation and Prediction of Bond Strength of GFRP-bar Reinforced Concrete Using Artificial Neural Network Optimized with Genetic Algorithm". *Composite Structures* 161:441–452.

Zhang, G., B. E. Patuwo, and M. Y. Hu. 1998. "Forecasting with Artificial Neural Networks: the State of the Art". *International Journal of Forecasting* 14 (1): 35–62.

Zhang, J., and A. C. Sanderson. 2009. "JADE: Adaptive Differential Evolution with Optional External Archive". *IEEE Transactions on Evolutionary Computation* 13 (5): 945–958.

Zhong, J., and W. Cai. 2015. "Differential Evolution with Sensitivity Analysis and the Powell's Method for Crowd Model Calibration". *Journal of Computational Science* 9:26–32.

Zhong, J., W. Cai, L. Luo, and H. Yin. 2015. "Learning Behavior Patterns from Video: a Data-driven Framework for Agent-based Crowd Modeling". In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, edited by P. Y. Gerhard Weiss, 801–809. Richland, South Carolina, USA: International Foundation for Autonomous Agents and Multiagent Systems.

Zhong, J., N. Hu, W. Cai, M. Lees, and L. Luo. 2015. "Density-based Evolutionary Framework for Crowd Model Calibration". *Journal of Computational Science* 6:11–22.

AUTHOR BIOGRAPHIES

WENCHAO YI is a Research Fellow in the School of Computer and Science Engineering, Nanyang Technological University (NTU), Singapore. She received her Ph.D. degree from the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, China, in 2016. Her research interests include: Computational Intelligence, Machine Learning, and Agent-based Modeling and Simulation. Her email address is wenchaoyi@ntu.edu.sg.

JINGHUI ZHONG is an Associate Professor in the School of Computer and Science Engineering, South China University of Technology, Guangzhou, China. He received his Ph.D. degree from the School of Information Science andTechnology, Sun YAT-SEN University, China, in 2012. His research interests include: Computational Intelligence, Machine Learning, and Agent-based Modeling and Simulation. His email address is jinghuizhong@gmail.com.

SINGKUANG TAN is a Research Assistant in the School of Computer and Science Engineering, Nanyang Technological University (NTU), Singapore. He received his B. Eng degree, from the School of Computer and Science Engineering, Nanyang Technological University (NTU), Singapore, in 2007. His research interests include: Computational Intelligence, Machine Learning, and Agent-based Modeling and Simulation. His email address is singkuang@ntu.edu.sg.

WENTONG CAI is a Professor in the School of Computer and Science Engineering, Nanyang Technological University (NTU), Singapore. He received his Ph.D. in Computer Science from University of Exeter (UK) in 1991. His expertise is mainly in the area of Modeling and Simulation (particularly, modeling and simulation of large-scale complex systems, and system support for distributed simulation and virtual environments). He is an associate editor of the ACM Transactions on Modeling and Computer Simulation (TOMACS) and an editor of the Future Generation Computer Systems (FGCS) and The Journal of Simulation (JOS). His email address is aswtcai@ntu.edu.sg.

NAN HU is a Research Scientist in the Institution of High Performance Computing, Agency for Science Technology and Research, Singapore. He received his B. Eng degree, and Ph.D. degree from the School of Computer and Science Engineering, Nanyang Technological University (NTU), Singapore, in 2007 and 2014, respectively. His research interests include: Computational Intelligence, Machine Learning, and Agent-based Modeling and Simulation. His email address is hun@ihpc.a-star.edu.sg.