

Using Genetic Algorithms to Optimize Social Robot Behavior for Improved Pedestrian Flow

Bryce D. Eldridge

Electrical and Computer Engineering Department
Colorado State University
Fort Collins, CO, U.S.A.
bryce@engr.colostate.edu

Anthony A. Maciejewski

Electrical and Computer Engineering Department
Colorado State University
Fort Collins, CO, U.S.A.
aam@engr.colostate.edu

***Abstract** – This paper expands on previous research on the effect of introducing social robots into crowded situations in order to improve pedestrian flow. In this case, a genetic algorithm is applied to find the optimal parameters for the interaction model between the robots and the people. Preliminary results indicate that adding social robots to a crowded situation can result in significant improvement in pedestrian flow. Using the optimized values of the model parameters as a guide, these robots can be designed to be more effective at improving the pedestrian flow. While this work only applies to one situation, the technique presented can be applied to a wide variety of scenarios.*

Keywords: crowd dynamics, genetic algorithms, social robots

1 Introduction

Crowd dynamics has long been a subject of interest in a variety of fields, including architecture, transportation, emergency escape design, and event planning. Improving crowd dynamics has the potential to save lives in situations where the behavior of the crowd itself becomes a threat to individuals. The goal of this research is to provide insight into the potential of employing robotic agents to reduce bottlenecks and improve pedestrian flow.

Attempts to introduce robots into crowds in different situations have been made [1], and while it has been shown that robots can improve crowd flow in some situations, many questions remain to be answered concerning the form of the robots themselves. This form includes both visual and behavioral aspects, which include size, shape, vocal cues, and movement [2,3]. The exact nature of the robots' interaction with the crowd is also variable, and can be modified by changing these attributes. Significant research has been performed on human-robot interaction and the effect that these different characteristics have on human response in the field of social robotics [4,5].

Because of the difficulty associated with empirically testing pedestrian flow in emergency situations, simulation provides a useful and important tool. However, one must have an accurate mathematical model of these social interactions. In order to accomplish this, we used the social

force model, which was first introduced by Helbing and Molnar, [4].

We first describe the underlying model in detail. We then describe an example problem situation and discuss a baseline simulation. This is followed by a detailed description of the genetic algorithm technique applied to the example problem scenario. Finally, the results are analyzed and conclusions are presented.

2 Social Force Model

The social force model for modeling pedestrian flow was first introduced by Helbing and Molnar [6] and expanded to include robots in [1]. This model is basically an application of particle dynamics to the simulation of pedestrian crowds. Each person or robot is treated as a circular particle with a particular mass and radius. The interactions between objects in the simulation are modeled as forces. In each iteration, the forces on each particle are summed, and then Newton's equation is solved to determine the acceleration, which is then used to determine the velocity and position of the particle.

Each object in the simulation can interact with all of the other objects. For example, there will be a force acting on each person due to the walls, and a separate force acting on each person due to every other person. The person-person force, for example, models the tendency for people to keep a minimum amount of personal space. While the person-person and person-wall interactions are fixed by human nature, the parameters that model the interactions between people and robots can be controlled to some extent by changing the form and behavior of the robot. One of the goals of this paper is to determine, through simulation, optimal values for these parameters. This will give an indication as to the form and behavior of the robot required to achieve the desired interaction. The implementation of the robots is left open for future research.

The total force on each particle is given by

$$f_i = f_i^S + \sum_{\substack{j=1 \\ i \neq j}}^N f_{ij}^I + \sum_{k=1}^M f_{ik}^W + \sum_{h=1}^R f_{ih}^C \quad (1)$$

where f_i^S is the self-driven force for the i th particle, f_{ij}^I is the repulsive interaction force on particle i due to particle j , f_{ik}^W is the repulsive force on particle i due to wall k , and f_{ih}^C is the cohesive force on particle i due to robot h . In this model there are N particles, M walls, and R robots. The self-driven force is simply a model of the particle's desire to achieve a specified velocity. This force is modeled by

$$f_i^S = \frac{m_i (s_i \hat{e}_i - v_i)}{\tau_i} \quad (2)$$

where m_i is the mass of the i^{th} individual, s_i is the desired speed, \hat{e}_i is the desired direction, v_i is the current velocity, and τ_i is a parameter that determines how fast the particle responds. The desired direction is set by an error term between the current position of the particle and the desired end location.

These simulations utilize two basic types of interaction forces. The *repulsive force* is modeled as follows:

$$f_R = [A \exp(-d_{ij}/B) + k g(d_{ij})] \hat{n}_{ij} - [\kappa g(d_{ij}) \Delta v'_{ij}] \hat{t}_{ij} \quad (3)$$

$$d_{ij} = \|x_i - x_j\| - r_i - r_j \quad (4)$$

where A is the magnitude, d_{ij} is the distance between object i and object j , r_i is the radius of the i^{th} particle, and B is a parameter that affects the rate of decay of the force. In high-density situations, physical contact can occur, and k and κ are used to model a compression term and a tangential friction term. Whether or not the particles are in contact is determined by the function $g(x)$, which is zero if x is positive and one otherwise. The terms n_{ij} and t_{ij} are the normal and tangential components of the vector between the two particles. The other important force is the *cohesive force*, which is modeled as:

$$f_C = C \exp\left(\frac{-(d_{ij} - D)^2}{E}\right) \quad (5)$$

where C is the peak magnitude, D controls how far away from the center of the object this peak occurs, and E affects the rate of decay of the force.

Using the nominal parameters given in [7], Figure 1 shows a graph of the repulsive force and the cohesive force versus distance. This provides a reasonably accurate model of the behavior of pedestrian crowds in the real world. Using this model, we then chose a specific pedestrian crowd situation to simulate.

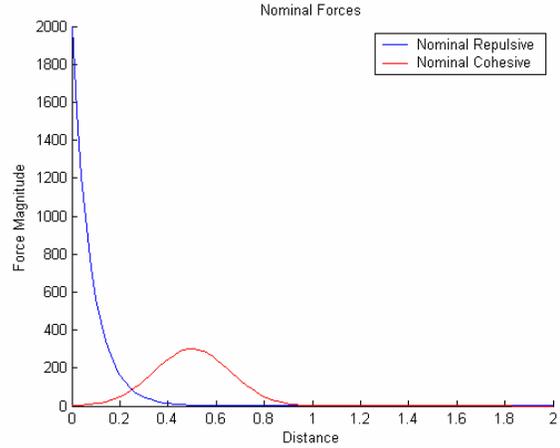


Figure 1 – This plot shows the nominal repulsive force that is exhibited between all objects, i.e., people, robots and walls. The nominal cohesive force is between the robots and the people in the hallway.

3 Problem Statement

3.1 Problem Situation

For our example problem we consider a very simplified exit scenario where individuals can take one of two paths to exit a hallway. The geometry for this situation is shown in Figure 2, which consists of a straight, six meter wide hallway with two different sized openings at the end. The end point of the split wall in the center of the hallway is variable in order to control the size of each exit. When one of the exits becomes small, the efficiency of the crowd leaving the hallway drops significantly because of blockages in the narrow part of the hallway.

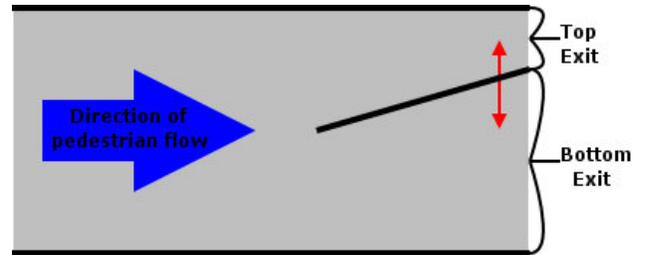


Figure 2 – This figure shows the geometry of the example situation. Pedestrians flow from left to right, and the red arrow indicates the variability of the exit sizes.

These blockages result in a significant portion of the pedestrian crowd being unable to exit the hallway in a reasonable time, as shown in Figure 3. If this were a real situation, such as a fire in a crowded building, it could mean that those people's lives would be in danger. However, introducing social robots into this situation could reduce or eliminate the blockage, and reduce the overall exit time of the people in the hallway.

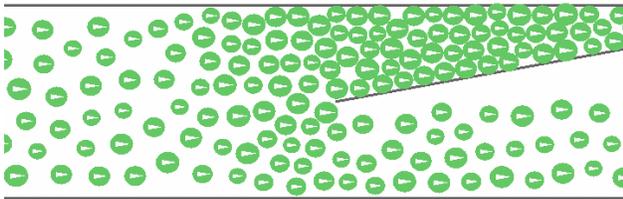


Figure 3 – This figure shows a simulation of the hallway with no robots. The white arrows inside each green circle indicate the desired direction of travel. The blockage in the upper part of the hallway is apparent in this figure.

3.2 Measure of Flow Efficiency

To provide a numerical measure of the efficiency of pedestrian flow, we calculate how close the individuals in a given area are to achieving their desired velocity. This number is then averaged over all particles within a six-meter wide window centered on the starting point of the split wall. The average of all the particles is then averaged over ten simulation runs for a particular set of parameters, with each run having a different set of random initial starting locations. In particular, the efficiency for a group of particles is calculated with the following equation

$$Efficiency = \frac{\left(\sum_{i=1}^N \hat{e}_i \cdot v_i / N \right)}{s} \quad (6)$$

where N is the number of pedestrians in the window, \hat{e}_i is the desired direction of travel, v_i is the actual velocity, and s is the desired speed. This measure of efficiency generally ranges from zero to one, although it is possible for particles to achieve an efficiency greater than one if their actual velocity exceeds their desired velocity. The first ten seconds of the simulation were discarded to eliminate the transient effects of the crowd first entering the split wall area. Other measures such as the particle exit rate were also computed, however, they yielded results comparable to the average efficiency.

To measure the improvement of adding social robots, a baseline case was simulated, genetic algorithms were implemented to optimize the interactions, and then the baseline situation was re-simulated with the new parameters.

4 Baseline Simulation

Figure 4 shows a graph of the efficiency versus the ratio of the two openings. When the ratio is small, the top exit is significantly smaller than the bottom exit and a bottleneck forms, which results in the blockage shown in Figure 3. The steep increase in efficiency that occurs at a ratio of 0.15 is due to the fact that the small top opening becomes wide enough to allow two people to pass through

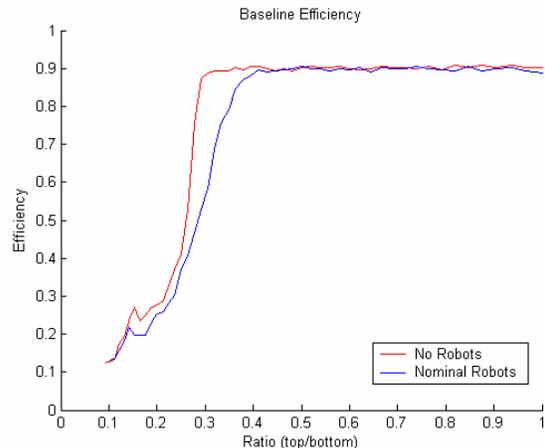


Figure 4 – This plot shows the efficiency as the end point of the split wall is moved from the top to the center position. A low ratio indicates a small exit size, and the plot clearly shows how the efficiency drops off as the exit narrows.

side by side instead of allowing only one person to exit. Similarly, a steep increase in efficiency occurs when the top opening becomes wide enough for three people to pass through side by side at a ratio of 0.25. Increasing the ratio beyond 0.3 has no appreciable effect on the efficiency. This may be due to the fact that typically no more than six people pass through the unobstructed hallway side by side. As can be seen from the graph, merely introducing robots with nominal cohesive forces into this situation actually decreases the efficiency slightly. This is because the robots become another object that must exit the hallway and, since they are larger than the people, they further impede the pedestrian flow.

When the split wall is horizontal and the two openings are the same size, a maximum efficiency of around 0.9 is achieved. An efficiency of 1.0 is not achieved because of the effects of people bumping into the end of the split wall.

5 Genetic Algorithm

Clearly the results of the previous section show that robot behavior parameters that were optimal for other pedestrian flow scenarios are not optimal for the case studied here. Unfortunately, determining optimal parameters is difficult because the search space is very large and difficult to describe mathematically in its entirety. However, genetic algorithms (GAs) offer a promising way to find near-optimal solutions. In order to experiment with different genetic algorithm techniques, the GALib software package was integrated into the crowd simulation software [8]. GALib allowed us to quickly design and implement a GA for this particular situation. We designed two similar approaches, designated GA1 and GA2 in this paper.

A GA functions by defining a *genome* and a *fitness measure*. In this case the genome is the set of parameters to simulate, and the fitness measure indicates how successful

that particular genome was at solving the problem. The GA begins by generating and evaluating an initial population of random genomes. Next, a new population is created from the old one by several methods. In these simulations both a *crossover* operation, which creates a new individual by combining two others, and a *mutation* operation, which creates a new individual by randomly changing one element of a previous genome, were used. The new population then replaces the original population, and the GA starts over again at the evaluation stage. The evolution continues until a specified convergence condition is reached, which in this case was 50 generations with no significant improvement.

5.1 Genome structure

The first GA (GA1) randomly sent robots to either the top exit or the bottom exit, using a total of 10 robots. The top robots had one common set of interaction parameters, while the bottom robots had a separate common set of interaction parameters. The second GA (GA2) only used four robots, and allowed them to stop at a location inside the split section of the hallway. Each robot in this case had its own set of interaction parameters. The desired direction of travel, \hat{e}_i for robot i , is computed as

$$\hat{e}_i = \frac{l_i - x_i}{\|l_i - x_i\|} \quad (7)$$

where l_i is the destination location and x_i is the current position of the robot. The interaction parameters were structured into a genome using the genes shown in Table 1.

Table 1 – Gene Descriptions

Gene	Description
PR_A	person-robot force 'A' parameter (Eq. 3)
RP_A	robot-person force 'A' parameter (Eq. 3)
RR_A	robot-robot force 'A' parameter (Eq. 3)
RW_A	robot-wall force 'A' parameter (Eq. 3)
PR_B	person-robot force 'B' parameter (Eq. 3)
RP_B	robot-person force 'B' parameter (Eq. 3)
RR_B	robot-robot force 'B' parameter (Eq. 3)
RW_B	robot-wall force 'B' parameter (Eq. 3)
C	cohesive force 'C' parameter (Eq. 5)
D	cohesive force 'D' parameter (Eq. 5)
E	cohesive force 'E' parameter (Eq. 5)
T/B	top/bottom exit for robots in GA1
l	destination location for robots in GA2 (Eq. 7)

5.2 Fitness measure

The fitness measure chosen in this research was the average efficiency defined in section 3.2. Genomes with a higher efficiency were used to create the next population, whereas genomes with a lower efficiency were discarded in each generation.

Table 2 – Common GA Parameters

Parameter	Value
GA Population Size	50
Mutation Probability	0.05
Crossover Probability	0.9
Simulation Length (seconds)	60
Number of Runs	10

5.3 GA Parameters

Some common parameters, shown in Table 2, were constant for all genetic algorithm runs. Due to the variability of the efficiency depending on the starting locations of the objects, each case was simulated in runs of 60 seconds, with the average of all runs resulting in the final score for that case. All genomes in a particular generation used the same set of starting locations.

The crossover operation was a standard crossover in which two parents resulted in two children using a random crossover point. The mutation operator picked a parameter at random from the genome and replaced it with a new value chosen at random from a specified range. Elitism, or always retaining the best individual ever found, was used in all simulations.

6 Optimized Results

Figure 6 shows the performance of the genetic algorithms. The first GA achieved a maximum score of 0.5 and improved pedestrian flow overall, but failed to eliminate the blockage in the hallway. The second GA achieved a maximum score of approximately 0.88 and was successful at relieving this blockage, significantly increasing the efficiency.

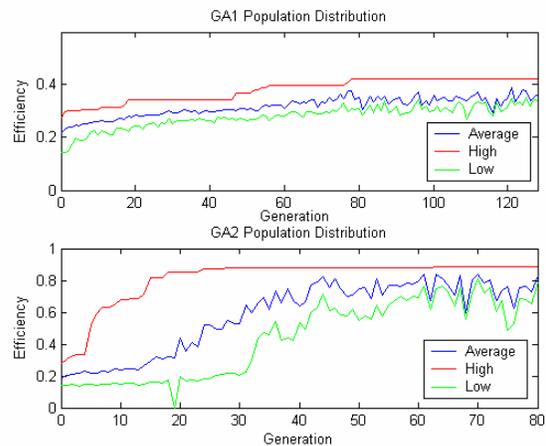


Figure 6 – This plot shows the efficiency distribution of the population for each generation of each GA.

Figure 7(a) shows a snapshot of the hallway for the best genome of the first GA. This solution sent all of the robots towards the bottom opening with a cohesive force to pull the crowd through.

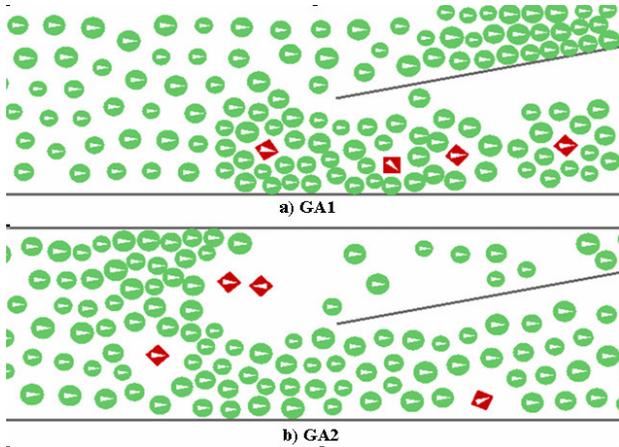


Figure 7 – This figure shows the optimized simulations for GA1 and GA2.

Figure 8 shows the graphs of the forces with respect to distance from the best genome found by the first GA. Since the solution only sent robots to the bottom opening, we are not concerned with the forces associated with the top robots. The bottom robots had high robot-person forces, which helped push them forward since the majority of the crowd was behind the robots. The bottom robots also had low person-robot forces, which is to be expected since they attempted to collect people around them. They also had a significant robot-wall force, which helped them stay in the center of the hallway. Since they are attempting to use their cohesive forces to pull groups of people through the bottom exit, this would allow the maximum amount of space for the group to pass through.

The cohesive forces for the first GA are consistent with the general strategy. The bottom robots have a large cohesive force compared to the nominal values, which helps them to gather people together.

Figure 7(b) shows a snapshot of the optimized solution for the second GA. In this case, the genetic algorithm found that the best solution was to place stationary robots with large repulsive forces at the entrance to the top exit. This allows only a small number of people into the top exit, which eliminates the blockage seen in the baseline by forcing the people into a single file line. The other two robots are given a cohesive force and directed through the bottom exit.

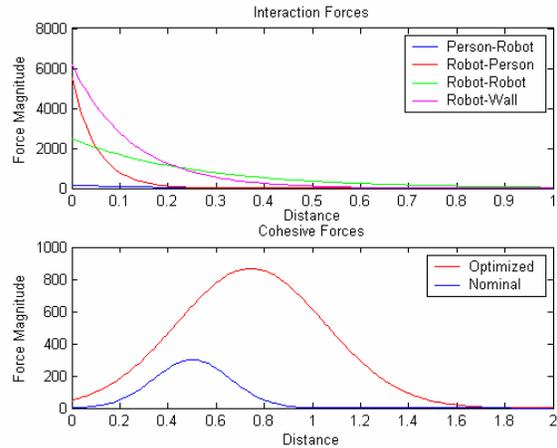


Figure 8 – These plots show the magnitude of the repulsive and cohesive forces versus distance in meters for GA1.

Figures 9 and 10 show graphs of the repulsive and cohesive forces with respect to distance for the best genome of the second GA. The two stationary robots have higher person-robot and robot-person repulsive forces than the moving robots, indicating that they attempt to keep the pedestrians away from them. This creates the empty bubble in Figure 7(b), which prevents most of the people from entering the top hallway. For the stationary robots, the robot-robot and robot-wall forces are considerably different between the two individual robots, which indicate that these forces are not important for the solution. This is because the force that keeps the robots stationary overrides any force that would cause them to repel each other or move away from the walls. The average cohesive force for the stationary robots is much lower than the nominal case, indicating that these robots do not try to collect pedestrians around themselves, which is consistent with the general strategy in this case.

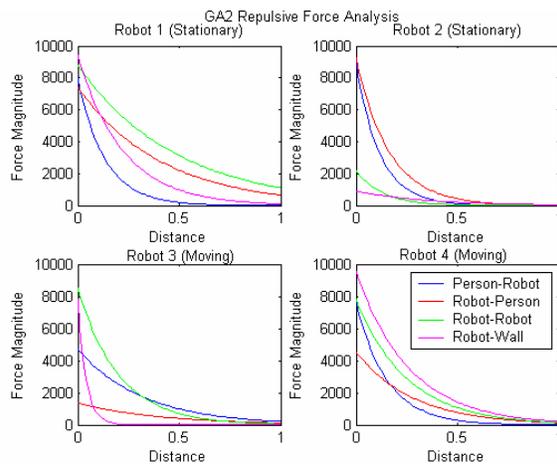


Figure 9 – These plots show the magnitude of the repulsive forces versus distance in meters for each of the robots in GA2.

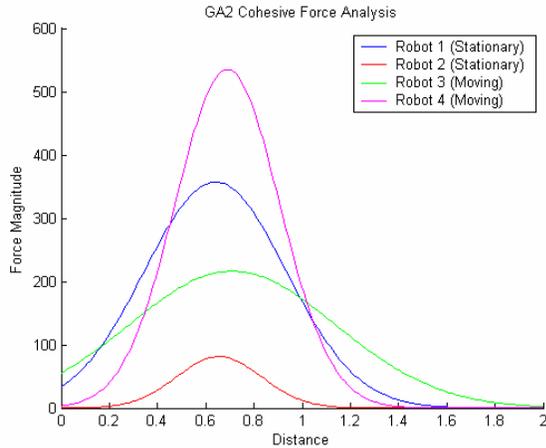


Figure 10 – This plot shows the magnitude of the cohesive forces versus distance for each of the robots in GA2.

The moving robots have high robot-robot and robot-wall forces, which have the effect of spreading them out and keeping them in the center of the bottom hallway, similar to the first GA. The average cohesive force for the moving robots is much higher than that of the stationary robots, and also higher than the nominal case.

Figure 11 shows a graph of the efficiency as the size of the exits are changed, using the new optimized parameters. It is clear that the second optimized case is a significant improvement over the baseline. The optimized parameters also do not significantly degrade the performance at high ratios, which means that this solution is useful in a wide range of exit ratios.

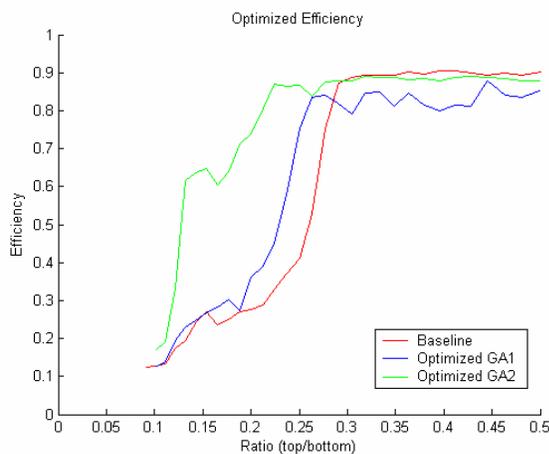


Figure 11 – This plot again shows the efficiency as the end point of the split wall is moved from the top to the center position. It is clear that GA2 improved the situation dramatically, especially at low ratios.

7 Conclusions

Simulations indicate that the introduction of social robots into crowded situations has great potential for improving pedestrian flow. The robots were able to completely eliminate the large blockages in the top part of the hallway. However, the effectiveness of these robots depends to some extent on their starting location in the crowd. It is interesting to note that the GA found an optimal solution in a relatively large search space. This illustrates the potential of genetic algorithms for finding nontrivial solutions to these types of problems.

Future work might include modifying the mutator algorithm for the GA to “fine-tune” the individual parameters by incrementally changing them instead of picking a new value at random from a specified range. Different measures of fitness could also be used, and different geometries could be optimized and compared.

8 References

- [1] J. A. Kirkland and A. A. Maciejewski, “A simulation of attempts to influence crowd dynamics,” *IEEE Int. Conf. Systems, Man, and Cybernetics*, pp. 4328-4333, Washington, DC, Oct. 5-6, 2003.
- [2] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu, “Robovie: A robot generates episode chains in our daily life,” *32nd Int. Symp. Robotics*, pp. 1356-1361, April 19-21, 2001.
- [3] T. Kanda, H. Ishiguro, T. Ono, M. Imai, and R. Nakatsu, “Development and evaluation of an interactive humanoid robot ‘Robovie’,” *2002 IEEE Int. Conf. Robotics and Automation*, Vol. 2, pp. 1848-1854, Washington DC, May 11-15, 2002.
- [4] T. Fong, I. Nourbakhsh, and K. Dautenhahn, “A survey of socially interactive robots” *Robotics and Autonomous Systems*, Vol. 42, pp. 143–166, 2003.
- [5] J. A. Kirkland, A. A. Maciejewski, and B. Eldridge, “An Analysis of Human-Robot Social Interaction for Use in Crowd Simulation,” *Robotics: Trends, Principles, and Applications*, Vol. 15, Proc. of the 9th Int. Symp. Robotics and Applications, pp. 319-324, Seville, Spain, June 28-July 1, 2004.
- [6] D. Helbing and P. Molnar, “Social force model for pedestrian dynamics,” *Physical Review E*, Vol. 51, No. 5, pp.4282-4286, May 1995
- [7] D. Helbing, I. Farkas, and T. Vicsek. “Simulating dynamical features of escape panic,” *Nature*, Vol. 407, pp. 487-490, September 28, 2000.
- [8] M. Wall, “GAlib: A C++ Library of Genetic Algorithm Components,” MIT, lancet.mit.edu/ga/, 1996.