

Towards a Computational Model of Social Comparison: Some Implications for the Cognitive Architecture

Natalie Fridman, Gal A. Kaminka
The MAVERICK Group
Computer Science Department
Bar Ilan University, Israel
{fridman, galk}@cs.biu.ac.il

Abstract

We investigate a general cognitive model of group behaviors, based on Festinger's social comparison theory (SCT), a prominent social psychology theory. We describe two possible implementations of SCT process at an architectural level, on the basis of the Soar cognitive architecture. The first, which seems to follow directly from Festinger's social comparison theory, treats the SCT process as an uncertainty-resolution method. The second, takes a different approach, in which an SCT process is constantly active, in parallel to any problem solving activity. We present the implementation of these approaches in the Soar cognitive architecture and argue that one is more suitable for modeling crowd behaviors. In previous work, we have shown that SCT covers a variety of pedestrian movement phenomena. In this paper we present the use of the SCT model in generation of imitational behavior in loosely-coupled groups. Based on experiments with human subjects, we show that SCT generates behavior in-tune with human crowd behavior.

1. Introduction

Models of crowd behavior facilitate analysis and prediction of the behavior of groups of people, who are in close geographical or mentally similar states, and are affected by each other's presence and actions. Existing models of crowd behavior, in a variety of fields, leave many open challenges. In social sciences and psychology, models often offer only qualitative description, and do not easily permit algorithmic replication. In computer science, models are often simplistic, and typically not tied to specific cognitive science theories or data. Moreover, existing computer science models often focus only on a specific phenomenon (e.g. flocking, pedestrian movement), and thus must be switched depending on the goals of the simulation.

In our previous work [7], we presented a model of crowd behavior, based on social comparison theory (SCT) [5], a popular social psychology theory that has been continuously evolving since the 1950s. The key idea in this theory is that humans, lacking objective means to evaluate their state, compare themselves to others that are similar. We believe that social comparison is a general cognitive process underlying the social

behavior of each individual in crowd. However, it was described as a stand-alone algorithm, with no discussion of how it should be integrated into the action-selection processes of the agent. Moreover, the model was evaluated almost entirely in the domain of synthetic pedestrian movement, without comparison to human crowd behavior.

In this paper we describe the implementation and adaptation of the SCT model in the Soar cognitive architecture, and provide a detailed description of its use in modeling imitational behavior. We describe two implementations of SCT process at an architectural level. The first, which seems to follow directly from Festinger's social comparison theory, treats the SCT process as an uncertainty-resolution method, i.e., as a weak (read: general) problem-solving method, which is *social*. The second takes a different approach, in which an SCT process is constantly active, in parallel to any problem solving activity. We argue that the latter approach, in which comparison is a continuous process, is more suitable for modeling crowd behaviors.

In addition, we evaluate the use of SCT in generation of imitational behavior in studies with human subjects. We show that SCT generates behavior in-tune with human crowd behavior: The subjects ranked SCT to be a middle-ground between completely individual behavior, and perfect synchronized ("soldier-like") behavior. Independently, human subjects gave similar rankings to short clips showing human crowds.

2. Background and Motivation

Social psychology literature provides several views on the emergence of crowds and the mechanisms underlying crowd behaviors [16, 1, 26, 20]. These views can inspire computational models, but are unfortunately too abstract to be used algorithmically. In contrast, computational crowd models tend to be simplistic, and focus on specific crowd behaviors (e.g, flocking). A common theme in all of them is the generation of behavior from the aggregation of many local rules of interaction, e.g. [21, 29].

Social psychology. A phenomenon observed within crowds, and discovered early in crowd behavior research, is that people in the crowd act similar to one another, often acting in a coordinated fashion as if governed by a single mind [16, 1, 26, 20, 2]. However, this coordination is achieved with little or no verbal communication.

There are several psychological theories that explain this coordinated behavior. For example, Le Bon [16] noted that individuals seem to lose their individuality (in terms of personality and thought) when becoming part of a crowd. Contagion Theory carried this further, emphasizing a view of the crowd as a "Collective Mind" that transforms an individual into becoming identical with the others in the crowd. Thus, according to Contagion Theory, the crowd as a collective causes an individual to behave in a manner similar to others [16].

Convergence Theory [1] states that crowd behavior is a product of the behavior of like-minded individuals. According to Allport [1], individuals become a part of the crowd behavior when they have a "common stimulus" with people inside the crowd. Allport's explanation of crowd homogeneous behavior is that similar people act in similar ways; otherwise they would not be a part of the same group. However, individual behavior affected by the behavior of his surrounding, thus, according to Allport, "the individual in the crowd behaves just as he would behave alone, only more so."

Additional explanations of coordinated crowd behaviors [26, 20] suggest that this coordination emerges because people in the crowd share a common social identity. Unlike Allport's individualistic behavior of people in crowds, Social Identity theory combines together the societal aspects with individual aspects.

Berk explains crowd behavior using decision theory [2]. According to Berk each individual tries to maximize her reward and minimize her cost and the crowd behaviors is no exception for this. Berk's explanation of coordinated behavior of crowds is that according to a minimax strategy, the greater the number of participants that engage in specific action, the less will be an individual cost for engaging in this action. Thus, each individual selects the action of the majority.

Different theories provides different explanations on what derives the individual behavior as being part of the crowd. However, all agree that individual behavior is affected by behavior of others when he or she is part of the crowd which displays coordinated crowd behavior.

Computational models. Work on modeling crowd behavior has also been carried out in other branches of science, in particular for modeling and simulation. However, only a few models have been validated against human data [4, 14, 10]

Reynolds [21] simulated bird flocking using simple, individual-local rules, which interacted to create coherent collective movement. There are only three rules: avoid collision with neighbors, match velocity with neighbors and stay close to the center of gravity of all neighbors. Each simulated bird is treated as a particle, attracted and repelled by others.

Blue and Adler [3] used Cellular Automata (CA) in order to simulate collective behaviors, in particular pedestrian movement. The focus is again on local interactions: Each simulated pedestrian is controlled by an automaton, which decides on its next action or behavior, based on its local neighborhoods. Blue and Adler showed that this simple rule results in the formation of lanes in movement, similarly to those formed in human pedestrian movement [28].

Helbing et al. [11, 12, 10] also focused on simulating pedestrian movement. Each entity moves according to forces of attraction and repulsion. Pedestrians react both to obstacles and to other pedestrians. They observed phenomena of self-organization in collective motion which can be caused by interaction among pedestrians similarly to the human pedestrian movement [28]. By self-organization, it means that there are some behavioral phenomena which were not planned: for example, creation of lane formation in pedestrian movement.

Kretz [14] proposes the Floor field-and-Agent based Simulation Tool model (FAST) which is a discrete-space and discrete-time model for pedestrian motion. The FAST model can be classified as an extension of Probabilistic Cellular Automata (PCA). The FAST model has been validated against human data. In particular, the model simulation results of evacuation scenario was compared to results of evacuation exercise at a primary school.

A key problem with these models is that the algorithms they provide change with the crowd phenomenon modeled. Thus, they are not presented at an architectural level. For instance, many models for crowd behavior utilize cellular-automata (CA), which differ between domains. One CA model for pedestrian movement [3] uses a set of 6

IF-THEN rules which work in parallel for all cells, to simulate the movement of pedestrians in cells. Another CA model for evacuation [27] uses different set of rules while the actions and perceptions of each entity are similar to those used in the pedestrian model. But the algorithmic computation is different.

Our work differs from those described above in that we aim to develop a general cognitive model of simulating group behaviors, based on psychology and implementable as an architectural mechanism. We have already shown that our model covers pedestrian movement phenomena as was presented in our previous work [7, 8], together with initial results on imitational behavior. In this paper, we describe the evaluation of SCT model on imitational behavior in loosely-coupled groups. We discuss the full set of results, and the evaluation methodology, in detail. Moreover, we describe two implementation of the SCT process at an architecture level, on the basis of the Soar cognitive architecture. The first which seems to follow directly Festinger's theory, treats the SCT as an uncertainty resolution method. The other is an alternative approach which treats the SCT as an constantly active process. We argue that the second approach, in which comparison is a continuous process, is more suitable for modeling crowd behaviors.

3. A Model of Social Comparison

Our research question deals with the development of a computerized cognitive model which, when executed individually by many agents, will cause them to behave as humans do in groups and crowds. The hypothesis underlying this research is that such a model should be a part of the cognitive architecture, rather than a specific task.

We took Festinger's social comparison theory [5] as inspiration for the social skills necessary for our agent. According to social comparison theory, people tend to compare their behavior with others that are most like them. To be more specific, when lacking objective means for appraisal of their opinions and capabilities, people compare their opinions and capabilities to those of others that are similar to them. They then attempt to correct any differences found.

We believe that social comparison theory may account for some characteristics of crowd behavior, as noted by Le Bon and Allport:

Contagion Theory [16]. Using social comparison, people may adopt others' behaviors. Festinger writes " [5, p. 124]: "The existence of a discrepancy in a group with respect to opinions or abilities will lead to action on the part of members of that group to reduce the discrepancy".

Convergence Theory [1]. One implication of SCT is the formation of homogeneous groups. Festinger notes " [5, p. 135]: "The drive for self evaluation is a force acting on persons to belong to groups, to associate with others. People, then, tend to move into groups which, in their judgment, hold opinions which agree with their own".

Festinger [5] presents social comparison theory (SCT) as an explicit set of axioms. The following subset of axioms (re-worded) are particularly relevant:

- When lacking objective means for evaluation, agents compare their state to that of others;
- Comparison increases with similarity;

- Agents take steps to reduce differences to the objects of comparison.

In our previous work, we turned these abstract axioms into a concrete algorithm. The algorithm is described in [7], and we provide only a brief description here (Algorithm 1).

Algorithm 1 SCT ($O, A_{me}, S_{min}, S_{max}$)

- 1: $S \leftarrow \emptyset$
 - 2: **for all** $A_o \in O$ **do**
 - 3: **if** $S_{min} < Sim(A_{me}, A_o) < S_{max}$ **then**
 - 4: $S \rightarrow S \cup A_o$
 - 5: $A_c \leftarrow \operatorname{argmax}_{A_c \in S} (Sim(A_{me}, A_o))$
 - 6: $D \leftarrow$ differences between me and agent A_c
 - 7: $a \leftarrow \operatorname{SelectAction}(D)$
 - 8: Apply action a with its Gain (Eq. 1) to minimize differences in D .
-

Each observed agent A_i is taken to be a tuple of k state features $A \equiv \langle f_1^A, \dots, f_k^A \rangle$. For each such agent, we calculate a similarity value $Sim(A_{me}, A_o)$, which measures the similarity between the observed agent A_o and the agent carrying out the comparison process A_{me} . The agent A_c with the highest such value within the bounds (S_{min}, S_{max}) is selected for the comparison. Then the comparing agent selects an actions (a) to reduce the discrepancy. We use a gain function $Gain(Sim(A_{me}, A_c))$ (Eq. 1) for the selected action a , which translates into the amount of effort or power invested in the action. For instance, for movement, the gain function would translate into velocity; the greater the gain, the greater the acceleration.

$$Gain(Sim(A_{me}, A_c)) \equiv \frac{S_{max} - S_{min}}{S_{max} - Sim(A_{me}, A_c)} \quad (1)$$

4. SCT Implementation in Soar

One of the open questions that we are coping with, is how to implement the SCT process at the agent architecture level. Soar serves us in this paper as a representative cognitive agent architecture.

We implemented SCT in the Soar cognitive architecture [19]. Soar was connected to the GameBots virtual environment [13]. Here, multiple agents, each controlled by a separate Soar process (each executing SCT) can interact with each other in a dynamic, complex, 3D virtual world (see Figure 1).

A detailed discussion of Soar’s role as a cognitive architecture is beyond the scope of this paper. We provide a very brief overview here, and refer the interested reader to [19, 17, 18, 15] for additional details.

Soar has two components: A graph-structured working memory, and a set of user-defined production rules that test and modify this memory. Efficient algorithms maintain the working memory by executing rules that match existing contents. All the



Figure 1: Soar agents in the GameBots environment. Each agent has limited field of view and range, and may move about and turn.

agent’s knowledge, sensor readings, and decisions are recorded in the working memory. Soar operates in a classic sense-think-act cycle, which includes a decision phase in which all relevant knowledge is brought to bear to propose, and then select, an *operator*, that will then carry out deliberate mental (and sometimes physical) actions. Once the operator finishes its actions, it is automatically de-selected (terminated), and the cycle repeats. Unlike simple production rules, whose effects on working memory are temporary, operator-induced the actions of rule firings on working memory (and in turn, on physical actions) are persistent, even after the operator has been de-selected. Overall, a Soar agent’s behavior is the result of the sequential selection of operators, each performing an action on the environment and/or internal memory.

In the following sections, we propose two approaches for such implementation. The first approach directly follows Festinger’s social comparison theory. According to Festinger, the SCT process is a response to uncertainty (“lack of objective means to evaluate opinions”). However, an alternative approach is to view the SCT process as an on-going process at the architecture level. Such a view requires that the agent will be monitoring other agents all the time, and is thus more computationally expensive. It also seems to contrast with Festinger’s theory.

It may appear easy to dismiss the implementation question as insignificant. However, the implementation choice carries significant implication: As SCT processes inherently rely on knowing about the behavior of others, the implementation question raises a more fundamental question about where modeling of others (e.g., using plan recognition) occurs in cognition: Is it a problem-solving activity, or is it carried out all the time, at an architectural level.

4.1. Implementation of the SCT as Problem-Solving Activity in Soar

A key novelty in Soar is that it automatically recognizes situations in which the decision-phase is stumped, either because no operator is available for selection (*state no-change impasse*), or because conflicting alternatives are proposed (*operator tie impasse*). When an impasse is detected, a subgoal is automatically created to resolve it. Thus, social comparison theory as described by Festinger seems to naturally fit Soar’s impasse-driven operation. In particular, Festinger describes the trigger to using

comparison as a situation in which people are unable to evaluate their opinions and capabilities, which seems to match an impasse situation.

We thus may treat the social comparison process as a new kind of uncertainty-resolution method, i.e., a method for impasse-resolution (problem-solving). Our goal in this section is to determine a general way to describe social comparison processes in Soar, in such a way that they can be used for solving a wide variety of problems.

A snapshot from a log showing Soar using SCT implementation as Problem-Solving Activity process (here, to decide on movement) is shown below. Soar's decision cycles are denoted by numbers before colons. In the first and second decision cycles, operators called *init* and *explore-decision*, respectively, are selected by Soar. Then, more than 20 different instantiations of an operator called *elaborate-target* are proposed by the system, one for each available navigation point for our agent. Each such operator promotes our agent towards its target and specifically in this case towards different available locations. Soar is faced with the task of choosing one among different operators for execution. Since it cannot decide, an operator-tie impasse is declared; see the line marked

```
3: ==> S: S3 (operator tie)
```

This triggers our social comparison process, which is carried out, in sequence, by the following operators: (i) *sct-init*, which sets up the new state, and copies relevant information. (ii) *sct-add-entities*, which copies information about other agents for use in ranking operators. *rank-item* then calculates a rank for all proposed operators, based on associated agents and their own choices. Finally *select-item* selects the highest-ranking operator and makes the decision. Indeed the last decision cycle (#8 in the log) shows a specific instance of the *elaborate-target* was chosen.

```
1:    0: 02 (init)
root is active ->proposed child : explore-decision ->by : root
2:    0: 04 (explore-decision)
->proposed child : elaborate-target ->by : explore-decision

->proposed child : elaborate-target ->by : explore-decision
[.....additional proposals for elaborate-target .....]
->proposed child : elaborate-target ->by : explore-decision

3:    ==>S: S3 (operator tie)
4:    0: 027 (sct-init)
5:    0: 028 (sct-add-entities)
6:    0: 051 (rank-item)

7:    0: 068 (select-item)
SCT Done.   Chose 021   Name: elaborate-target
8:    0: 021 (elaborate-target)
elaborate-target is active

[.....Soar continues .....]
```

4.2. Implementation of the SCT as an on-going process in Soar

A different way of treating SCT process is as on-going process at the architecture level that is active in parallel to problem-solving activity. Thus, SCT was implemented as secondary parallel thread within Soar (Figure 2). At every cycle, operators are proposed (and selected) by Soar based on their suitability for a current goal (e.g., through means-end analysis), and also based on their suitability for SCT. Thus SCT-proposed operators compete with the task-oriented operators for control of the agent. By setting Soar’s decision preferences to prefer the SCT-proposed operators, we get a very social agent. Conversely, by preferring the task-oriented operators, we get an individual choice agent which makes its decisions independently of its peers.

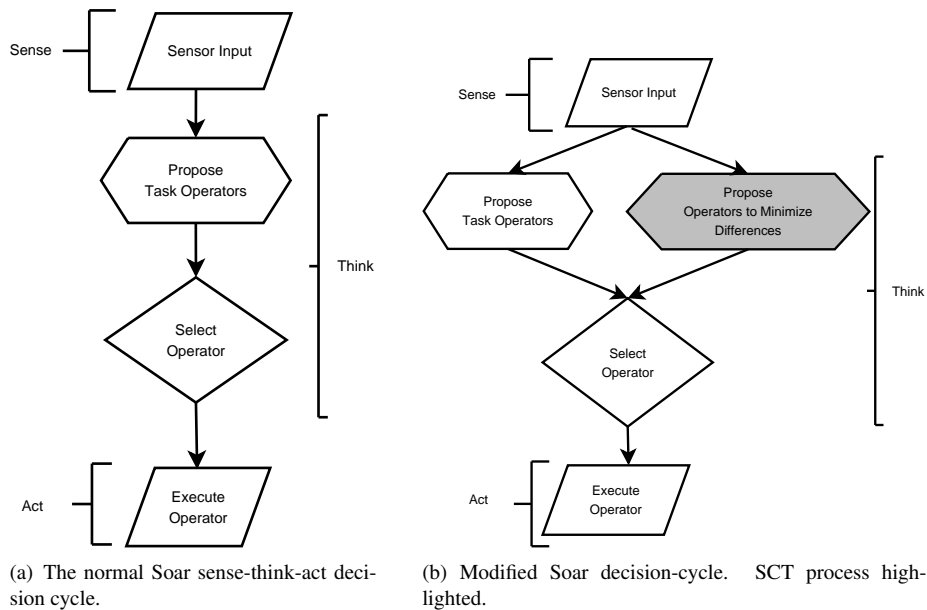


Figure 2: The Soar sense-think-act decision cycle, SCT process highlighted.

The SCT thread proposed operators by following the algorithm described previously in Section 3, though in a way that is adopted for Soar’s decision cycle: At every cycle, for each observed agent and for each difference, the SCT process would propose an operator that would minimize the difference. Then, a set of preference rules is triggered that ranks the proposals based on feature weight. Additional rules prefer the most similar agent (that is still not sufficiently similar).

Figure 3 presents a snapshot of Soar’s graph-structured working memory during one of the experiments. *S1* represents the root. In this example, there are several operators that were proposed by Soar and matched the current state of the agent (*O18*, *O21*, *O8*, *O1*, *O19*, *O20*). According to this approach SCT process proposes operators which minimized the differences to the observed agents in parallel to problem-solving activity. In this example, the SCT operators are *O20*, *O19* and *O18* which minimized


```

(S1 ^events E1 ^goal G1 ^inputs I2 ^io I1
  ^me yes ^name |BISBot| ^operator O18+ ^operator O21+ ^operator O8+
  ^operator O1+ ^operator O19+ ^operator O20+ ^outputs I3 ^recipe R11
  ^rotation left ^sct-memory M5 ^state S1 ^superstate nil ^team R1
  ^timer 93773.1 ^type state)
(I1 ^input-link I2 ^output-link I3)
(M5 ^sct S4 ^time-out 93602.7)
(R11 ^name one ^root R12 ^rsc R15)
(E1 ^event E7 ^event E8 ^event E6)
(G1)
(R1 ^name team ^root R2 ^tsc R5)
(I2 ^audio A1 ^objects O3 ^query-results Q1 ^self S2)
(I3)
(O21 ^entity E5 ^loc L13 ^name turn-to-entity-memory ^rotation R10
  ^type sct-memory)
(O20 ^amount 9216. ^entity E3 ^name turn-to-entity ^type sct)
(O19 ^amount -1. ^entity E4 ^name turn-to-entity ^type sct)
(O18 ^amount 0. ^entity E5 ^name turn-to-entity ^type sct)
(O8 ^dparams nil ^me yes ^name explore-decision ^sc R13)
(O1 ^name wait)

```

Figure 3: Soar’s memory snapshot

the differences in direction between three observed agents (*E3*, *E4* and *E5*). The problem-solving activity process proposes task oriented operators which promote the agent toward its current goal (in this example: *O21*, *O8*, *O1*). The figure shows that the SCT operators and task-oriented operators are competing for control of the agent. At the end, only one is selected.

We now present a snapshot from a log showing Soar using SCT implementation as on-going approach. When no SCT operators are available (agent do not observe any other agents), the task-oriented operators took control. In this example, explore-decision, elaborate-no-target and wait are task-oriented operators (lines #2,#3 and #4 in the log). When agent observe other agents, the SCT process proposes operators (ex. turn-to-entity) in parallel to task oriented process. Based on different preference rules, Soar selects one operator for the execution. In our example, according to Soar working memory (Figure 3) the following operators are proposed *O18*, *O21*, *O8*, *O1*, *O19*, *O20*, however, the *O20* (turn-to-entity) which is SCT operator is selected for the execution (line #7 in the log).

```

1:      O: O2 (init)
2:      O: O4 (explore-decision)
3:      O: O5 (elaborate-no-target)
4:      O: O1 (wait)
5:      O: O6 (turn-to-entity)
6:      O: O1 (wait)

[.....Soar continues .....]

7:      O: O20 (turn-to-entity)
8:      O: O1 (wait)

[.....Soar continues .....]

```

Implementing SCT as on-going process at the architecture level may appear to contradict Festinger’s theorizing that social comparison comes into play only when people

are at an impasse. However, this is not the case. For instance, by setting Soar’s decision preferences to prefer SCT-proposed operators only when no task-oriented operators are available, one gets the behavior predicted by Festinger’s theory.

5. Modeling Imitational Behavior

An attractive feature of social comparison is its hypothesized prevalence in human group behavior, i.e., its generality across different behaviors. Indeed, we believe that the SCT model we present in this paper is sufficiently general to account for a wide variety of group behaviors. This is in contrast to many existing computational models, that typically focus on specific tasks. In previous work [7, 8] we evaluated the use of the SCT model in generation of different pedestrian movement phenomena. Here, we provide additional evidence for such generality by describing the application of the SCT model to the problem of generating imitational behaviors in loosely-coupled groups.

In the next section, we describe the experiment setup. We then briefly describe a pilot experiment comparing the two SCT implementation approaches (Section 5.2), based on which one of the approaches is dropped from further consideration. We then describe experiments examining the remaining approach in simulating a crowd of virtual agents (Section 5.4), and contrast this with the behavior of a human crowd (Section 5.5).

5.1. Experimental Setup

Unlike individual imitation, where one individual imitates a role model, crowd imitational behavior spreads across a group of individuals who dynamically select role models for imitation. Imitation in a crowd occurs more loosely, as the imitation role models do not necessarily intend to play their role, and indeed may not even know that they are being imitated. Also, the imitators potentially switch their comparison targets from one moment to the next. Psychology literature describes such imitational behavior as one of the keystones of crowd behaviors [16].

In order to simulate imitational behavior we used position and direction as the agents’ feature set, as these are obvious observable features. For each observed agent and for every difference found, the SCT process proposes a corrective operator to be performed in order to minimize the difference in the selected feature. In this task, the corrective operators were ‘move-to’ (minimizing distance to the observed agent, correcting position differences) and ‘turn-to’ (imitating angle of the observed agent, turning to the same direction as observed agent).

In addition to the proposed SCT operators, Soar also proposes operators based on their suitability for the current goal, and based on a monitoring mechanism which proposes operators seeking information. In this task, goal operators were ‘turn-to’ (a random angle); the monitoring mechanism operators turned towards previously seen agents.

5.2. A Pilot Experiment: Comparing SCT Implementations

We began evaluating the SCT implementations by running a small-scale pilot experiment, in which we presented to three subjects the resulting simulated behavior of both approaches (SCT as a problem-solving approach and SCT as on-going approach) and afterwards ask their opinion regarding the simulated behavior. All subjects claimed that SCT as a problem-solving approach provide unrealistic behavior, and in particular that the agent's behavior was completely individual, and the agents did not appear to coordinate or influence each other in any way. It became clear that the problem-solving approach simply would not do. Below, we describe the two implementations for this experiment, and explain the results.

5.2.1. On-Going SCT

For our experiments, several basic task-oriented operators and SCT operators were implemented, to allow the agents to move about, turn towards each other, measure distances to others, etc. Thus one thread of control, always running, is in control of the agent's actions towards whatever tasks it was given. The additional thread of control which is also always running is the SCT process.

Here an addition to the SCT model became necessary. Suppose an agent X decided to turn towards the same angle as an agent Y that is next to it. For instance, suppose agent X , facing West, sees agent Y looking North. Agent X now imitates agent Y , and also turns to the North. Due to the limited field-of-view of X , it would lose track of Y once it makes the turn. From that point on, it could no longer keep track of Y , to minimize additional differences. This would cause it to become overly reactive, turning about immediately to seek Y again, or to select a different operator altogether (now that Y could no longer be imitated).

We thus found it necessary to utilize two mechanisms: (i) a memory mechanism that keeps track of the whereabouts of agents, once seen; and (ii) a monitoring mechanism that occasionally would turn towards remembered agents, to provide an update on their state (for the purpose of comparison). Both of these mechanisms (memory and monitoring) are of course present in many cognitive architectures [23, 22], and are not necessarily linked to SCT. We thus leave discussion of such mechanisms outside of this paper.

We used Soar preference rules to rank the feature weights such that the position feature gets higher priority than direction. This means that a closest agent is considered to be more similar, however the chosen feature for correction is direction. The S_{max} value was unbounded, which means that there is no such thing as too similar. In our case Soar can propose corrective operator with value equal to zero if there is no correction to make with respect to the observed agent. We used additional Soar preference rules to give higher priority to exploration mechanism operators than to goal operators. Thus, each agent prefers the SCT operators ('turn to') and in the case when there are no seen agents (i.e. there is no proposed SCT turn-to operator) an agent will prefer the exploration mechanism operators, and only afterwards the goal operators. The resulting simulated behavior has the agents standing in their initial locations, turning to some direction (often those of some other agent near them) or doing nothing.

5.2.2. *SCT as Problem Solving*

Implementing the SCT as problem-solving process in Soar architecture proved an interesting challenge. In particular, we quickly discovered that in the initial implementation, the reason agents using it did not display social behavior was that the SCT process (as a problem solving activity) was rarely, if ever, called. In imitational behavior, the task-oriented operator that promote agent towards its current goal, is a turn-to operator which orients the agent toward a direction of interest. This operator is constantly available for the agent, since the agent can always turn, thus the "state no-change impasse" where no operator is proposed, never takes place. So we could not rely on the SCT problem solving method to be called on state no-change impasses.

We thus switched to an alternate trigger for the SCT process. It could be triggered as a response to an operator tie impasse, where there is a conflict, i.e., when task-oriented operators compete with each other. Given that the agent could turn in any direction, Soar proposed multiple turn-to operators (one for each separate target angle). These competed with each other, creating a tie-impasse. Here, two approaches are possible: To use preference rules (which prefer one operator over another) to prefer one operator over the others (in which case, no impasse occurs, and thus no SCT process), or to allow an impasse to occur. However, given that turn-to operators are always possible, and are thus constantly proposed, the impasses would never end. The Soar agent would be forever executing SCT processes to try to keep up with impasses.

Regardless of which tactic we tried, the resulting simulated behavior was individual: The virtual agents ignored each other, and only performed their task-oriented operators. This is what caused the subjects to so decidedly reject this approach.

5.2.3. *Additional support for On-Going SCT*

There is other evidence for rejecting the approach of social comparison as a problem-solving activity, which is triggered only on impasse, i.e., only when the agent becomes stuck in its goal-oriented activity. We discuss this now.

First, from psychological literature, we learn that people are always aware of other people's behavior, and indeed elaborations on social comparison theory expanded its view on when comparison takes place. For example, Hakmiller [9] and Singer [24] expanded the theory and demonstrated that people tend to confirm or reassure that their actions or beliefs are the correct ones, by comparing themselves to others after making their decisions. Thus according to this approach people tend to use social comparison *in parallel to their decision making process*.

Second, results in other domains (pedestrian movement, gathering and crowding [6]), show that even when used as a stand-alone algorithm, the SCT process must be constantly active to produce reasonable results. Whenever it is used only when the agent is "stuck", the quality of the simulation drops, and sometimes disappears altogether (i.e., the simulation fails).

Finally, treating SCT as on-going process is a more general approach, which includes the SCT as a problem-solving activity approach as its special case and provides a compatibility with Festinger's theory. However, the reverse is not true. Thus the on-going approach is in fact more general.

5.3. Evaluation of imitational behavior: Methodology

We conducted experiments to evaluate whether SCT can indeed generalize to account for imitational behavior in groups. Unlike the pedestrian movement domain, where clear measures are available for objective measurement of the success of a model (e.g., flow, lane changes), imitational behavior does not have clear standards of evaluation.

We propose a method for evaluation of imitational behavior. We propose a questionnaire composed of general questions and specific tasks related questions. The general questions can be used as a common method for evaluation of all kinds of imitational behaviors. We rely on experiments with human subjects, which judged the human crowd behavior and the resulting SCT behavior in comparison to completely individual behavior (i.e., arbitrary decisions by each agent, independent of its peers), and to completely synchronized behavior (i.e., all agents act in complete unison).

The first hypothesis underlying the experiments was that groups controlled by SCT would generate behavior that would be ranked somewhere in-between the individual and perfect-coordination models, i.e., that SCT would generate behavior that would be perceived as coordinated, but not perfectly so. Another hypothesis is that human crowd behavior would also be ranked somewhere in-between the individual and perfect-coordinated behaviors.

To examine the first hypothesis, we created three screen-capture movies of 11 Soar agents in action. All movies were shot from the same point of view, and showed the agents in the same environment. In all screen-capture movies there is one blue agent that stands in front and turns up to 90° left or right. All others are red agents that act according to one of the models.

In one movie (*individual*), the red agents act completely independently of each other, randomly choosing an angle and turning to it. In another (*unison*), the red agents act in almost perfect coordination, turning towards the same angle as the blue agent almost instantaneously (small timing differences result from asynchronous responses of the simulated environment). Finally, in the *SCT* movie, the red agents act according to our model as described above.

These experiments were carried out using 12 subjects (ages: 18–40, mean: 28; male: 6; additional 4 subjects dropped due to technical reasons). Each subject was given a brief description of the appearance of the environment and agents, sometimes aided by a snapshot from a movie (e.g., as in Figure 1). The subjects were told that the purpose of the experiment was to evaluate the use of perception models embedded in the agents; that there was a red dot—visible to the agents but not to the subjects—that moves about on the walls surrounding the group. The agents’ goal is to individually locate this dot, and then track it in place by turning around. The purpose of the cover story was to focus the attention of the subjects away from group behavior and imitation, so as to not bias the results. After the description, the movies were shown to the subject.

After each movie, the subjects were asked to fill a short questionnaire (described below) based on what they saw. Each movie was shown only once. The order of presentation of movies was randomly selected for each subject, to control for learning and order effects. The questionnaire included the following questions:

1. If there is only one red dot in the room, to what degree did all agents see it? (1 -

- nobody saw the red dot; 6 - all agents saw it)
2. To what degree were the movements of the agents random? (1 - not random at all; 6 - very random)
 3. To what degree was there cooperation between the agents? (1 - no cooperation at all; 6 - full cooperation)
 4. To what degree was there agreement between the agents? (1 - no agreement at all; 6 - full agreement)
 5. To what degree were the agents coordinated in terms of the direction of their movements? (1 -no coordination at all; 6 - fully coordinated)
 6. How quickly did the agents find the red dot? (1 - dot not found at all; 6 - immediately found)
 7. To what degree were the agents related to each other? (1 - no relation at all; 6 - tight relation)
 8. Do you see any leaders? If so, how many? (1-11) (1- one leader; 11 - all agents are leaders, i.e., no leader).

In this experiment, the subjects were asked to grade the movies on an ordinal scale of 1–6, with 1 being a low score (typically associated with more individual behavior), and 6 being a high score (typically associated with perfect unison). In order to keep consistency in presentation of results, the scale of the second question (Non-Random) was reversed. The results of the last question (Number of leaders) are presented separately due to inconsistency in scale with other questions.

5.4. Results: Virtual Agents

In general, the responses to the questions in this experiment have placed SCT between the individual and unison models. Results are summarized in Figure 4(a) and 4(b). The questions in Figure 4(a) are associated with agents' performance on a given task. In the presented questionnaire the number of questions are 1, 3, 4 and 6. Figure 4(b) refers to more general questions (i.e. the same questions that were used in human crowd movie). In the questionnaire the relevant numbers of questions are 2, 5, and 7. The categories in the X-axis correspond to questions given to the subjects. The Y-axis measures the median result. Each bar correspond to compared model and as explained above we compare SCT model to Individual and Unison models.

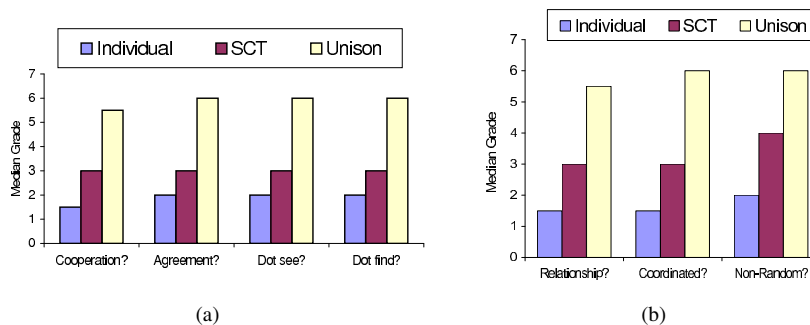


Figure 4: Results of questionnaire on agents performance.

The results clearly demonstrate that the SCT model lies in between the individual and perfect-unison model. While in some questions it appears to be somewhat closer to the individual model, it is significantly different from it (ex. question #3: $t(11) = 2.17$ $p = 0.02$, one-tailed t-test, question #6: $t(11) = 3.36$ $p < 0.01$, one-tailed t-test).

Figure 5(a) shows the results for the question on the number of leaders. The median result for the individual was 11 (i.e., every agent is a leader, or in other words, no leader). For the unison model, the median result was 1. For the SCT model, the median result was 3. In this question the SCT model result is very close to the Unison model. According to t-test (one-tailed) the SCT model significantly differs from the Individual model ($t(11) = -2.24$, $p = 0.02$). However, in comparison to Unison model there is no significance found ($t(11) = 0.51$, $p = 0.3$).

We conducted an additional experiment, in which static images—snapshots from the movies—were shown to subjects. However, as opposed to screen-capture movies where the subject answered the questionnaire after seeing each movie only once, in this experiment the subjects answered the questionnaire while watching still images.

We took three snapshots from each movie. The snapshots were taken at the same time slot from the beginning, middle and the end of the movies. After filling the screen-capture movies questionnaire, the snapshots from the movies were shown to the subjects. The question presented to the subjects was: “If there can be more than one red dot in the room and each agent is known to be fixating on a dot, how many red dots are in the room?” The subjects were asked to grade each snapshot on an ordinal scale of 1–11 with 1 being a low result (i.e., all agents look at the same red dot) and 11 being a high result (i.e., the number of red dots as the number of agents meaning all agents look at different dots).

The results of this experiment are summarized in Figure 5(b). Again the categories in the X-axis correspond to question given to the subjects. The Y-axis measures the average of median results that belong to each model.

Again the results demonstrate that the SCT model lies in between the individual and perfect-unison model and it significantly differs from the individual model ($t(11) = -2.86$, $p < 0.01$, one-tailed t-test) and from perfect-unison model ($t(11) = 2.58$, $p = 0.012$, one-tailed t-test).

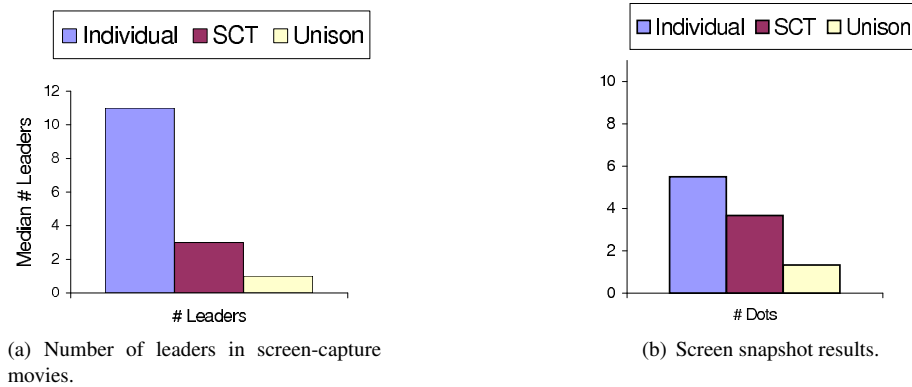


Figure 5: Additional results for the simulated agents.

5.5. Results: Human Crowd

Another hypothesis underlying the experiments is that human crowd behavior would also be ranked somewhere in-between the individual and unison models. To examine this, we search for a human crowd movie where individuals perform the same action as in simulated agents movies. We used a news clip movie which shows people, grouped together, standing and waiting for some event to occur. The only action they perform in the movie is to turn occasionally.

The experiment was carried out using 12 human subjects. These were not the same subjects who participated in the first experiment (sections 5.1 – 5.4). Each subject, after viewing a human crowd movie (Figure 6(a)) was asked to fill the same questionnaire as in previous experiments. However, since in the human crowd movie there was no cover story about red dot, there were some irrelevant questions that were dropped out. The remaining questions are more general and not tied to a specific task.

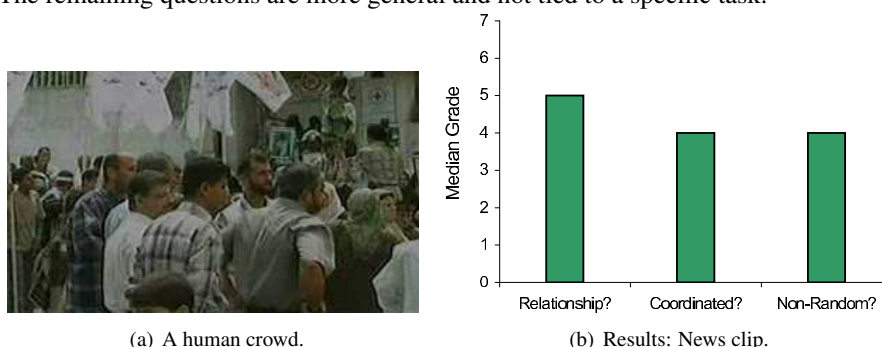


Figure 6: **Human crowd.**

Results are summarized in Figure 6(b). As in previous results, the categories in the X-axis correspond to questions given to the subjects and the Y-axis measures the median result.

We compare the human crowd results to the individual and perfect-unison models results. It appears to be significantly different from the individual model in all questions ($t(22) = 7.20$ $p < 0.01$, $t(22) = 5.89$ $p < 0.01$, and $t(22) = -2.4$ $p < 0.05$, respectively; t-test, one-tailed). However, in comparison to the perfect-unison model, the results of the coordination and non-random questions are significantly different ($t(22) = -3.59$ $p < 0.01$, and $t(22) = 4.98$ $p < 0.01$, respectively; t-test, one-tailed). The results of the relationship question shows no significant difference between the perfect-unison and the news-clip movie ($t(22) = -0.13$, $p = 0.44$).

In response to the question “Do you see any leaders? If so, how many?”, the median result in human crowd movie was 1.5. It is significantly different from the individual model ($t(22) = -3.71$, $p < 0.01$, t-test, one-tailed) but not in comparison to the perfect-unison ($t(22) = -0.30$, $p = 0.38$). When the subjects were asked to qualitatively discuss their answer to this question, many subjects reported that they don’t see any leader, however “one must be present outside of the view of the movie, since the crowd is waiting for something or someone”. However, when they were asked to refer to only people seen in the movie, the answer was that there were several subgroups in the seen crowd. While this qualitative answer is similar to the answer we received in

asking similar questions about the simulation movies, we do not believe that this necessarily suggests that the SCT model is completely accounting for realistic behavior. In the future, we will focus more explicitly on the issue of subgroups, by adding the following question to the questionnaire: “Are there any subgroups? If so, how many?”.

6. Conclusions

This paper presented a model describing crowd behavior, inspired by Festinger’s social comparison theory [5]. The model intuitively matches many of the characteristic observations made of human crowd behavior, in particular as described by major crowd behavior theories.

We presented in detail two implementation approaches to social comparison processes, in Soar cognitive architecture. The first approach treats social comparison as a problem solving activity, to be triggered by uncertainty in the agents’ task-oriented reasoning. The second approach treats social comparison as an on-going processes, taking place in parallel to any task-oriented reasoning and actions. In early experiments, it was shown that the latter approach is superior, and we discuss the reasons for this choice in depth.

We report on experiments with human subjects, to evaluate the use of the on-going SCT process in modeling imitational behavior in a group. Though there is a lack of objective data against which the model can be evaluated, results of experiments with human test subjects are promising and seem to match intuitions as to observed behavior. The subjects ranked SCT to be a middle-ground between completely individual behavior, and perfect synchronized (“soldier-like”) behavior. Independently, human subjects gave similar rankings to a short news clip showing human crowds.

The research reported in this paper raises several implications for cognitive architecture and their structure, as it pertains to social reasoning.

- First, accepting that social comparison takes place in parallel to problem solving activity implies that agent modeling—the process by which an agent keeps track of other agents, monitors their behavior, and possibly infers their intention—is carried out at an architectural level, in parallel. Based on this, it would seem that agent-modeling, which forms the basis for social comparison, is in some sense like learning: Always present, always taking place. The agent can choose, perhaps, to what degree to follow up on its results, but the agent cannot turn it off.
- Second, accepting that social comparison takes place constantly, in parallel to task-oriented reasoning, we are now faced with the challenge of investigating the timing of comparison: When do humans follow up on the results of the comparison, and when do they ignore it? Does the tendency to follow up on social comparison change between people? What affects this tendency? etc.
- Third, there is of course much more depth to social comparison, as the basis for cognitive modeling. For instance, we know from psychology literature that

the comparison process itself is more complex than the process captured in Algorithm 1. For instance, psychology tells us that the process of selecting a target for comparison is much more involved than simply picking the most similar agent [25].

We hope to pursue these implications and directions for further research in our future work. We are particularly interested in continuing to investigate cognitive models of social behavior and social reasoning.

Acknowledgments. This research was supported in part by Israeli Science Foundation grant #1357/07, and by IMOD.

- [1] F. H. Allport. *Social Psychology*. Boston: Houghton Mifflin, 1924.
- [2] R. Berk. A gaming approach to crowd behavior. *American Sociological Review*, 1974.
- [3] V. J. Blue and J. L. Adler. Cellular automata microsimulation of bidirectional pedestrian flows. *Transportation Research Record*, pages 135–141, 2000.
- [4] W. Daamen and S. P. Hoogendoorn. Experimental research of pedestrian walking behavior. *Transportation Research Record*, pages 20–30, 2003.
- [5] L. Festinger. A theory of social comparison processes. *Human Relations*, pages 117–140, 1954.
- [6] N. Fridman. Modeling crowd behavior based on social comparison theory. Master’s thesis, Bar Ilan University, 2007.
- [7] N. Fridman and G. A. Kaminka. Towards a cognitive model of crowd behavior based on social comparison theory. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI-07)*, 2007.
- [8] N. Fridman, G. A. Kaminka, and M. Traub. First steps towards a social comparison model of crowds. In *International Conference on Cognitive Modeling (ICCM-09)*, 2009.
- [9] K. L. Hakmiller. Threat as a determinant of downward comparison. *Journal of experimental social psychology*, 2:32–39, 1966.
- [10] D. Helbing. Traffic and related self-driven many-particle systems. *Reviews on Modern Physics*, 73, 2001.
- [11] D. Helbing and P. Molnar. Self-organization phenomena in pedestrian crowds. In F. Schweitzer, editor, *Self-organization of Complex Structures: From Individual to Collective Dynamics*, pages 569–577. Gordon and Breach, 1997.
- [12] D. Helbing, P. Molnar, I. J. Farkas, and K. Bolay. Self-organizing pedestrian movement. *Environment and Planning B*, 28:361–384, 2001.

- [13] G. A. Kaminka, M. M. Veloso, S. Schaffer, C. Sollitto, R. Adobbati, A. N. Marshall, A. Scholer, and S. Tejada. GameBots: A flexible test bed for multiagent team research. *Communications of the ACM*, 45(1):43–45, January 2002.
- [14] T. Kretz. *Pedestrian traffic: simulation and experiments*. PhD thesis, Universität Duisburg-Essen, 2007.
- [15] J. E. Laird. Extending the soar cognitive architecture. In *Proceedings of the Artificial General Intelligence Conference*, Memphis, TN, 2008.
- [16] G. Le Bon. *The crowd: A study of the popular mind*. Dunwoody, Ga., N.S. Berg, 1968 edition, 1895.
- [17] J. F. Lehman, J. Laird, and P. Rosenbloom. A gentle introduction to soar, an architecture for human cognition. In S. Sternberg and D. Scarborough, editors, *Invitation to Cognitive Science*, volume 4. MIT Press, Cambridge, MA, 1996.
- [18] J. F. Lehman, J. Laird, and P. Rosenbloom. A gentle introduction to soar, an architecture for human cognition: 2006 update. Technical report, University of Michigan, 2006.
- [19] A. Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, Massachusetts, 1990.
- [20] S. Reicher. The psychology of crowd dynamics. In *Blackwell handbook of social psychology: Group processes*. Oxford: Blackwell, 2001.
- [21] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques (SIGGRAPH-87)*, pages 25–34, New York, NY, USA, 1987. ACM Press.
- [22] D. D. Salvucci. Modeling driver behavior in a cognitive architecture. *Human Factors*, 2006.
- [23] D. D. Salvucci, E. R. Boer, and A. Liu. Toward an integrated model of driver behavior in cognitive architecture. *Transportation Research Record: Journal of the Transportation Research Board*, 2001.
- [24] J. E. Singer. Social comparison: progress and issues. *Journal of experimental social psychology*, 2:103–110, 1966.
- [25] J. Suls and L. W. (Edt). *Handbook of social comparison: theory and research*. New York: Plenum Press, 2000.
- [26] H. Tajfel and J. C. Turner. The social identity theory of intergroup behavior. In *Psychology of intergroup relations*. Chicago: Nelson-Hall, 1986.
- [27] P. C. Tissera, M. Printista, and M. L. Errecalde. Evacuation simulations using cellular automata. *Journal of Computer Science and Technology*, 2007.

- [28] M. Wolff. Notes on the behaviour of pedestrians. *In People in Places: The Sociology of the Familiar*, pages 35–48, 1973.
- [29] K. Yamashita and A. Umemura. Lattice gas simulation of crowd behavior. In *Proceedings of the International Symposium on Micromechatronics and Human Science*, pages 343–348, 2003.