Sasanka Sekhar Chanda

Professor, Strategic Management,
Indian Institute of Management Indore
B-Tech (IIT Kharagpur 1994), PGDM (IIM Lucknow 1998), FPM (IIM Calcutta 2013)
[PGDM may be considered equivalent to M.B.A. Likewise FPM is equivalent to PhD.]

Sasanka Sekhar Chanda's research interests are in theory development by computational simulation modeling using genetic algorithm and Kauffman's NK model, encompassing autonomous strategic behavior and complexity, uncertainty, failure and renewal. Earlier, Sasanka worked in the industry in a range of roles spanning engineering, consulting, and management over a period of fifteen years.

TEACHING

Program	Subject
Undergraduate	√ An Introduction to Thinking in Complexity
M.B.A.	√ Corporate Strategy √ Strategic Analysis of Business Events (based on C. K. Prahalad's work) √ Artificial Intelligence in Human Resource Management
Doctoral	√ Philosophical Moorings of Social Science Research √ Theory of the Firm

RESEARCH

1. Chanda SS, Ray S (2011) 'Generic strategies in dynamic environments'. *Academy of Management Annual Meeting* (*AOM*), BPS Division, 2011, San Antonio, Texas.
2. Chanda SS, Ray S, Das R (2011) 'Developmental programmes, microcredit and Gandhian innovation: Pillars of a bottom of pyramid strategy?' *Strategic Management Society* (*SMS*) Special Conference 2011, San Diego.
3. Chanda SS, Ray S (2011) 'Do managers add value in any environment?' *Journal of Management Studies (JMS) Alternative Conference*, October 2011, Hong Kong, SAR.
4. Chanda SS, Ray S (2013) 'Why do strategic projects fail in MNCs? A resource dependence perspective'. *SMS* India Special Conference at Mohali, India, December 2013.
5. Chanda SS (2013) 'Comprehensiveness in making strategic decisions: Boon or bane?' *SMS* India Special conference at Mohali, India, December, 2013.
6. Chanda SS, Ray S (2015) Formal theory development by computational simulation modelling: A Tale of two philosophical approaches. ***Decision***, 42(3): 251–267. DOI: 10.1007/s40622-015-0096-y. https://link.springer.com/article/10.1007/s40622-015-0096-y
7. Chanda SS (2015) *CEO Cognition in Strategy Research*. Available at SSRN: http://dx.doi.org/10.2139/ssrn.2586215.

8. Chanda SS, Ray S (2015) Optimal exploration and exploitation: The managerial intentionality perspective. ***Computational and Mathematical Organization Theory***, 21(3): 247–273. DOI: 10.1007/s10588-015-9184-y
9. Chanda SS, Ray S (2016) Learning from project failure: Globalization lessons for an MNC. ***Thunderbird International Business Review***, 58(6): 575–585. DOI: 10.1002/tie.21776
10. Chanda SS (2017) Inferring final organizational outcomes from intermediate outcomes of exploration and exploitation: The complexity link. ***Computational and Mathematical Organization Theory***, 23(1): 61–93. DOI: 10.1007/s10588-016-9217-1 (https://rdcu.be/5wsj)
11. Chanda SS (2017) 'Ontology and epistemology of conceptual replication of computational simulation modelling research'. *Academy of Management Annual Meeting* (*AOM*), RM Division, 2017, Atlanta, Georgia.
12. Chanda SS, Ray S, McKelvey B (2018) The continuum conception of exploration and exploitation: An update to March's theory. ***M@n@gement***, 21(3): 1050–1079. https://management-aims.com/index.php/mgmt/issue/view/189
13. Chatterjee A, Chanda SS, Ray S (2018) Administration of an organization undergoing change: Some limitations of the transaction cost economics approach. ***International Journal of Organizational Analysis***, 26(4): 691–708. DOI: IJOA-07-2017-1202 https://www.emeraldinsight.com/eprint/PFGBTPPAGSZHAJ92RNFD/full
14. Chanda SS (2018) 'When are exploration and exploitation orthogonal constructs, when do they form ends of a continuum?' *SMS* Special Conference in Hyderabad (*SMS*).
15. Chanda SS (2018) 'Corporate strategy as order creation in far-from-equilibrium conditions'. 20th Annual Convention of the *Strategic Management Forum* (*SMF*), IIM Tiruchirappalli.
16. Chanda SS, McKelvey B (2018) *A Computational Study Explaining Processes underlying Phase Transition*. Available at *arXiv*: https://arxiv.org/abs/1810.04036
17. Chanda SS, Nargundkar R. (2019) 'Why keep promises when contracts are incomplete?' Asian *Academy of Management Conference (AAOM*), Bali, Indonesia.
18. Chanda SS, Miller KD (2019) Replicating agent-based models: Revisiting March's exploration-exploitation study. ***Strategic Organization***, 17(4): 425–449.
19. Chanda SS (2019) 'Does a biased media pose challenges to democratic functioning**?**' *International Conference on Operations Research & Decision Sciences* (*ICORDS*) IIM Visakhapatnam, India.
20. Chanda SS, Yayavaram S (2020) *An Algorithm to Find Superior Fitness on NK Landscapes under High Complexity: Muddling Through*. Available at *arXiv*: https://arxiv.org/abs/2006.08333
21. Chanda SS, McKelvey B (2020) Back to the basics: Reconciling the continuum and orthogonal conceptions of exploration and exploitation. ***Computational and Mathematical Organization Theory***, 26(2): 175–206 DOI https://doi.org/10.1007/s10588-020-09311-y. https://rdcu.be/b4tNx
22. Banerjee DN, Chanda SS (2020) *AI Failures: A Review of Underlying Issues*. Available at *arXiv*: https://arxiv.org/abs/2008.04073
23. Chanda SS, Ray S (2021) *Why Do Strategic Projects Fail?* Available at SSRN: https://ssrn.com/abstract=3836325
24. Chanda SS (2021) *An Algorithm to Effect Prompt Termination of Myopic Local Search on Kauffman-s NK Landscape*. Available at *arXiv*: https://arxiv.org/abs/2104.12620

25. Chanda SS, Yayavaram S (2021) *Overcoming Complexity Catastrophe: An Algorithm for Beneficial Far-Reaching Adaptation under High Complexity*. Available at *arXiv*: http://arxiv.org/abs/2105.04311
26. Chanda SS, Burgelman RA (2022) 'The role of autonomous experimentation in organizational knowledge creation: A computational study'. *Conference in honor of Bill McKelvey*, UCLA, June 18.
27. Chanda SS (2023). 'Indoctrination with the metaphor of the iterated prisoner's dilemma: Some issues and a way forward'. *PRME* India World Tour Research Development Workshop. March 3.
28. Yayavaram S, Chanda SS (2023) Decision making under high complexity: A computational model for the science of muddling through. ***Computational and Mathematical Organization Theory*** 29: 300–335. https://doi.org/10.1007/s10588-021-09354-9 [https://rdcu.be/c5uF8]
29. Chanda, Sasanka Sekhar (2023) *A Constructor Theory-based Approach for Computer Code Model Validation: The Crucial Role of an Effort-directing Feedback Mechanism*. Available at SSRN: https://ssrn.com/abstract=4468005 or http://dx.doi.org/10.2139/ssrn.4468005.
30. Chanda SS (2023). An update to Duncan's theory: Uncertainty as a function of complexity, environmental turbulence, and managerial approach to decision-making. *India Strategy Conference 2023*, IIM Bangalore, December, 2023.
31. Chanda SS, Banerjee DN, (2024) Omission and commission errors underlying AI failures. ***AI & Society*,** 39: 937–960. https://doi.org/10.1007/s00146-022-01585-x [https://rdcu.be/c5uFR]
32. Chanda SS (2025) *A Framework to Assess the Potential Impact of Adoption of an AI System in Business Functions in a Company*. Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5042416
33. Chanda SS (2025) Anticipating a renaissance in organization and management theories. ***International Journal of Complexity in Leadership and Management***. 4(1): 1–11.
34. Chanda SS (2025) Enabling Creative Destruction: A View from the Perspective of Managerial Control. ***Journal of Contemporary Business Research***. 1(1): 8–22. [https://journals.sagepub.com/doi/10.1177/3049513X241233064]
35. Chanda SS, Burgelman RA (2025) *Diversity can be the basis for continued organizational order: A computational proof of Prigogine's conjecture bridging physical and social systems***.** Available at *Research Square*. [https://www.researchsquare.com/article/rs-6521388/v1]
36. Chanda SS (2025) The *Effect of Sporadic Dormancy on Adaptation under Natural Selection: A Formal Theory*. Available at *arXiv*: https://arxiv.org/abs/2506.22758.
37. Chanda SS, Ray S (2025) Is comprehensiveness in making strategic decisions always helpful? ***Australian Journal of Management*** 50(4): 1011–1036.
38. Chanda SS (2026) "The post bell-curve world: Is frequent feedback to employees always helpful?" *12th Management Doctoral Colloquium* (12th MDC) organized by Vinod Gupta School of Management, IIT Kharagpur (January 28 – 30).
39. Burgelman RA, Chanda SS (2026) Autonomous strategic behavior, organizational learning and top management support: Re-examining field research with computational modeling. ***Strategic Management Review***. 7(1). [https://www.strategicmanagementreview.net/assets/articles/Burgelman.pdf]
40. Burgelman RA, Chanda SS (2026 / Forthcoming) Strategic exploitation challenges and organizational learning: Integrating computational and field research findings. ***Strategic***

Management Review.
[https://www.strategicmanagementreview.net/assets/articles/Burgelman%20and%20Chanda.pdf]

41. Burgelman RA, Chanda SS (2026 / Forthcoming) *Evolutionary View of Strategy-Making. Oxford Research Encyclopedia.*

Appendix: MATLAB program code replicating March (1991) Chanda & Miller (2019)

```
%%% TRANSLATION OF PROF. MARCH'S CODE FROM BASIC TO MATLAB. MY HEARTFELT
%%% THANKS TO LATE PROF. MARCH FOR MAKING THE CODE AVAILABLE TO ME. SASANKA.


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% In order to run distinct cases of March 1991, Figure 1 ... 5, please
%% comment out the code for the other cases in the flower boxes below.
%% The flower boxes are identified as FB01 … FB05. In a given run, the
%% code inside only one flower box should be uncommented, contents of all
%% other Flower Boxes should be commented out by placing a '%' at the
%% beginning of each line of code. At the end of simulation, the results
%% are to be found in the variable p4_eka for Figures 1,2 & 4 and in the
%% variable p4_aock for Figure 5. For Figure 3 (FB03] the results are in
%% two containers, p4_eka (org code knowledge) and p4_fig3 (average
%% knowledge of slow and fast learners and average individual knowledge).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%

%% To get March's results, the variables flag_neg, flag_2_step and
%% flag_0_guess must be set to one. flag_neg represents negative marking.
%% For example, if out of 10 total beliefs of an entity, 7 are correct (with
%% respect to the standard of the external reality) 1 is wrong and 2 are
%% '0', (i.e. cannot be determined to be wrong or right), the logic of
%% Prof. March's code would assign a score of 6/10, i.e., implementing
%% negative marking for the wrong belief. In contrast the publication text
%% (correctly) states that scoring is on the "proportion of correct beliefs".
%% Probably above was just a coding mistake, occurring due to
%% multiplication of the reality and org. code (or member knowledge)
%% vectors instead of counting the number of matches one by one.
%% Since this went unnoticed, the other two fixes described below
%% became necessary to make the curves behave.
%% Flag_2_step represents a 2-step update of a member's
%% non-conforming belief to the org code's non-zero belief. The text of
%% March's paper suggests that, when a member's belief is not conforming to
%% the (non-zero) belief of the organizational code, it will get updated to
%% the organizational code's belief with a probability p1. However, Prof.
%% March's code implements (effectively) a two-step update when a member's
```

%% belief is not conforming to the (non-zero) belief of the organizational
%% code. Accordingly, in case of such non-conformance, the member's belief
%% is updated to 0 with a probability p1. Only '0' beliefs of members get
%% updated to the org-code's (non-zero) belief with probability p1.
%% Flag_0_guess = 1 represents the idea that, when the org code is
%% selecting elites, a given member's zero belief is randomly guessed as
%% '-1' or '1' (with equal probability), and belief scores are computed
%% based on this perceived belief set. This mechanism is not given in
%% the paper. This was definitely unintended, since it violates the closed
%% system assumption that applies to Figures 1, 2 and 3.

```
dim_reality = 30; %% M
N = 50; %% number of members in the organization / group
%%iterations = 80; %% mc_steps: TEST MODE
iterations = 10000; %%

%% implies initial popln have (-1,0,1) with 1/3 probability
prejudice = 2/3 ;
REAL = 0.50; %% probability bit value of the Reality is 1 or -1

beliefs = zeros(N, dim_reality); %% BELIEF | N rows dim_reality columns
score = zeros(N, 1 ); %% SCORE | col vector
init_reality_str = ones(1, dim_reality); %% REALWORLD | row vector
collective = zeros(1, dim_reality); %% COLLECTIVE | org_code row vector
dim_sum = zeros(1, dim_reality) ;
%% SUM | initializing container to keep elites' belief sum

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Necessary additional (undocumented) parameter in March 1991 code
p_interpret = 0.50 ;
%% probability of interpretation of '0' in member belief as -1 or 1

%% Additional parameters from study of March's code. The three flags below
%% need to be set to value '1' to have replication of March's results. A
%% value of '0' in any flag will show the results that transpire when the
%% undocumented feature (w.r.t. text of the 1991 paper) is absent.

%% '1' implies negative marking for false beliefs; '0' => no negative marking
flag_neg = 1;

%%'1' implies 2 step update of non-conforming member beliefs; '0' implies 1 step update
flag_2_step = 1;

%% '1' implies members' 0 bits are randomly assigned '1' or '-1' in choosing elites.
flag_0_guess = 1;
%% '0' in flag_0_guess implies, members' 0 bits are ignored in choosing %% elites
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FB01 : To replicate Figure 1 of March 1991
%%  p4_eka will contain the output
TT = 250; %% period_choice: Figure 1 & 2 only
p1 = [0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9]; %% SOCIALIZATION: Figure 1
p2 = [0.1 0.5 0.9]; %% LEARNING: Figure 1 only
p3 = 0; %%TURNOVER
p4 = 0 ; %% TURMOIL: Figure 1, 2, 3, 4
flag_soc = 0; %% '0' implies no heterogeneous learning: Fig 1, 4, 5
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FB02 : To replicate Figure 2 of March 1991
% % p2_eka will contain the output for heterogeneous p1.
%% The output for homogeneous p1 is to be obtained from FB01 (middle col)
% TT = 250; %% period_choice
% p1 = [0.2 0.3 0.4 0.5 0.6 0.7 0.8]; %% AVERAGE SOCIALIZATION RATE
% p2 = 0.5; %% LEARNING:
% p3 = 0; %%TURNOVER:
% p4 = 0 ; %% TURMOIL:
% flag_soc = 1; %% '1' implies heterogeneous learning: Fig 2, 3

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FB03 : To replicate Figure 3 of March 1991
% O/P SL-FL-AL will be in p1_fig3 / p2_fig3 / p3_fig3. code =>p4_eka


% TT = 20; %% period_choice
% %% corresponds to 0-100% fraction of members with p1 = 0.90.
% p1 = [0.1 0.18 0.26 0.34 0.42 0.5 0.58 0.66 0.74 0.82 0.9];
% p2 = 0.5; %% LEARNING:
% p3 = 0; %%TURNOVER:
% p4 = 0 ; %% TURMOIL:
% flag_soc = 1; %% '1' implies heterogeneous learning: Fig 2, 3

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FB04 : To replicate Figure 4 of March 1991
% %%% Output will be in the variables p3_eka, p4_eka
% TT = 20; %% period_choice
% p1 = [0.10 0.90]; %% SOCIALIZATION
% p2 = 0.5; %% LEARNING:
% p3 = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1]; %% TURNOVER: Figure 4
only
% p4 = 0 ; %% TURMOIL:
% flag_soc = 0; %% '0' implies no heterogeneous learning

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FB05 : To replicate Figure 5 of March 1991
%% output will be in variable p4_aock for Figure 5
% TT = 100; %% period_choice
% p1 = 0.50; %% SOCIALIZATION
% p2 = 0.5; %% LEARNING:
% p3 = [0 0.10]; %%TURNOVER:
% p4 = 0.02 ; %% TURMOIL:
% flag_soc = 0; %% '0' implies no heterogeneous learning

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

y_prejudice = prejudice/2;
z_prejudice = 1 - y_prejudice;
```

```
p1_cases = size(p1, 2);
p2_cases = size(p2, 2);
p3_cases = size(p3, 2);
p4_cases = size(p4, 2);

eka = zeros(1, p1_cases);
knowledge01 = zeros(TT, 1);
ock = zeros(TT, iterations);
aock = zeros(p1_cases, TT);

knowledge02 = zeros(iterations, 1);

p1_fig3 = zeros(3, p1_cases);


%%%% For Fig 2 & Fig 3 compute fraction of slow learners
if flag_soc == 1 %%% Implements heterogeneous learning.

    %%%%%%%%%%%%% x_points for p1_mixed %%%%%%%%%%%%%%%%%%%%
    % fraction with p1 = 0.9 is (1/8) * { ( het_mat /0.1) -1 }
    p1_1_fraction =  1 - (1/8) * (    ( p1 /0.1) - 1       ) ;
    %%  Above will be a vector of size p1_cases

    slow_learners_row = round(N * p1_1_fraction);
    slow_p1 = 0.10; %% For Figure 2 and Figure 3
    fast_p1 = 0.90; %% For Figure 2 and Figure 3

else
    %% will signify homogeneous learning
    slow_learners_row = (-1)* ones(1, p1_cases);

end; %if flag_soc == 1



for p4_ind = 1:1:p4_cases
    set_p4 = p4(p4_ind);
for p3_ind = 1:1:p3_cases
   set_p3 = p3(p3_ind);
for kk = 1:1:p2_cases
    set_p2 = p2(kk);
    for jj = 1:1:p1_cases
        set_p1 = p1(jj);

        if flag_soc == 1
          slow_learners = slow_learners_row(jj);
        end; %% if flag_soc == 1


        %% begin of monte carlo iterations
        equi_know = 0; %% EQUIKNOW
        time_to = 0; %% TIMETO

        for ll = 1:1:iterations

            %% populate initial_reality_string & org_code knowledge vector

            rand01 = rand(1, dim_reality);
            for idx01 = 1:1:dim_reality
                init_reality_str(idx01) = 1; %%% initialization
                if rand01(idx01) < REAL
```

```
                init_reality_str(idx01) = -1;
            end;

            collective(idx01) = 0; %% all bits of org_code have 0
        end; %% for idx01 = 1:1:dim_reality
        %%clear rand01;

        %% populate belief set of members of the organization
        rand02 = rand(N, dim_reality); %% supply of random numbers
        for idx01 = 1:1:N
            for idx02 = 1:1:dim_reality
                beliefs(idx01, idx02) = 0; %%% initialization
                if rand02(idx01, idx02) < y_prejudice
                    beliefs(idx01, idx02) = 1;
                elseif rand02(idx01, idx02) > z_prejudice
                    beliefs(idx01, idx02) = -1;
                else
                    beliefs(idx01, idx02) = 0;
                end; %% if rand02(idx01, idx02) < y_prejudice

            end; %% for idx02 = 1:1:dim_reality
        end; %% for idx01 = 1:1:N
        %%clear rand02;

        %% Begin of Time Steps
        marker = 0;
        idx00 = 1:1:TT; %% initializing container to 0
        knowledge01(idx00) = 0;

        for T = 1:1:TT
            marker = marker + 1;

            %% compute knowledge of org code, relative to reality
            if flag_neg == 1
              knowledge = init_reality_str * collective' ;
            else
                knowledge = 0;
                for i =1:1:dim_reality
                    if collective(i) == init_reality_str(i)
                        knowledge = knowledge + 1;
                    end;
                end;
            end;
%% note: above need to be modified downstairs to address TURMOIL (p4)
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% compute knowledge score of members, based on perceived beliefs
            beliefstar = beliefs;
            rand03 = rand(N, dim_reality); %% supply of random numbers
            for idx01 = 1:1:N
                score(idx01) = 0; %% re-initialization / refresh!!
                for idx02 = 1:1:dim_reality

                    if beliefstar(idx01, idx02) == 0
                      if flag_0_guess == 1
                        if rand03(idx01, idx02) > p_interpret
                            beliefstar(idx01, idx02) = -1;
                        else
                            beliefstar(idx01, idx02) = 1;
                        end;
```

```
                        else
                            %% do nothing: Members' '0' beliefs will have
                            %% nothing to contribute in determination of
                            %% elites
                        end; %% if flag_0_guess == 1

                    end; %% if beliefstar(idx01, idx02) == 0

                    if flag_neg == 1
    %% keep adding to a member's score for reality-beliefstar bit matches
    %% Penalize wrong beliefs by subtracting from the score
                    score(idx01) = score(idx01) +
init_reality_str(idx02) * beliefstar(idx01, idx02) ;

                    else %% note, reality str will never have 0 values.
           %% So, chance of scoring a 0-0 match by mistake does not exist.
                  if init_reality_str(idx02) == beliefstar(idx01, idx02)
                            score(idx01) = score(idx01) + 1;
                        end;

                    end; %% if flag_neg == 1

                end; %% for idx02 = 1:1:dim_reality
            end; %% for idx01 = 1:1:N
            %%clear rand03;
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            %% Effect member learning by socialization (p1)
         %% initializing container to keep elites' belief sum to 0
            idx00 = 1:1:dim_reality;
            dim_sum(idx00) = 0;

            rand04 = rand(N, dim_reality); %% supply of random numbers
            for idx01 = 1:1:N

                if flag_soc == 1
                    if slow_learners > 0
                        if idx01 <= slow_learners
                            set_p1 = slow_p1;
                        else
                            set_p1 = fast_p1;
                        end; %% if idx01 < = slow_learners
                    elseif slow_learners == 0
                        set_p1 = fast_p1;

                    end; %% if slow_learners > 0

                end; %% if flag_soc == 1

              for idx02 = 1:1:dim_reality

                  if collective(idx02) == 0
                      %% do nothing
                  else
               temp01 = collective(idx02) * beliefs(idx01, idx02);

                        switch temp01

          case 0 %% member's bit is 0, update member to org_code value

                                if rand04(idx01, idx02) < set_p1
```

```
                    beliefs(idx01, idx02) = collective(idx02);
                      end; %% if rand04(idx01, idx02) < set_p1
                  case 1
                      %% values match, do nothing
                  case -1
                      if rand04(idx01, idx02) < set_p1

                        if flag_2_step == 1
            %% values don't match update member bit value to 0
                          beliefs(idx01, idx02) = 0;
                        else
%% 1 step update of non-conforming member bit to non-zero org code value.
                    beliefs(idx01, idx02) = collective(idx02);
                        end; %% if flag_2_step == 1

                      end; %% if rand04(idx01, idx02) < set_p1


              end; %%switch temp01


          end; %% if collective(idx02)

      end; %% for idx02 = 1:1:dim_reality

      if score(idx01) > knowledge
%% In dim_sum we accumulate the sum of beliefs of all elites, for each dim.
          for idx03 = 1:1:dim_reality
  dim_sum(idx03) = dim_sum(idx03) + beliefstar(idx01, idx03);
          end; %% for idx03 = 1:1:dim_reality

      end; %% if score(idx01) > knowledge

    end; %% for idx01 = 1:1:N
    %%clear rand04;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%% EFFECT LEARNING BY ORGANIZATIONAL CODE
  rand05 = rand(N, dim_reality); %% max majority of N possible
    for idx03 = 1:1:dim_reality

        if dim_sum(idx03) == 0
            %% do nothing
        elseif dim_sum(idx03)  > 0 %% POSITIVE case

            if collective(idx03) == 1
               %% do nothing
            else

                for idx04 = 1:1:dim_sum(idx03)
                    if rand05(idx04, idx03) < set_p2
                        collective(idx03) = 1;
                        break;
                    end; %% if rand05(idx04, idx03) < set_p2

                end; %% for idx04 = 1:1:sum(idx03)

            end; %% if collective(idx03) == 1

        else %% dim_sum(idx03) < 0 NEGATIVE case
            if collective(idx03) == -1
```

```
                %% do nothing
            else
                temp02 = (-1)* dim_sum(idx03);
                for idx04 = 1:1:temp02
                    if rand05(idx04, idx03) < set_p2
                        collective(idx03) = -1;
                        break;
                    end; %% if rand05(idx04, idx03) < set_p2

                end; %% for idx04 = 1:1:temp02

            end; %% if collective(idx03) == -1

        end; %% if dim_sum(idx03) == 0

    end; %% for idx03 = 1:1:dim_reality
    %%clear rand05;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   %%% Effect of Turmoil
   if set_p4 > 0
     for idx03 = 1:1:dim_reality

         if rand() < set_p4
     init_reality_str(idx03) = (-1)* init_reality_str(idx03);
         end; %% if rand() < set_p4

     end; %% for idx03 = 1:1:dim_reality
   end; %% if set_p4 > 0
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Effect of Turnover
if set_p3 >0
    if T < TT
        for idx01 = 1:1:N

            if rand() < set_p3
                rand06 = rand(1, dim_reality);
                for idx02 = 1:1:dim_reality
                    beliefs(idx01, idx02) = 0;
                    if rand06(idx02) < y_prejudice
                        beliefs(idx01, idx02) = 1;
                    elseif rand06(idx02) > z_prejudice
                        beliefs(idx01, idx02) = -1;
                    else
                        beliefs(idx01, idx02) = 0;
                    end; %% if rand06(idx02) < y_prejudice

                end; %% for idx02 = 1:1:dim_reality

            end; %% if rand() < p3

        end; %% for i1 = 1:1:N

    end; %% if set_p3 >0
end; %% if T < TT

    knowledge01(T) = knowledge;
    ock(T, ll) = knowledge;

    %% score calc assumes all TT periods are run
```

```
                if flag_soc == 1 && T == TT
                    indivs_score(ll,:) = score';
                end;

            end; %% for T = 1:1:TT

            %% capture the end-of-period knowledge avg over dims
            knowledge02(ll) = knowledge01(TT)/ dim_reality;
%% assumes all timesteps are executed
        end; %% for ll = 1:1:iterations
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %%%%%%%%%%%% Compute Results
        if flag_soc == 1
            indivs_score_col = mean(indivs_score); %% row mean
            if slow_learners > 0 && slow_learners < N
                score_low = 0;
                for pp = 1:1:slow_learners
                    score_low = score_low + indivs_score_col(pp);
                end; %% for pp = 1:1:slow_learners
                score_low_avg = score_low / ((slow_learners)*dim_reality);

                score_high = 0;
                for pp = (slow_learners + 1):1:N
                    score_high = score_high + indivs_score_col(pp);
                end; %% for pp = 1:1:slow_learners
           score_high_avg = score_high/ ((N - slow_learners)*dim_reality);

                %% average score across all org members
               overall_score_avg = sum(indivs_score_col) / (N*dim_reality);

            elseif slow_learners == 0
                score_low_avg = 0;
                score_high_avg = sum(indivs_score_col) / (N*dim_reality);
                overall_score_avg = score_high_avg;
            elseif slow_learners == N
                score_high_avg = 0;
                score_low_avg = sum(indivs_score_col) / (N*dim_reality);
                overall_score_avg = score_low_avg;

                %%% not coding for subsequent stacking
            end; %% if slow_learners > 0 && slow_learners < N
            p1_fig3(1, jj) =  score_low_avg;
            p1_fig3(2, jj) =  score_high_avg;
            p1_fig3(3, jj) =  overall_score_avg;

        end; %% if flag_soc == 1

        aock(jj,:) = mean(ock, 2)'/ dim_reality; %% TT cols

        know_per_iteration = sum(knowledge02)/iterations;
        eka(jj) = know_per_iteration;
        %% eka(jj) = equi_know/ (dim_reality * iterations);

    end; %% for jj = 1:1:p1_cases

    if kk == 1
        p2_eka = eka;
        p2_aock = aock;
        p2_fig3 = p1_fig3;
    else
```

```
        p2_eka = [p2_eka; eka];
        p2_aock = [p2_aock; aock];
        p2_fig3 = [p2_fig3; p1_fig3];
    end; %% if kk == 1

end; %% for kk = 1:1:p2_cases

if p3_ind == 1

    p3_eka = p2_eka;
    p3_aock = p2_aock;
    p3_fig3 = p2_fig3;
else
    p3_eka = [p3_eka; p2_eka];
    p3_aock = [p3_aock; p2_aock];
    p3_fig3 = [p3_fig3; p2_fig3];

end; %% if p3_ind == 1


end; %% for p3_ind = 1:1:p3_cases

if p4_ind == 1
    p4_eka = p3_eka;
    p4_aock = p3_aock;
    p4_fig3 = p3_fig3;
else
   p4_eka = [p4_eka; p3_eka];
   p4_aock = [p4_aock; p3_aock] ;
   p4_fig3 = [p4_fig3; p3_fig3];
end;

end; %% for p4_ind = 1:1:p4_cases

clear rand* idx* belief* *cases temp* know* score* ock ll jj kk T;
clear dim_sum y_prejudice z_prejudice time_to marker;
clear slow_learners_row init_reality_str equi_know collective;
```