Sasanka Sekhar Chanda

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

Sasanka Sekhar Chanda is a faculty in strategic management in Indian Institute of Management Indore. His research interests are in strategic decision-making, managerial intentionality, and complexity theory. Earlier, Sasanka worked in the industry in a range of roles spanning engineering, consulting, and management for a period of fifteen years. Some of his key publications are in *Computational and Mathematical Organization Theory*, *Decision, M@n@gement* and *Strategic Organization*.

Teaching

Post Graduate Program:
Strategic Management II focusing on Corporate Strategy
A Perspective on C. K. Prahalad's Strategic Thinking
Executive Post Graduate Program:
Artificial Intelligence in Management and Business
Doctoral Program:
Theory of the Firm
Strategy Process Research
Philosophical Moorings of Social Science Research, focusing on developing a research question

Publications in peer reviewed journals

7. Chanda S. S., and Miller K. D. (2018). Replicating agent-based models: Revisiting March's exploration-exploitation study. *Strategic Organization.* DOI: 10.1177/1476127018815295. https://journals.sagepub.com/doi/pdf/10.1177/1476127018815295

Description. In this paper we show that the graphs underlying the March (1991) study — introducing exploration and exploitation through an organizational learning metaphor — were generated using computer code that had three additional features not mentioned in the publication text. Removal of these undocumented features puts March's theory on firmer footing and opens up the genetic algorithm platform for multi-level theory development in a wide variety of topics in organization and management studies.

What is accomplished? *This paper conclusively replicates March (1991) and suggests that the appropriate form of the March (1991) model for future work extending the model is one where the assumptions existing only in March's code—i.e. absent from the publication text—*

are taken off. An entire field of theory development is opened up by virtue of the general availability of the genetic algorithm platform for developing multi-level theory.

6. Chanda S. S., Ray S., and McKelvey B. (2018). The continuum conception of exploration and exploitation: An update to March's theory. *M@n@gement*, 21(3): 1050–1079. http://www.management-aims.com/download.php?id=401&l=en&f=en_1538596574.pdf

Description. In this paper we show that for the continuum conception of exploration and exploitation, March's (1991) result (Figure 2, p. 77) that more exploration is always desirable reverses if we use a lower stock of collective human capital (CHC) than that assumed in March's experiments. Our research indicates that a section of extant research is mistaken in assuming that March's formal model for the continuum conception suggests an inverted U-shaped relation between the extent of exploration and organizational outcome. Instead, the level of CHC determines whether it is rewarding to focus on exploration or exploitation. Thus, the formal model supports managerial intentionality towards exploratory and exploitative innovation through appropriate choice of the level of CHC. We call for a new "balance" discussion, focusing on the determinants of the minimum level of the non-preferred activity from among exploration and exploitation.

What is accomplished? *This paper effectively provides the missing half of March's theory regarding the continuum conception of exploration and exploitation.*

5. Chatterjee A., Chanda S. S., and 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

Description. In this paper we build theory highlighting the dark side of transaction cost economics (TCE) theory. Specifically we discuss the deleterious consequences that ensue when TCE is used to administer organizations undergoing change. The dysfunctions arise owing to (a) TCE's inappropriate overreliance on managerial foresight (b) TCE's inability to handle interaction between transactions, given that TCE uses a transaction between dyadic parties as the unit of analysis and (c) TCE's inability to distinguish between shirking and honest mistakes.

What is accomplished? *This paper contributes to the literature that highlights negative impacts of governing organizations by the TCE approach. Specifically, it suggests reasons why large change projects fail and why radical innovation has virtually dried up in multinational firms governed on TCE principles.*

4. Chanda S. S. (2017). Inferring final organizational outcomes from intermediate outcomes of exploration and exploitation: The complexity link. *Computational and Mathematical Organization Theory*, 23(1): 61–93. (https://rdcu.be/5wsj) DOI: 10.1007/s10588-016-9217-1

In this paper I suggest an approach to derive probability of organizational success from an intermediate construct, the level of accumulated organizational knowledge. Extent of matches between sub-samples from the org. code knowledge and the environment (or external reality) enable this computation. Thereby, outputs of managerial efforts are better discerned even if environmental perturbations exist. The paper builds on the elaboration by Mosakowski and McKelvey (1997) that the resource-based-view (RBV) of the firm is, in fact, not a tautology, as is discernible when intermediate constructs are separated from final outcome constructs.

What is accomplished? *Complexity is introduced as a key construct connecting intermediate outcomes with final outcomes. This enables implementing better accountability of managers.* [Mosakowski E, McKelvey B (1997) Predicting rent generation in competence-based competition. In: Heene A, Sanchez R (eds) Competence-based strategic management. Wiley, New York].

3. Chanda S. S., and 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

Description. In this paper we suggest that as far as configuration and implementation of information technology solutions for companies is concerned, the design skills clearly lie in countries like India, on account of greater familiarity with a wider variety of business processes compared to the Western countries. This is quite the opposite of what transpired when Western countries shifted manufacturing to China, keeping design to themselves.

What is accomplished? *The paper highlights that higher variety in service delivery in the emerging markets confers higher service design and configuration skills to emerging market designers, breaking a groupthink that all design must occur in the West.*

2. Chanda S. S., and 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

Description. This paper suggests a different answer to the question what is an optimal mix of exploration and exploitation. Posen and Levinthal (2012, *Management Science*) say that the optimal proportion is 50:50. Posen and Levinthal (2012) assume that exploration and exploitation are two ends of a continuum. They use a single-agent bandit model to derive their answer. In our paper we model exploration and exploitation as orthogonal constructs following a formalization given in March (1991). We show that multiple exploration: exploitation mixes attain the same optimal outcome. Thus, managerial intentionality is feasible: managers do not have to adopt one 'right' mix of exploration and exploitation. Our work demonstrates Prigogine's principle, that diversity can be a source of continued order. It further shows that a moderate rate of inflow of diverse, un-vetted knowledge helps firms combat Knightian Uncertainty.

What is accomplished? *The paper establishes that managers orienting their organization towards exploitative innovation can do equally well as managers orienting their organization towards exploratory innovation. It also shows that the prescription regarding appropriate managerial action is vastly different in open systems—where the objective is to fashion orderly structures in far-from-equilibrium conditions—compared to the prescription from research that mandates reaching equilibrium in a closed system as its main purpose.*

[Posen HE, Levinthal DA (2012) Chasing a moving target: Exploitation and exploration in dynamic environments. Management Science 58(3):587–601].

1. Chanda S. S., and 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

Description. In this article we distinguish two streams of research for theory development by computational simulation modeling: a critical realist stream that seeks to investigate outcomes by inverting one or more key assumptions in a dominant agent-based model, and a scientific realist stream in the semantic conception tradition that seeks to extend theory or build theory by new constructions in well-known agent-based models, preserving key assumptions.

What is accomplished? *The paper lays down a roadmap for researchers wishing to work on theory development by computational simulation modeling (TDCSM). It provides guidance to editors for suitably assessing a TDCSM manuscript.*

Other Research

III). Chanda S. S., McKelvey, B. (2018) A computational study explaining processes underlying phase transition. arXiv.org > physics > arXiv:1810.04036 https://arxiv.org/abs/1810.04036

In this article we demonstrate a mechanism for phase transition that has the potential to challenge the dominant Ising model, or inform where the Ising model fails. Here, phase transition is defined as attainment of very widely differing final value on an outcome of interest, on account of small differences in initial conditions. We unearth an elegant mechanism by going deep into results akin to phase transition found in a genetic algorithm model. The mechanism involves initial accumulation of incorrect knowledge (or harmful chemicals / vectors) OR correct knowledge (or beneficial chemicals / vectors) owing to initial difference in concentration, followed by positive feedback loops where (a) a virtuous cycle leads to high accumulation of correct knowledge (or beneficial chemicals / vectors) and (b) a vicious cycle leads to high accumulation of incorrect knowledge (or harmful chemicals / vectors).

II). Chanda, S. S. (2016) Corporate Strategy as order creation in disequilibrium, IIM Indore Technical Note, Technical Note, AY 2016-17, TN/01/2016-17/SM

In this article I interpret Professor C. K. Prahalad's work as a call to fashion business organizations as orderly structures that survive and prosper in far-from-equilibrium conditions, departing from an economics-led view of trying to find equilibrium points for firms to function on. The idea that orderly structures can thrive in far-from-equilibrium conditions was elaborated by Ilya Prigogine in his Noble Prize acceptance lecture. On account of an inability to accommodate irreversibility in the theoretical models, the economics-led view is of limited use in management studies. As noted by Professor Bill McKelvey and others, agent-based models are eminently suited to incorporate irreversibility as well as avoid being prisoner to other unpalatable and inappropriate assumptions of mathematics.

I). Chanda, S. S. (2015) CEO cognition in strategy research. http://dx.doi.org/10.2139/ssrn.2586215.

In this article I argue that the mental maps of CEOs get shaped by the experiences they accumulate by meeting various stakeholders as part of their job role, e.g., (a) the TMT (b) Financial Analysts in the Wall Street (c) Regulators (d) Media (e) Board (own firm) (f) Shareholders (g) Debt and Bond holders (h) Employees of the firm (i) Management Consultants engaged by the firm (j) Trade Associations, (k) Board of Director membership in other firms.

Appendix: MATLAB program code replicating March (1991), related to #7 above.

```
%%% 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: Figures 1,2,3,4,5
%%iterations = 10000; %%

prejudice = 2/3 ; %% implies initial popln have (-1,0,1) with 1/3 probability
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.

flag_neg = 1; %% '1' implies negative marking for false beliefs; '0' => no negative marking
flag_2_step = 1; %% '1' implies 2 step update of non-conforming member beliefs; '0' implies 1 step
update
flag_0_guess = 1; %% '1' implies members' 0 bits are randomly assigned '1' or '-1' in choosing elites.
%% '0' in flag_0_guess implies, members' 0 bits are ignored in choosing
%% elites
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
%% FB01 : To replicate Figure 1 of March 1991

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
```

```
% %
% 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

% 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
%
% 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

% 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;

[v_unused p1_cases] = size(p1);
[v_unused p2_cases] = size(p2);
[v_unused p3_cases] = size(p3);
```

```
[v_unused p4_cases] = size(p4);

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)
           idx00 = 1:1:dim_reality; %% initializing container to keep elites' belief sum to 0
           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; %%% initialization
                        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, results in row vector
        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
```